



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Short-Term Electricity Demand Prediction in Great
Britain Using Machine Learning**

Name: Mehul Tank

Student ID: 2410391

University: University of Wales Trinity Saint David Birmingham

Course: MSc in Data and Artificial Intelligence

Module: Master's Project

Supervisor: Dr Abdallah Naser

Declaration

I, ... MEHUL TANK declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Mehul Tank

Date: 05/01/2026

Form of Consent

I, ... MEHUL TANK hereby consent that my Project, submitted in candidature for the MSc in Data and Artificial Intelligence degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: Mehul Tank

Date: 05/01/2026

Acknowledgements

I want to give a big thank you to Dr Abdallah Naser, my supervisor, for all the help he gave me with my research from beginning to end. His expert knowledge of machine learning & power system analysis, as well as all his constructive input & support, was so valuable to me along the way and helped improve this dissertation.

I also want to give a big thank you to the University of Wales Trinity Saint David Birmingham for providing me with the necessary computational resources and academic facilities to finish my research successfully.

Lastly, I want to send my thanks to the publicly available data source providers [National Energy System Operator (NESO), British Menu and Running Schedule (BMRS), ESPENI and Open-Meteo] for the data they supplied that allowed me to perform a complete analysis of Great Britain's electricity systems.

Abstract

In Great Britain, the electricity sector is transforming into a more sophisticated, dispersed, and time-varying form as a result of rapid decarbonisation, the increasing penetration of renewables, and the changing patterns of electrical consumption. The transition from conventional fossil fuels to weather-dependent energy generation from wind and solar technologies has introduced increasing uncertainty in supply and demand, making accurate and timely predictions critical to the reliable and efficient operation of the grid.

This dissertation is intended to review the use of advanced machine learning techniques for the prediction of short-term electricity demand in Great Britain with reliance upon publicly available datasets for the period of January 2020 through December 2023. The analyses are based on four comprehensive datasets that can provide an in-depth data quality analysis: NESO Historical Demand, BMRS Fuels Mix, ESPENI Systems Indicators, and Open Meteo Weather Variables, creating a unique dataset comprised of variables that link demand with weather and operation. The study design included creating and validating three different prediction models, LSTM Neural Networks, XGBoost, and Prophet, with a focus on one- and twenty-four-hour-ahead electricity demand prediction. Prediction accuracy was measured with traditional statistical metrics (RMSE, MAE) as well as mean absolute percentage error.

The results demonstrate that machine learning predictive modelling techniques effectively capture temporal and non-linear dependencies in addition to providing more accurate predictions and providing system operators with practical implications in a developing GB electricity generation system, considering the substantial amount of electrical generation due to renewable resources.

Contents

Declaration.....	2
Form of Consent.....	2
Acknowledgements	3
Abstract.....	4
Abbreviations	8
List of Figures.....	9
List of Tables.....	10
Chapter 1: Introduction	11
1.1 Research Background	11
1.1.1 GB Electricity System.....	12
1.1.2 Data-driven Transition	13
1.1.3 Predicting Challenges in GB.....	13
1.2 Problem Statement.....	13
1.3 Aim and Objectives.....	14
1.4 Research Questions	15
1.5 Significance of the Study	15
1.6 Scope and Limitations.....	15
1.7 Contribution	16
1.8 Dissertation Structure.....	16
1.9 Chapter Summary	17
Chapter 2: Literature Review	18
2.1 Introduction.....	18
2.2 Conceptual Background of Electricity Demand Prediction.....	18
2.2.1 Role in Planning, Operation, and Markets.....	18
2.2.2 Demand Drivers and Predicting Horizons	19
2.3 Predicting Approaches	20
2.4 Data Sources and Feature Engineering	21
2.4.1 Data Sources	21
2.4.2 Feature Engineering Strategies	23
2.5 Research Gaps in Existing Literature	24
2.6 Conceptual framework.....	26

2.7 Justification for Model Selection	27
2.8 Chapter Summary	27
Chapter 3: Methodology.....	28
3.1 Introduction and Research Design	28
3.2 Data Sources and Preprocessing	29
3.2.1 Overview of Data Sources	29
3.2.2 Data Extraction, Cleaning, and Time Alignment	31
3.2.3 Merging Datasets	32
3.2.4 Exploratory Data Analysis	33
3.3 Feature Engineering and Selection	33
3.3.1 Temporal and Calendar Features.....	34
3.3.2 Weather, Fuel Mix, and Lag Features	35
3.4 Models, Training, and Validation.....	35
3.4.1 Modelling Strategy.....	36
3.4.2 Prophet Model.....	36
3.4.3 XGBoost Model	36
3.4.4 LSTM with Attention Model	37
3.4.5 Training Procedure.....	37
3.5 Tools and Reproducibility	38
3.6 Chapter Summary	38
Chapter 4: Results.....	39
4.1 Introduction.....	39
4.2 Model Implementation.....	39
4.2.1 Prophet Setup and Hyperparameters.....	39
4.2.2 XGBoost Pipeline, Feature Engineering, and Training.	40
4.2.3 LSTM with Attention Architecture, Sequence Creation, and Training.....	43
4.3 Global Performance Summary.....	45
4.4 One-Hour-Ahead (H1) Results	47
4.4.1 Prophet H1 Prediction.....	47
4.4.2 XGBoost H1 Prediction	48
4.4.3 LSTM H1 Prediction.....	49
4.5 Twenty-Four-Hour-Ahead (H24) Results	50
4.5.1 Prophet H24 Prediction.....	50

4.5.2 XGBoost H24 Prediction	51
4.5.3 LSTM H24 Prediction.....	52
4.6 Explainability Results	53
4.6.1 XGBoost SHAP Analysis.....	53
4.6.2 LSTM Attention Profiles.....	56
4.7 Chapter Summary	57
Chapter 5: Discussion	58
5.1 Introduction.....	58
5.2 Overview of Main Findings	58
5.3 Interpretation in Contradiction to Research Questions	58
5.4 Links to Literature and Theoretical Expectations	59
5.5 Implications at the Practical Level and the System Level	60
5.6 Methodological Thoughts and Limitations	60
5.7 Chapter Summary	61
Chapter 6: Conclusion	62
6.1 Introduction.....	62
6.2 Restatement of Aim and Objectives.....	62
6.3 Main Conclusions	62
6.4 Knowledge and Practice Contribution	63
6.5 Limitations	63
6.6 Recommendations.....	64
6.6.1 Practice Recommendations	64
6.6.2 Future Research Recommendations.....	64
6.7 Chapter Summary	64
References.....	65
Appendix A: Additional Figures	69
Appendix B: Additional Tables	81

Abbreviations

GB	Great Britain
UK	United Kingdom
NESO	National Energy System Operator
BMRS	Balancing Mechanism Reporting Service
ESPENI	Electricity System Performance and National Indicators
PV	Photovoltaic
EV	Electric Vehicle
DNO	Distribution Network Operator
ESO	Electricity System Operator
STLP	Short-Term Load Prediction
LSTM	Long Short-Term Memory
XGBoost	Extreme Gradient Boosting
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MW	Megawatt
API	Application Programming Interface
SHAP	SHapley Additive exPlanations

List of Figures

Figure 1.1: Dissertation Structure Diagram	16
Figure 2.1: Conceptual Framework of the Project	26
Figure 3.1: Quantitative Experimental Design Workflow	28
Figure 3.2: Workflow for Merging Datasets.....	32
Figure 4.1: Data Preparation for Prophet	39
Figure 4.2: Prophet Configuration	40
Figure 4.3: Feature Engineering for XGBoost	41
Figure 4.4: XGBoost Model Training.....	42
Figure 4.5: Sequence Creation for LSTM.....	43
Figure 4.6: LSTM with Attention Model Definition	44
Figure 4.7: LSTM with Attention Model Training	45
Figure 4.8: Performance of the overall model (H1 and H24)	47
Figure 4.9: The distribution of actual and predicted demand of Prophet H1	48
Figure 4.10: XGBoost H1 – actual vs predicted demand (last 7 days).....	49
Figure 4.11: LSTM H1 – actual vs predicted demand on the test set.....	49
Figure 4.12: Prophet H24 – distribution of actual and predicted demand.....	51
Figure 4.13: XGBoost H24 – actual vs predicted demand (last 7 days).....	52
Figure 4.14: LSTM H24 – actual vs predicted demand on the test set	52
Figure 4.15: SHAP summary for the XGBoost H1 model.....	54
Figure 4.16: SHAP summary for the XGBoost H24 model.....	55
Figure 4.17: Average attention weight profile for the LSTM over a 24-hour lookback ..	56

List of Tables

Table 2.1: Comparison of Models: Strengths and Limitations 21

Table 2.2: Summary of Datasets and Their Roles in Short-Term Demand Prediction ...23

Table 2.3: Identified Research Gaps and Corresponding Contributions 25

Table 3.1: Overview of Datasets, Sources, and Roles in the Study 30

Table 3.2: Data Preprocessing and Feature Engineering Workflow 32

Table 3.3: Categorisation of Engineered Features for Demand Prediction Models 34

Table 4.1: Test-set performance by model and horizon (MW) 46

Chapter 1: Introduction

The electricity sector of Great Britain (GB) is currently in a phase of considerable overhaul resulting from climate change imperatives, rapid technological change, and changing trends in electricity production and consumption [3], [4]. This transition from fossil fuel-based generation to low-carbon and renewable energy sources has produced unprecedented complexity in the operation and planning of power systems [3]. Concurrently, the rapid electrification of transport, heating, and business and industrial processes has created new and more volatile demand patterns on the electricity grid [62]. This changing landscape makes accurate prediction of short-term electricity demand essential for the continued operation of an economically efficient and environmentally sustainable power system. Predicting the demand for electricity has traditionally been accomplished using linear statistical methods, including ARIMA (Autoregressive Integrated Moving Average Models) and various regression techniques, which were effective up until recently in stable systems. As electricity generation and consumption continue to evolve into a more decentralised system of energy supply and demand and therefore become less stable and much more affected by nonlinear systems such as weather, these traditional methods are becoming less reliable. Machine learning and artificial intelligence are the latest technological developments that can help improve the prediction accuracy of electricity demand by determining the relationships between large, complex datasets much more accurately than ever before [9], [12], [21]. In this dissertation, the research looks at the possible usages of advanced ML algorithms, including Long Short-Term Memory (LSTM), XGBoost and Prophet, to forecast short-term electricity demand in Great Britain with a higher level of accuracy.

This chapter includes an overview of the research context, an outline of the research problem, an outline of the research objectives, and a discussion of the significance and limitations of the research.

1.1 Research Background

Across the globe, nations are taking steps towards meeting their climate targets by incorporating a greater share of renewable energy sources such as wind and solar energy into their economies and decreasing their reliance on coal, oil and natural gas [37]. For example, in the UK, offshore wind capacity, distributed solar PV systems (DSS), and battery storage systems have increased dramatically in a very short space of time, changing how we view and operate electricity systems [37]. Due to the inherent variability of renewables, weather situations create increased uncertainty in both the generation and balancing of consumer demand for electricity [41]. As the world continues to transition to a cleaner, more sustainable energy system, the demand for accurate short- to medium-range prediction metrics (1 hr and 24 hrs ahead) will continue to grow and evolve as consumers implement electric vehicles, heat pumps, and smart home technologies into their homes, thereby creating a need for increasingly robust prediction tools & techniques that provide high-fidelity prediction metrics [40], [42]. The rapid development and adoption of high-speed Internet, coupled with significant advances

in data availability and computational power, has enabled users to obtain, store, analyse and leverage large volumes of high-resolution electricity & weather data. Decision-making on operations is currently supported by the use of data-driven methods, especially those that use machine learning, as many organisations are beginning to adopt machine learning methods as a means of making better operational decisions [9], [39]. These data-driven machine learning models have the capability to overcome limitations of the current statistical models in the area of power systems by learning from large quantities of historical data to produce accurate and reliable predicting results, while also identifying complex patterns, which are typically difficult to predict using the current methods [2], [5].

This research is conducted in the larger context of power systems, data science, and sustainable energy management and focuses on how, as electricity networks become increasingly complex and decentralised, the need for more reliable and timely prediction of electricity grid supply and demand growth becomes critical for efficient operation. Predicting failures can result in inefficient dispatch of generation sources, increased balancing costs, increased carbon emissions, and, in extreme situations, potential threats to grid or system security [38]. This research will help identify effective and scalable prediction methods that can withstand the challenges of today's renewable-dominated electricity grid systems through the use of publicly available data sources and advanced machine learning methodologies. Ultimately, it will provide support for academic research into the operation of modern electricity systems and supply chains and will support the practical implementation of these methods in the management of electricity and energy resources.

1.1.1 GB Electricity System

Electricity is generated, transmitted, and distributed in Great Britain via an integrated electricity system. Generation of electricity is composed primarily of low-carbon technologies with greater emphasis on a combined usage of both offshore/onshore wind, nuclear, solar, hydroelectric, and gas-fired power stations [3]. The National Energy System Operator (NESO) oversees the transmission system, which contains a high-voltage electricity network throughout England, while each region has several Distribution Network Operators (DNOs) that distribute the lower-voltage electricity. The electricity generation mix for Great Britain has recently significantly changed from being primarily coal-based to becoming increasingly renewable-based [38], [60], [69]. Offshore wind generation has become the backbone of the electricity generation system by supplying the large quantities of low-carbon electricity being used; on the other hand, wind generation and solar generation are both variable and difficult to predict with a high degree of accuracy, resulting in increasing difficulty in balancing electricity generation in real-time [20]. Therefore, NESO places a significant amount of reliance on short-term demand prediction in balancing electric generation [31]. Short-term prediction impacts decisions regarding unit commitment, dispatch of flexible generation, activation of balancing services, and interconnection flows to neighbouring countries. With this increased complexity, the demand for highly accurate predicting tools has increased tremendously.

1.1.2 Data-driven Transition

Digital transformation is taking place in the energy sector with increasing use of data analytics and automation and the application of artificial intelligence. The introduction of smart meters, advanced sensors, and modern communication technologies has enabled utilities to receive high-resolution, real-time data on both electricity demand and system performance. As well as providing high-resolution information about demand and performance, these developments have also allowed for the application of machine learning models that can automatically identify patterns and correlations from large amounts of data [67]. Various machine learning methods, including deep learning and ensemble models, have been shown to be very effective for a range of predicting applications, while conventional statistical models cannot address the complexities of non-linear relationships and interactions among several different variables, including weather conditions, calendar effects, and consumer behaviours [2]. The use of data-driven approaches to electricity demand prediction allows for the integration of diverse types of data, such as market data, historical system demand records, fuel mix data, and meteorological predictions. This research aims to advance this transition to data-driven prediction through the development and evaluation of a reproducible prediction pipeline using publicly available datasets and more recent machine learning architectures.

1.1.3 Predicting Challenges in GB

Great Britain's electricity demand prediction faces several challenges due to the increasing share of renewable energy and growing variability/uncertainty within the electric system [3]. With the growing use of renewable generation comes an increase in demand uncertainty related to weather conditions, including temperature, wind speed, and cloud cover that can impact how much electricity is needed for heating/cooling and lighting [46]. Another area of complexity contributing to unpredictable demand patterns with more variability is the electrification of transport and heat use. Charging of electric vehicles (EVs) and operation of heat pumps creates additional peaks and valleys in total electricity consumption that traditional time-series models find difficult to account for. Additionally, several other factors not captured with statistical models can cause changes in consumer consumption patterns, including behaviour modification, economic conditions, and government policy intervention; therefore, more advanced predicting techniques that are dynamic and learn from historical data are now required for accurate electricity demand prediction [2]. Machine learning predicting models fit this requirement very well, as they are capable of processing massive amounts of data and developing non-linear relationships far more accurately and efficiently compared to more conventional statistical predicting approaches.

1.2 Problem Statement

While accurate short-term electrical demand prediction for GB is extremely important, traditional predicting techniques have not been able to keep up with the increased complexity

of GB's electrical supply system that is caused by the growth of renewable resources [51]. Traditional time-series models employ ARIMA-type approaches, regression, or basic time-based techniques to make short-term predictions, but they have difficulty capturing non-linear characteristics, regime changes, and the strong influence of weather and calendar-related variables on customer demand [5], [52]. The main problem presented by this study centres around finding ways to improve the ability of currently used predicting techniques to produce accurate short-term electrical demand estimates (1 and 24 hours out) in an increasingly renewable-rich and dynamic GB electric supply system [53]. Inaccurate predictions lead to inefficient system balancing, increased operating costs, increased carbon emissions due to increased reliance on and use of unnecessary backup generation, and ultimately, endangered system reliability [54]. Currently, there is a gap between research done on available models through research that is both reproducible and comparable to multiple advanced types of machine learning models within the unique context of Great Britain. For the most part, research has been conducted on individual models (e.g., LSTM, XGBoost) using private, proprietary data sources that are not easily reproduced. This study proposes a systematic method for the development, training, and testing of LSTM, XGBoost, and Prophet models using publicly available datasets [34].

1.3 Aim and Objectives

Aim

This research is to explore comparative models of "machine learning" to predict short-term electricity demand in Great Britain in order to find the model which is accurate and reliable for use in operations.

Objectives

This research has six key objectives:

1. Integrate the NESO Historical Demand Data, Elexon BMRS, ESPENI, and Open-Meteo datasets.
2. Generate temporal, meteorological, and calendrical features for use in the prediction.
3. Employ differing methods for training and validating the LSTM, XGBoost, and Prophet models.
4. Analyse the performance of each model type using RMSE, MAE, and MAPE metrics.
5. Use explanation methods to produce explanations for the XGBoost and LSTM models.
6. Publish the code and results of the research so the code is available as open-source and replicable by others.

1.4 Research Questions

To inform this project, the following research question will guide this study:

1. How well do a variety of different machine learning models (LSTM, XGBoost, Prophet) predict short-term electricity demand patterns in Great Britain through publicly accessible sources?
2. Which of the three types of ML produces the best-performing short-term prediction accuracy?
3. How do weather-related features, time-related features and holiday/weekday versus weekend calendar features influence the results given by the different ML models used?
4. How can I effectively utilise prediction results from different ML model types practically?

1.5 Significance of the Study

This research is important to scholars, practitioners, and professionals. Academically, it provides an important contribution to the evolving literature around energy prediction through a comprehensive evaluation of state-of-the-art machine learning methods for a renewable-dominated national power grid. The research has also developed a reproducible and transparent predicting pipeline from publicly available datasets and tools that are used by data scientists around the world. These components also ensure that the recommendations made from this study align with the best practices in data science, including the need for interpretability and validation of predictive models and upholding principles of openness and transparency in scientific research. Additionally, this study's results will benefit system operators, energy traders, and policymakers by enhancing the accuracy of short-term electric demand prediction. Accurate prediction will also lead to greater efficiency of system operations, decreased costs of balancing, decreased carbon emissions, and enhanced integration of renewable energy sources.

1.6 Scope and Limitations

This study aims to predict short-term electricity demand for Great Britain and aims to predict one hour and one day ahead. The study will utilise publicly available datasets from NESO Historic Demand Data, Elexon BMRS, ESPENI, and Open-Meteo. These datasets were obtained from official websites, and the data were collected programmatically via APIs. The study utilises a quantitative, experimental design and is limited to the use of three specific models: LSTM, XGBoost, and Prophet. The research excludes both long-term capacity planning and long-term market price prediction. While the models used in this study will be rigorously evaluated, it is anticipated the results may be impacted by limitations of the collected dataset(s), such as poor data quality, missing value(s), and time resolution of the datasets available. Additional limitations associated with the research methodology are as follows: the

training of deep learning models and the potential for overfitting, despite using cross-validation and robust evaluation protocols, will pose a challenge.

1.7 Contribution

This study has developed a data-driven, reproducible framework for short-term electricity demand prediction in Great Britain using publicly available datasets, NESO Historic Demand Data, Elexon BMRS FUELHH, ESPENI, and Open-Meteo, and combining them in a way that few GB demand prediction studies have done. The framework contains a complete end-to-end data pipeline, consisting of data cleansing, programmatic merging of the four datasets, and development of features from the data such as DayOfWeek, Month, Year, Date, BankHoliday, and is_weekend. Additionally, this research has compared the three leading current machine learning predicting approaches (LSTM, XGBoost, and Prophet) on a unified experimental design to predict one (1) hour ahead and twenty-four (24) hours ahead electricity demand. Furthermore, an evaluation framework to assess the performance of these models relative to baseline models and using multiple accuracy metrics has been created.

1.8 Dissertation Structure

This dissertation is structured into six chapters.

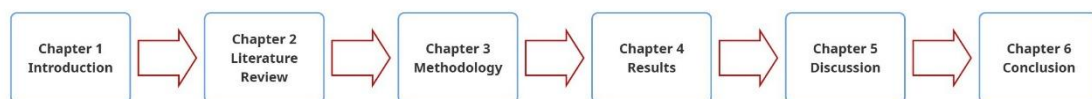


Figure 1.1: Dissertation Structure Diagram

Chapter 1 introduces the research context, background, aims, and significance of the study.

Chapter 2 provides a comprehensive review of the academic literature related to electricity demand prediction, traditional and modern modelling approaches, and machine learning applications.

Chapter 3 outlines the research methodology, including data collection processes, preprocessing steps, feature engineering techniques, model development, and validation procedures.

Chapter 4 presents the experimental results and the comparative performance of the predicting models.

Chapter 5 offers a critical discussion of these findings, linking them to existing research and highlighting key insights.

Finally, **Chapter 6** concludes the dissertation by summarising the main contributions, acknowledging limitations, and proposing directions for future research.

1.9 Chapter Summary

This chapter is dedicated to examining Great Britain's electric power market and how its evolution towards zero carbon, electrification, and increased use of digital technologies has led to increasing importance, for the purpose of maintaining a safe, reliable, and cost-effective electric power grid, to accurately predict electricity consumption in the short term. In particular, it highlights the fact that conventional statistical techniques will be challenged by the non-linear relationship between energy consumption and weather conditions as well as the effects of human behaviour in a grid that is predominately composed of renewable energy sources and is operated from many different locations. As a result, this chapter demonstrates the need for developing machine learning algorithms, including LSTM, XGBoost, and Prophet, that could enhance the accuracy of short-term electricity consumption projections to one hour and twenty-four hours ahead by utilising publicly available data on Great Britain's electricity market and weather conditions in conjunction with machine learning methods. The chapter establishes the main research problem focus, articulates the primary research goal, and builds an extensive set of research objectives and specific research questions for investigating model performance, temporal, weather, and calendar factors, and the understanding and interpretation of predicted results for operational application. Additionally, through the chapter, the academic and practical import of the research study is established, the project's scope and limitations are outlined, the major contributions therein to the existing literature are provided, and finally, an overview of the structure of the research thesis is presented. The structure of the research thesis is presented as a series of subsequent chapters that delineate the literature review and methodology to present results and discussions and conclude with recommendations for further investigation.

Chapter 2: Literature Review

2.1 Introduction

Electricity demand predictions are necessary within modern electric grids to assist operators and market players with the optimal scheduling of resources, reserve management, and maintaining electric grid security in increasingly volatile environments [2], [20]. In GB, the rapid transition toward a decarbonised generation portfolio combined with increased electrification of transportation and heating has increased the need for frequent and accurate short-term electricity demand prediction from sub-daily horizons, especially one hour ahead and twenty-four hours ahead [3]. The combination of ambitious climate & energy policies and scenarios being created by National Grid ESO is anticipated to result in significant increases in flexible demand and variable renewable generation [3], [15]. As a result, the complexity and importance of short-term load prediction have greatly increased. This chapter provides a review of existing research regarding the prediction of electricity demand, focusing on the different types of methods, data, and challenges related to GB.

There are three main objectives of this literature review. The first goal is to define the essential terms and functions of short-term load prediction as it relates to power system planning and operations. The second objective is to examine existing techniques that are related to load prediction using time series analysis through traditional statistical methods [16], [23], machine learning [6], [18], [19], deep learning [7], [21], [40], and the various benchmarking techniques and feature-engineering techniques that have been implemented on load and associated energy time-series data. The final goal is to detail some of these research gaps that are motivating this current project, including the limited use of publicly available Great Britain datasets for developing fully reproducible loading prediction models, and no systematic comparison has been performed between LSTM (long-short term memory) [7], [45], [48], XGBoost (extreme gradient boosting) [35], [51] and Prophet [10], [14], [49] loading predictive models for one-hour and twenty-four-hour time horizons. In this review, load prediction is defined as the process of predicting the future consumption of electrical energy based upon the historical consumption data and other explanatory information (e.g., weather conditions, calendar effects) related to the load prediction issue [33]. In the broad scope of load prediction, the term ‘short-term load prediction’ (STLP) typically describes time horizons of a few minutes to several days, and therefore short-term load prediction is different than medium- or long-term prediction used for capital planning and capital investment decisions.

2.2 Conceptual Background of Electricity Demand Prediction

2.2.1 Role in Planning, Operation, and Markets

Electricity demand predictions are utilised to make decisions about how to operate and plan electric supply systems [2]. System operators will use short-term load prediction to schedule how to turn on generators, how many megawatts to produce each hour, how much generating capacity to hold in reserve to cover unexpected outages, and how to manage transfers of

electricity over interconnections [31]. Wind and solar have become significant players on the supply side of electricity supply systems in Great Britain, which makes it very complicated for system operators to accomplish these tasks because now the system operators must deal with higher levels of volatility (and therefore uncertainty) of electricity supplies and a greater risk of having to redirect pumped hydroelectric power when the wind stops blowing or the sun goes down at certain times [28]. A demand prediction is also used by generators and retailers to determine how much electricity to bid into wholesale and balancing markets [17]. As a result, demand prediction will also affect profitability and risk exposure of these market participants, because they impact their ability to hedge their exposure to changes in demand by making accurate demand predictions [17]. Long-term load predictions are also used by utility companies and municipalities to plan for or expand their generation capacity and for reinforcing their networks for reliability and cost efficiency, but this is not a primary focus of this dissertation.

Predictions covering one hour and twenty-four hours fall within the short term for purposes of the present study. One-hour prediction helps match in-week demand-supplies, assists with developing a reserve programme for the next 24 hours, and assists with automated generation controls within the weekly scheduling network where large errors can occur due to time-sensitive opportunities [31]. Twenty-four-hour prediction is a central component of the schedules and clearing process of the next 24 hours because the information in the prediction drives how to discharge economically and maintain the system for the next operating day [31]. A number of transmission network operators within Great Britain use both types of predictions, including 24-hour advance, because the modelling approaches and accuracy level are often different for each.

2.2.2 Demand Drivers and Predicting Horizons

There are various factors that affect the demand of electricity, such as weather, economic and societal status, technology, government policies, and human behaviour [40]. Cooling, heating and lighting needs are highly sensitive to weather variables like temperature, humidity, wind and solar irradiance and are thus the most crucial inputs of short-term load prediction (STLP) models [8], [40], [42]. Both the present and the future electricity consumption are also influenced by the socioeconomic factors such as population growth, income levels, industrial activity and urbanisation [61]. The technological shifts, like the use of heat pumps, electric cars, rooftop solar and smart appliances, change the consumption patterns and provide flexibility and more behind-the-meter generation [62]. Besides this, government policies and regulations also affect demand based on incentives and requirements, such as time-of-use tariffs and energy efficiency or decarbonisation goals [3], [4].

The projection of demand at various time periods has been found to exhibit different demand drivers. Recent load history and short-term weather inputs dominate very short-term predictions compared to day-ahead predictions that rely on daily behavioural patterns based on calendar effects and temperature [8]. In turn, hourly and daily STLP are regarded as primary operational functions of utilities [44]. The operational conditions and inertia drive hourly

prediction, whereas the daily cycles and weather-sensitive demand drive the 24-hour prediction [8]. During these periods, the literature finds three typical challenges, namely volatility and uncertainty, nonlinearity, and shifting demand patterns [8]. Weather variability has led to volatility because renewable generation is uncertain; stochastic consumer behaviour and uncertain prediction with a longer horizon increase prediction uncertainty [20]. Complex interactions between the variables determine nonlinearity, e.g., temperature-load relationships and saturation effects that are not well represented in linear models [1]. The shifting demand patterns are due to long-term trends such as decarbonisation, embedded generation and more electrification, which result in structural breaks and decreased reliability of past calibration [37], [68].

2.3 Predicting Approaches

The development of STLP has transformed older methods of prediction loads in the short term to a multitude of machine learning and deep learning techniques [2]. In the past, operational prediction was based on statistical models, including linear regression, ARIMA and Exponential Smoothing (ETS) [8], [16]. Linear regression models have been used to relate electricity demand to the major drivers, which include temperature and calendar effects, and ARIMA-type models have been used to model the autocorrelation and seasonality based on lagged and moving average terms [16]. Other extensions, like ARIMAX and transfer function models, blend the regression with time-series frameworks and are appreciated as being interpretable and theoretically based [16]. The ETS models, especially those in state-space form, can be updated recursively with respect to the level, trend and seasonality and be further developed to do probabilistic prediction [11]. But their dependence on linear assumptions and the dimensions of features restricts them to nonlinear relationships of a complex nature in contemporary, weather-sensitive and decentralised energy systems [1], [8]. As a result, their performance tends to decline as the demand trends change because of some new data sources on the weather conditions or new technologies that emerge [5].

Machine learning techniques deal with these constraints by allowing nonlinear mapping of a set of inputs and outputs [18]. Neural networks can learn complicated relationships among load, weather and temporal variables without predefined functional forms but are susceptible to overfitting, unstable training and low interpretability [1]. SVM and SVR are kernel-based methods that should be offering nonlinear modelling as well as a convex optimisation guarantee and can be competitive with ensemble methods such as random forests and gradient boosting when tuned appropriately [18], [19]. Gradient boosting algorithms minimise the error residuals, and XGBoost has proven to be a powerful and regularised boosting algorithm to solve the load and energy prediction tasks [6], [13].

The deep learning also improves prediction precision by multi-layer and sequence progressive designs. RNNs and LSTMs explicitly specify the temporal requirements, and LSTMs resolve the vanishing gradient problem and show good performance, including household to system-level electricity demand prediction [7], [31], [47]. Hybrid CNN-LSTM models enhance the accuracy because they extract local temporal features and then model sequences [32]. More

recently, attention mechanisms and transformer models, including Temporal Fusion Transformers, have shown state-of-the-art performance on the most pertinent temporal and contextual information and provide better interpretability [21], [56]. Although transformer architectures are relatively expensive to calculate and have not been widely applied to operations, they form the best benchmark compared to other STLP research [21].

Model type	Algorithm	Advantage	Weakness
Machine learning	XGBoost	Captures nonlinear relationships and handles diverse, incomplete tabular features robustly.	Often needs careful hyperparameter tuning and can be sensitive to configuration choices.
Deep learning	LSTM	Learns long-range temporal dependencies and complex nonlinear patterns in sequential load data.	Requires substantial data and computation, and training can be slow and harder to interpret.
Benchmark	Prophet	Provides an interpretable, decomposable model of trend, seasonality, and holiday effects.	Less effective at modelling complex interactions than LSTM or XGBoost on rich feature sets.
Classical	ARIMA	Simple to implement and effective when relationships are mostly linear and stationary.	Performs poorly with nonlinear, non-stationary demand and many external predictors.

Table 2.1: Comparison of Models: Strengths and Limitations

Table 2.1: This table summarises the main predicting model types considered in section 2.3 and outlines their key advantages and weaknesses for short-term electricity demand prediction.

2.4 Data Sources and Feature Engineering

2.4.1 Data Sources

The fertile ground for energy prediction studies has been a result of quality public datasets available to researchers [43]. For Great Britain, the National Grid ESO's Historic Demand Data portal hosts half-hourly system demand data across many years, which has been used in

numerous studies, both academic and operational [31]. These studies illustrate that GB's half-hourly electrical demand can be derived from publicly available sources, thereby supporting the premise of open datasets being capable of enabling reproducible research [59]. Furthermore, there are many other GB-specific datasets which provide additional context to system operations, including but not limited to carbon intensity data, fuel mix data, and operational statistics available via ESO's Digitalisation Strategy [38]. The Elexon BMRS FUELHH dataset is a half-hourly breakdown of generation strategies broken down by fuel type which will enable researchers to assess how fuel mix and embedded generation affect net demand and overall system operations [30]. The ESPENI dataset combines electricity and weather data across several countries, including GB, providing an invaluable opportunity for researchers to analyse the differences between countries and develop comparisons as well as benchmarking methodologies [28]. Exogenous inputs for load forecasting models are often derived from Global Reanalysis Products (GRPs) and technology-based (open) API services (OpenMeteo) to access historical medium- to high-resolution weather data, such as temperature, relative humidity, precipitation, solar radiation, and wind speed/direction. The OpenMeteo dataset in this study contains queried data from one location that is representative of the region of Great Britain (latitude: 54, longitude: -2). This coordinate was selected using publicly available geospatial reference data to represent the centre of the population and energy network in the GB Power System. The selected location will serve as a valid proxy for weather conditions throughout the GB Power Network and facilitate consistent alignment of the created time series with national-level demand and generation datasets while minimising the feature set for machine learning models, UK bank holiday calendars, solar generation estimates, and other open-source datasets [24], [26], [27], [29]. Further enhance the available space of features for data-driven modelling.

Dataset	Provider / Coverage	Key Variables	Research Role
NESO Historic Demand	National Grid ESO, GB, half-hourly	System demand (MW)	Main target series for 1-hour and 24-hour STLF
BMRS FUELHH	Elexon, GB, half-hourly by fuel	Generation by fuel type, embedded generation	Operational context / optional features
ESPENI	Zenodo half-hourly GB	Load, generation	Cross-checked load and weather; additional context

Dataset	Provider / Coverage	Key Variables	Research Role
Open-Meteo	Open weather APIs	Temperature, wind, solar irradiance, etc.	Meteorological predictors

Table 2.2: Summary of Datasets and Their Roles in Short-Term Demand Prediction

Table 2.2: Lists the main public datasets relevant to GB short-term demand prediction and shows how each contributes to the feature space used later in the thesis.

2.4.2 Feature Engineering Strategies

Feature engineering converts data in its raw format (e.g., time series data) into predictors of demand that reflect the underlying drivers of demand [2]. Some predictors will fall into the following categories: Meteorological, temporal, operational, any associated interaction terms or lagged variables. Meteorological features are typically composed of temperature (or apparent temperature), wind speed, solar irradiance, humidity, etc.; many studies perform transformations on the meteorological features (for example, heating degree days, cooling degree days, etc.) or otherwise apply some type of non-linear function to correctly capture the threshold effects [40], [42]. Common examples of temporal features include HourOfDay, DayOfWeek, Month, Season, and indicators for weekends and holidays (which exhibit some degree of predictability about consumer behaviour) [26]. Operational features can capture fuel-mix shares, net imports, renewable penetration, and carbon intensity, which all impact system conditions and are generally believed to be correlated with demand [30], [58]. UK load prediction (among others) can be improved by incorporating comprehensive net imports datasets into the load-predicting model, as these additional variables (related to system conditions) are more likely to provide improved predictive performance [58]. Finally, lagged load values and moving averages still provide the basis for many load-predicting models, as these features capture autocorrelation and local temporal context and are especially valuable in making very short-term predictions [16]. Recent trends emphasise reproducibility as well as adherence to the FAIR Principles (Findable, Accessible, Interoperable and Reusable); therefore, researchers are encouraged to make datasets, code, and the complete feature engineering pipeline publicly available [25]. By integrating the NESO Historic Demand, BMRS FUELHH, ESPENI, and the open weather data sets available for Great Britain (GB), it becomes possible to develop comprehensive, integrated data resources which can be used to effectively compare various models on multiple predicting horizons [28], [29], [30], [31].

2.5 Research Gaps in Existing Literature

Although there is a lot of research done on electricity demand prediction, some significant gaps are still present, especially concerning Great Britain (GB) and the area of this dissertation [44]. To begin with, it is difficult to find articles that use publicly available data and open-source models, and many of the most successful methods are based on proprietary or underdocumented data [43]. Transparency in data processing and feature engineering is usually inaccessible even in the case of using public data [43]. Moreover, there is scanty research that has incorporated a combination of several open GB datasets, such as NESO Historic Demand, BMRS FUELHH, ESPENI data and OpenWeather API, into a complete and reproducible data processing chain [28], [29], [30], [31].

Another gap is related to the comparative analysis of forecasting models. Despite the widespread use of LSTM, XGBoost and Prophet in various forecasting tasks, including load, heat demand and emissions prediction [34], [36], the existing literature tends to do comparisons of these models on various datasets, time horizons and evaluation methods, which do not allow drawing significant conclusions regarding the comparative performance of the models when short-term electricity demand forecasting is required in GB [34]. The comparisons in the past have often been on irrelevant areas like web traffic or even building-level consumption instead of national demand of electricity using publicly accessible system-wide data [34]. Likewise, the use of machine learning and deep learning models in a controlled experimental study with the same data set and conditions across GB-focused research has seldom been done [50], [61].

The other important research gap is associated with feature richness and model interpretability. Most of the literature, though, incorporates simple calendar and temperature; few incorporate combined calendar, weather and holiday, weekday/weekend and operational variables of fuel mix and electricity imports in a single framework [40]. Besides, the interpretability methods, including SHAP values of tree-based models and attention visualisation of sequence models, are currently being integrated in STLP studies [55], [56], [57] only recently. As a result, research that can compare the predictive power and feature effect of one-hour-ahead and twenty-four-hour-ahead demand prediction in GB is required [44]. Current GB studies are mostly around the long-term demand, or price prediction, operational case studies, or individual end-use, including households or buildings [44], [62], [63], [64], [65], [70], and lack a system-level short-term forecasting that uses publicly available data [43].

The dissertation fills such gaps by building an integrated public dataset based upon NESO Historic Demand, BMRS FUELHH, ESPENI and Open-Meteo data; engineering extensive temporal and weather characteristics; and comparing systematically LSTM, XGBoost and Prophet with established benchmark models at one-hour-ahead and twenty-four-hour-ahead forecasts [28], [29], [30], [31].

Research Gap in Literature	Implication
Limited GB studies using only public system-level and weather data	Builds dataset exclusively from open sources (NESO, BMRS, ESPENI, Open-Meteo, UK holidays)
Few systematic comparisons of LSTM, XGBoost, and Prophet on the same GB data	Implements all three models on a single integrated dataset and identical horizons
Limited exploration of rich feature sets (weather, calendar, holidays, weekends)	Engineers and evaluates a comprehensive feature space for 1-hour and 24-hour prediction
Limited interpretability analysis for STLF	Applies model-specific explainability tools (e.g., SHAP, component plots)

Table 2.3: Identified Research Gaps and Corresponding Contributions

Table 2.3: Links the main research gaps from Section 2.5 with the design of the empirical work, clarifying how the thesis contributes to GB short-term electricity demand prediction literature.

2.6 Conceptual framework

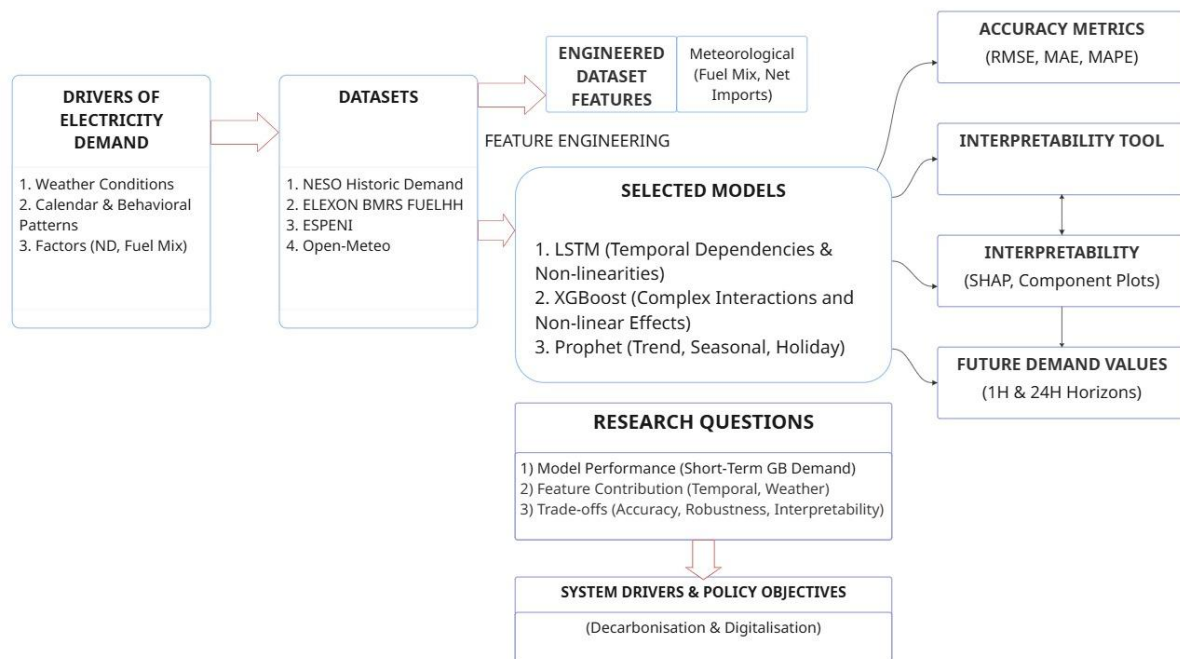


Figure 2.1: Conceptual Framework of the Project

This project describes how to model weather, calendar and behavioural patterns to predict electricity demand in Great Britain. These drivers include weather conditions, calendar events (e.g., holidays), social influences (e.g., the effect of school vacations on residential electricity use), and fuel type used in generation (i.e., mix of coal, gas and renewables). Data for these drivers comes from historic electricity demand, generation and weather service datasets. Next, we convert the raw data into temporal, meteorological and operational features, which we then use as inputs into three different predicting models (LSTM, XGBoost, and Prophet) to generate hourly and 24-hour-ahead demand predictions. Predictions generated from each model capture their own characteristics (e.g., time-series trends, seasonality, non-linear effects) and will be evaluated using RMSE, MAE and MAPE as indices of accuracy and then further interpreted using SHAP and the component plots of the models to identify the features that influence the prediction results. Finally, the entire framework directly connects to our research interests regarding how well these models perform, which features most impact the forecast and how to balance prediction accuracy, robustness and interpretability in the ongoing transition to a digital, decarbonised GB electricity economy.

2.7 Justification for Model Selection

The dissertation focuses on LSTM, XGBoost, and Prophet because together they cover the main ways of thinking about short-term load prediction in GB. LSTM is chosen as the deep-learning workhorse: it is built for sequential data, can learn complex daily and weekly patterns, and naturally combines weather and calendar inputs in a single model, which is important for a changing, weather-sensitive system. XGBoost fills a different role as a powerful, tree-based ensemble for tabular data, using gradient boosting and regularisation to handle many engineered features and giving useful feature-importance and SHAP explanations for meteorological and calendar effects. Prophet adds a simpler but highly interpretable baseline that decomposes load into trend, multiple seasonality, and holiday impacts, which fits well with GB's strong weekend and bank-holiday patterns and is easy to deploy in practice. By applying all three models to the same public GB dataset for one-hour-ahead and twenty-four-hour-ahead prediction, the study can compare their accuracy, robustness, and interpretability in a fair and transparent way.

2.8 Chapter Summary

The chapter provides an overview of the history of short-term electricity demand prediction, especially in Great Britain. It begins by explaining the need for the accuracy of the hourly and daily demand predictions of the functioning of the power systems and electricity markets and outlines the key variables of demand variability, including weather conditions, technological change, and government or regulatory policies. The chapter continues briefly and describes the three most popular forecasting methods (statistical, machine learning and deep learning) and their strengths and weaknesses available to address the more weather-sensitive dynamic demand patterns. The chapter offers a description of how the publicly available Great Britain datasets, including NESO Historic Demand, BMRS FUELHH and ESPENI data, as well as open-source weather APIs, can be integrated with engineered temporal, weather and operational characteristics to make short-term demand prediction reproducible. It next identifies the gap in research, like the number of studies that have comparatively considered LSTM, XGBoost and Prophet on the same public datasets and the lack of studies that have examined rich features and interpretability examined in short-term load prediction. Finally, the chapter presents one conceptual framework and justifies the selection of LSTM, XGBoost and Prophet as complementary models to each other to give a foundation of empirical analysis in the remaining chapters.

Chapter 3: Methodology

3.1 Introduction and Research Design

The transition of Great Britain towards a carbon-neutral economy has introduced an unprecedented period of change within the national electricity distribution network (grid system). Between 2020 and 2023 the amount of offshore wind capacity being produced increased significantly. Simultaneously there was an exponential growth in the adoption rate of distributed solar PV systems and battery storage technologies within the electricity grid system. The increasing use of Electric Vehicles (EVs) for transportation, heat pumps for heating purposes, and the electrification of industrial activities has created unprecedented levels of volatility in electricity demand [3]. Such volatility has resulted from the combination of variable weather conditions, changes in consumer behaviour and the changing generation patterns of renewable energy generators. Linear predictive models (e.g., ARIMA), which were historically effective in predicting loads associated with conventional coal- and gas-fired generation, are now unable to adequately address the rapidly evolving non-linear demand patterns associated with renewable energy generation [5]. For instance, in instances of high levels of wind-generated electricity and significant drops in demand, linear models will generally not provide an appropriate prediction of corresponding peak loads occurring during intraday timeframes [22].

This chapter presents a comprehensive and repeatable methodology to predict short-term electrical demand in Great Britain (GB) over two timeframes: shortly (1 hr ahead) and further (24 hrs ahead). This includes meeting the research objectives outlined in Chapter 1 by integrating 4 separate GB datasets into a single 30-min interval data set, developing features which capture the temporal, meteorological, renewable energy and lag effect of GB electrical demand, employing 3 different prediction techniques (Prophet, XGBoost and LSTM with an attention mechanism), evaluating the accuracy of each technique using Root Mean Squared Error (RMSE), Median Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE), providing an explanation of prediction performance via SHAP values and attention mechanisms and providing source code and results in an open source format to facilitate additional research and operational application.

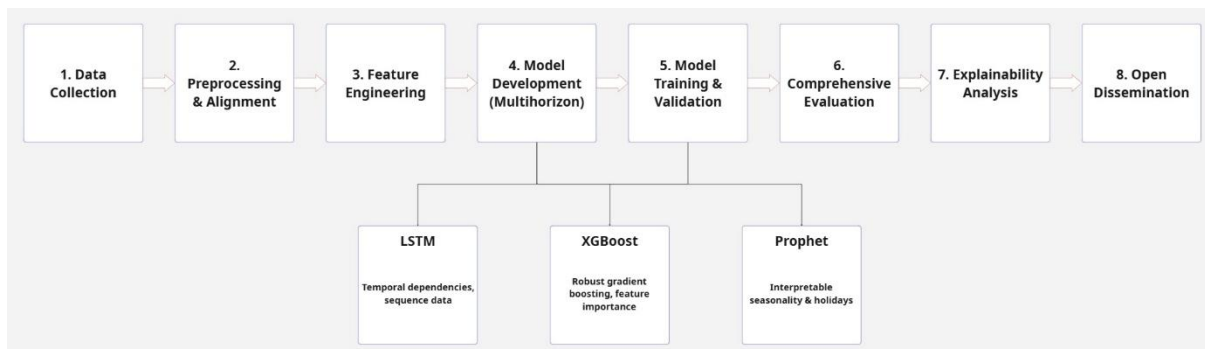


Figure 3.1: Quantitative Experimental Design Workflow

A structured approach informed by quantitative experimental design is used to develop the methodology. First, comprehensive collections of data are extracted, cleaned, and prepared using a structured feature engineering process to incorporate temporal, weather, and generated renewable resources. The primary pipeline within this research consists of training and validating three different modelling architectures: (1) Long Short Term Memory (LSTM), which is capable of learning from sequencing patterns or time series data; (2) an XGBoost for robustly performing predictive modelling with predictable features and explainability; and (3) Prophet, to model seasonal and calendar variations or seasonality. The ending steps ensure that the entire study maintains research rigour through the inclusion of a comprehensive list of evaluation metrics, a clear methodology for explainability analysis using SHAP and visual representation of the attention scores for the selected model architecture, as well as an openly distributed code set.

3.2 Data Sources and Preprocessing

3.2.1 Overview of Data Sources

To conduct this research, four separate subsets of data were collected. Each dataset included different variables, offering a unique perspective into how the GB electricity system operated over time. The selected datasets were useful due to their granularity, importance, and the fact that they provide access to customers’ data from January 1, 2020, to December 31, 2023 [28], [29], [30], [31]. Across all datasets, there are a total of 70K+ half-hourly and hourly measurements that were concatenated programmatically into a master file named NEFW_DT.csv using the "pandas" package in Python. The master file retains the native half-hourly and hourly level of detail necessary for capturing generator output and demands that occur during the day.

Dataset	Source	Coverage	Main Purpose (summary)	Brief Description (summary)
NESO Historic Demand Data	NESO Historic Demand Data Portal	Half-hourly GB electricity demand, 2020-01-01 to 2023-12-31	Baseline system load and non-generation components (e.g., interconnectors, pumping) for training and analysis	High-fidelity CSVs of half-hourly NETS flows, including national demand, transmission system demand, embedded wind/solar, interconnector flows, and storage pumping.

Dataset	Source	Coverage	Main Purpose (summary)	Brief Description (summary)
BMRS FUELHH Dataset	Elexon BMRS FUELHH Portal	Half-hourly GB generation by fuel type, 2020-01-01 to 2023-12-31	Derive net demand and analyse variability of renewable and conventional generation.	Granular 30-minute time series of generation by fuel (CCGT, nuclear, biomass, coal, hydro, imports), used to study mix, ramping, and residual demand.
ESPENI Dataset	ESPENI on the Zenodo Portal	Half-hourly GB load, generation, and weather, 2020-01-01 to 2023-12-31	Pre-processed bridge dataset aligning demand, generation, and weather for cleaner modelling and weather-demand validation	Composite time series harmonising ESO and Elexon data, correcting known issues and synchronising timestamps to align grid variables with meteorological conditions.
Open-Meteo Historical Weather Data	Open-Meteo API	Hourly GB weather at (54, -2), 2020-01-01 to 2023-12-31	Quantify weather sensitivity and reconstruct renewable potential (wind/solar) for prediction and performance assessment.	Reanalysis-based hourly weather (temperature, wind, precipitation, humidity, cloud cover, shortwave radiation) for the GB centroid, used as exogenous forecasting input.

Table 3.1: Overview of Datasets, Sources, and Roles in the Study

Table 3.1: The above table provides an overview summarising the four types of data that were publicly accessible and had been used in the study undertaken between January 1, 2020, and December 31, 2023: These included information about the GB electricity usage/demand, electricity production/generation and localised meteorological (weather) data; when combined with one another, these datasets will enable researchers to build a single data set that is time-

matched into alignment and therefore compatible with developing sophisticated machine learning algorithms.

3.2.2 Data Extraction, Cleaning, and Time Alignment

To make a half-hourly time series from multiple raw datasets for the use of a machine-learning model, data extraction, cleaning, and alignment were completed. The three datasets, NESO, ESPENI, and FUELHH, provided half-hourly recorded measurements and thus were merged directly from their similar half-hour timestamps and created the primary spine of the master file. When adding the Open-Meteo weather dataset, since it is available at an hourly resolution, the hourly weather record attributes were merged into the corresponding pairings of half-hourly data by using the date field and hours of the weather record. For those cases where an hourly weather/temperature did not exist for a specific half-hourly time slot, they were replaced by (0) as a neutral placeholder that would not have affected the merging of valid generation or demand rows. In this manner, the combination of three half-hourly datasets and one hourly dataset resulted in a singular, full and aligned master table containing 70K+ rows that could then be used in the downstream modelling phases.

Process	Methods Applied	Purpose	Implementation
Extraction	Python scripts via URLs/APIs	Automated, reproducible retrieval	Extract_Datasets.py → Merge_Datasets.py Clear timestamped filenames
Datetime Standardisation	pd.to_datetime(utc=True)	Eliminate time zone ambiguities.	All datasets converted to consistent UTC datetime
Missing Value Imputation	ffill() for short gaps Flags for long exceptional periods	Preserve temporal continuity.	extreme weather flagged, not removed
Outlier Detection	Z-score > 3 thresholds	Identify real events vs. errors.	Flagged (not removed) to maintain integrity
Categorical Conversion	Binary flags (is_weekend, BankHoliday_flag)	Machine learning compatibility	DayOfWeek, Month, and behavioural shifts encoded

Table 3.2: Data Preprocessing and Feature Engineering Workflow

Table 3.2: The preprocessing steps applied to the raw NESO, BMRS, ESPENI and Open-Meteo datasets that were applied before modelling are summarised in the table. The main purpose of each of these steps is to provide an automated and reproducible way to retrieve data on a consistent time scale and to produce a format that is suitable for machine learning while maintaining the integrity of the original operational events.

3.2.3 Merging Datasets

The cleaned and aligned datasets were combined using the common datetime_utc field and pandas outer join capability in order to create NEFW_DT.csv, which contains about 70K+ rows after eliminating lost periods within the dataset. Model-specific scaling was performed on the merged datasets after combining. LSTM used MinMaxScaler for scaling in order to maintain gradient stability; raw scales were used to provide information to the XGBoost model, since tree-based models are invariant to monotonic transformations, and Prophet may normalise or standardise the regressors without explicit scaling functions. The final dataset after considering the lag and rolling feature construction and after potentially removing the “NaN” or inaccessible information resulted in above 70K+ usable rows and approximately ~100+ features.

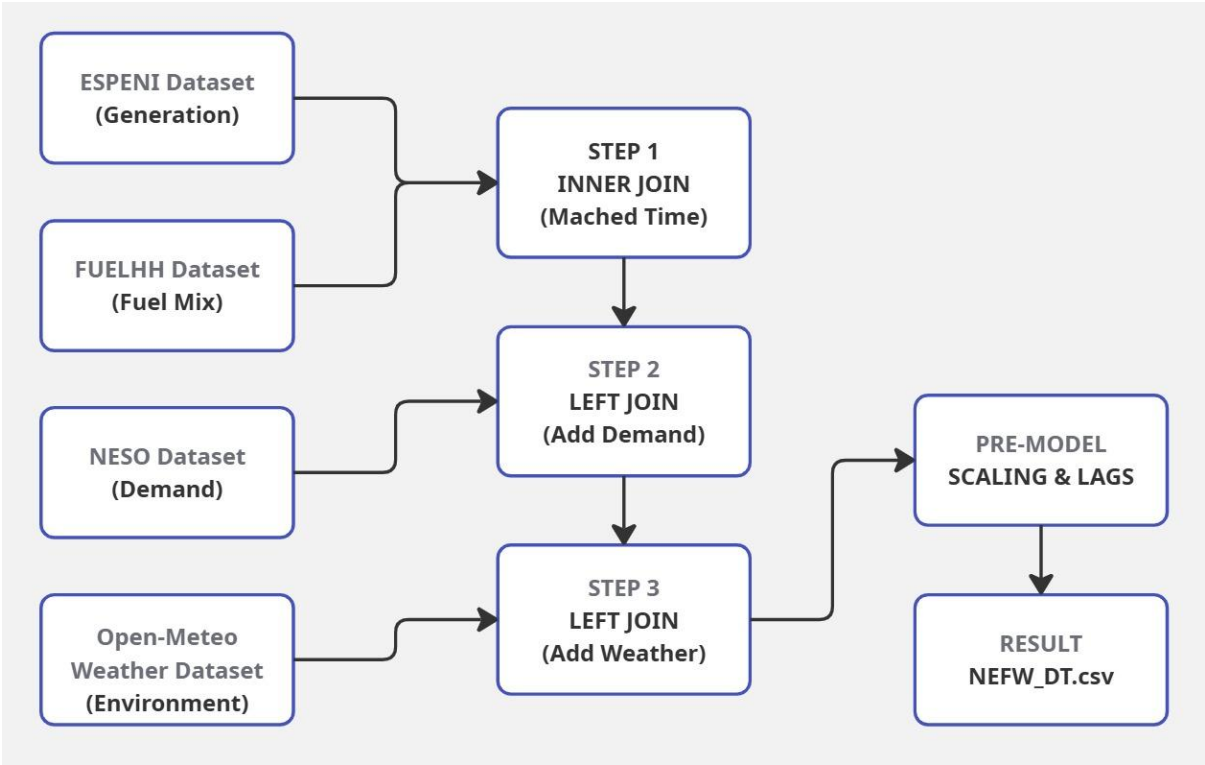


Figure 3.2: Workflow for Merging Datasets.

In the figure above is a diagram of an integration pipeline that contains multiple levels of integration to bring four datasets (Generation, Fuel Mix, Demand, and Weather) together into one time series. The first step is to create a central "core" dataset using an inner join that combines the generation and fuel mix datasets. Then, the next two datasets (Demand and Weather) are joined using left joins to ensure the greatest possible retention of all recorded instances, even when there are connectivity issues between the two or more datasets. After combining the datasets, all data is cleaned and transformed to create the High-Dimensional NEFW_DT.csv form designed for model training that is the result of all previous processing steps.

3.2.4 Exploratory Data Analysis

An exploratory data analysis of the four public datasets was conducted prior to combining them into the NEFW_DT master file to gain a clear understanding of the characteristics of the data structure, coverage, and quality of the data. Analyses were conducted on the NESO demand data, BMRS FUELHH fuel mix data, ESPENI System Indicators, and Open-Meteo weather series. For example, the dataset from January 2020 to December 2023 was examined to ensure complete continuous coverage and identify gaps and extreme values and the evolution of structural changes over time. The datetimes were all standardised to UTC to eliminate time zone confusion, and all missing data for shorter periods of time (up to 90 mins) were filled using forward fill, while missing data for more than 90 minutes due to severe weather events were recorded as exceptional periods rather than omitted because they present the opportunity to record real system stress events. A Z-score cut-off value of 3 was used to identify outliers, but all observations that were anomalous but plausible were retained to preserve the underlying dynamic nature of demand. The process of analysing temporal patterns in the data led to the creation of calendar-related features, such as hour of day, day of week and binary indicators for weekends and UK bank holidays [26], to capture daily and weekly recurring profiles [69].

3.3 Feature Engineering and Selection

By applying the feature engineering techniques to transform all half-hourly and hourly data collected from NESO, BMRS, ESPENI, and Open-Meteo into 100+ predictive signals used for GB electricity demand, Add_features.py (Feature Engineering) was able to produce a set of features via lagged autocorrelations and configuration of cyclic patterns using sine/cosine transformations, along with weather-related variables and fuel mix variables [40]. Using this approach, we were able to integrate nonlinear interdependencies between the variables that are typically lost through the use of traditional ARIMA models, allowing us to address some very important questions in regard to renewable heavy grids. The event used to generate rolling windows was also used to identify "NaN" values, leading to the creation of a dataset consisting of 70,032 observations of high-quality data. Therefore, this approach allowed us to develop a more accurate dataset which allows NESO to balance its operations while also providing a

bridge for high-performance machine learning to learn from the large amount of variance present in the data collected over time.

Feature Group	Example Features	Main Role in Model
Temporal & Calendar	hour_sin, hour_cos, dayofweek_sin/cos, month_sin/cos, dayofyear, is_weekend_flag, BankHoliday_flag	Capture weekly, annual patterns and holiday effects.
Weather	temperature_2m, apparent_temperature, wind_speed_10m, shortwave_radiation, humidity, cloud_cover, precipitation	Model heating/cooling demand and renewable potential
Fuel Mix & Grid	FUEL_WIND, FUEL_SOLAR, FUEL_CCGT, FUEL_NUCLEAR, interconnector flows, embedded wind/solar	Capture supply-side context and net demand effects.
Lag & Rolling Stats	lag_1h, lag_6h, lag_24h, lag_336, roll_mean_3h, roll_mean_24h	Encode autocorrelation and smooth short-term noise.

Table 3.3: Categorisation of Engineered Features for Demand Prediction Models

Table 3.3: The information in the table highlights the key categories of engineered characteristics that can be used to predict load with the forecasting models. The combination of temporal, weather, fuel composition, and lagged variables will allow for the forecasting models to account for demand seasonality, weather sensitivity, supply-side issues, and autocorrelation that are needed for an accurate prediction of GB load on a short-term basis.

3.3.1 Temporal and Calendar Features

Both calendar and time features (from the control in our remark to consider many of the time features) represent and describe predictably repeating and continuously recurring cycles that form a basis for electricity demand. Using any one or more of the time's features, we can appreciate when that event will occur. The use of these features allows for the cyclical nature of the electricity usage to be predicted as follows: There will be two cycles within each day (the two peaks of demand), as well as various cycles at other times; and we can also identify the cyclical nature of demand for both heating and cooling. The cyclical transformation of sine

and cosine functions also provides effective mathematical descriptions for preventing the misleading discontinuities inherent in raw ordinal characterisations and reinforces the continuous relationship between temporally adjacent periods, so that from the perspective of a machine learning model there will be very little or no confusion with respect to the value of time from 23:00 to 00:00 (and conversely). Examples of these types of variables are binary flag variables that are representative of the sharp behavioural discontinuities between weekends and common holiday days in the UK (as provided through official Gov.uk sources) [26], where use of industrial and commercial electric supply drops significantly while domestic uses may significantly increase as a result of domestic changes in consumer behaviour. Establishing and using a time variable together with the various features of temperature, supply-related (weather-related or otherwise), and therefore predictive elements, it helps create a more complete predictive model.

3.3.2 Weather, Fuel Mix, and Lag Features

To represent the complex interdependencies of Great Britain's Renewable Integrated Grid, the model comprises three sets of variables: weather data (from Open-Meteo), fuel mix data (from BMRS) and autocorrelation data, as critical exogenous/timing factors. Weather Data (Temperature_2M & Wind_Speed_10M) are direct measurements of how sensitive the grid is to changes in weather and provide a measure of the amount of renewable generation capability available. Autocorrelation data are used to quantify short-term persistence and seasonal patterns that characterise demand at different time horizons; data from earlier in the week would therefore be used to predict demand on the coming weekend [22]. The model also uses BMRS fuel-mix per cent and interconnector flows to give supply-side context to the extent to which embedded renewable generation (i.e., "invisible" renewables) is suppressing visible (i.e., "transmission") demand. Finally, the model applies smoothing processes to reduce the impact of high-frequency noise in order to highlight any underlying trend; this is done through the application of rolling statistical windows, which generates new feature variables that will enable the model to effectively generalise beyond simple persistence baselines.

3.4 Models, Training, and Validation

An ensemble of three complementary, reliable forecasting methodologies provides a production-level multi-horizon prediction capability with respect to $H=2$ (the hour for the next settlement, i.e., prediction for 60 minutes) and $H=48$ (the full trading day, i.e., a prediction of 24 hours) using Jupyter notebook-based implementations with dedicated documentation. The unique aspect of this sophisticated tri-ensemble methodology is that each of the three methods produce seasonally oriented predictions which allow for statistical properties of seasonality and holidays to be incorporated into the prediction using Prophet's interpretability attributes [10]; combine XGBoost's empirically superior ability to predict with tabular feature sets having a lot of structured data with production service-level explainability as provided by SHAP; and utilize LSTM with attention to model complex time sequence relationships over extended

periods of time reflecting the behaviour of intermittent renewable generation driven by weather patterns on a rapidly decarbonising electricity network such as the electricity network in Great Britain [56]. Further, the implementation incorporates strict chronological validation splits which mirror what would exist in real-world situations where actual operational prediction would be conducted; furthermore, all random seeds for validation splits are permanently fixed, thus allowing for the scientific evaluation of various statistical, ensemble tree-based and deep time series methodologies across a diverse range of conditions to be fully reproducible [14], [66].

3.4.1 Modelling Strategy

Every model type in this ensemble strategy is based on specific hurts and strengths with regard to the GB Electricity Sector demand forecasts [46]. First, the prophet's ability to provide seasonality components, holiday effects for all 34 UK public holidays and trend estimation for each of the half-hourly (48-period daily seasonality) requirements are clearly benefits of the additive decomposition framework method used in this approach. Next, XGBoost provides the best-performing empirical estimates on our complete engineered tabular features and provides the industry-standard SHAP-based explainable model required for stakeholders to have confidence in the predictions made. Finally, our LSTM+Attention architecture is characteristic of its extended period of temporal dependency and should be used for growing renewable-based volatility patterns that exist with respect to modern grid operations [7].

3.4.2 Prophet Model

Prophet, developed by Facebook, is a time series regression model that uses a mathematical formula to decompose the electrical load into parts that are easier to understand, such as its long-term trend and daily (48), weekly, and yearly components [10]. It has components for known holiday effects and other input parameters that help in making predictions of future electrical loads. This allows for a transparent way to validate production predictions. Prophet incorporates all UK holiday lists from 2020 to 2023 provided by Gov.UK [26], allowing it to be applied in practice. In addition, because Prophet supports 30-minute (settlement period) electrical load data, its ability to model daily load patterns through 48 periods allows for accurate modelling of time series patterns across a 70,000+ settlement load dataset; however, care has been taken to use appropriate values for the holiday and seasonality prior scales to reduce overfitting due to the limited number of holiday observations relative to the number of settlement periods [11].

3.4.3 XGBoost Model

The XGBoost algorithm uses a wrapper from scikit-learn called MultiOutputRegressor. This wrapper uses the XGBRegressor to create ensembles of H many trees through gradient boosting at each of the prediction horizons (i.e., if $H=2$, then there will be 2 trees as part of that

configuration, and if $H=48$, there will be 48 trees). The benefits provided by this algorithm are both parallelised computation and empirical robustness to the noise inherent in the real-world data collected in practice, along with a mathematical basis for addressing the issue of how complex feature interactions occur utilising optimally constructed energy-domain hyperparameters [35]. Bias-variance trade-offs are optimised in configuration to provide a small learning rate (0.05) during training, which allows for stable results across an extensive 1000 iterations of trained ensembles, while controlling both tree complexity ($\text{max_depth}=6$) and L1/L2 regularisation ($\text{reg_alpha}=5$; $\text{reg_lambda}=1$), as well as using stochastic subsampling to improve generalisation performance (i.e., subsampling of 80% of patients and 90% of features utilised within trees). When monitored by evaluating the validation RMSE, early stopping at validation prevents any overfitting on the chronologically split validation data [52].

3.4.4 LSTM with Attention Model

The multiple-layer LSTM+Attention model for deep learning takes in 336 half-hourly timesteps that make up a complete 7-day history (a 7-day lookback window) and uses 50+ distinct features chosen based on their relative importance to the dependent variable [7]. Combining multiple stacked layers of LSTM with a custom temporal attention method allows the model to compute relevance weights dynamically for each time period (i.e., historical demand for this week relative to past demand), which enables the model to allocate more computational resources to the most relevant time periods for short-term ($H=2$ forecast horizon) vs. long-term ($H=48$ forecast horizon) predictions. Use of Dropout (0.2) as a regularisation technique prevents model overfitting, while the context vectors derived from attention, concatenated with the LSTM's last hidden states, create a richer representation of the input into the downstream (dense) layer of the model. The model uses $\text{MinMaxScaler}(0,1)$ for numerical stabilisation of the gradient during training on 47538 parameters in the $H=2$ configuration and 50528 parameters in the $H=48$ configuration [47].

3.4.5 Training Procedure

In order to create a valid and representative model to predict how well a model will work in practice, a method was developed using 70% of the data collected between Jan 2020 and Oct 2022 to train the models; 15% of the data collected from Oct 2022 to May 2023 was used as a period to conduct validation by selecting hyperparameters and stopping during early periods of the training process. Finally, the last 15% of the data collected from May to Dec 2023 would be used to evaluate the final model and therefore provide the best prediction without any of the bias associated with operational performance prediction. In order to ensure that results could be repeated using all available libraries, the use of predetermined random seeds (i.e., $\text{np.random.seed}(42)$, $\text{tf.random.set_seed}(42)$, $\text{random_state}=42$) was essential to enabling reproducibility. With respect to computational efficiency, the time required to train the models varied based upon the selected architecture; XGBoost completed a parallelised version of the training within 2 to 5 minutes; LSTM required 30 minutes of training utilising CPU

acceleration to enable the processing of the sequence data; and Prophet rapidly returned performance of < 5 minutes, with all of the architectures providing sub-second inference latency, which is a requirement for real-time operational use in the balancing mechanisms of a National Energy System Operator [53].

3.5 Tools and Reproducibility

Python 3.12, Jupyter notebooks, and open-source libraries (pandas/numpy: Data; scikit-learn: Preprocessing; Prophet: Stats; XGBoost: Trees; TensorFlow/Keras: LSTM + Attention; SHAP: Explainability; Matplotlib/Seaborn: Visuals) are used in this pipeline. Reproducibility was assured through fixed seeds (42) and versioned datasets.

3.6 Chapter Summary

This chapter describes the development of a new, consolidated, cleaned, and data-rich data set from four publicly available data sets (NESO, BMRS, ESPENI, and Open-Meteo), the creation and training of three complementary forecasts (Prophet, XGBoost, and LSTM with attention), and the evaluation of these forecasts with metrics and explainability tools (RMSE, MAE, MAPE, SHAP, and attention visualisation). This method is based around a modular, open, and totally replicable approach, mirroring the way forecasting works for today's weather-based societies and nations who are moving to net zero emissions.

In Chapter 4 (Results), the empirical results will be presented, including the comparison of Prophet, XGBoost, and LSTM's performances at the two-prediction horizon (1 hour and 24 hours), along with revealing the information the models contain regarding what drives the short-term demand for electricity in Great Britain.

Chapter 4: Results

4.1 Introduction

The chapter contains a critical review of three predicting approaches, Prophet, XGBoost, and LSTM with attention, applied to predicting short-term Great Britain electricity demand: H1 (one hour ahead, with 2 additional 30-minute periods) and H24 (24 hours ahead, with 48 more half-hour periods) predicting. All the models are trained and assessed on the same half-hourly and hourly data between January 2020 and December 2023, with a chronological split of 70 per cent training, 15 per cent validation and 15 per cent testing to avoid time-leaking issues. The models are evaluated on three standard error measures: root mean squared error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE). It will begin its analysis with a global performance comparison of models and horizons, then dwell more on the results of one-hour-ahead performance and twenty-four-hour-ahead performance, and finally, the results of explainability will be provided using SHAP values of XGBoost and attention weights of LSTM.

4.2 Model Implementation

XGBoost, Prophet and LSTM were tested on the same half-hourly and hourly Great Britain demand data (NEFW_DT.csv), where a chronological divide is set to include training, validation and test sets 70/15/15 to avoid information leakage. Two models are used (H1 (next hour, i.e., next two half hours) and H24 (next 24 hours, i.e., next 48 half hours)) that forecast the total demand (ELEC_POWER_TOTAL_ESPENI_MW).

4.2.1 Prophet Setup and Hyperparameters

Prophet was defined on a univariate frame with the column ds being the timestamps and target demand in y and was set up by converting the NGEM UTC timestamp to a time zone naive datetime and remapping the column of demand accordingly. The exogenous regressors were automatically selected by computing the absolute correlation of all possible weather, fuel, interconnector flow and aggregate power variables with the target and retaining the best few (e.g., synchronous and fossil generation, coal and hydro) that were then added to the model using add_regressor.

```
# 1. LOAD AND PREPARE DATA
df = pd.read_csv('NEFW_DT.csv')

# Convert to datetime and STRIP TIMEZONE (Prophet requirement)
df['ds'] = pd.to_datetime(df['ELEC_NGEM_startTime_utc_datetime'], utc=True).dt.tz_localize(None)
df['y'] = df['ELEC_POWER_TOTAL_ESPENI_MW']
```

Figure 4.1: Data Preparation for Prophet

The final Prophet specification was able to have daily, weekly, and annual seasonality; UK public holidays using `add_country_holidays(UK)`; and tuned prior scales (changepoint prior scale 0.1, seasonality prior scale 10, holidays prior scale 10) and was afterwards fitted on the training subset itself. The H1 and H24 predictions were generated by generating short future prediction data structures by test set data and then calling the model. Predict and compute RMSE, MAE, and MAPE by a simple custom function which uses a safe indexing technique of both the actual and the predicted series by converting them to NumPy arrays.

```
# 4. PROPHET CONFIGURATION & TRAINING
# Daily, Weekly, and Yearly seasonality enabled with manual tuning as requested.
model = Prophet(
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=True,
    changepoint_prior_scale=0.1,
    seasonality_prior_scale=10.0,
    holidays_prior_scale=10.0
)

# Add UK Holidays
model.add_country_holidays(country_name='UK')

# Add Selected External Regressors
for regressor in selected_regressors:
    model.add_regressor(regressor)

# Fit the model on the Training Set
model.fit(train_df[['ds', 'y'] + selected_regressors])
```

Figure 4.2: Prophet Configuration

4.2.2 XGBoost Pipeline, Feature Engineering, and Training.

All the features are used by the XGBoost pipeline, which first incurs much preprocessing: the categorical calendar features (e.g., day of week, month, bank holiday flag, etc.) are numericised, the rest of the non-numeric data are numericised and cleaned, infinities and missing values are forward /backfilled, and non-useful all-NaN variables are pruned. A filtered collection of temporals, weather, fuel mix, interconnectors flow and embedded generation and aggregate system sums are then assembled into the final collection of features, and all features are scaled by MinMaxScaler to the [0,1] range.

```

def engineer_features_ultimate(path):
    print("Step 1: Loading data...")
    df = pd.read_csv(path)

    # Standardize Datetime
    df['datetime'] = pd.to_datetime(df['ELEC_NGEM_startTime_utc_datetime'])
    df = df.sort_values('datetime').set_index('datetime')

    print("Step 2: Creating Time & Seasonality features...")
    # Cyclical Time (Better for XGBoost than raw numbers)
    df['hour_sin'] = np.sin(2 * np.pi * df.index.hour/23.0)
    df['hour_cos'] = np.cos(2 * np.pi * df.index.hour/23.0)
    df['month_sin'] = np.sin(2 * np.pi * df.index.month/12.0)
    df['month_cos'] = np.cos(2 * np.pi * df.index.month/12.0)

    # Binary Holiday flag
    df['is_bank_holiday'] = (df['BankHoliday'] != 'No').astype(int)

    print("Step 3: Creating Lags & Rolling Trends...")
    # Memory (Lags) - These are the MAPE killers
    lags = [1, 2, 48, 336]
    for lag in lags:
        df[f'target_lag_{lag}'] = df[TARGET].shift(lag)

    # Rolling Mean (The missing feature from previous error)
    df['rolling_mean_4h'] = df[TARGET].shift(1).rolling(window=8).mean()

    # Historical Benchmark (Typical demand for this hour/day)
    # Using 'dayofweeknum' which exists in your CSV
    df['benchmark'] = df.groupby(['dayofweeknum', df.index.hour])[TARGET].transform('mean')

    # --- FINAL FEATURE SELECTION ---
    # We use lowercase for engineered ones and CSV names for existing ones
    features = [
        # Engineered Memory/Trends
        'target_lag_1', 'target_lag_2', 'target_lag_48', 'target_lag_336',
        'rolling_mean_4h', 'benchmark',

        # Time/Cyclical
        'hour_sin', 'hour_cos', 'month_sin', 'month_cos',
        'is_bank_holiday', 'dayofweeknum', 'dayofyear',

        # Weather (From CSV)
        'temperature_2m', 'wind_speed_10m', 'shortwave_radiation', 'cloud_cover',

        # Grid State (From CSV)
        'FUEL_WIND', 'FUEL_CCGT', 'FUEL_NUCLEAR', 'FUEL_BIOMASS'
    ]

    # Drop the first week because lags need 336 rows of history
    data = df[[TARGET] + features].copy()
    data = data.dropna(subset=[f'target_lag_{lag}' for lag in lags])

    # Final cleanup of any small gaps
    data = data.ffill().bfill()

    print(f"Success! Total Rows: {len(data)} | Total Features: {len(features)}")
    return data

# Execute
df_ml = engineer_features_ultimate(CSV_PATH)

```

Figure 4.3: Feature Engineering for XGBoost

```

# Define the parameter grid for RandomizedSearchCV
param_grid = {
    "learning_rate": [0.01, 0.05, 0.1],
    "max_depth": [4, 6, 8],
    "subsample": [0.7, 0.8, 0.9],
    "colsample_bytree": [0.7, 0.8, 0.9],
    "n_estimators": [500, 1000, 1500],
    "reg_lambda": [0, 1, 5],
    "reg_alpha": [0, 1, 5],
}

# Initialize base XGBoost regressor
xgb_model = xgb.XGBRegressor(objective="reg:squarederror", n_jobs=-1, random_state=42)

# Set up RandomizedSearchCV (tuning on first horizon/target for simplicity)
search_H1 = RandomizedSearchCV(xgb_model, param_grid, n_iter=20,
                               scoring="neg_mean_squared_error", cv=3, verbose=1)
search_H1.fit(X_train_H1, y_train_H1.iloc[:,0]) # Tune using the first target

# Print best parameters found
print("Best params for 1-Hour Horizon:", search_H1.best_params_)

# Repeat tuning for 24-Hour Horizon (optional, but added for completeness)
search_H24 = RandomizedSearchCV(xgb_model, param_grid, n_iter=20,
                                scoring="neg_mean_squared_error", cv=3, verbose=1)
search_H24.fit(X_train_H24, y_train_H24.iloc[:,0]) # Tune using the first target

# Print best parameters found
print("Best params for 24-Hour Horizon:", search_H24.best_params_)

```

✓ 16m 28.4s

Fitting 3 folds for each of 20 candidates, totalling 60 fits
 Best params for 1-Hour Horizon: {'subsample': 0.8, 'reg_lambda': 1,
 Fitting 3 folds for each of 20 candidates, totalling 60 fits
 Best params for 24-Hour Horizon: {'subsample': 0.7, 'reg_lambda': 1,

```

def train_model(X, y, name, best_params):
    print(f"Training XGBoost: {name} with tuned parameters...")
    model = MultiOutputRegressor(xgb.XGBRegressor(
        **best_params,
        objective="reg:squarederror",
        n_jobs=-1,
        random_state=42
    ))
    model.fit(X, y)
    return model

# Train models using the best parameters from tuning
model_H1 = train_model(X_train_H1, y_train_H1, "1-Hour", search_H1.best_params_)
model_H24 = train_model(X_train_H24, y_train_H24, "24-Hour", search_H24.best_params_)

```

✓ 13m 59.6s

Training XGBoost: 1-Hour with tuned parameters...
 Training XGBoost: 24-Hour with tuned parameters...

Figure 4.4: XGBoost Model Training

The features with labels of lagged targets are made to accommodate both horizons, e.g., multiple-hour lagged features (e.g., 1, 2, 48, and 336 steps), a 4-hour rolling mean, and cyclic encodings of hour, day of week, and month; they are joined with the exogenous variables and separated chronologically into 70% train, 15% validation, and 15% test with no shuffling. Each horizon is trained with an independent XGBoost regressor of the training set with the help of the validation set to prevent premature termination, and the performance on the held-out test set is measured using the original MW scale in terms of RMSE, MAE, and MAPE; SHAP

values are then computed on the test features to gain insight into the importance and impact of features.

4.2.3 LSTM with Attention Architecture, Sequence Creation, and Training

The LSTM attention model accepts the same scaled multivariate input as XGBoost but rearranged into sequences: after the data has been preprocessed, the data matrix is transformed into 48 (24 hours) long sequences, and a `create_sequences_multi_horizon` function returns a 3D tensor and two target arrays, one each of H1 (horizon 2) and H24 (horizon 48). The weights of the 48 time steps are learnt with a custom temporal AttentionLayer, and to train a stacked architecture, the `create_lstm_attention_model` function is defined with two LSTM layers (256 units and `return_sequences=True`) and dropout and layer normalisation blocks, the attention layer, a dense layer, and a linear output neurone.

```
def create_sequences_multi_horizon(data, target_idx, lookback, forecast_horizons):
    """
    Create sequences for multiple forecast horizons
    """
    x = []
    y_horizons = {h: [] for h in forecast_horizons}

    for i in range(lookback, len(data) - max(forecast_horizons)):
        x.append(data[i-lookback:i, :])
        for horizon in forecast_horizons:
            y_horizons[horizon].append(data[i + horizon - 1, target_idx])

    return np.array(x), {h: np.array(y_horizons[h]) for h in forecast_horizons}

forecast_horizons = [2, 48] # H1 and H24
target_idx = df.columns.get_loc(TARGET_COL)

x, y_horizons = create_sequences_multi_horizon(data, target_idx, lookback, forecast_horizons)

print(f"Shape of x: {x.shape}")
print(f"Shape of y (H1): {y_horizons[2].shape}")
print(f"Shape of y (H24): {y_horizons[48].shape}")

✓ 0.6s

Shape of x: (70037, 48, 91)
Shape of y (H1): (70037,)
Shape of y (H24): (70037,)
```

Figure 4.5: Sequence Creation for LSTM

The sequences are chronologically separated into 70 per cent training, 15 per cent validation and 15 per cent test with the help of a special splitter that preserves the temporal pattern, and two models are instantiated where H1 and H24 were of the same structure, but they had different targets. The two models are provided with Adam (learning rate 1×10^{-3}), loss of the mean squared errors, early stopping, reducing the learning rate and best model checkpointing; the predictions on all splits are inversely transformed to the MW and analysed with the help of a helper that calculates the RMSE, MAE and MAPE.

```

# Cell 8: Custom Attention Layer
class AttentionLayer(Layer):
    """
    Custom Attention mechanism for interpretability.

    This layer computes attention weights across the temporal dimension,
    allowing the model to focus on important time steps in the input sequence.
    """
    def __init__(self, **kwargs):
        super(AttentionLayer, self).__init__(**kwargs)

    def build(self, input_shape):
        self.W = self.add_weight(
            name="att_weight",
            shape=(input_shape[-1], 1),
            initializer="normal",
            trainable=True
        )
        self.b = self.add_weight(
            name="att_bias",
            shape=(input_shape[1], 1),
            initializer="zeros",
            trainable=True
        )
        super(AttentionLayer, self).build(input_shape)

    def call(self, x):
        e = tf.keras.backend.tanh(tf.keras.backend.dot(x, self.W) + self.b)
        a = tf.keras.backend.softmax(e, axis=1)
        output = x * a
        return tf.keras.backend.sum(output, axis=1)

    def compute_output_shape(self, input_shape):
        return (input_shape[0], input_shape[-1])

```

✓ 0.0s

Figure 4.6: LSTM with Attention Model Definition

```

# Cell 9: Build LSTM Models with Attention (Unidirectional)
def create_lstm_attention_model(input_shape, name="LSTM_Attention", lstm_units=128):
    """
    Build Standard LSTM model with Attention mechanism and regularization.
    """
    inputs = Input(shape=(input_shape[1], input_shape[2]), name='input')

    # First LSTM Layer - return_sequences=True is required to pass to Attention
    x = LSTM(lstm_units, return_sequences=True, name='lstm_1')(inputs)
    x = Dropout(0.25, name='dropout_1')(x)
    x = LayerNormalization(name='layer_norm_1')(x)

    # Second LSTM Layer
    x = LSTM(lstm_units // 2, return_sequences=True, name='lstm_2')(x)
    x = Dropout(0.2, name='dropout_2')(x)
    x = LayerNormalization(name='layer_norm_2')(x)

    # Attention Layer
    # Note: If your AttentionLayer is a custom class, keep using it.
    # If not, ensure you have defined it or use a GlobalAveragePooling1D
    # as a simple alternative if the custom layer is missing.
    x = AttentionLayer(name='attention')(x)

    x = Dense(64, activation='relu', name='dense_1')(x)
    x = Dropout(0.2, name='dropout_3')(x)

    outputs = Dense(1, name='output')(x)

    model = Model(inputs=inputs, outputs=outputs, name=name)

    # Using Adam with a decay or specific Learning rate
    optimizer = tf.keras.optimizers.Adam(learning_rate=1e-3)
    model.compile(optimizer=optimizer, loss='mse', metrics=['mae'])

    return model

# Creating models with the specific time horizons
# Note: Ensure X_train is defined from your previous splitting cells
model_H1 = create_lstm_attention_model(X_train.shape, name="H1_1Hour_Ahead", lstm_units=256)
model_H24 = create_lstm_attention_model(X_train.shape, name="H24_24Hour_Ahead", lstm_units=256)

print("Unidirectional LSTM Models created successfully!")
print("\nH1 Model Summary:")
model_H1.summary()

print("Unidirectional LSTM Models created successfully!")
print("\nH24 Model Summary:")
model_H24.summary()

```

Figure 4.7: LSTM with Attention Model Training

4.3 Global Performance Summary

This section presents the global performance of the models in the prediction horizon using the common error measures of the RMSE, MAE and MAPE in megawatts. The condensed findings presented in Table 4.1 allow us to directly assess the behaviour of the Prophet, XGBoost and the LSTM with attention to predicting (H1) vs. (H24) with the same data division and target definition. On comparing these measures with one another, the table does not just identify the

model that produces the lowest average error at each horizon, but also the strength of the one that is very short term is substituted with a full day ahead demand prediction.

Model	Horizon	RMSE (MW)	MAE (MW)	MAPE (%)
Prophet	H1	694.13	677.94	2.69
XGBoost	H1	598.42	448.64	1.58
LSTM	H1	1598.61	1216.73	4.03
Prophet	H24	845.22	750.55	3.17
XGBoost	H24	1163.17	851.37	3.05
LSTM	H24	2124.32	1596.73	5.45

Table 4.1: Test-set performance by model and horizon (MW)

Table 4.1 summarises the test-set performance of all models at both horizons using RMSE, MAE, and MAPE in the MW scale.

XGBoost has a better prediction accuracy (H1) with a smaller RMSE, MAE, and MAPE than Prophet and the LSTM with attention. Prophet is not very weak at H1, and the LSTM demonstrates much bigger errors. In the (H24) horizon, both Prophet and XGBoost are easily performing well, and the error level is more or less similar. Conversely, the LSTM once again has much larger RMSE, MAE, and MAPE, which proves it is not as effective as the other two models in short-term and day-ahead demand prediction in this study.

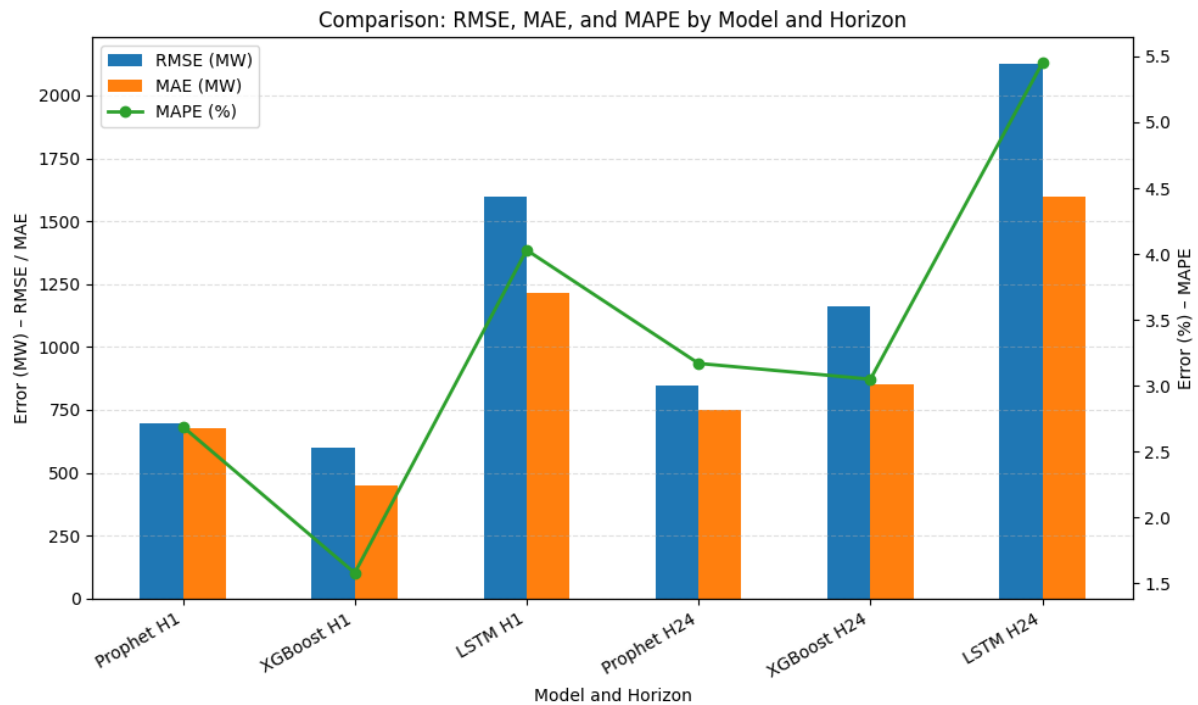


Figure 4.8: Performance of the overall model (H1 and H24)

Figure 4.8 indicates the effectiveness of each model in predicting demand of electricity at (H1) and (H24). The blue bars and orange bars are the magnitude of the error in MW, and the green line is the error percentage. On H1, XGBoost contains the minimum bars and lowest green point, and this translates to being the most accurate. In the case of H24, Prophet is the one that has the least amount of errors, with the LSTM model having the largest amount of errors at both horizons.

4.4 One-Hour-Ahead (H1) Results

This part examines the performance of the three models in making one-hour electric demand projections in Great Britain. At this short range, demand tends to flow in a smooth manner, yet it can change rapidly in the morning and evening peaks. The combination of the error measures and plots in the analysis both upholds the degree of accuracy of the models when the average case is taken and the extent to which they capture these rapid changes. The following subsections are devoted to each of the models and describe the way these models address the H1 task. Collectively, they provide a vivid comparison of the very short-term prediction models.

4.4.1 Prophet H1 Prediction

This section explains the performance of Prophet when it is used to predict the demand of electricity an hour ahead. The comparison of the actual and the predicted distributions indicates that Prophet represents the general form as well as the dispersion of H1 demand in a rather

acceptable manner. The predicting curve is, however, shifted slightly, indicating that Prophet predicts a slightly lower demand than that which exists. The error values also depict this behaviour; the error values are also reasonable but not the best among the models. The findings can be used to demonstrate the strengths of Prophet as well as its minor systematic bias at this short horizon.

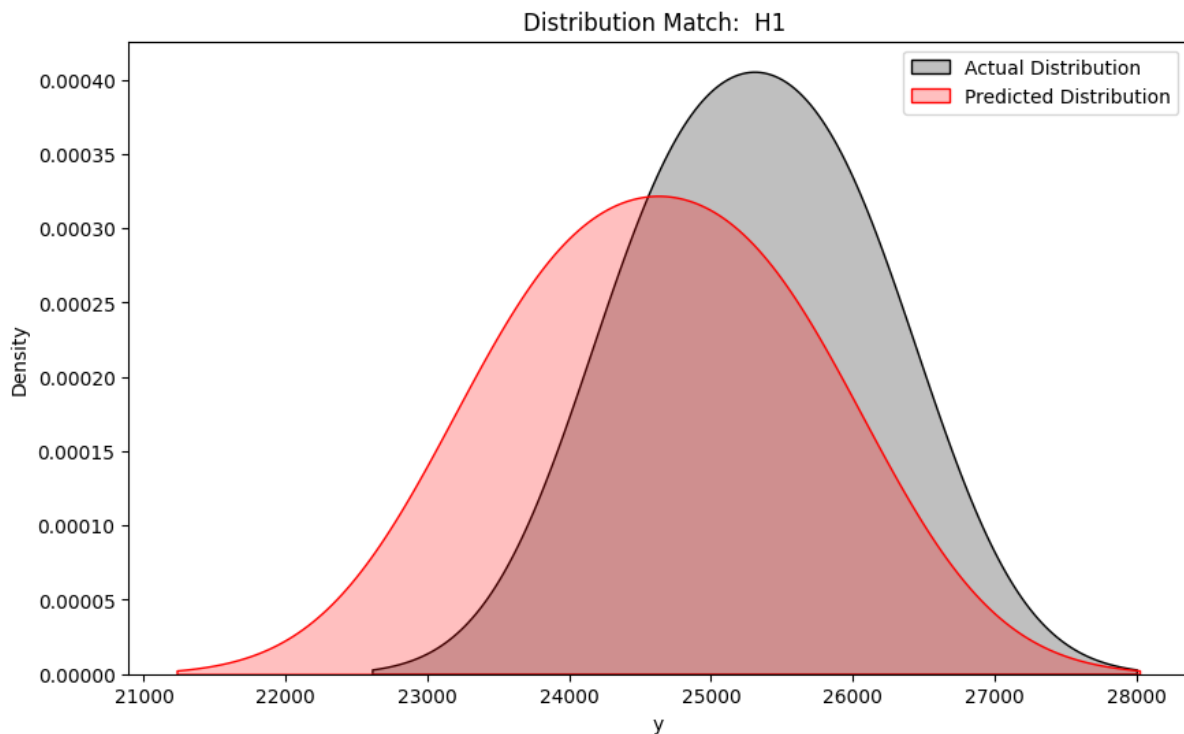


Figure 4.9: The distribution of actual and predicted demand of Prophet H1

Figure 4.9 draws a comparison between the actual values of H1 demand and the Prophet prediction. The grey curve indicates the actual demand, and the red curve indicates the Prophet forecasts, both of which constitute smooth bell curves. The curves are rather coinciding, and it is an indication that Prophet represents the overall variability and dispersion of demand. But it is observed that the red curve is somewhat displaced, and this means that Prophet is likely to forecast a slightly lower level of demand as compared to the level that was observed at the H1 horizon.

4.4.2 XGBoost H1 Prediction

This part is a description of the performance of XGBoost to predict the demand one hour in advance. It has the fewest errors at H1, indicating that it is the best model to be used in this horizon. As time series plots can show, XGBoost also follows the time and the amplitude of daily peaks and troughs closely. The feature analyses indicate that the recent demand alongside calendar, fuel, and weather variables have been very important in this high performance.

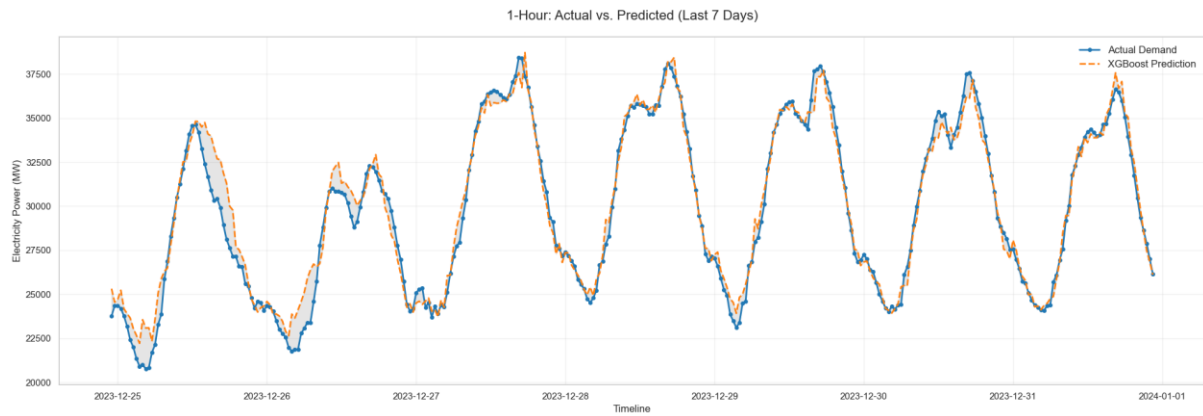


Figure 4.10: XGBoost H1 – actual vs predicted demand (last 7 days)

In Figure 4.10, the XGBoost H2 prediction is plotted against the actual electricity demand during the past seven days of the test set. The blue line (actual demand) is nearly parallel to the dashed orange line (prediction), and this indicates the XGBoost is highly sensitive with regard to capturing the time and magnitude of the daily peaks and troughs. Between the two lines, small gaps occur on some intervals, though they are insignificant in contrast to the match. This graphic information backs up the low H1 errors documented in XGBoost and confirms its harshness in the short-term prediction.

4.4.3 LSTM H1 Prediction

In this subsection, the behaviour of the LSTM with attention on the H1 horizon is examined. The model takes in handy various generation and flow variables which have a close relation to the future demand. Nevertheless, its forecasts tend to deviate from the actual values, particularly where there are high peaks and low troughs. Consequently, its error values are evidently much more than Prophet and XGBoost, though the correlation with the true series in total is high. These results indicate that the LSTM will learn the overall trend and is not very specific in predicting what will happen in a one-hour time frame.

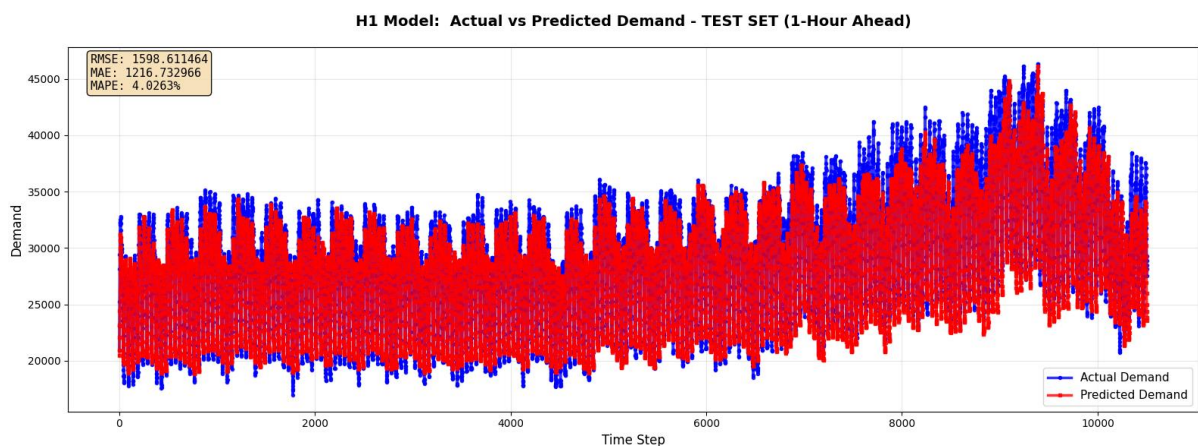


Figure 4.11: LSTM H1 – actual vs predicted demand on the test set

The LSTM predictions of the one-hour-ahead demand are presented in Figure 4.11 against the actual demand of the entire H1 test set. The red line is the LSTM prediction, and the blue line is the actual demand, and despite the overall direction of the trend of the model and the daily cycles, there is usually a difference between the two curves, particularly at peaks and troughs. The text box records an RMSE of approximately 1599 MW, an MAE of approximately 1217 MW and an MAPE of approximately 4.0 per cent, which indicates that errors are significantly greater than with Prophet and XGBoost at H1. These findings suggest that the LSTM would be able to capture the general trend but fail to give accurate short-term predictions.

4.5 Twenty-Four-Hour-Ahead (H24) Results

In this section, the performance of the model is studied on a twenty-four-hour-ahead (H24) horizon in which the entire demand profile in a day is predicted. Here the same errors that were used in H1 are used to compare Prophet, XGBoost, and the LSTM with attention. The findings demonstrate that Prophet and XGBoost maintain rather reasonable accuracy at 24 hours ahead, and the LSTM commits a bigger error. Such visual tools as distributions and hour-of-day errors are used to demonstrate how prediction behaviour is throughout the entire day. This enables one to see the difference in model performance between very short-term and day-ahead prediction.

4.5.1 Prophet H24 Prediction

This subcategory idles on the 24-hour prediction by Prophet, and its capacity to explain the 24-hour demand cycle form. The distributions of the actual and the predicted values are near to each other, and both have two peaks as they should; that is, the normal daily distribution. Hour of the day error indicates that the greatest errors are made during the hours when demand is changing fast, i.e., the early morning and the early evening. The residual plots on the horizon make sure that the underestimation and overestimation are more common in the evening peak. This is preceded by a component breakdown which is used to show how the daily, weekly, trend and holiday effects add together to provide the H24 prediction.

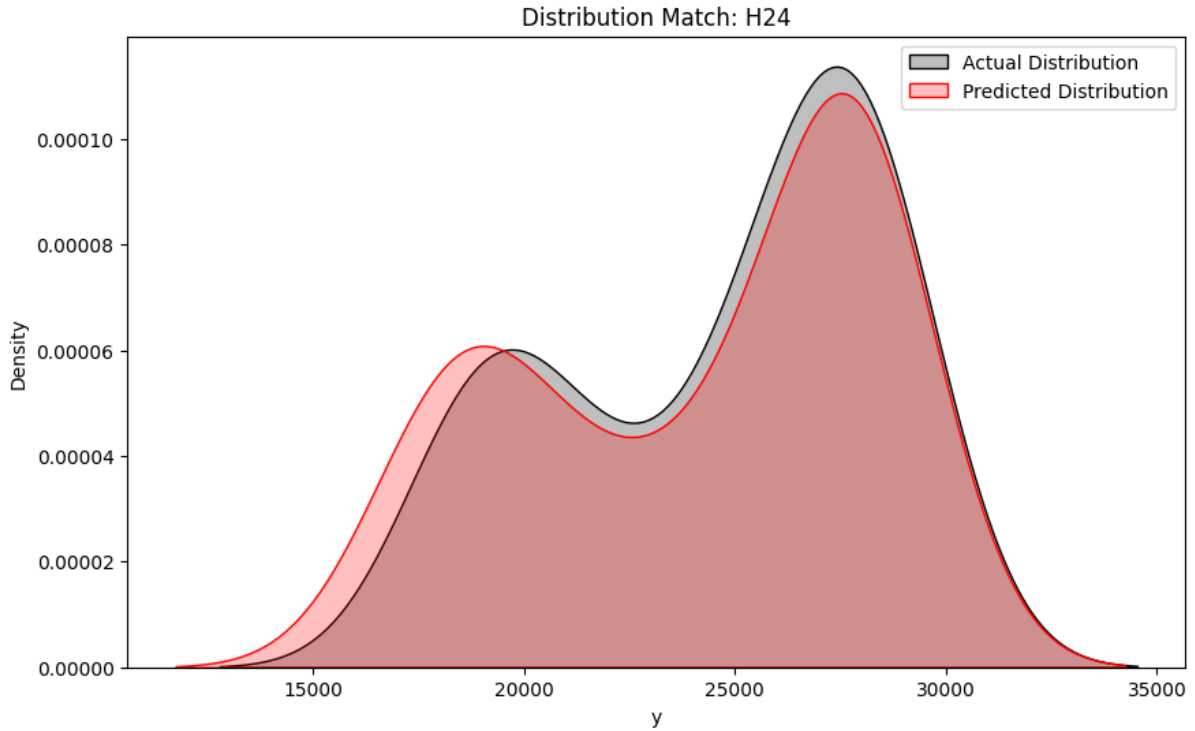


Figure 4.12: Prophet H24 – distribution of actual and predicted demand

Figure 4.12 reveals that there is a comparison between the distribution of the actual twenty-four hours ahead of (H24) demand and the distribution of the Prophet prediction. The grey curve shows the distribution of the real values of demand, and the red curve is the distribution of the predicted values. The two curves are seen to possess a clear two-peak structure that occurs parallel to the underlying daily patterns. That the curves coincide in the whole range would mean that not only does Prophet reproduce the averageness of the level of demand, but also does it do the variability and the multiple modes of the H24 distribution. The curves have minor deviations, indicating the fact that Prophet slightly over- or underestimates certain levels of demand, which are not significant with regard to the overall fit.

4.5.2 XGBoost H24 Prediction

In the current subsection, the author will consider the 24-hour-ahead predictions of the XGBoost and the effectiveness of the latter at this horizon (24 hours). The plot of several of the test days demonstrates that the times of the daily peaks and troughs and their magnitude can be followed reasonably in XGBoost. Both correlation and feature importance results indicate prevailing values of recent values of demand with longer values of lags and calendar equipment. Additional refinements that are made by fuel and weather characteristics help XGBoost to create realistic day-ahead characterisations.

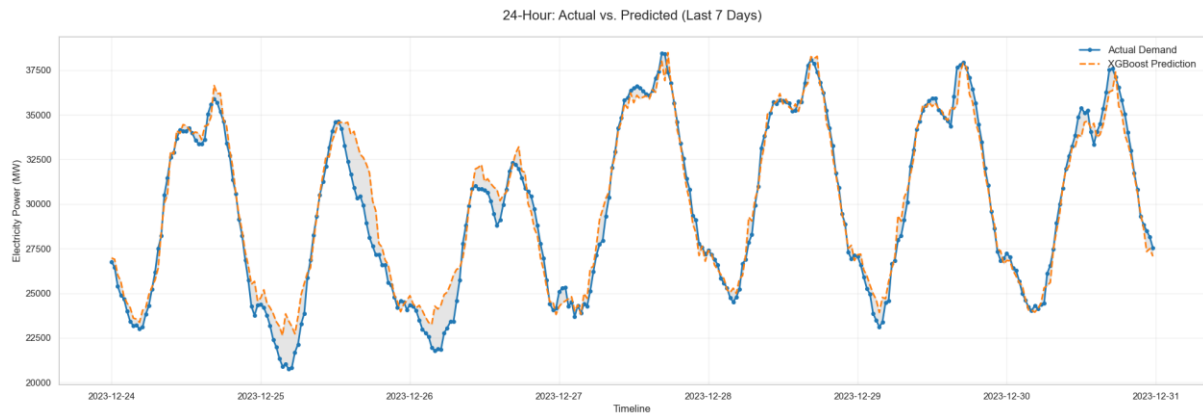


Figure 4.13: XGBoost H24 – actual vs predicted demand (last 7 days)

Figure 4.13 compares the day-ahead predictions against the real electricity demand during the previous seven days of the test set by the XGBoost model. The XGBoost (dashed orange) prediction is right behind the solid blue line (actual demand) and indicates the time and the overall size of peaks and troughs on a daily basis. There are large deviations on some days, but on the general trend, the model is slightly overfitting or underfitting the peak height. This value indicates that XGBoost can provide quite good 24-hour-ahead profiles, even though Prophet is slightly better on average at this horizon.

4.5.3 LSTM H24 Prediction

The H24 prediction of the LSTM is discussed in this subsection, and the model must be able to peep far into the future using its sequence input. Other important features include synchronous generation, hydro generation, coal generation and fossil generation, which are highly correlated with the future demand. The projected profile of the day ahead is, however, normally diverse with the actual profiles, especially the high-demand days. The error of H24 values is quite large compared to Prophet and XGBoost, which are poorly performing. The residual and scatter plots show that the errors are widely distributed, in particular at the higher load levels; hence, it is simple to understand the reason behind the high RMSE and MAPE.

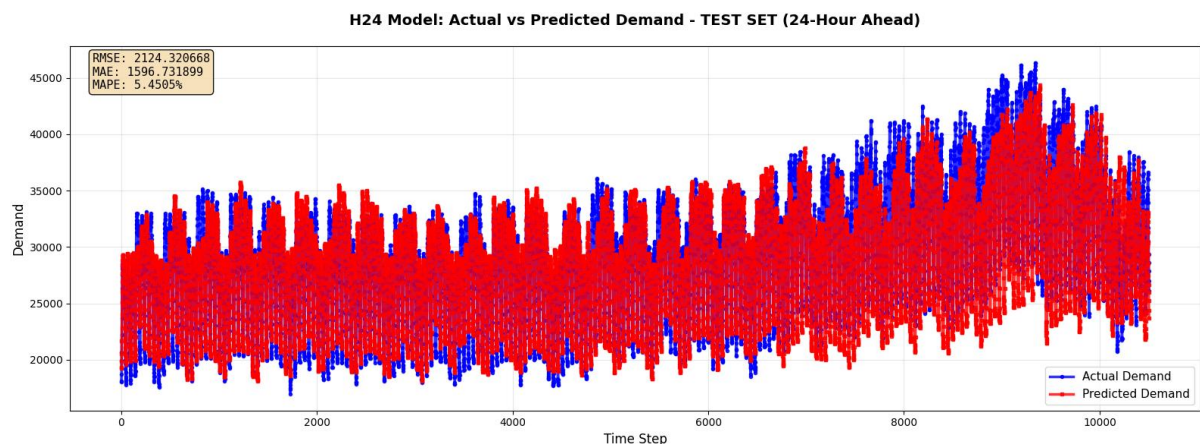


Figure 4.14: LSTM H24 – actual vs predicted demand on the test set

The actual demand compared with the results of the LSTM 24-hour-ahead prediction is plotted in figure 4.14. The actual load is represented by the blue line, and the LSTM prediction by the red line. Although the model follows the general direction of the uptrend and the daily trends, the two curves tend to diverge, especially on larger peaks and troughs. These differences are significant, which is testified to by the text box numeric values (RMSE 2124 MW, MAE 1596 MW, MAPE 5.45%). This measure shows that the LSTM is worse than the H1 horizon, and it provides a considerably orchestrated prediction compared to Prophet and XGBoost.

4.6 Explainability Results

This part provides the explainability analysis in order to understand the models better in terms of making predictions. In XGBoost, SHAP figures are used to measure the amount of contribution of each feature that is either adding to or subtracting from individual predictions. Attention weights used in the case of LSTM with attention show the time steps in the 24-hour window that the model has been most attentive to. Components in the case of Prophet show the relationship between the trend term, daily term, weekly term, and holiday term with the actual demand. The combination of these tools makes the models more transparent and allows them to associate their numerical performance with understandable and clear trends.

4.6.1 XGBoost SHAP Analysis

SHAP plots are used in this section to give an explanation of the way XGBoost uses its features at H1 and H24. In MW, the summaries rank features in terms of importance and show the impact a high or low value of any feature would have on the forecast. The most powerful aspect is always the latest value of a demand, followed by the longer-term lags, hour of day, day of week and fuel mix variables. The difference in the H1 and H24 SHAP patterns is representative of the trade-off between short-term memory and the temporalisation of XGBoost increasing with horizon. The following discussion may be applied to comprehend the usefulness of XGBoost and the way in which its predictions reflect known drivers of electricity demand.

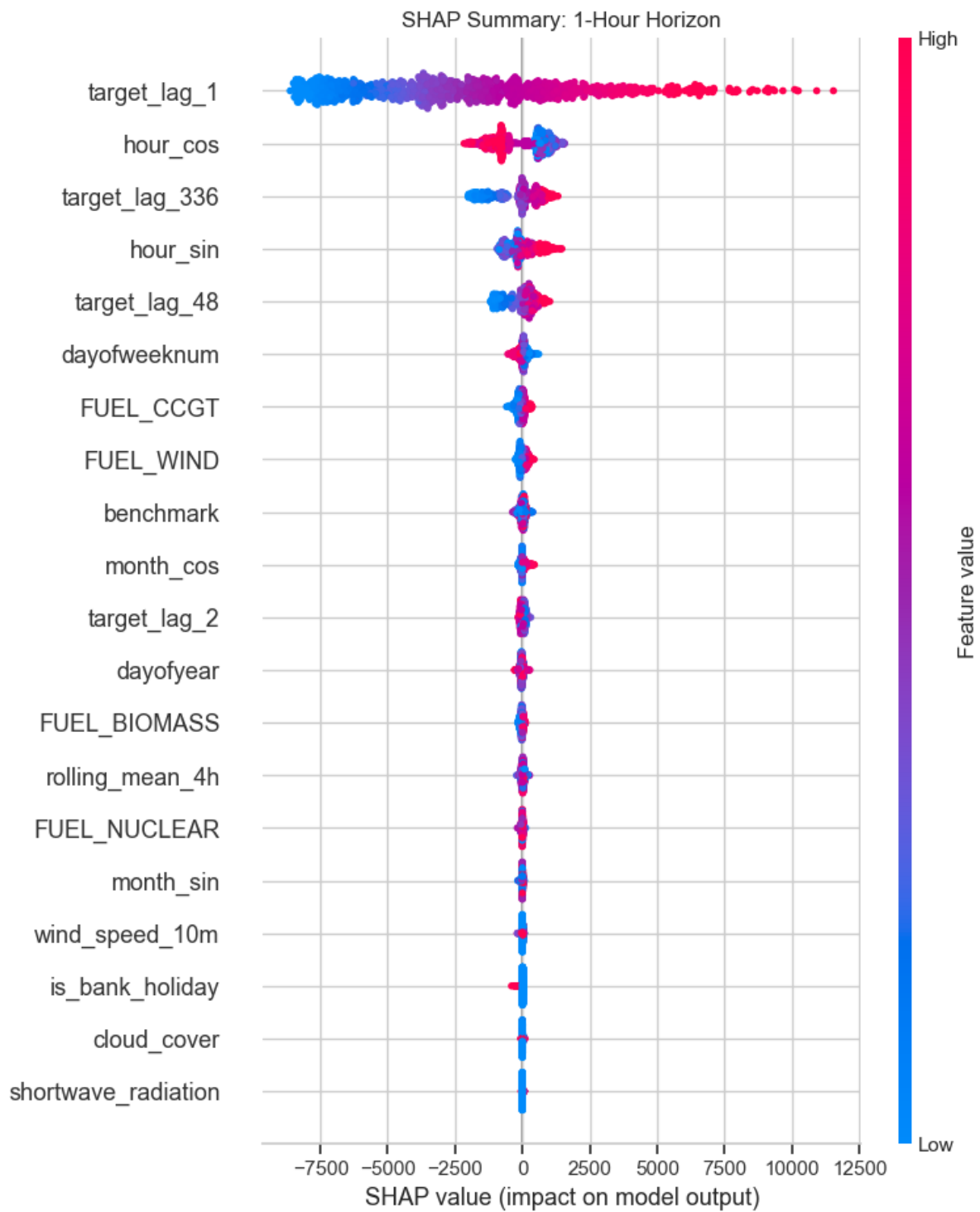


Figure 4.15: SHAP summary for the XGBoost H1 model

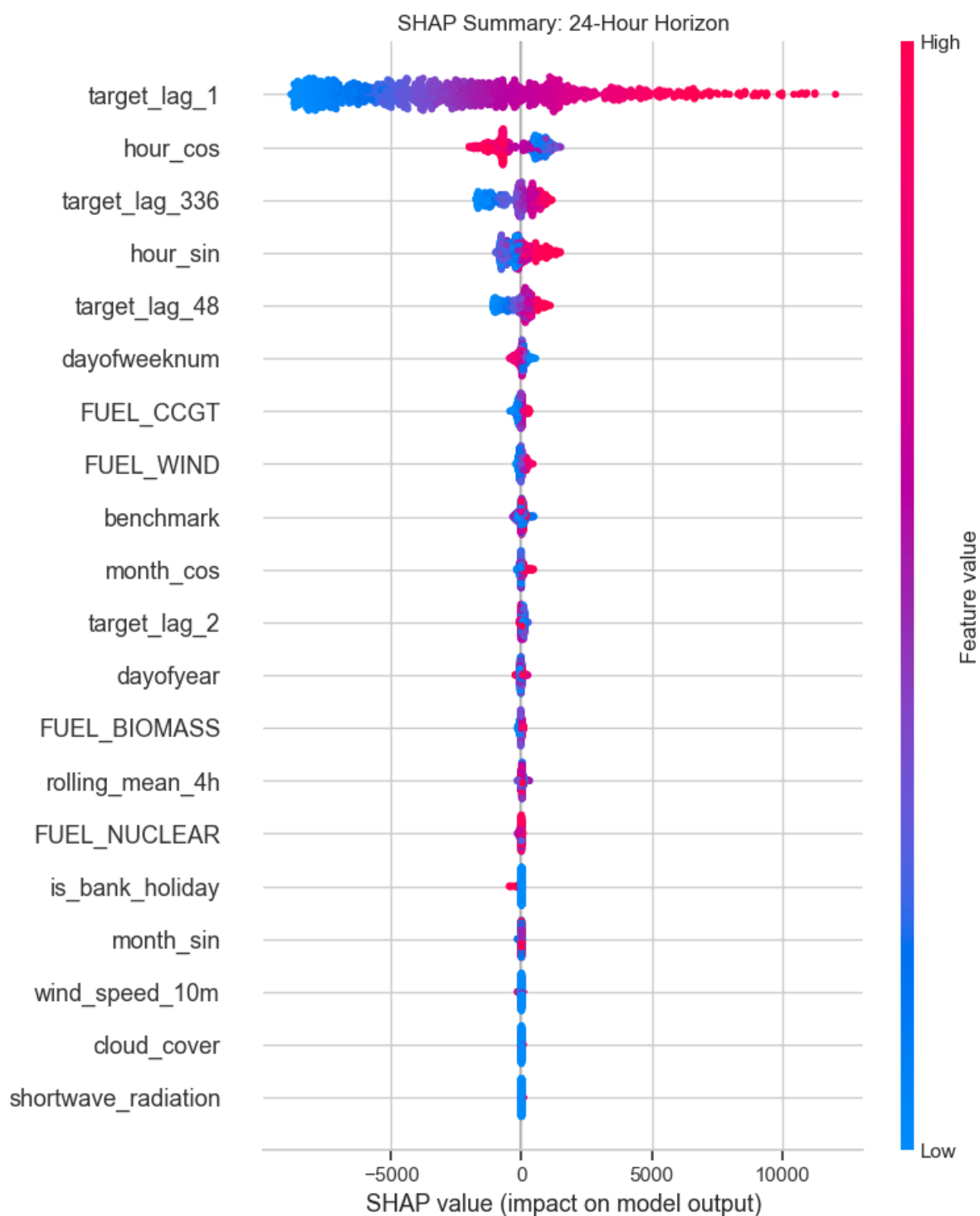


Figure 4.16: SHAP summary for the XGBoost H24 model

The SHAP summary figures are used to display the way the XGBoost models use their input features to make predictions, which in both horizons has a row per feature ranked by importance and a dot per row depicting the extent to which that feature changes an individual prediction up or down in MW, with the dots coloured either red or blue depending on whether the feature value is high or low, respectively. At H1 and H24, the most recent load (target_lag_1) is the major driver, with high recent demand augmenting the forecast and low

recent demand weakening it, with further consistent adjustments given by time of day encodings (hour_cos and hour_sin), longer lags (48 and 336 steps), day of week, and generation variables (gas, wind, and biomass at H24).

4.6.2 LSTM Attention Profiles

This subsection considers the attention mechanism in the LSTM and the way the attention mechanism gives weight to various aspects of the past 24 hours. Mean attention curves of H1 and H24 indicate that the greatest weights are towards the more recent hours before the forecast. The weights decrease in the middle of the window and increase again towards the end of 20-24 hours before it indicates the role of the previous day at the same time. This behaviour is appropriate in the concept that electricity demand is related to the present situation as well as daily trends. The attention profiles thus provide an intuitive perspective on what past times are the most effective for the LSTM to make predictions.

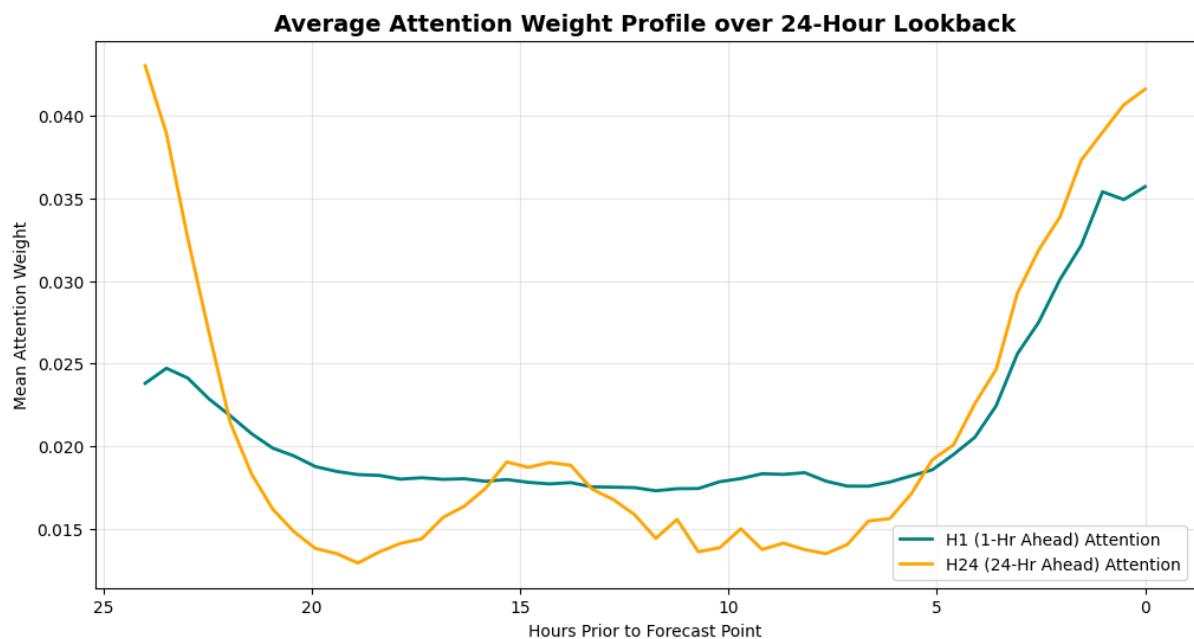


Figure 4.17: Average attention weight profile for the LSTM over a 24-hour lookback

Figure 4.17 demonstrates the importance distribution of the attention mechanism of LSTM across the last 24 hours during one-hour-ahead (H1) and 24-hour-ahead (H24) predictions. Both lines have a high value close to the latest few hours preceding the forecast time, i.e., the model is based mostly on the very recent demand history to make its predictions. The weights of attention are significantly lower in the middle of the lookback window, followed by an increase again at approximately 20-24 hours before, which indicates that the LSTM attracts some information at the same time during the previous day. This is the natural trend with electricity demand in that the most recent load and the profile of the last day are useful information in short-term prediction.

4.7 Chapter Summary

In this chapter, Prophet, XGBoost, and an LSTM with attention were compared to one-hour-ahead (H1) and twenty-four-hour-ahead (H24) predictions of electricity demand in Great Britain, on shared half-hourly and hourly data with a chronological split of 70/15/15. XGBoost was also the most accurate at H1, and Prophet and XGBoost were also accurate at H24, and the LSTM had significantly higher errors at both horizons. The visual and error-based analysis revealed that Prophet and XGBoost closely follow the daily peaks and troughs, but the LSTM primarily follows broad trends. Explainability analyses with SHAP, attention, and Prophet components revealed the importance of recent load, calendar effects, fuel mix, and seasonal patterns, which shows that tree-based and decomposition-based models are more appropriate in this research than the deep learning architecture used in the experiment.

Chapter 5: Discussion

5.1 Introduction

This chapter is a critical discussion of the key empirical results as far as the research questions are concerned, the broader literature on short-term electricity demand prediction, and the methodological decisions used in the research. Following are the interpretation of the model performance at one-hour-ahead (H1) and twenty-four-hour-ahead (H24), the role of features, and the reflection of the practical and methodological implications.

5.2 Overview of Main Findings

The findings indicate that XGBoost performs the best in one-hour-ahead prediction, whereas XGBoost and Prophet perform well at the twenty-four-hour horizon, which is manifested by lower values of RMSE, MAE, and MAPE than that of the LSTM with attention. The LSTM, on both horizons, is more prone to errors and tends to smooth peaks, suggesting that, under this model and data, tree-based ensemble and decomposable time-series models would be more successful in representing GB system-level dynamics than the used deep learning model.

Such results directly answer the first two research questions because the presented findings show that machine learning models trained only on publicly available data can provide practically useful short-term predictions of Great Britain and that both XGBoost and Prophet are more successful than the LSTM implementation within the proposed unified experimental framework. The findings hence endorse the Chapter 2 hypothesis that contemporary machine learning and time series models can triumph over conventional linear techniques when making short-term predictions of loads in complex systems dominated by renewable sources.

5.3 Interpretation in Contradiction to Research Questions

The initial research question was what the predictive accuracy of various models of GB short-term demand is using open data. The relatively low percentage error at H1 and H24 of XGBoost and Prophet, along with the capability to recreate daily peaks and troughs based on their usage, with minimal bias, points to the fact that these models can be used in operational decision support during the period under study. Specifically, high H1 performance means that recent history of performance, along with high-frequency time aspects, offers sufficient structure to make an accurate short-term prediction despite the use of only publicly available data.

The second research question was which model family out of three gives the highest accuracy. The empirical results are quite clear, with XGBoost dominating at the one-hour horizon and XGBoost and Prophet as the topmost at twenty-four hours, with the LSTM with attention not doing as well with its higher representational ability. This fact supports the claim made in the literature that, with gradient boosting, engineered tabular features present, gradient boosting

could be more effective than more complex neural architectures, especially when the quantity of training data is small and model tuning resources are limited.

The third research question concerned the impact of weather, time and calendar attributes. The XGBoost models of SHAP analysis suggest recent load, hour of day, day of week, and bank holiday indicators be considered significant factors, whereas weather variables, including temperature and wind speed, impact more significantly at the H24 horizon, where aggregated behavioural patterns prevail. The decomposed components of Prophet also highlight the daily and weekly seasonality and holiday effects, and the LSTM attention weights reveal that the very recent demand and the same hour of the day before are used as the core sources of prediction skill in GB.

5.4 Links to Literature and Theoretical Expectations

This stronger performance for XGBoost and Prophet runs somewhat in the same theme as research that emphasises the need for incorporating autocorrelation, meaningful calendar variables, and interpretative seasonality for short-term load prediction on the system level. As explained in Chapter 2, linear models are weak against the non-linear influences caused by climactic conditions, the integration of renewables into the energy network, and changing behaviour, whereas boosting models are less sensitive to these forces than complex neural models.

Relative to the literature, the comparative weakness of the LSTM model involving attention seems inconsistent concerning the improvements attributed to deep learning for building-level and fine-grained load scales. It might be attributed to the fact that the GB national aggregate demand time series is already smoothed, and the major processes follow robust and systematic daily and weekly patterns, leaving less scope for the exploitation of long-term dependencies by the deep sequence model, as the major forecasting determinants are already encompassed within the lagged demands and various day attributes. Moreover, the LSTM model is known to be more prone to the risk of overfitting, which has adversely affected its performance in the current research as well.

Taken together, these results suggest that, where rich feature sets and high levels of seasonal structure are present, ensemble trees and decomposable time series models can potentially provide a better tradeoff of accuracy, robustness, and interpretability than more complex models of the type used in deep learning. This is in line with more recent research emphasis suggesting that model selection research can very much be driven by characteristics of available data, rather than basing model selection on the principle of always preferring deep models over others.

5.5 Implications at the Practical Level and the System Level

System-operations-wise, the one-hour-ahead predictions generated by XGBoost are accurate, and the day-ahead performances of XGBoost and Prophet are competitive, which directly affects the reserve scheduling, unit commitment, and short-term market planning in Great Britain. The relatively low cost of computation and the quick time of inference of such models are appropriate to incorporate in real-time or near-real-time decision-support tools, unlike the much costlier training process involved in the LSTM.

The component and feature importance analyses are also a benefit in terms of providing insight into the drivers of demand. The fact that the key role of the lagged demand and calendar variables is established proves that operators still should invest in the high-quality and high-frequency demand measurement and specific calendar modelling, which is dependent on weekends and bank holidays. The quantifiable value of the fuel-mix-related and interconnection-associated variables suggests that integrated demand, generation, and cross-border flows can be used to enhance predictions of net system load and that there are arguments within the literature that support the use of unified, open data platforms in power system operation.

5.6 Methodological Thoughts and Limitations

The analysis of results should be discussed within the framework of multiple methodological limitations. The paper purposely uses exclusively publicly available data, such as NESO Historic Demand, BMRS FUELHH, ESPENI, and Open-Meteo, and develops a vast feature set of temporal, meteorological, and operational features, which enhances the transparency and reproducibility but does not necessarily correspond to the completeness or the quality of pre-processing proprietary operational data. Other possible problems with performance in the models would have been residual values (such as missing values and measurement noise) and imperfect weather representation, especially of extreme events, that would have impacted the more sensitive LSTM.

Besides, all models were trained using a single LSTM architecture and a pragmatic, not exhaustive, hyperparameter optimisation strategy to ensure the same and a tractable, experimentable framework. Consequently, the metrics reported can be interpreted as representative of efficiently but not optimally tuned instances of every algorithm family, and that in this context, there is also room to make as much progress through deep learning as through architectural variation, regularisation, and more extensive search. Lastly, each of the models was trained and tested based on data within a certain timeframe, and thus structural shifts in the GB demand patterns, policy or use of technology after the time period might require retraining and might result in a change in comparative performance in the future when used.

5.7 Chapter Summary

The experimental results have been interpreted in the chapter, and the findings show that XGBoost and Prophet provide more reliable and practically useful short-term electricity demand prediction of Great Britain than the applied LSTM with attention with the use of fully public datasets. These results have been associated with the short-term estimation of the load in the literature, the temporal and calendar characteristics have been identified as central to the results, and the implications of the research for system operators and methodological issues surround the quality of data, model specification and tuning. These lessons will form the foundation of the final chapter that will summarise the general efforts of the research and make clear conclusions and suggestions on how future research should be conducted.

Chapter 6: Conclusion

6.1 Introduction

This chapter combines the aim, objectives, approaches, and significant results obtained in the dissertation to restate the manner by which the research enquiries were addressed and provides recommendations for practice on the prediction of short-term electricity demand in Great Britain by using machine learning models.

6.2 Restatement of Aim and Objectives

The general objective of the research was to investigate and compare various machine learning frameworks– LSTM, XGBoost, and Prophet– to predict short term electricity demand in Great Britain with only publicly available data to find models that are both precise and workable. To achieve this, the research:

- Combined NESO Historic Demand, BMRS FUELHH, ESPENI and Open Meteo into one dataset of half-hourly and hourly of the period 2020-2023.
- Created a wide range of functionality that consisted of temporal, meteorological, and operational drivers.
- XGBoost, Prophet trained, and attention-based LSTM with one hour of future (H1) and twenty-four hours of future (H24) horizons applied with a 70/15/15 chronological split.
- Comparable models are evaluated and analysed based on RMSE, MAE, and MAPE and explainability with SHAP, Prophet components, and LSTM attention profiles.
- These measures were used to make sure that the models were tested equally under a single, reproducible experimental setup based on open data.

6.3 Main Conclusions

One of the first conclusions is that precise and practically useful short-term predictions about GB can be made with the use of only publicly available datasets, provided that these are wisely combined, cleared, and extended by proper features. The combined NEFWDT dataset and feature engineering pipeline demonstrate that the open data can be used to predict the performance at the level appropriate to operational decision-making.

The second conclusion is that XGBoost and Prophet are the best predictors compared with the implemented LSTM with attention at both horizons, where XGBoost yields the lowest error at horizon H1, and XGBoost and Prophet generate similar but rather small error at horizon H24. This shows that gradient boosting and decomposable time series models are a better balance in accuracy, robustness, and computational efficiency in a system-level GB with a strong calendar structure with a rich tabular feature space than the deep learning architecture being tested.

A third conclusion is in regard to the drivers of demand coded in the models. SHAP and component analyses point to recent load, hour of the day, day of the week, holiday flags and chosen fuel mix variables as the most significant predictors, with weather variables becoming more significant at the day-ahead horizon. The profile of LSTM attention also supports that history very close and identical to the hour on the day before charge has the greatest weight, which supports the opinion that short-term time structure and calendar effects are the major factors dictating forecast skill in GB.

6.4 Knowledge and Practice Contribution

On an academic level, the dissertation presents an entirely documented and reproducible pipeline that incorporates four large-scale publicly available GB databases and compares LSTM, XGBoost and Prophet on the same data, horizons and metrics. This directly considers the areas of missing information on the use of public information, comparative models of systematisation, rich features, and interpretability in GB-centred short-term load prediction studies.

To practise, the results show that XGBoost and Prophet are good alternatives to operational implementation in short-term predicting systems in GB, with sub-second inference time and interpretable outputs that can aid in reserve scheduling, unit commitment, and short-term trading decisions. The explainability findings also enable system operators to gain a better understanding of the interaction between demand, weather, and fuel mix variables, which can be used to make better use of forecasts in a decarbonising, renewable-dominated grid.

6.5 Limitations

These conclusions are limited in a number of ways. It is limited to three model families and three years, 2020-2023: the introduction of structural changes in the demand patterns, policy, or technology outside the timeframe would change the performance and would need retraining or adjusting the model. Moreover, a single LSTM design and practical hyperparameter were considered; thus, deep learning findings cannot be considered to cover the entire design space and could be enhanced with further tuning or different designs.

Despite a well-curated processing, the dataset still reflects the flaws of the original sources in the form of missing values, noise in measurements, and an approximation to GB-wide weather with a single location. These problems are especially extensive for more complicated models like LSTMs, and it is likely that this contributed to the relative underperformance realised in this research.

6.6 Recommendations

6.6.1 Practice Recommendations

- It is suggested that the system operators and analysts should focus on feature-rich, tree-based and decomposable models (including XGBoost and Prophet) to predict short-term GB demand, particularly when transparency and quick retraining are needed.
- High-quality, high-frequency demand data and fine calendar modelling should be maintained in the forecasting teams, and the fuel mix and interconnector data should be incorporated in the forecasting processes because the variables prove to enhance accuracy.
- Since the results with public data are rather good, organisations should also adopt open data and open code more, based on the reproducible pipeline that has been proven during this dissertation.

6.6.2 Future Research Recommendations

- Future research needs to cover a wider area; it has better deep learning models, such as lighter LSTMs, CNN-LSTM hybrids, and transformer-based models, with more systematic hyperparameter optimisation, to re-evaluate the possibility of deep learning to predict the GB level with short-term forecasting.
- Further studies would expand the feature space to more detailed data on the spatial weather, demand-side response measure, and economic variables and measure their incremental value on top of the already detailed temporal and operational features that are applied here.
- Lastly, additional research is needed on the model in the presence of change in regimes, e.g., extreme events or changes in electrification patterns, and ensemble strategies that use XGBoost, Prophet and more sophisticated deep learning models to capitalise on their complementary advantages.

6.7 Chapter Summary

This chapter has reiterated the aim and objectives of the dissertation and demonstrated how they have been fulfilled through the creation of an open reproducible pipeline for short-term electricity demand forecasting in Great Britain. It has summarised the major inferences of XGBoost and Prophet being better than the used LSTM, that the time and calendar characteristics are the biggest predictors, and that public data could be utilised to support operationally significant accuracy, but has also accepted the major limitations of data, model range, and fine-tuning. The practical and research suggestions presented herein give a clear direction to the practitioners and scholars to further develop the work, perfect the forecasting tools, and continue to expand the existing evidence on machine learning in the contemporary, renewable, abundant power systems.

References

- [1] H. S. Hippert, C. Pedreira, and R. C. Souza, "Neural networks for short term load forecasting: A review and evaluation," *IEEE Trans. Power Syst.*, vol. 16, no. 1, pp. 44–55, 2001.
- [2] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, 2016.
- [3] National Grid ESO, "Future Energy Scenarios 2024."
- [4] Ofgem, "Decarbonisation Action Plan," 2020.
- [5] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [6] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, 2016.
- [7] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [8] J. W. Taylor and P. E. McSharry, "Short term load forecasting methods: An evaluation," *Int. J. Forecast.*, vol. 22, no. 1, pp. 1–17, 2006.
- [9] Y. Ryu et al., "Deep neural network-based demand forecasting for electricity consumption," *Energies*, vol. 10, no. 3, p. 381, 2017.
- [10] S. J. Taylor and B. Letham, "Forecasting at scale," *PeerJ Preprints*, 2017.
- [11] A. Harvey and S. Koopman, *Structural Time Series Models*. Oxford Univ. Press, 2000.
- [12] J. Lago et al., "Forecasting day ahead electricity prices with deep learning," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 1–10, 2020.
- [13] Y. Li et al., "Short term load forecasting using improved XGBoost model," *Energy*, vol. 219, p. 119592, 2021.
- [14] Meta (Open-Source Team), "Prophet forecasting tool," 2024.
- [15] National Grid ESO, "Digitalisation Strategy 2022." [Online].
- [16] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, 5th ed. Wiley, 2015.
- [17] R. Weron, "Electricity price forecasting: A review," *Int. J. Forecast.*, vol. 30, pp. 1030–1081, 2014. [Online].
- [18] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2000.
- [19] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, 2001.

- [20] T. Hong et al., “Global energy forecasting competition 2012,” *Int. J. Forecast.*, vol. 30, no. 2, pp. 357–363, 2014.
- [21] J. Lim et al., “Temporal fusion transformers for interpretable multihorizon forecasting,” *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764, 2021.
- [22] J. Zhao et al., “Hybrid ARIMA–LSTM load forecasting,” *Energy AI*, vol. 1, p. 100013, 2020.
- [23] P. Brockwell and R. Davis, *Time Series: Theory and Methods*, 2nd ed. Springer, 2016.
- [24] Copernicus Climate Change Service (ECMWF), “ERA5 reanalysis dataset,” 2024.
- [25] FAIR Principles, “GO FAIR Initiative,” 2024.
- [26] Gov.UK, “UK Bank Holidays Dataset,” 2025.
- [27] Sheffield Solar PV Live, “UK Solar Generation Estimates,” 2024.
- [28] J. Lago, “ESPENI: Electricity system and weather data integrated dataset,” *Zenodo*, 2020. [Online].
- [29] Open Meteo, “Open Meteo weather API / historical data,” Open Meteo, 2025.
- [30] Elexon BMRS, “Half hourly generation outturn by fuel type (FUELHH) — BMRS API Documentation,” Elexon, 2025.
- [31] National Grid ESO, “Historic Demand Data Portal,” 2025.
- [32] A. K. Farsi *et al.*, “On short-term load forecasting using machine learning techniques and a novel parallel deep LSTM–CNN approach,” *IEEE Access*, vol. 9, pp. 36720–36733, 2021.
- [33] I. E. E. A. A. Al-Musay *et al.*, “Energy load forecasting with machine learning: Models, metrics, and future directions,” *Preprint*, Aug. 2025.
- [34] T. V. T. Ngoc, N. T. L. Giang, and H. H. Nguyen, “Comparing Prophet, XGBoost, and LSTM models for web traffic forecasting,” *DiVA Portal*, pp. 29–39, Sep. 2024.
- [35] L. Zhang and D. Jánošík, “Short-term electrical load forecasting based on XGBoost model,” *Energies*, vol. 18, no. 19, p. 5144, 2025.
- [36] K. T. Jones, “Time series forecasting for power plant emissions: LSTM, XGBoost, and SARIMA,” *Medium*, Oct. 2025.
- [37] I. Staffell, “Measuring the progress and impacts of decarbonising British electricity,” *Energy Policy*, vol. 102, pp. 463–475, 2017.
- [38] National Grid ESO, Environmental Defense Fund Europe, and WWF, “GB carbon intensity forecast methodology,” *NESO Public Document*, Jan. 2018.
- [39] A. Lotfipoor, M. K. Rahmani, and M. Wetter, “Short-term forecasting of residential electricity demand using CNN–LSTM,” in *Proc. USIM Conf.*, 2020.

- [40] S. K. Moon, H. Kim, and H. Cho, "Short-term electric load forecasting based on weather and social effects using LSTM," *Energies*, vol. 11, no. 12, p. 3283, 2018.
- [41] S. A. Fayaz and R. Kavasseri, "Short-term load forecasting of electric power systems using wavelet transform and weather-sensitive artificial neural networks," *Electric Power Systems Research*, vol. 122, pp. 140–147, 2015.
- [42] M. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term load forecasting with weather information for a district heating system," *Energy*, vol. 35, no. 12, pp. 4727–4733, 2010.
- [43] Kaggle, "UK electricity consumption prediction – time series," 2023.
- [44] NESO, "Peak demand forecasting literature review (WP1)," *Technical Report*, 2020.
- [45] N. Di Lascio *et al.*, "Electricity demand forecasting through natural language processing and LSTM networks," *arXiv preprint*, arXiv:2309.06793, 2023.
- [46] R. Gholizadeh, P. Mancarella, and J. Mutale, "Short-term electricity demand forecasting in Great Britain using deep recurrent neural networks," in *Proc. IEEE PES General Meeting*, 2020.
- [47] A. Lago, G. De Ridder, and B. De Schutter, "Short-term forecasting of regional electricity load with deep neural networks," *Electric Power Systems Research*, vol. 189, 2020, Art. no. 106664.
- [48] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A LSTM-based model," *Procedia Computer Science*, vol. 147, pp. 186–191, 2019.
- [49] S. Ghosh and A. Ghosh, "Day-ahead electric load forecasting using Prophet model: A case study," *Energy Reports*, vol. 6, pp. 789–795, 2020.
- [50] N. E. Sebastian, "A comprehensive analysis of UK electricity consumption using machine learning," M.Sc. thesis, Natl. College of Ireland, 2025.
- [51] Q. Le, T. Pham, and H. Mac, "PSO-XGBoost for short-term building heating load forecasting," *Energies*, vol. 12, no. 18, Art. no. 3451, 2019.
- [52] M. Mac *et al.*, "Heat load forecasting using XGBoost: A case study with real building data," *Energy and Buildings*, vol. 226, Art. no. 110347, 2020.
- [53] A. Shakeel, A. S. M. Sajid, M. Imran, and S. A. R. Zaidi, "District heating load forecasting with a hybrid model based on LSTM and XGBoost," *Journal of Cleaner Production*, vol. 421, Art. no. 138422, 2023.
- [54] P. Puning *et al.*, "Short-term load forecasting using SVR, deep neural networks and XGBoost under direct and recursive strategies," *Applied Energy*, vol. 253, Art. no. 113500, 2019.
- [55] S. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. NeurIPS*, 2017, pp. 4765–4774.

- [56] G. Guo, S. Yang, and Q. Li, "Attention-based LSTM for short-term load forecasting in smart grid," in *Proc. IEEE ISGT Europe*, 2019.
- [57] P. Kundur, E. Fulton, and L. R. Nielsen, "Explainable machine learning for short-term load forecasting: A SHAP-based analysis," *Electric Power Systems Research*, vol. 211, Art. no. 108340, 2022.
- [58] A. Yusuf, Y. A. Sha'aban, J. Palmer, and A. Al Mohammed, "Predictive models for short-term load forecasting in the UK's power system using a comprehensive net imports dataset," *PLOS One*, vol. 19, no. 4, Art. no. e0297267, 2024.
- [59] Y. A. Sha'aban, J. Palmer, and A. Al Mohammed, "Calculating Great Britain's half-hourly electrical demand from publicly available data," *Energy Strategy Reviews*, vol. 38, Art. no. 100743, 2021.
- [60] J. W. Taylor, "Forecasting frequency-corrected electricity demand to support frequency control," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3330–3339, 2016.
- [61] D. A. Bala, "Forecasting United Kingdom's energy consumption using machine learning and hybrid models," *Energy Exploration & Exploitation*, vol. 42, no. 1, pp. 1–26, 2024.
- [62] Y. P. Li, H. Fan, and J. Wang, "Dynamic electricity demand prediction for UK households," *Energy Procedia*, vol. 62, pp. 186–193, 2014.
- [63] J. Foster, "Electric load forecasting with increased embedded generation," Ph.D. thesis, Queen's Univ. Belfast, 2019.
- [64] B. Drysdale, "Flexible demand in the GB domestic electricity sector," Ph.D. thesis, Cardiff Univ., 2014.
- [65] A. Aboelata, "A hybrid framework for estimating daily electricity demand in UK dwellings," Ph.D. thesis, Cardiff Univ., 2023.
- [66] S. M. A. Karim *et al.*, "Machine learning and deep learning prediction models for time series: A comparative analytical study for the use case of the UK short-term electricity prices," *Energy Sustainability and Society*, 2024.
- [67] M. Qureshi, S. A. Raza, and M. Khan, "Deep learning-based forecasting of electricity consumption," *Scientific Reports*, vol. 14, Art. no. 5432, 2024.
- [68] X. Liu and Z. Lin, "Impact of COVID-19 pandemic on electricity demand in the UK based on multivariate time series forecasting with Bidirectional Long Short-Term Memory," *Energy*, vol. 227, p. 120455, Jul. 2021.
- [69] Department for Energy Security and Net Zero, *Energy Consumption in the UK* dataset, last updated 18 Dec. 2025.
- [70] M. Al Shimmari and D. Wallom, "Short-term load forecasting using UK non-domestic businesses to enable demand response aggregators' participation in electricity markets," *Oxford University Research Archive*, University of Oxford, Oxford, UK, 2023.

Appendix A: Additional Figures

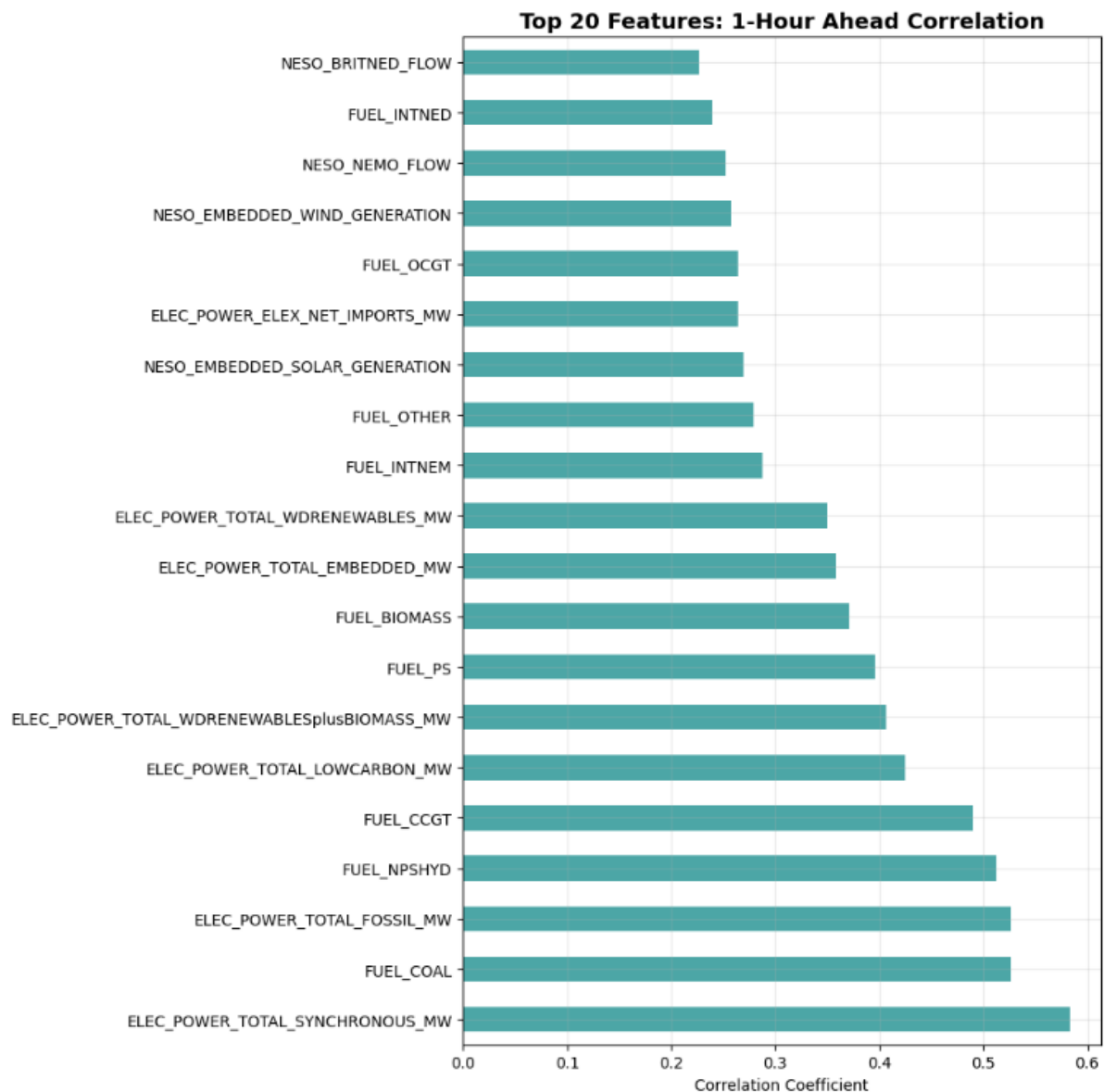


Figure A.1: Top 20 correlated input features for the LSTM H1 model

The next most significant factors impacting one hour's relay at LSTM (see Figure A.1) are all designated as the top twenty evaporation and emission data inputs for future load. To help you understand how each of these plays a role in forecasting, please note the bar graphs depict the amount of connection that exists between the collection of current generation or flow information (i.e., synchronous, coal, gas, and renewable power) and future rates of energy use. The lower the height of the bars, the higher their relationship to future demand, thus they give valuable insight into conditions on the network for the LSTM model to utilize effectively in the development of its forecast for one hour's ahead load. Key fuel types as well as the total

amount of power created are vital contributors to the short-term historical demand pattern that the LSTM applies to creating the one-hour relay prediction.

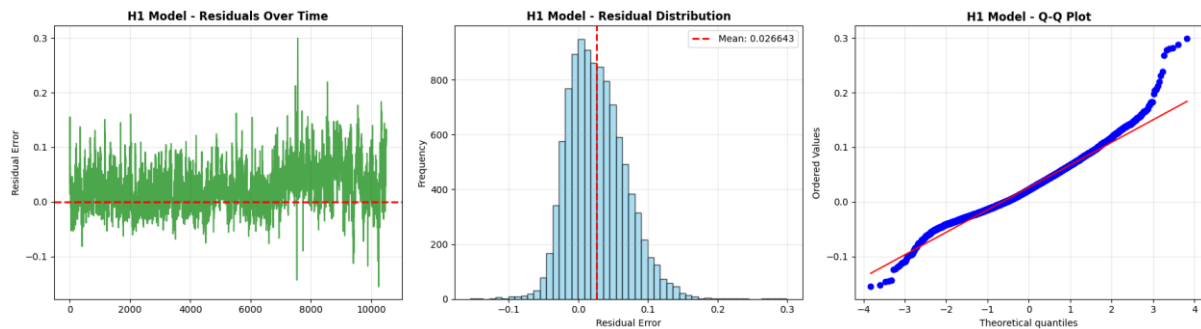


Figure A.2: LSTM H1 residual analysis

There are 3 graphs that comprise the summary of LSTM errors for 1 hour ahead. First, the graph on the left illustrates the residual values over time. Although they are mostly scattered around the zero line, there is still a high frequency of spikes or large residual errors during certain timeframes where there were either large positive or negative predictions made. Second, the histogram on the left shows that most of the residuals lie close to the middle of the graph where there is a very small positive mean value; however, there is a wide range of residual values which can explain the high RMSE and MAE values. Lastly, the Q-Q plot shows that there are many more extremes occurring in the tails than a basic Gaussian assumption would lead one to believe; therefore, the residuals do exhibit characteristics of being non-normally distributed.

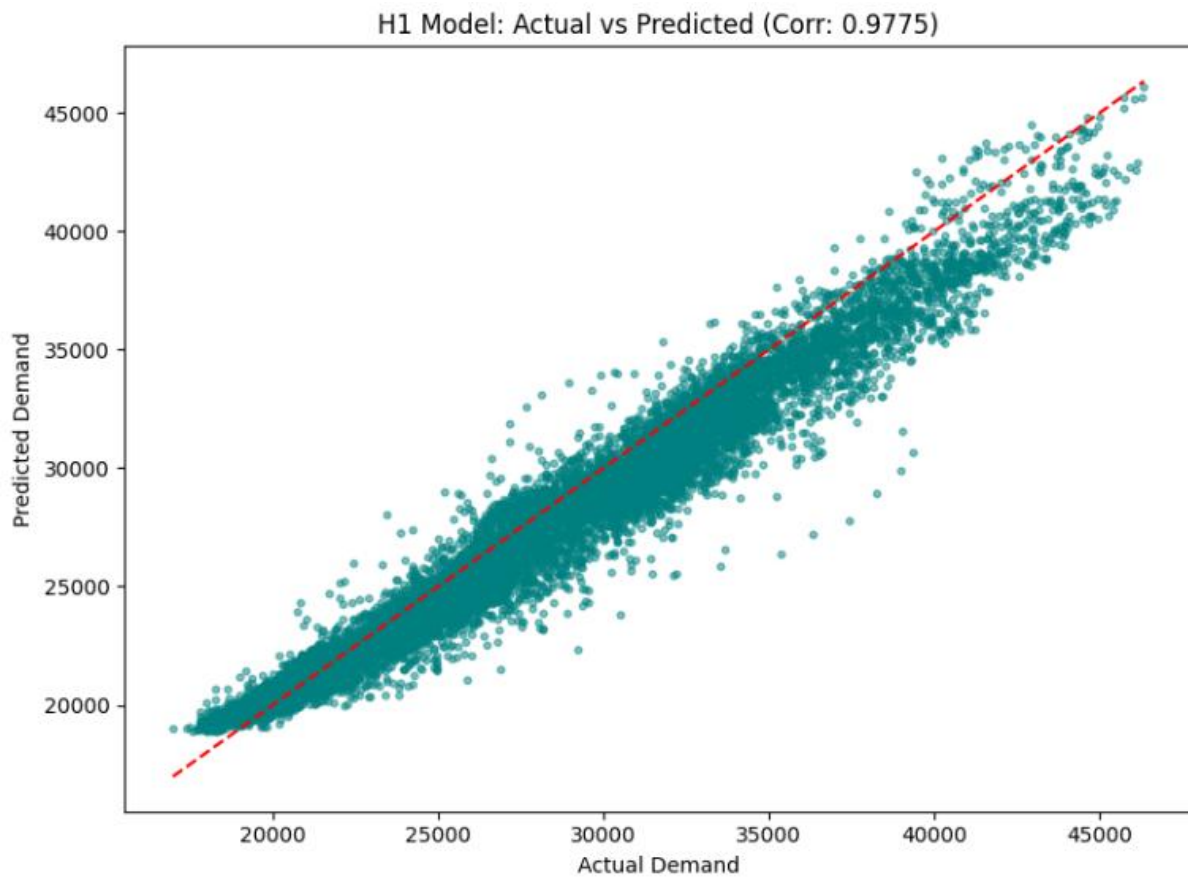


Figure A.3: LSTM H1 – scatter plot of actual vs predicted demand

In Figure A.3, the scatter plot shows the comparison of actual vs predicted demand using LSTMs for the next hour of prediction compared to the actual values. The dataset contains a point for each time frame. The dashed red line depicts the perfect correlation between the two (or perfectly matched demand and prediction). Going by their closeness to this perfect correlation line and their correlation coefficient (approx. 0.98), we would expect that the model accurately predicts actual vs predicted demand. However, as can be seen on the scatter plot, particularly at higher demands, there is still quite a bit of variance between predicted and actual due to this higher RMSE and MAE compared to both Prophet and XGBoost.

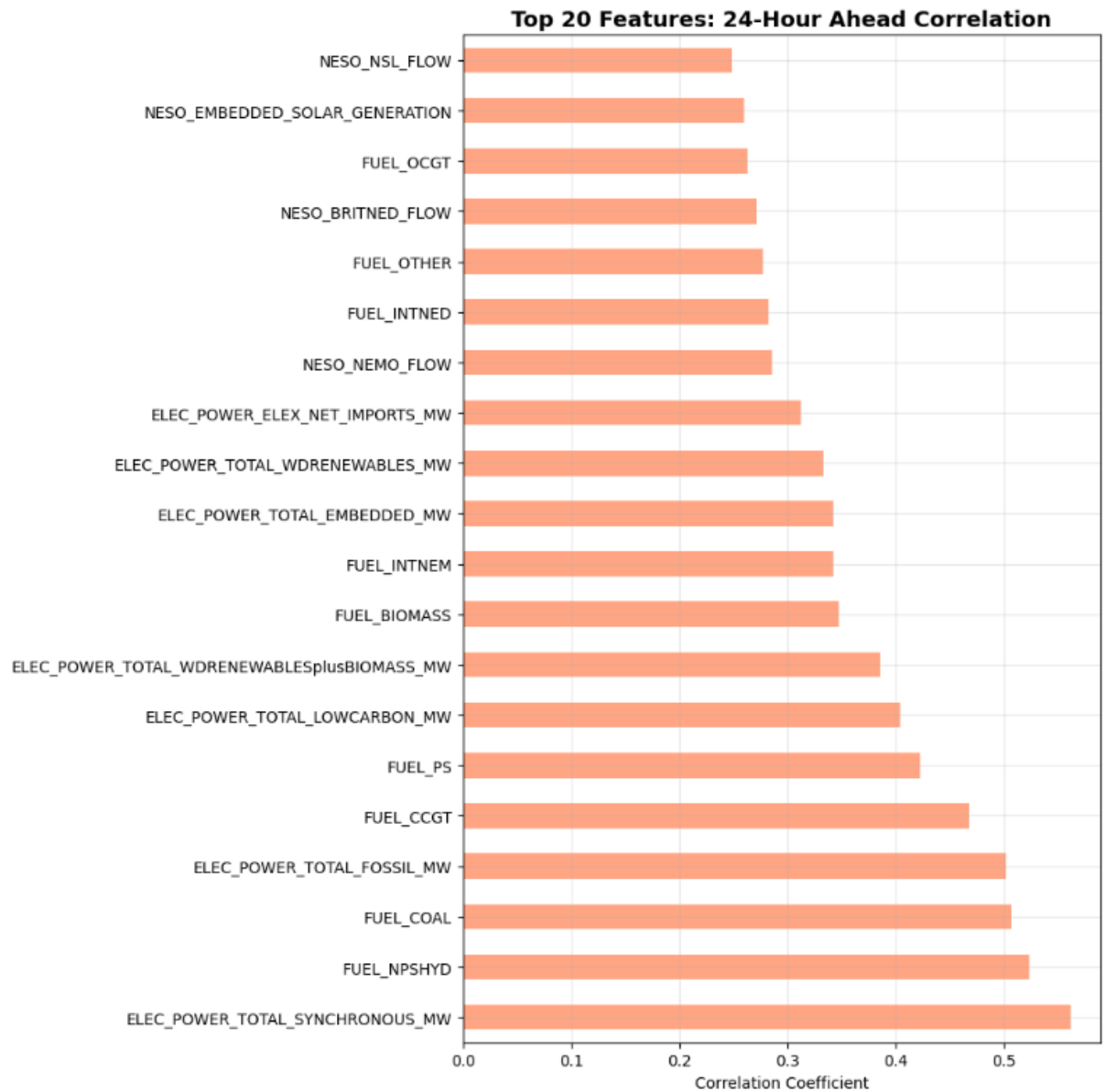


Figure A.4: Top 20 correlated input features for the LSTM H24 model

The 20 input variables used by the LSTM model which are correlated most strongly with the model's 24-hour ahead target can be found in Figure A.4. The 20 input variable relationships to the 24-hour ahead target are depicted by the 20 bars in Figure A.4. The bar lengths represent the degree of correlation (positive or negative) between the input variable and the 24-hour ahead target demand levels. The three lowest bars in Figure A.4 correspond to the three input variable types with the highest correlation coefficient (i.e., synchronous, hydro, coal and total fossil generation). As such, the LSTM model likely considers these three types of generation as having the most influence in predicting future load demand at longer timeframes.

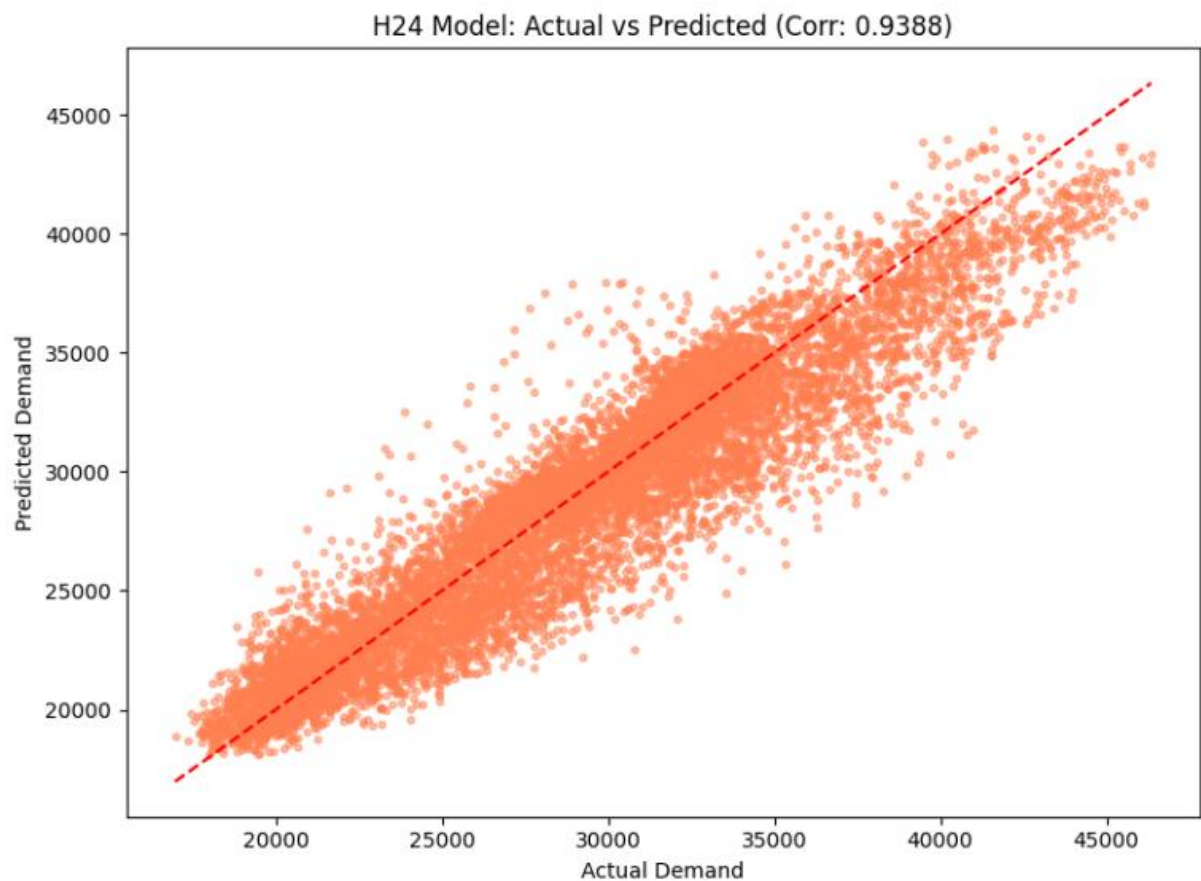


Figure A.5: LSTM H24 – scatter plot of actual vs predicted demand

LSTM 24 hour ahead predictions (Orange Dots) plotted on Figure A.5 against actual demand for all test points mostrar clear upward trend with correlation coefficient about 0.94 between 24h AHEAD predictions of LSTM vs actual demands indicating LSTM model captures a mostly linear relationship between actual vs predicted load but with a large scatter from the line of best fit; consequently errors (from LSTM) for day ahead are greater than for both Prophet and XGBoost especially with higher demand levels.

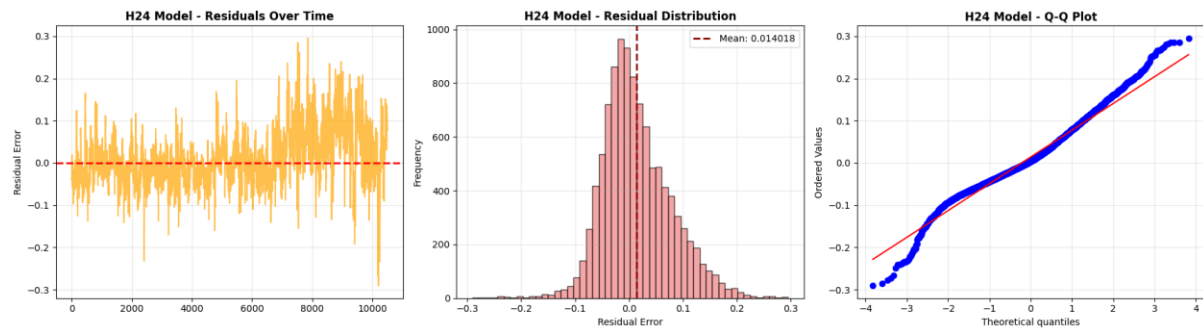


Figure A.6: LSTM H24 – residual analysis

The residuals of a Long Short-Term Memory (LSTM) predictive model at 24 hours into the future are illustrated in three separate diagnostic plots in Figure A.6. The left plot displays the residuals through time, revealing a steady increase in variability as well as periodic systematic under & over prediction from zero. The centre histogram shows residuals with an average near the upper portion of the histogram, while displaying high overall variability. The right Q–Q plot indicates that extreme residuals happen at rates higher than one would obtain from a normal distribution; therefore, they help to explain the high RMSE and MAPE for this model.

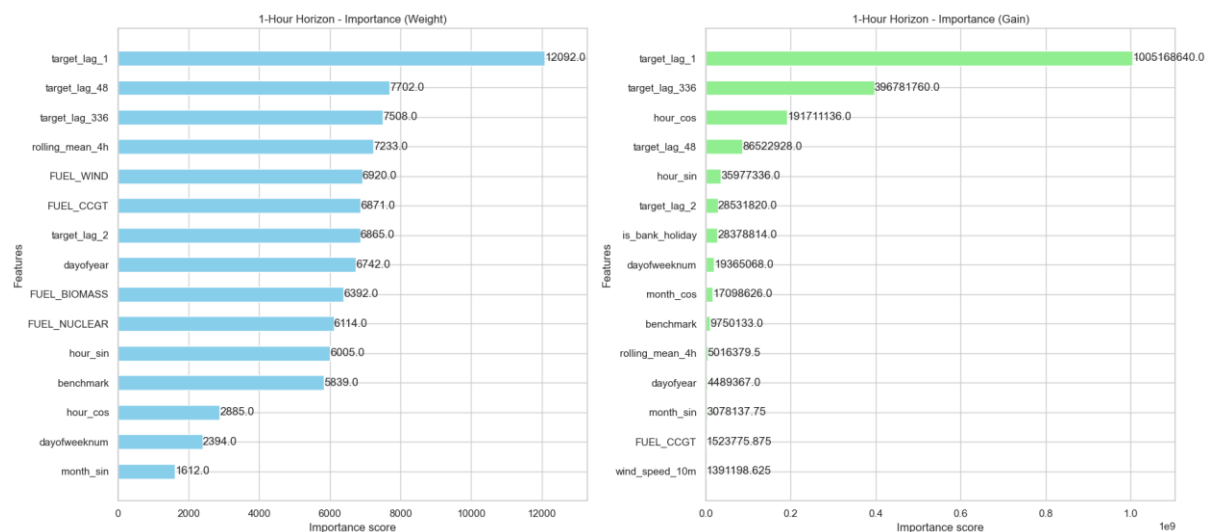


Figure A.7: XGBoost H1 feature importance (weight and gain)

Figure A.7 indicates the relative importance of the input variables for the one-hour ahead XGBoost model based on two criteria: (1) how frequently a particular variable appeared in trees, and (2) the amount of increase in performance that variable brought to the model when used. As can be seen from both graphs, the most significant input variables for this model are lagged loads from the previous hours (specifically times $t-1$ to $t-30$) and longer lagged loads (the longer-, the greater- the lag value). Additionally, the calendar variables Hour of Day, Day of Week, and some measures pertaining to Fuel are also very influential, albeit not nearly as

much as the lagged load input variables. Therefore, Figure A.7 supports that the one-hour ahead based XGBoost model does use a combination of historical demand trends/output and demand trend predictors/useful information derived from the history of the load time series.

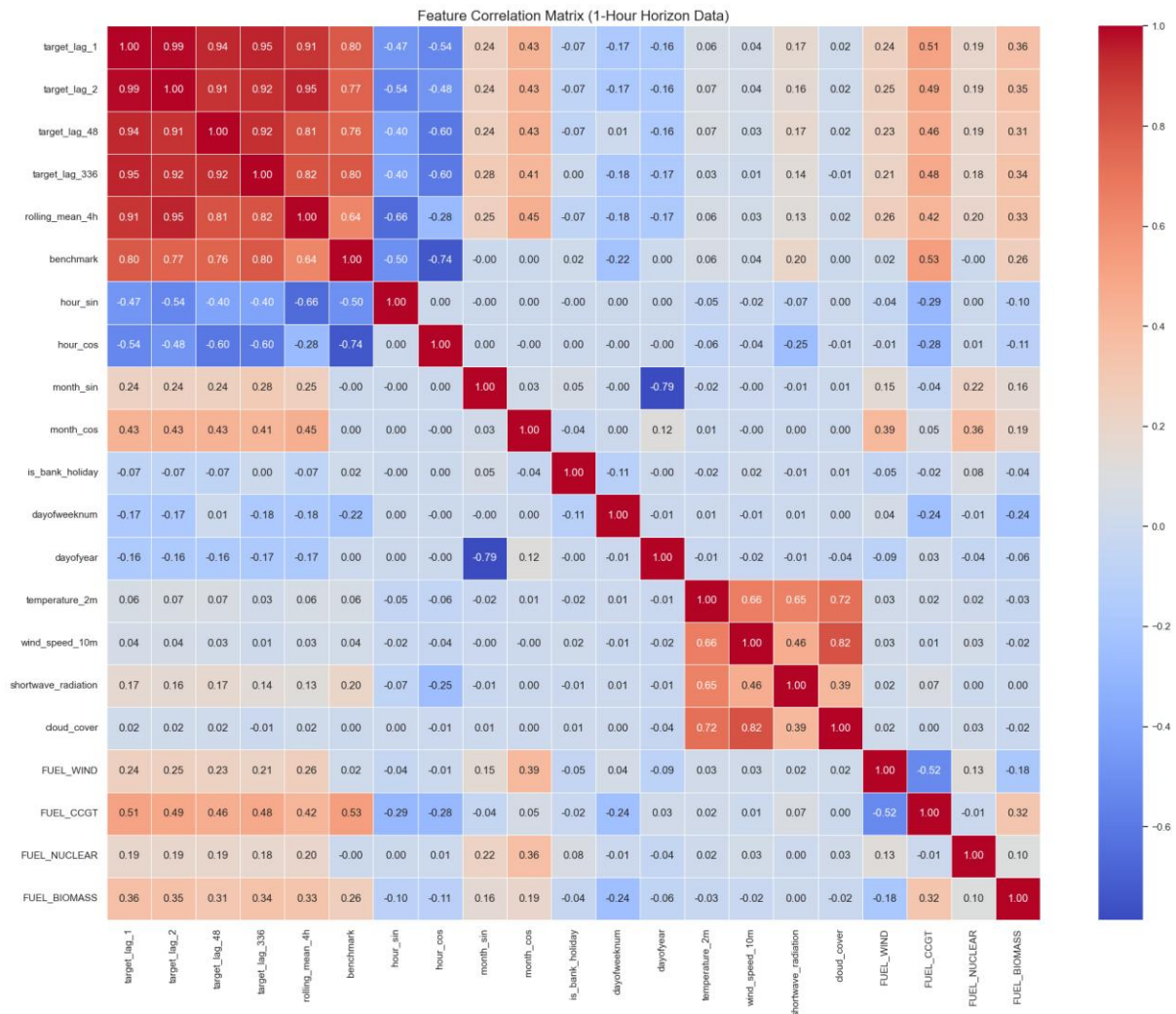


Figure A.8: Feature correlation matrix for H1 (XGBoost input data)

In figure A.8 we see a correlation between every single feature being passed into the XGBoost one hour ahead model. Darker red cells represent strong positive relationships between those two features while darker blue cells represent strong negative relationships between those two features. The recent lag targets as well as the rolling means show very high correlations to the current load, making them excellent predictors. Calendar features are less correlated to each other than weather and/or fuel variables. However, both of these feature groups provide helpful additional insights into how load patterns have changed over time.

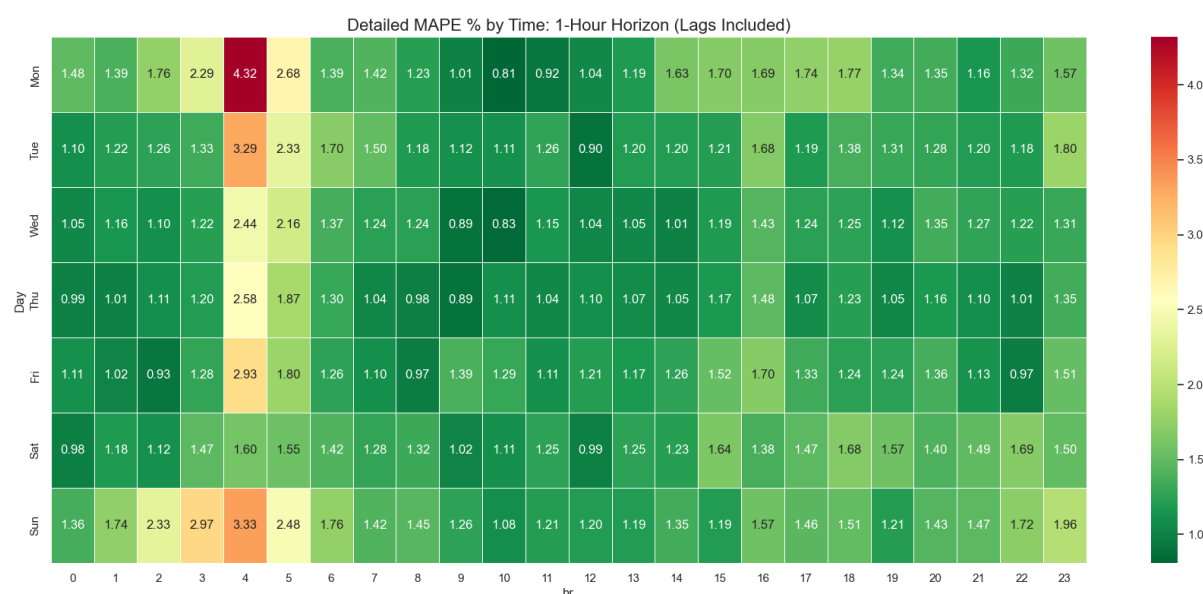


Figure A.9: Detailed H1 MAPE by hour and day (XGBoost)

The hourly mean absolute percent errors (MAPE) for XGBoost will be shown in a heat map in Figure A.9. In general, there is a dominant amount of Dark Green cells in the heat map showing lower errors in most hours, whereas, in comparison, several light yellow/orange and red cells indicate higher errors occurring at certain hours for certain days of the week. Although many hours will be unpredictable to predict by demand when the demand becomes more difficult to predict, there are only a few hours when XGBoost has lower prediction accuracy. Therefore, XGBoost has consistently shown prediction accuracy over time (H1); the only area(s) where prediction accuracy decreases will be for a small number of specific time periods.

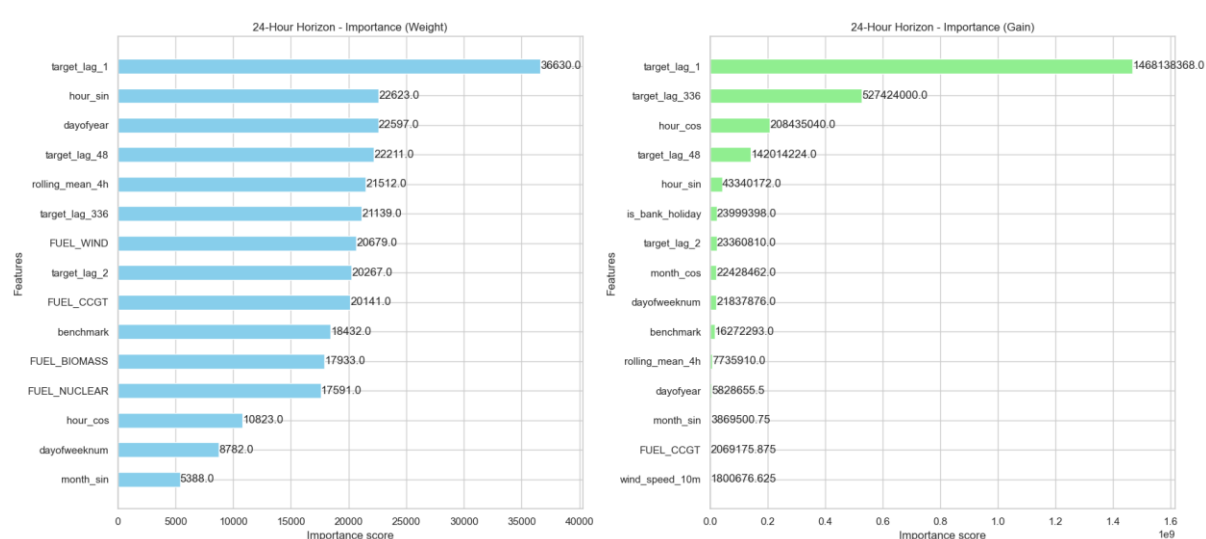


Figure A.10: XGBoost H24 – feature importance (weight and gain)

As seen in Figure A.10, the two views illustrating the importance of features to the XGBoost 24 hour ahead model show that the most important feature was how often it was included in trees (on the left side), and the amount of improvement related to that feature for H24 model (on the right side). The feature that contributed the most was the value of the most recent load, (target_lag_1) followed by other lags at a distance (336, 48) from the most recent value as well as hour of the day and day of the week calendar features. Fuel and weather variables had a small, but still significant, gain which enabled the model to better define the expected shape of the profile for the day ahead. The data shown in this figure clearly demonstrates that XGBoost models are dependent upon the most recent demand information and provide additional predictive ability through incorporating both the structure of the calendar and the operational characteristics of the system at H24.

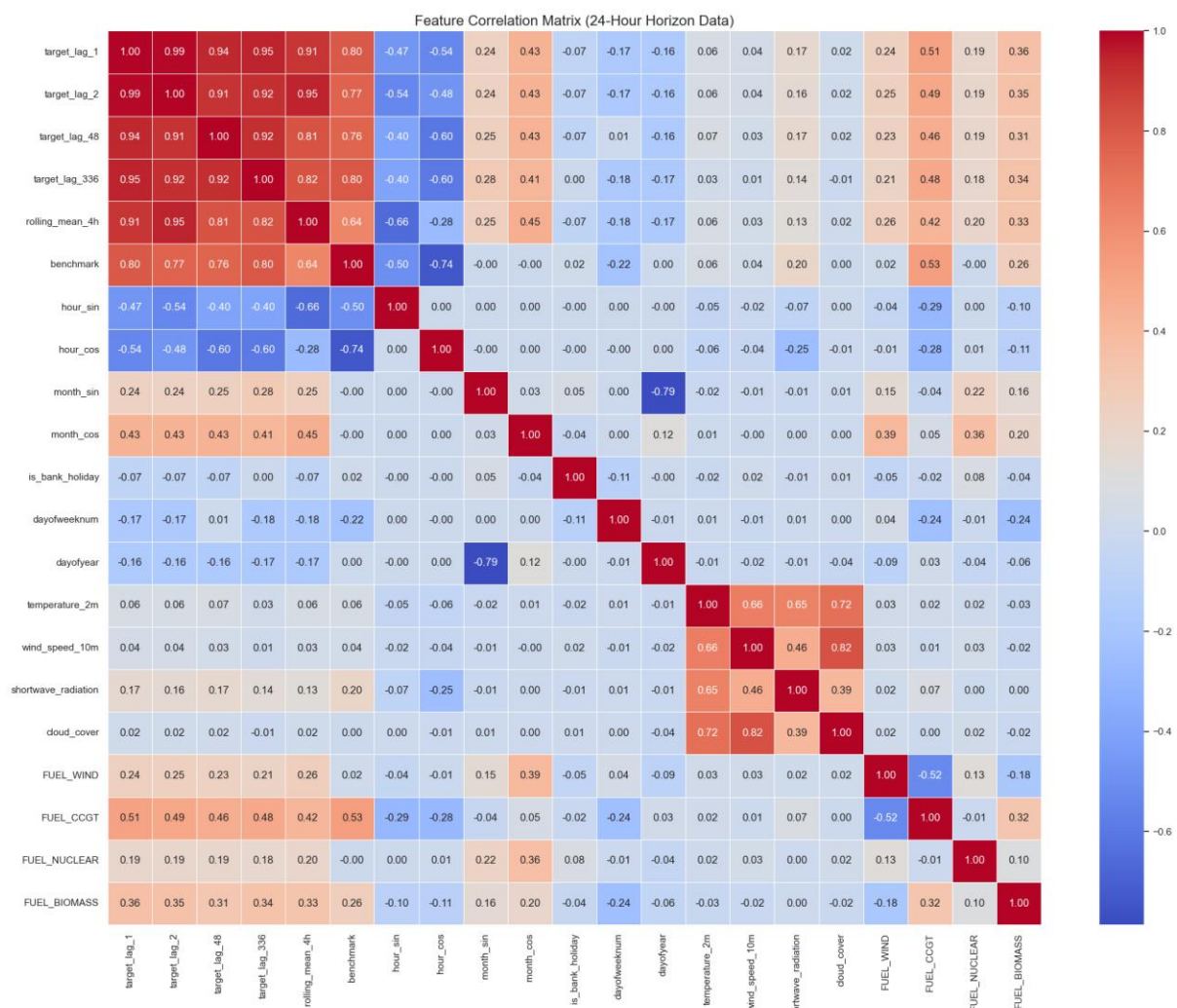


Figure A.11: XGBoost H24 – feature correlation matrix

Figure A.11 shows the correlations between all of the input features used in the XGBoost 24-hour ahead prediction models. The strong red blocks in the upper left of the matrix demonstrate

that the target lags as well as the rolling 4 hours mean are very highly correlated to one another and to the current load, thereby confirming their importance as predictors. Correlations between calendar and weather/fuel variables are much less; yet several do exhibit some degree of relationship (e.g., fuel types to clouds/radiation) that may be beneficial additional information. Overall, the correlation matrix illustrates that XGBoost utilizes a combination of features that are highly correlated as well as those that have low correlation in order for it to predict using both short-term memory and larger conditions of the system at the day-ahead time frame.

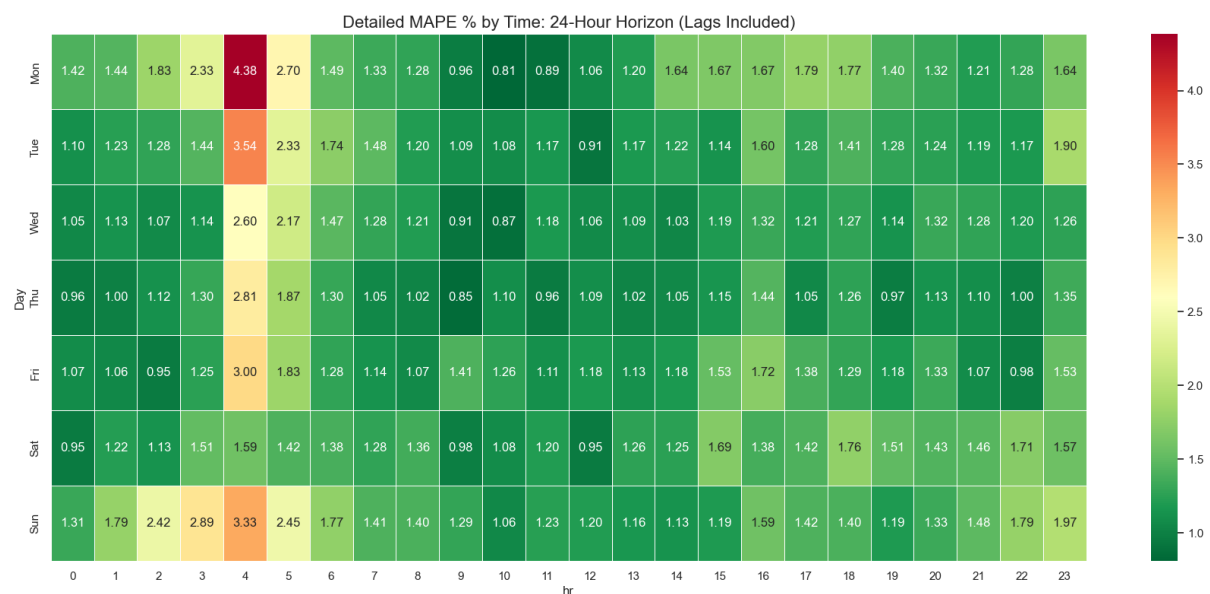


Figure A.12: XGBoost H24 – Detailed MAPE heatmap by hour and day

Most cells in the heat map are dark green or light green, which means that there was only a small percentage error for many different times or days. In addition, there is a band of high yellow-red error (greater than 20% error) between approximately 4:00 a.m. and 5:00 a.m. on weekday mornings and Sundays, during which the model is less successful in predicting electricity demand. In general, the results from the heat map indicate that XGBoost has fairly high accuracy with its 24-hour ahead (H24) electricity predictions, with only a few exceptions where errors increase to high levels.

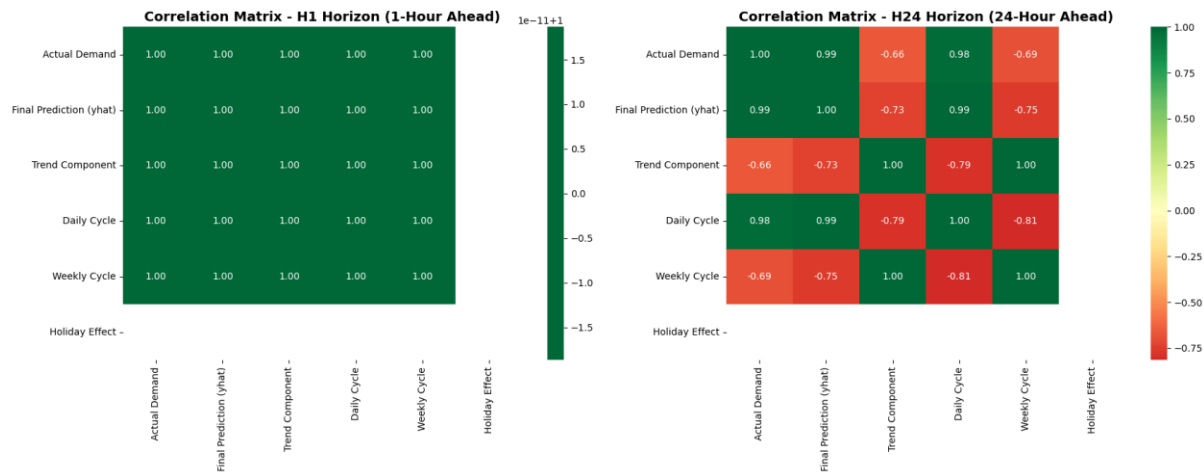


Figure A.13: Prophet component–outcome correlation at H1 and H24

How Prophet's models and demand correlates at 1-hour (H1) and 24-hours (H24) ahead is shown in Figure A.13. Utilize the left heatmap to see how, at H1, all the components (final prediction/trend/daily/weekly/holiday) have nearly complete correlation both with themselves and with Actual Demand - due to the short-term nature of the Model and the close proximity between predicted and Actual Load. The right heatmap illustrates the increased variation seen between components at the H24 timeframe, whereby the Final Prediction remains highly correlated to Actual Demand, whereas Daily & Weekly Cycle have moderate negative to strong positive correlations indicating that these seasonal components account for most of the 24-hour forecasting component structure. The Trend and Holiday components of the model have much weaker correlations to both the Final and Actual Demand component of Prophet's longer-term horizons, but they are still considered relevant in determining Prophet's overall longer term prediction behaviour.

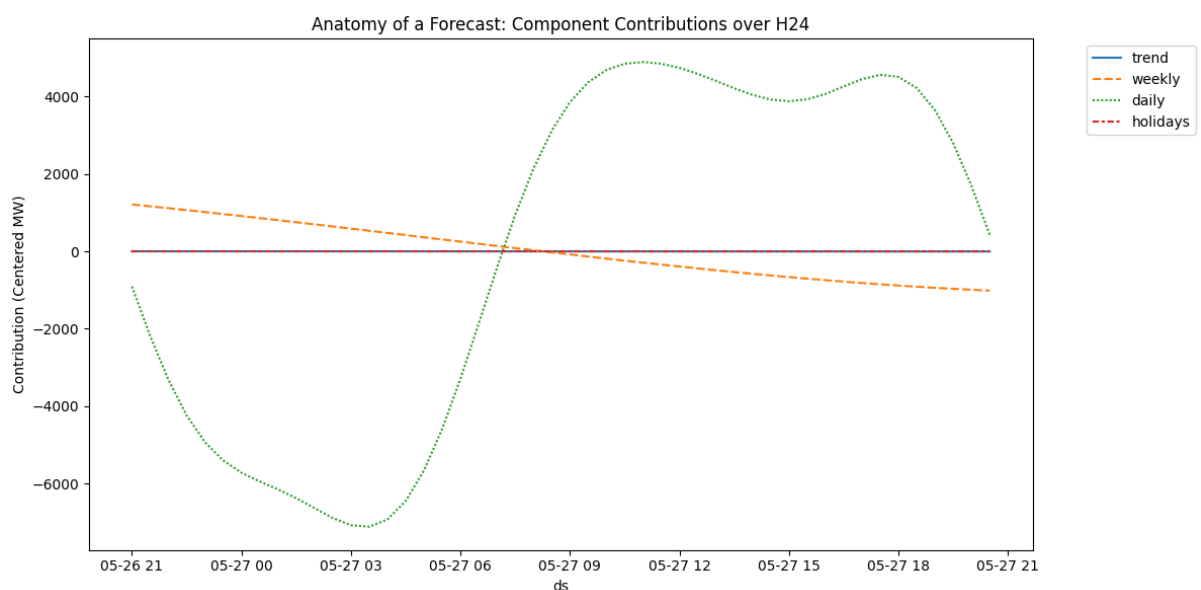


Figure A.14: Prophet H24 – Component Contributions over the Prediction Horizon

In figure A.14, Prophet is shown breaking down a single day's prediction into the four primary components which affect it: trend, weekly (seasonal) pattern, daily (seasonal) pattern, and holiday. The daily green trend dominates, causing the drop overnight and higher day-time demand levels. The orange weekly trend exhibits a much slower rate of change compared to the others and provides less of an impact on the overall trend, while the trend and holiday components have nearly no impact. This provides evidence that the H24 prediction is primarily driven by the daily cycle compared to changes associated with the other components.

Appendix B: Additional Tables

Model	Final Feature Count	Key Preprocessing	Target Setup
XGBoost/ LSTM	100+ features	Drop timestamps, encode categorical, fillna(0), numeric dtype	MultiOutputRegressor for H=2,48 steps
Prophet	6 regressors	ds= ELEC_NGEM_startTime_utc_datetim e, y=ELEC_POWER_TOTAL_ESPENI _MW	temperature_2m, embedded wind/solar, flags

Table B.1: Supervised Learning Setup

XGBoost and LSTM developments are built on large, cleaned feature sets; both contain 100 or more independent variables. The feature sets consist of many encoded categorical variables, numeric cast values, and filled in missing fields utilizing a method of gap-fill. Both use multi-output prediction techniques for 2-step and 48-step prediction processes.

Model	Strengths	Best For
Prophet	Transparent seasonality, holidays, trends	Interpretable additive patterns
XGBoost	Tabular data performance, SHAP explainability	Complex nonlinear interactions
LSTM + Attention	Long temporal dependencies, sequence patterns	Weather-driven renewable volatility

Table B.2: Modelling Strategy

Prophet's transparent trend, seasonality and holiday effects makes Prophet the best choice for interpreting additive demand patterns. XGBoost uses SHAP to interpret tabular data and is able to represent many feature interactions in a single model. LSTM with attention enables long-term dependence on many different temporal features and is a good fit for volatility driven by both weather and renewables.

Item	Value / Description
Raw rows, columns	~70k+ half-hourly & hourly rows, ~100+ original variables
Key feature groups	Temporal, weather, fuel mix, interconnectors, totals
Target variable	ELEC_POWER_TOTAL_ESPENI_MW

Table B.3: Dataset and Feature Engineering Summary

The NEFW_DT dataset is summarized. The columns in this summary demonstrate the architecture and characteristics of the NEFW_DT dataset and present an overview of primary feature sets used across all models. Readers will realize that all three experiments based on the LSTM, XGBoost and Prophet models utilize similar base targets and target groups, thus providing for a fair comparison among the models in subsequent sections.

Component	H1 / H24 Setting
Input shape	48 time steps $\times$ 91 features
Core layers	2 LSTM layers (256, 128 units) + Attention
Training setup	70/15/15 split, 15 epochs, batch 64

Table B.4: LSTM-Attention Model Configuration (H1 and H24)

The above table provides an overview of how both the 1 hour and 24 hour Ahead LSTM models use similar architectures in terms of actual Data Sequencing Length, Number of Layers in

model structure etc. Therefore, rather than including low level configuration data (i.e. Network Architecture), the main write-up will concentrate on the performance and interpretative elements of the respective models.

Aspect	Details
Engineered features	Lags (1, 2, 48, 336), 4-hour rolling mean, benchmark
Additional predictors	Time cycles, holiday flag, weather, fuel mix
Split sizes (H1 example)	70% train, 15% validation, 15% test (~69k rows total)

Table B.5: XGBoost Feature Set and Splits

This table documents the final features used along with the dates (timeframes) for XGBoost baselines at each horizon's date (timeframe(s)). The results show tree-based modeling relies more on memory and seasonality trends compared to those based on deep-learning models that are actually trained using time-split-based data (as per their respective "Chronological" layout).

Layer / Component	Configuration	Purpose
Input	Shape (48, 91)	24 hours of history, 91 features per step
LSTM 1	256 units, return_sequences=True	Capture short-term temporal patterns
Dropout 1	0.25	Reduce overfitting on first LSTM layer
LayerNorm 1	LayerNormalization	Stabilise activations over time steps

Layer / Component	Configuration	Purpose
LSTM 2	128 units, return_sequences=True	Learn higher-level sequence representations
Dropout 2	0.20	Additional regularisation
LayerNorm 2	LayerNormalization	Normalise second LSTM outputs
Attention	Custom AttentionLayer	Focus on most informative time steps
Dense	64 units, ReLU	Non-linear projection to compact features
Dropout 3	0.20	Regularise dense representation
Output	Dense(1)	Predict demand at chosen horizon
Optimizer	Adam, lr = 1e-3	Stable gradient-based training
Training	70/15/15 split, 15 epochs, batch size 64	Consistent temporal training regime

Table B.6: LSTM-Attention Architecture (H1 and H24)

Included in this table are complete details on the LSTM Attention network stack configuration and training for the 1 and 24 hour prediction models. These models combine recurrent layers, normalisation and attention in order to capture the temporal characteristics of the multivariate input sequence.

Hyperparameter	Example value (from tuning)	Purpose in electricity-demand model
n_estimators	e.g. 300–500	Number of boosting trees

Hyperparameter	Example value (from tuning)	Purpose in electricity-demand model
max_depth	e.g. 4–8	Controls tree complexity / interactions
learning_rate	e.g. 0.03–0.1	Step size for each new tree
subsample	e.g. 0.7–0.9	Row sampling for robustness
colsample_bytree	e.g. 0.7–0.9	Feature sampling per tree
min_child_weight	e.g. 1–5	Minimum leaf “mass”, combats overfitting
gamma	e.g. 0–2	Split penalty, encourages simpler trees
reg_lambda	> 0	L2 regularisation on leaf weights
Objective	“reg:squarederror”	Squared error loss for regression
Wrapper	MultiOutputRegressor	Direct multi-step prediction (t+1...t+H)

Table B.7: XGBoost Key Hyperparameters (H1 and H24)

This table presents an overview of the most important XGBoost hyperparameters that were adjusted through a random search. Additionally, it provides details about how each hyperparameter affects the balance between accuracy, complexity, and generalisation for both the short term and the long-term demand predicting models.

Component	Configuration / Setting	Purpose
Growth	Linear (default)	Capture overall long-term trend
Seasonality	Daily, weekly, yearly	Model strong intra-day and calendar cycles

Component	Configuration / Setting	Purpose
Holidays	UK bank holiday indicators from dataset	Capture demand shifts on special days
Additional regs.	Selected weather or grid variables as regressors	Incorporate exogenous effects on demand
Changepoints	Automatic detection (default settings adjusted if needed)	Allow trend breaks over 4-year period
Training window	2020-01-01 to prediction cut-off	Fit on full historical series

Table B.8: Prophet Components and Configuration

This table presents the prophet configuration used as a transparent statistical baseline for the production of interpretable electrical demand predictions. The configurations represent how trends are combined with multiple seasonality, holidays and regressors in order to provide electrical demand predictions that can be interpreted.