REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

ENHANCED ENERGY DEMAND MANAGEMENT IN ELECTRIC DISTRIBUTION NETWORKS USING LSTM-XGBOOST MODEL

Falah Hasan Dakheel DAKHEEL

Doctor of Philosophy

Supervisor

Asst. Prof. Dr. Mesut ÇEVİK

İstanbul, 2026

ENHANCED ENERGY DEMAND MANAGEMENT IN ELECTRIC DISTRIBUTION NETWORKS USING LSTM-XGBOOST MODEL

Falah Hasan Dakheel DAKHEEL

Electrical and Computer Engineering

Doctor of Philosophy

ALTINBAŞ UNIVERSITY

2026

The dissertation titled ENHANCED ENERGY DEMAND MANAGEMENT IN ELECTRIC DISTRIBUTION NETWORKS USING LSTM-XGBOOST MODEL prepared by FALAH HASAN DAKHEEL DAKHEEL and submitted on 12/01/2026 has been **accepted unanimously** for the degree of PhD in Electrical and Computer Engineering.

Asst. Prof. Dr. Mesut ÇEVİK
Supervisor

Thesis Defense Committee Members:

Asst. Prof. Dr. Mesut ÇEVİK	Department of Electrical-Electronics Engineering, Altınbaş University	
Asst. Prof. Dr. Timur İNAN	Department of Computer Engineering, Marmara University	
Asst. Prof. Dr. Hameed Mutlag Farhan FARHAN	Department of Computer Engineering, Altınbaş University	
Asst. Prof. Dr. Ameer Yalmaz Asaad ASAAD	Department of Computer Technologies, İstanbul Topkapı University	
Assoc. Prof. Dr. İzzet Paruğ DURU	Department of Medical Services and Techniques, İstanbul Gedik University	

I hereby declare that this dissertation meets all format and submission requirements for a PhD dissertation.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Falah Hasan Dakheel DAKHEEL

Signature

DEDICATION

To my parents, whose unconditional love and support have made this achievement possible. To my beloved wives, for their patience and unwavering encouragement throughout this journey. To my children, who inspire me every day to become a better version of myself. To my siblings, for their constant presence and support. To my dear friends, for their unwavering belief in me. And to my professors and advisors, whose guidance and mentorship were invaluable.

I humbly dedicate this thesis to all of them.

PREFACE

This dissertation includes content that has previously appeared in a peer-reviewed publication authored by the candidate:
Dakheel, F., & Çevik, M. (2025). Optimizing Smart Grid Load Forecasting via a Hybrid LSTM-XGBoost Framework: Enhancing Accuracy, Robustness, and Energy Management. *Energies*, 18(11), 2842. https://doi.org/10.3390/en18112842

Relevant content from the publication has been adapted throughout the dissertation to align with academic formatting and structural requirements. The reuse of this material complies with the Creative Commons Attribution (CC BY 4.0) license and adheres to academic integrity standards.

ABSTRACT

ENHANCED ENERGY DEMAND MANAGEMENT IN ELECTRIC DISTRIBUTION NETWORKS USING LSTM-XGBOOST MODEL

DAKHEEL, Falah Hasan Dakheel

Ph.D., Electrical and Computer Engineering, Altınbaş University

Supervisor: Asst. Prof. Dr. Mesut ÇEVİK

Date: 01/2026

Pages: 88

Accurately forecasting electricity consumption is essential for improving the reliability, operational efficiency, and resilience of energy systems, particularly given the rising integration of renewables and the advancing complexity of smart grids. This research introduces an innovative multilayered model for short-term electric load forecasting that combines Long Short-Term Memory (LSTM) networks with Extreme Gradient Boosting (XGBoost) to enhance accuracy and dependability.

The approach consists of three main steps: data cleaning and feature extraction, development of separate LSTM and XGBoost models, and combining them into a unified hybrid architecture. The Elia Grid dataset from Belgium was used in this study, containing high-resolution load data for 2022 captured at 15-minute intervals. The hybrid model leveraged the LSTM's strength in learning sequential dependencies, while XGBoost contributed by capturing non-linear residual patterns and extracting feature importance.

The proposed model underwent extensive testing to evaluate its performance against independent LSTM and XGBoost models. The hybrid model achieved its best results with a Root Mean Square Error (RMSE) of 106.54 MW and Mean Absolute Percentage Error (MAPE) of 1.18% and Coefficient of Determination (R^2) of 0.994. The sensitivity analysis showed that increasing the look-back window size improved model performance but the attention mechanisms did not enhance accuracy so they were removed from the final design.

In addition to outperforming traditional models, the proposed framework demonstrated strong generalizability, scalability, and interpretability, making it suitable for real-time energy management systems. The model was benchmarked against recent classical and deep learning (DL) models and showed competitive or superior results across multiple datasets.

This work contributes to the design of AI-based applications in smart grid management and enriches the literature on hybrid DL techniques for time series forecasting. Future research will explore expanding the model to multi-regional datasets, incorporating weather and socioeconomic variables, and deploying the system in real-world grid control environments.

Keywords: Energy Forecasting, Smart Grid, LSTM, XGBoost, Hybrid Model, Deep Learning, Time Series.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

ACF	:	Autocorrelation Function
AI	:	Artificial Intelligence
ARIMA	:	Autoregressive Integrated Moving Average
CEEMDAN	:	Complete Ensemble Empirical Mode Decomposition with Adaptive Noise
DL	:	Deep Learning
GAN	:	Generative Adversarial Network
GRU	:	Gated Recurrent Unit
IoT	:	Internet of Things
LLM	:	Large Language Model
LR	:	Linear Regression
LSTM	:	Long Short-Term Memory
MAE	:	Mean Absolute Error
MAPE	:	Mean Absolute Percentage Error
ML	:	Machine Learning
MW	:	Megawatt
PACF	:	Partial Autocorrelation Function
R^2	:	Coefficient of Determination
ReLU	:	Rectified Linear Unit
RMSE	:	Root Mean Square Error
RNN	:	Recurrent Neural Network
SARIMA	:	Seasonal ARIMA
SCADA	:	Supervisory Control and Data Acquisition
SMOTE	:	Synthetic Minority Over-sampling Technique

SVM	:	Support Vector Machine
TCN	:	Temporal Convolutional Network
VMD	:	Variational Mode Decomposition
XGBoost	:	Extreme Gradient Boosting

LIST OF SYMBOLS

X	:	Original Value (Before Normalization)
X_{norm}	:	Normalized Value Using Min-Max Scaling
y_t	:	Actual Observed Value at Time t
$\hat{y_t}$	:	Predicted Value at Time t
e_t	:	Residual Error at Time t
$\hat{y}_{final}$	:	Final Prediction Output of the Hybrid Model
$\hat{y}_{LSTM}$	:	Intermediate Prediction from the LSTM Model
f_{xgb}	:	Regression Function Learned by XGBoost
N	:	Number of Observations or Window Size (Used in RMSE, MAPE, and Rolling Statistics)
h_t	:	Hidden State of LSTM at Time t
x_t	:	Input Feature at Time t
h_{t-1}	:	Hidden State at Previous Time Step $t-1$
$f_k(x)$	:	Prediction of the k-th Decision Tree in XGBoost
T	:	Total Number of Trees in XGBoost
$\bar{y}$	:	Mean of Actual Values, Used in R^2 Formula

1. INTRODUCTION

1.1 BACKGROUND OF THE STUDY

The ability to extract valuable information from evolving datasets has become essential for all industries because data dominates our current world. Time series data holds significant importance because they contain timing information and frequently appear in real-world scenarios. The industrial sector's advancement has led to rapid data creation particularly time series data which serve essential functions across finance and healthcare and climate prediction and energy management and other domains [1], [2]. Time series forecasting serves as the essential tool for maximizing resource management efficiency and organizational productivity and operational decision facilitation. The energy sector requires demand forecasting to achieve proper demand-supply management and renewable integration and power failure risk control.

In the coming years, power consumption is likely to go up as the economy grows and the population expands in addition to the advent of smart grid technologies [3]. Smart grids are characterized by the combination of energy and communication tech which enables them to monitor in real time and thus make the supply more efficient and the system more robust. The merging of smart grids leads to complications due to the very nature of power supply being influenced by factors such as varying demand from electric vehicles and the different outputs of various energy sources. This also means the variability of consumer energy consumption [4]. The technique of load forecasting (LF) has become an unremovable component of energy systems. It participates in demand management and infrastructure planning very effectively and efficiently so indirectly it gives way to reliable and cheap power generation [5].

The progression of computational efficiency has made it possible for ML and DL to become the dominant techniques in time series forecasting. In difficult non-linear situations with highly fluctuating data, ML-based methods perform much better than the traditional linear models such as ARIMA and SARIMA [6]. The particular RNN model called LSTM networks is very good at the process of uncovering sequential patterns and time relations in data. The two main problems that still restrict its universal adoption are the model’s comprehensiveness and interpretability of the results [7], [8].

The current research presents a hybrid framework which unites LSTM's sequential learning capabilities with XGBoost's gradient boosting capabilities. The combined model targets improved forecasting accuracy and robustness through energy load prediction as its representative application.

1.2 PROBLEM STATEMENT

Energy industry is in the process of a change to a new model led by the growing use of renewable energy sources and digitalization of infrastructure [9]. Low carbon energy sources like solar and wind energy are stochastic in nature; hence, demand prediction is paramount for achieving balance and cost optimality for the power network [10]. Standard forecasting approaches are not particularly effective in adequately representing energy consumption behaviors as these are vibrant, nonlinear models and therefore do not efficiently predict energy consumption and therefore distribution and management infrastructure [11].

While methods in DL models like LSTMs suggest solutions, overfitting is still a challenge, and such models exhibit inefficiencies when scaling to large systems to accommodate high volumes of data [12]. To overcome these limitations, there have been attempts to integrate the two or combine some of the best features of both models [13]. The combined models have shown promise in producing better results in modeling both perfect and imperfect data [14], [15]. Nonetheless, the advancement and refinement of such models remain in nascent phases, necessitating additional investigation [16].

Moreover, the non-uniform and irregular features of energy production combined with the oscillating patterns of electric consumption pose a problem in keeping a real-time equilibrium between energy supply and demand [17]. The absence of financially feasible huge energy storage alternatives is still another problem, which leads to the issue of energy being wasted or fossil fuels being burned inefficiently during peak demand periods.

Foreseeing the near-future demand of energy very accurately is a very critical element in today's constantly changing smart grid scenario. Not only does it ensure the robustness of the system, but also it leads to the reduction of the overall power used and aids the management of energy distribution and generation [18]. In this research, the topic of integrated energy systems (IES) is a bit broader, but still, it mainly revolves around the short-term electricity demand prediction in smart grid properties.

1.3 RESEARCH OBJECTIVES

The principal objective of this investigation is to develop an integrated predictive framework that accurately interprets forecasts through the synergy between LSTM networks and the XGBoost algorithm, thus making it applicable to different fields.

a. Demarcating advantages and restrictions of time series models through performance evaluation on energy dataset.

b. Implementation of a new hybrid LSTM-XGBoost framework that will be superior in performance and will take advantage of temporal dependencies and feature importance.

c. The proposed model will be put to the test using real energy data sets for comparing its performance with standalone and conventional models.

d. The study will showcase the hybrid model's ability to be used not only in practical scenarios but also in other areas.

1.4 RESEARCH QUESTIONS

The research will be conducted with the guidance of the following questions:

a. In what ways does the combined LSTM-XGBoost model stand out from separate models with regards to the factors of accuracy and efficiency?

b. Which preprocessing and feature selection methods are the best for time series forecasting?

c. In what way does the length of the look-back window impact the forecasting power of the hybrid model in short-term load prediction?

d. What are the reasons that the attention mechanism does not have a positive effect on the accuracy of hybrid LSTM-XGBoost models, and what circumstances might allow for it to be advantageous?

1.5 SIGNIFICANCE AND CHALLENGES OF THE STUDY

The results of this research have a great impact on the future development of both academic and practical use of the time series forecasting. Combining the sequential learning power of

LSTM and the interpretability and robustness of XGBoost, a hybrid framework is introduced which is really helpful for researchers who want to go beyond the restrictions of separate models. The hybrid model constructed in this research can become the basis for later improvements that would try to achieve a good trade-off among precision, robustness, and clarity in Machine Learning (ML) forecasting models.

Short-term load forecasting is mainly a very important factor for the energy sector where it is indispensable to the management of the grid, the integration of renewable resources, and the saving of operating costs [18]. In this way, the better forecast accuracy obtained from the hybrid model is directly linked to dependable planning, resource distribution, and to minimizing energy waste. Apart from this, the model is able to deal with intricate, non-linear, and noisy time series data, thus it could be applied in other areas where forecasting is a necessity, such as technology monitoring, financial market forecasting, and climate variation studies.

The presented framework is not limited to the mentioned areas but it opens new possibilities for other fields that mainly depend on time series data. The applications in healthcare include predicting patient vital signs or disease progression, in finance the forecasting of stock prices and market trends, and finally in climate science by modeling weather patterns and climate change [19].

Forescasting through time series is a powerful yet difficult task. A large number of problems are encountered in the energy domain, for example, noise in the data and missing values, which are present in original datasets and cause difficulties in the model's estimation. One more challenge that can be mentioned is the huge dimensionality; many researchers have indicated that the existence of very correlated variables distorts the results and leads the model to be overly complex with respect to the categories. Another aspect of the problem is that different Azure ML models need to be trained for different datasets and that is why performing cross-validation is essential. Last but not the least, the issue of computational complexity, particularly the issue of DL models being resource-intensive, is often addressed by hybrid frameworks [19].

The hybrid LSTM-XGBoost model overcomes these issues by merging the LSTM's capabilities in time series forecasting with the feature importance assessment of XGBoost. The main breakthroughs are:

a. Attention Mechanisms: These mechanisms give emphasis to the time steps that are relevant, thus accuracy and interpretability are both improved.

b. Dynamic Feature Selection: XGBoost dynamically selects the most important features, which makes the model stronger.

c. Ensemble Learning: The merging of LSTM and XGBoost lowers variance and thus increases the reliability of the model.

Consequently, the study fills a large methodological void and coincides with the global movement of smart and data-driven infrastructure systems, which is the main point of the study. The investigation is designed in a manner that it can be utilized for different forecasting issues and thus is a solution with scalability and adaptability.

1.6 METHODOLOGY OVERVIEW

The methodology is based upon three major phases:

a. The historical database is made up of raw energy load data that needs to be aggregated, standardized, or normalized first before any kind of analysis can be performed. The method employs advanced interpolation techniques while maintaining the integrity of the data since it lacks some variables.

b. The model employs LSTM for capturing temporal dependencies and refines the predictions with XGBoost. The attention mechanism went through experimental evaluation but was dismissed for the final model due to its performance constraints.

c. The proposed hybrid model is assessed by means of RMSE, MAPE, and R^2 metrics while being juxtaposed with the standalone LSTM and XGBoost models. The model is subjected to further testing which embraces the investigation of the impact of the look-back window size on the results, performance evaluation of the attention mechanism and comparison with contemporary studies to ascertain its robustness and generalization.

1.7 THESIS OUTLINE

This thesis is structured as follows:

A thorough evaluation of existing literature in Chapter 2, particularly related to energy and smart grid systems, is undertaken and it discusses time series forecasting techniques ranging from traditional statistical models to the latest ones that are of ML and DL. The spotlight is placed on hybrid and ensemble methods, attention mechanisms, recent advancements like meta-learning, and a few other innovations that are related.

In the third chapter, the authors elaborate on the various stages involved in the construction of the suggested hybrid LSTM-XGBoost model, which includes data preprocessing, exploratory data analysis and feature engineering—lag variables, rolling statistics, and time-based features among others—model structure and training parameters. They also give the justification for the choice of the Elia Grid dataset, together with a description of the benchmarks (RMSE, MAPE, R^2) that are employed during the model evaluation process.

In Chapter 4, we reveal the findings made by the individual LSTM and XGBoost models which are then compared to those from the combined model. Also, a sensitivity analysis is conducted to evaluate the influence of both the look-back window size and the attention mechanisms on the model performance. The analysis is supported by the comparison of results with past benchmark studies and the application of various visualizations like time series plots, scatter plots, residual plots, histograms, and boxplots to represent the data.

Initially, an assessment of the results is provided in Chapter 5, however, the conclusions and suggestions for future research follow. We discuss the hybrid model's application in prediction from the standpoint of both theory and practice and not only offer hints on how to make the model more flexible and comprehensible but also effective in real-time contexts.

Contained in Appendix A is a sample from the Elia Grid dataset, along with the corresponding tables that were utilized in the development and evaluation of the model.

2. LITERATURE REVIEW

2.1 FORECASTING TECHNIQUES

Time series forecasting is among the top-ranking analytical methods that find application in various industries such as energy, finance, and healthcare. By analyzing historical data, companies are able to predict future trends and events, therefore, supporting their planning and decision-making processes. The ML and DL techniques have evolved the traditional ARIMA models to such a level that they are now considered to be the best forecasting solutions for different types of forecasting needs.

This chapter gives an extensive review of ML and DL algorithm-based time series forecasting techniques. It, furthermore, elaborates on the role of the LSTM networks in the learning paradigms of supervised and unsupervised—for instance, the hybrid model approach combined with various models to give better prediction, encoder-decoder structure, and so on.

A time series consists of data points that are recorded at the same intervals of time [20]. The primary goal of the time series analysis is to bring out the patterns, trends, and seasonal cycles in the data through time, which is a key method for such things as modeling the economy, predicting the market, and estimating energy consumption. But, working with actual data sets is far from easy since they are subjected to noise, incomplete information, and complex nonlinear patterns [21].

Researchers apply not only short-term but also long-term forecasting methods to overcome these issues. The use of statistical models such as AutoRegressive Integrated Moving Average (ARIMA) together with modern DL models like LSTM has shown great promise in capturing the temporal dependencies within the data. The application of supervised learning models is successful by training with labeled input-output pairs and manipulating regression models, decision trees, neural networks, exponential smoothing techniques, and ARIMA [22].

The ensemble method XGBoost, combining the three often referred to as the most powerful techniques, is still a great choice due to the fact that it unites several weak learners into a strong and accurate predictive model which performs very well even in complicated and noisy settings [23].

The marking of features or artefacts through unsupervised learning methods occurs without the use of training data since these methods identify hidden patterns in the data. The marking of features or artefacts through unsupervised learning methods occurs without the use of training data since these methods identify hidden patterns in the data. The applied methods include clustering algorithms such as K-means for grouping similar data points and anomaly detection for spotting rare events in the data [24], [25]. These techniques are good for analyzing data in order to find suspicious patterns, which are often found in temporal datasets produced by IoT systems or communication networks.

2.2 DEEP LEARNING AND HYBRID MODELS

Time series forecasting has adopted DL models as its primary choice because these models extract data features without requiring extensive feature engineering. The two most widely used models for sequence prediction are LSTM networks and GRU networks [26].

The Recurrent Neural Network variant LSTM maintains sequence data dependencies through its long-term memory system. The accuracy of LSTM exceeds traditional econometric models such as ARIMA when dealing with big data profiles that contain complex non-linear relationships [27]. LSTM architectures have shown effective application in forecasting electricity demand as well as modeling financial time-dependent data. The architectural structure and gating mechanism are further illustrated in Figure 2.1

[29].

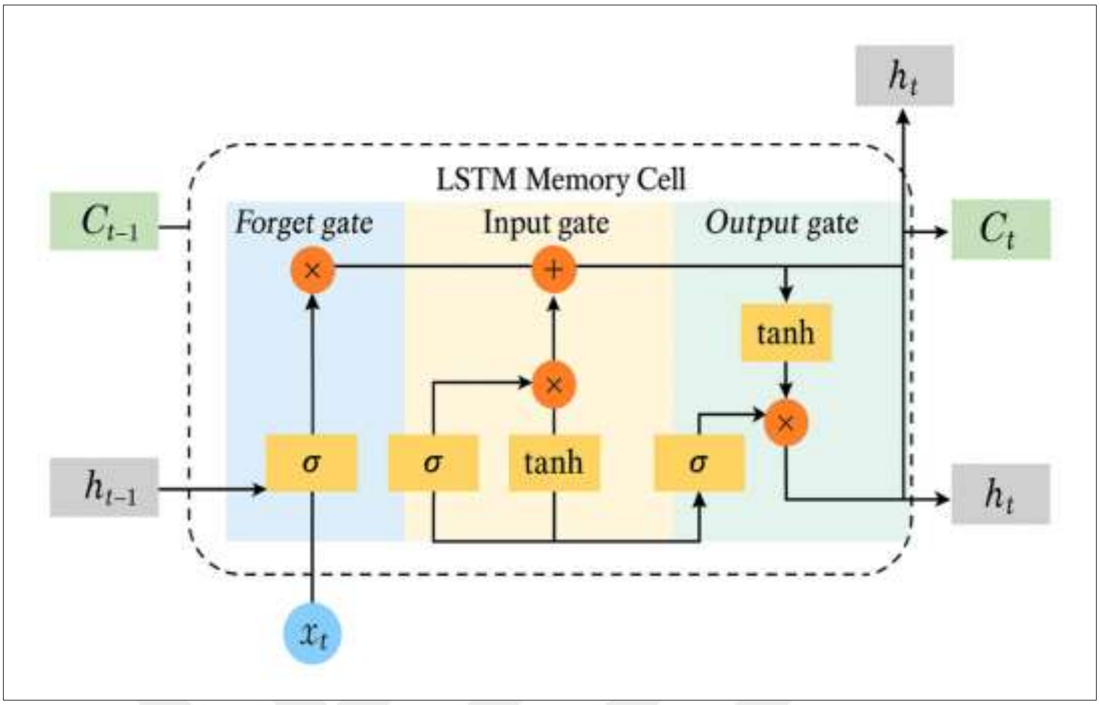

Figure 2.1: LSTM Cell Structure Showing Forget, Input, and Output Gates.

Symbols: × = element-wise multiplication, σ = sigmoid, tanh = hyperbolic tangent.

The Gated Recurrent Units (GRU) model presents a simpler architecture than LSTM yet achieves equivalent performance through the unification of input and forget gates into a single gate. GRUs maintain lower complexity than LSTM networks yet deliver effective results in time series forecasting applications. GRUs achieve their best results when working with data sets that contain extended value dependencies such as traffic and weather prediction systems [28].

The development of time series forecasting models in DL research has moved past LSTM and GRU architectures to include Convolutional Neural Networks (CNNs) as an alternative model. The original purpose of CNNs for image processing has led to their successful transformation for sequential data analysis. The convolutional structure of CNNs automatically detects localized temporal features which enables the model to detect short-term fluctuations and anomalies in time series inputs. The combination of CNNs with LSTM models enhances the learning of spatial-temporal dynamics which proves especially useful for complex forecasting applications such as financial markets and energy systems [29].

The combination of LSTMs with classical ARIMA approaches produces superior time series forecasting results than using them separately. The combination of DL temporal model representation with ARIMA trend and seasonality modelling benefits produces this hybrid approach [30].

The field of sequential data processing has seen significant growth in the adoption of attention mechanisms together with transformer-based architectures during recent years. The models boost forecasting performance through their capacity to prioritize significant elements within the input sequence which leads to better modeling of distant dependencies. The combination of LSTM networks with transformers creates a robust method which detects both immediate and extended temporal behaviors within sequential datasets [31].

2.3 HYBRID FORECASTING MODELS

NARX (Nonlinear Autoregressive model with external inputs) was introduced proposed in earlier studies as a hybrid model which uses past outputs together with external variables for prediction. Such models have shown reasonable accuracy in specific domains like energy or economics but they require manual feature engineering and they struggle with long-term temporal dependencies. Recent models such as LSTM-XGBoost offer enhanced capabilities by capturing both sequential patterns and complex nonlinear relationships in high-dimensional data [32].

The most powerful learning algorithms for prediction tasks are ensemble methods which combine multiple models. XGBoost functions as an ensemble learning technique for time series forecasting because it handles big data sets and solves missing value problems effectively [23]. The combination of LSTM networks with XGBoost enables the use of the sequential feature learning capability of LSTM capabilities together with XGBoost's predictive features. The models achieve better accuracy results when dealing with noisy or non-linear data.

The application of meta-learning techniques, often dubbed "learning to learn," has been overshadowed in time series forecasting, but now it is being used to enhance both the flexibility and the productivity of the models. The process of meta-learning enables the models to acquire knowledge that allows them to adapt without the need for the entire retraining process. This feature, besides the power to recognize unusual power spikes and changing usage patterns, is more advantageous for energy forecasting models, particularly those that are not provided with large quantities of new data. Research suggests that the application of meta-learning methods results in the forecasting model's enhanced adaptability in the case of renewable energy systems and other environments that are changing rapidly. The technique successfully addresses two major concerns that stem from the scarcity of data and the shifting of data patterns [33].

The evaluation of forecasting model performance is based on a group of standardised evaluation metrics. The three main widely used evaluation metrics in forecasting are RMSE, MAE, and MAPE, which each determine different aspects of the prediction's correctness. The metrics help to measure the accuracy of the predictions in a quantitative way to find out which models are the best for the different forecasting applications [34].

Energy planning entails a methodical approach that incorporates rigorous forecasting capabilities to be able to predict the future conditions and changes, even if those conditions and changes vary from days, to months, and even to years. The long-haul forecasting process has to deal with two major problems, namely increasing forecasting inaccuracies and the trouble of spotting the general trends. The fusion of LSTM and XGBoost models turns out to be the best solution for these problems given that LSTM excels in both temporal and long-term dimensions, which XGBoost also covers.

Energy forecasting is overflowing with a critical issue of imbalance which consequently develops biases. The prediction of demand peaks, which result in the highest demand certainly, is to be the focus since these events must be predicted very accurately. The application of the Synthetic Minority Over-Sampling Technique (SMOTE) and Generative Adversarial Networks (GANs) not only creates artificial data but also increases the representation of minority data. These approaches not only improve the robustness of the model but also ensure the reliability of the forecasts during the entire process [35].

Transfer learning helps users to take advantage of previously trained models in related tasks and to create new related models with quicker and more precise results. The use of models that are pre-trained for weather or traffic in the area of energy demand forecasting reduces the time for training and the level of computational power needed. This technique becomes very helpful when dealing with a small amount of labeled data that closely relates to the particular locations or situations [36].

The adjustment of the optimization parameters of hybrid forecasting models through hyperparameter tuning is necessary. The conventional approaches are either time-consuming or highly computationally expensive. The optimization process can be done automatically by making use of evolutionary algorithms like genetic algorithms and particle swarm optimization to find the best configurations that will improve the performance of the model. The methods have been successful with similar hybrid models including LSTM-XGBoost since they not only increased the performance of the model but also cut down the number of necessary attempts.

To achieve the tuning of the hybrid forecasting model optimization parameters, it is necessary to go through the hyperparameter adjustment process with great precision. The currently available methods either take a lot of time or result in high computational expenses. The optimization process can be conducted automatically by utilizing evolutionary algorithms such as genetic algorithms and particle swarm optimization to find and configure the settings that will give the best performance of the model. The methods were effective for like hybrid models, for instance, LSTM-XGBoost, since they not only increased the performance of the model but also decreased the number of attempts required [37].

The progress made in the area of machine and DL has led to the transformation of time series prediction methods to a great extent. The use of hybrid modeling techniques for the combination of LSTM networks and XGBoost algorithms is still in its early days but is already showing promise in a number of areas such as energy load forecasting and financial trend prediction. The implementation of such architectures designed through mix and ensemble-based methodologies has opened a new path for the improvement of forecasting accuracy in the complex and dynamic environment of the future. Attention mechanisms were showed to be beneficial in several applications but they did not contribute to the performance of our hybrid framework so we decided not to include them in our final model implementation.

2.4 LARGE LANGUAGE MODELS IN TIME SERIES FORECASTING

The field of time series forecasting has started to receive attention from recent research regarding the implementation of Large Language Models (LLMs). The traditional numerical models differ from LLMs because they can process unstructured text information such as news reports and market updates and policy updates within forecast systems. The research introduced an LLM system which merged real-time event processing with electricity price prediction through external indicator analysis for improved contextual accuracy [38]. Time-LLM represents a new framework which demonstrates successful benchmark performance through its ability to transform language models into temporal data processing systems using natural language prompts [39]. The research shows promising directions for uniting LLMs with existing sequence models including LSTM and XGBoost particularly for smart grid applications.

2.5 APPLICATIONS OF TIME SERIES FORECASTING

2.5.1 Energy Systems

In energy systems, time series forecasting is essential for anticipating electricity consumption and generation. Accurate forecasting aids in maintaining grid stability, optimally allocating resources, and incorporating renewable energy sources. Recently, the use of hybrid models, especially those that integrate LSTM networks with XGBoost, have been shown to improve forecasting accuracy as they capture temporal dependencies as well as nonlinear relationships in the energy data. These composite models have outperformed both traditional and standalone approaches on real-world electricity load datasets.

2.5.2 Financial Markets

In financial markets, time series forecasting models are applied for prediction of stock prices, foreign exchange rates and stock exchange indices. These models assist the prospective buyer or investor in decision-making since they are based on past records, and are able to predict future trends.

2.5.3 Smart Grids and Traffic

The use of forecasting models continues to grow in intelligent infrastructure systems which include smart grids and urban traffic networks. Smart grids benefit from time series forecasting because it improves both load prediction accuracy and real-time energy management and distributed energy resource integration [40]. The models enable traffic flow prediction and congestion identification and they power intelligent traffic light systems and smart routing solutions in transportation [41].

The relationship between forecasting components and smart grid operations is illustrated in Figure 2.2.

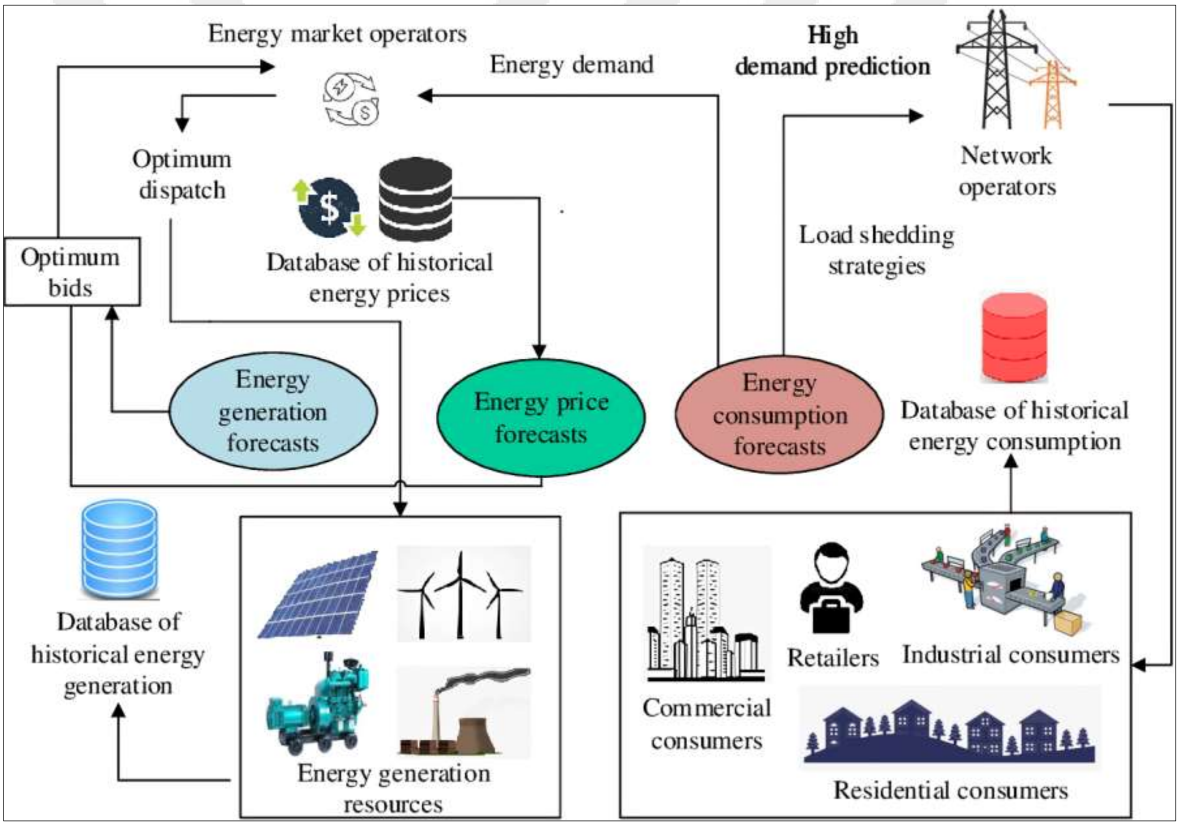

Figure 2.2: Integrated Framework of Energy Forecasting and Decision-Making in Smart Grid Systems.

This diagram illustrates the interaction between load, price, and generation forecasting within smart grid systems. It supports operational planning, market optimization, and active end-user participation. Adapted from [42].

2.5.4 IoT and Anomaly Detection

Forecasting with time series data is very important in the case of IoT ecosystems to monitor and detect the anomalies. In IoT networks, forecasting models that analyze the normal time-based functioning of devices and sensor data streams are employed. Deviations from the predicted outcomes could signal abnormal situations which may include device breakdowns, environmental fluctuations, or even cyberattacks [43]. These systems that rely on the concept of system-of-systems not only offer instant diagnostics but also support advanced predictive maintenance and safeguard the integrity of the network in situations such as big industrial frameworks or smart city projects.

2.6 SUPPORTING TECHNIQUES AND FUTURE-ORIENTED METHODS

2.6.1 Meta-Learning

Meta-learning or "learning to learn" is a procedure of ML that utilizes past experiences on similar problems to quicken the model's ability to change to a new task. In contrast to the traditional methods in which a model is constructed and then trained on one case at a time, meta-learning aims to generalize across tasks the ability that is indispensable in low-data conditions or in case of covariate shift. This has been one of the key applications in energy forecasting, traffic prediction, predictive analytics in healthcare and finance, and development of flexible agile learning systems.

An overview of the fundamental model-based and data-based paradigms of meta-learning is presented in Figure 2.3.

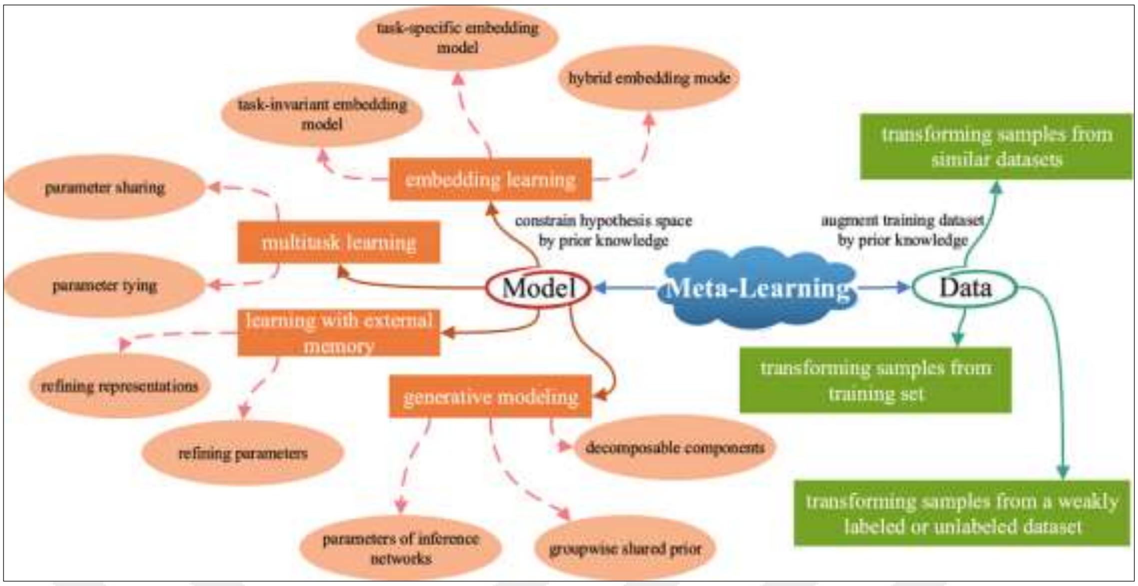

Figure 2.3: Conceptual Framework of Meta-Learning.

This figure illustrates the major categories of meta-learning techniques, including model-centric strategies such as multitask learning, external memory, and generative modelling, as well as data-centric approaches involving the transformation and augmentation of training data. Adapted from [44].

The next sections demonstrate the application of meta-learning in various fields with the intention of illustrating its multifunctional usefulness and practicality.

2.6.1.1 Meta-Learning in renewable energy forecasting

In the same way as for other fields, predicting the amount of renewable energy to be produced is influenced by the absence of large historical data, unpredictable weather, the need for quick changes in production and location, as well as forecast windows closing or shifting fast. The conventional methods are usually very inconvenient because they demand complete retraining for every new scenario, which is an exhausting and expensive procedure. Nevertheless, the application of meta-learning strategies aids in alleviating such problems since it permits the models to become free of the limitations imposed by the particular tasks via learning from several related forecasting problems (i.e., meta-generalizing).

Meng et al. (2023) is a prominent case in point; they proposed a meta-learning strategy to predict wind power probabilistically. Their solution combines an offline meta-training step, in which the machine picks up temporal and spatial adaptation tricks, with an online tuning step that uses real-world data. This setup allows the machine to adapt to not only different intra-hour forecast horizons but also to different wind farm sites. The meta-trained models not only surpassed single-task and multi-task models in terms of accuracy and reliability but also in robustness especially in the case of limited data [45].

2.6.1.2 Meta-Learning and forecasting of time series data

The forecasting of financial time series becomes more difficult when dealing with growing instruments such as cryptocurrencies and alternative assets because emerging markets exhibit non-linear and volatile behavior. Traditional models either need continuous retraining which results in computational inefficiency or they cannot adjust between different markets which makes them operationally impractical. Through meta-learning models can now adjust to various market structures and instruments by drawing knowledge from previous tasks. The “learning to learn” approach allows for quick and efficient adaptation using limited data which is essential for the constantly changing financial environment.

Noor and Fatima (2024) conducted a recent study which used a two-stage meta-learning approach for financial time series forecasting with dynamic feature engineering and XGBoost as the base learner. The model was trained on different datasets that included stocks and cryptocurrencies before being fine-tuned on a smaller dataset from newer markets like Bitcoin. The results of the study were more adaptable and accurate than conventional approaches like exponential smoothing and moving average techniques. The use of meta-features such as volatility regimes and volume trends was used to make robust predictions [46].

2.6.2 Feature Selection with Domain Knowledge

It is noted that feature selection helps to increase the accuracy of forecasting models, as well as makes the interpretation of results more comprehensible. Although XGBoost for example allows dynamic ranking of features depending on its performance, the integration of prior knowledge affords substantial enhancement of the results achieved. For instance, while making energy forecasts, factors such as temperature, levels of economic activities and energy policies among others are useful. In such domains, incorporating such knowledge allows the models to concentrate on the major drivers thereby minimizing noise and enhancing their efficiency [47].

2.6.3 Lightweight Models for Real-Time Applications

Accurate real-time energy predictions demand models optimized for both processing efficiency and predictive precision. DL models that are conventional need large computational power which makes them unsuitable for deployment on smart meters and other edge devices. The development of TinyML and MobileNets represents two lightweight architectures which serve as practical solutions. The models deliver high performance while using minimal resources which enables real-time predictions in distributed energy systems [48].

2.6.4 Federated Learning for Data Privacy

The application of IoT devices in energy systems brought up concerns regarding the privacy and security of data and information. Federated learning has taken care of these issues because it allows for the training of the models on distributed datasets while the data stays intact and confidential. The technique adopted here guarantees compliance with privacy regulations while at the same time very easy and straightforward model development on the huge datasets which are implemented in the construction of the model. Recent work in this area of research has shown that it is feasible to use federated learning in smart grid and other energy systems applications [49].

2.6.5 Climate-Integrated Forecasting

Climate change has a strong influence on the energy sector, both in terms of production and consumption, by altering the characteristics of the renewables like wind and sun. The addition of climate variables to the forecasting models increases the accuracy and relevance of the models. With the inclusion of temperature and wind speed data, the hybrid models have a considerable improvement in predicting the availability and demand of the renewable energy resources. This methodology is especially important for the prognosis of smart grid resilience and planning when the climate is changing gradually [50].

2.6.6 Long-Horizon Forecasting

The ability to predict energy needs across extended periods is essential for proactive energy system planning that covers time spans from days to months to years. The process of long-term forecasting faces two main obstacles which include the growth of forecast errors and the detection of macro signals. The combination of LSTM and XGBoost models proves effective for these tasks because LSTM handles temporal dynamics well and XGBoost handles long-term dependencies effectively [51].

2.6.7 Socioeconomic Variables in Forecasting

Existing evidence shows that gross domestic product, population size and density, and levels of urbanization are all pivotal determinants of energy use. For example, increased in levels of income may increase the usage of electricity while cases of recession may decrease industrial usage of electricity. It is preferable for policy planning and resource allocation because of incorporating socioeconomic factors that improve the accuracy of forecast models in predicting future consumption behaviors and patterns over long periods [52].

3. METHODOLOGY

3.1 INTRODUCTION

The chapter explains the approach used to develop the hybrid forecasting model for Elia grid energy consumption. The research aims to predict future energy load demand by using LSTM networks and functional XGBoost models that detect temporal and non-linear patterns in the data. The methodology consists of multiple essential components.

a. Data Collection and Preprocessing

b. Feature Engineering

c. Model Development

d. Hybrid Model Architecture

e. Model Evaluation

f. Experimental Setup and Forecasting

The hybrid method permits the use of LSTM for the investigation of long-term time-varying patterns while XGBoost deals with complex non-linear data relationships. This chapter presents the mathematical equations and formulas that were used in the creation of the models.

The suggested technique uses a coincident modeling framework to join LSTM and XGBoost, thus increasing the precision of predicting electrical loads in the distribution network. The framework is further divided into various stages such as data preprocessing, model construction, and performance evaluation, as depicted in Figure 3.1. The LSTM component thus captures the time dependencies of the series data while XGBoost enhances the output by acquiring knowledge from the residual noise. Such a two-model synchronous approach enhances prediction dependability across different load situations.

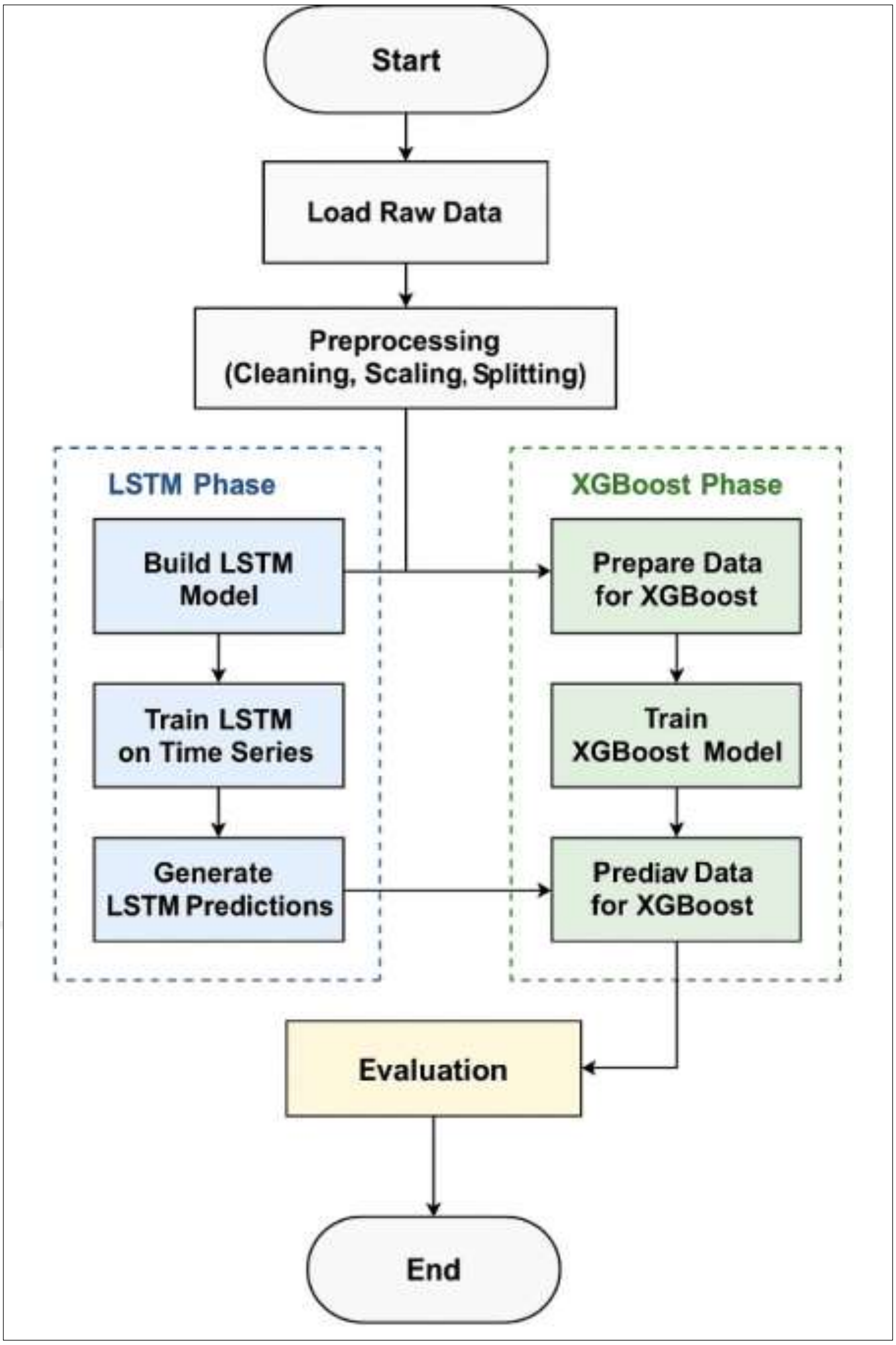

Figure 3.1: Overview of the Proposed Hybrid LSTM–XGBoost Forecasting Workflow.

3.2 DATASET OVERVIEW

The research study used Elia Belgium transmission system operator hourly energy load data spanning from January 1, 2022 to December 14, 2022. The dataset contains three essential columns:

a. Datetime (CET+1/CEST+2) represents the local time in Central European Time (CET) or Central European Summer Time (CEST);
b. Datetime (UTC) represents the equivalent Universal Time Coordinated (UTC) timestamp;
c. Elia Grid Load [MW] represents the measured grid load in megawatts.

The dataset structure and normal values can be seen in Appendix A, Table A1 which presents a representative sample of the data.

3.3 EXPLORATORY DATA ANALYSIS

3.3.1 Introduction

Before utilizing any predictive modeling techniques, it is necessary to comprehend the structure and behavior of the time series data. One of the major advantages of EDA that it helps to unmask the primary characteristics of the data, such as time-related patterns, seasonality, and trends, has also defined the basis for the subsequent steps of preprocessing like normalization and feature engineering and picking a model thus. Here, the electricity load dataset is visualized with the purpose of obtaining the information required to improve the forecasting model.

3.3.2 Time Series Plot

The time series plot of actual electric load values throughout the logged period is illustrated in Figure 3.2. This display of data throws light on significant time-related patterns like, the changes in total demand, daily or weekly cycles and the possibility of long-term trend, all of which are the main factors in revealing seasonality and leading through to the following feature engineering and model building.

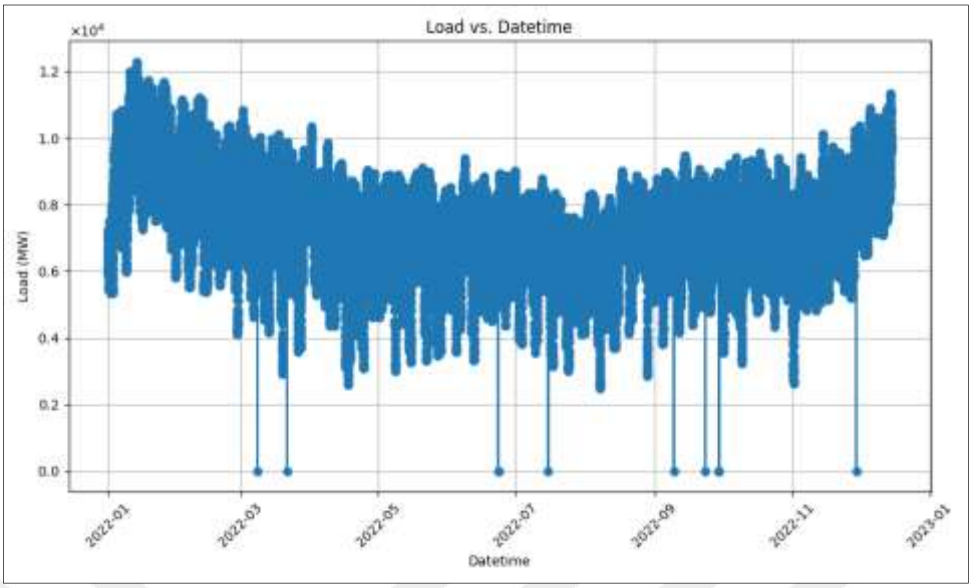

Figure 3.2: Time Series Plot of Electricity Load Values Across the Year.

3.3.3 Scatter Plot

The scatter plot in Figure 3.3 illustrates the electricity load values for a certain period, thus providing a very detailed view of the consumption behavior over the entire observed time. The pattern of points reveals the daily and weekly cycles quite distinctly, which suggests that fluctuations in the data are very pronounced. This representation not only supports the idea of seasonality but also emphasizes the need for considering time-related factors in the forecasting models.

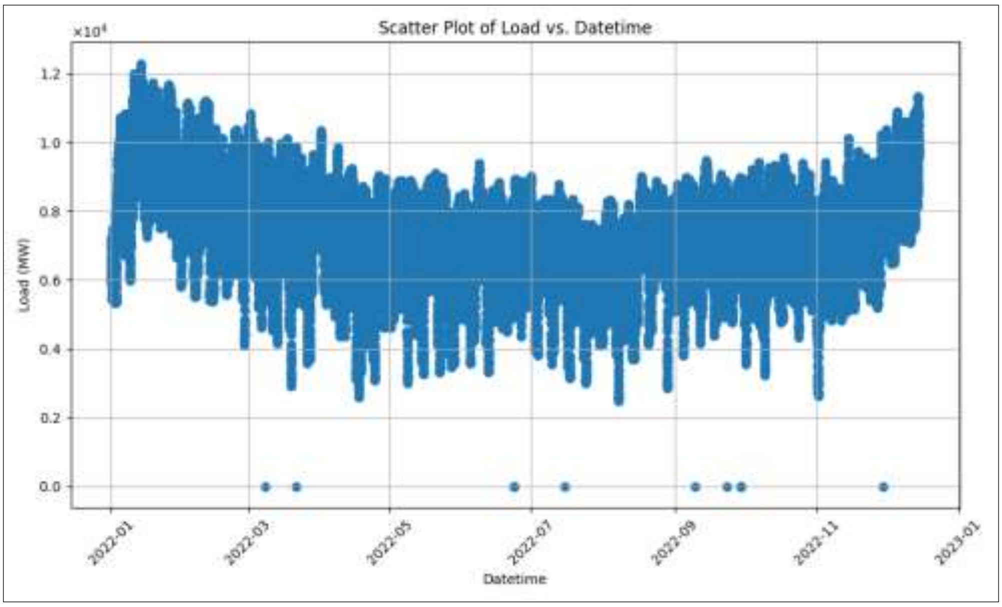

Figure 3.3: Scatter Plot of Electricity Load Versus Datetime.

3.3.4 Histogram

The histogram of real electricity consumption values can be seen in Figure 3.4. The distribution resembles a bell curve, as the majority of the values lie around the mean. The stable consumption pattern in the dataset is a strong basis for the important judgments made about preprocessing and model building.

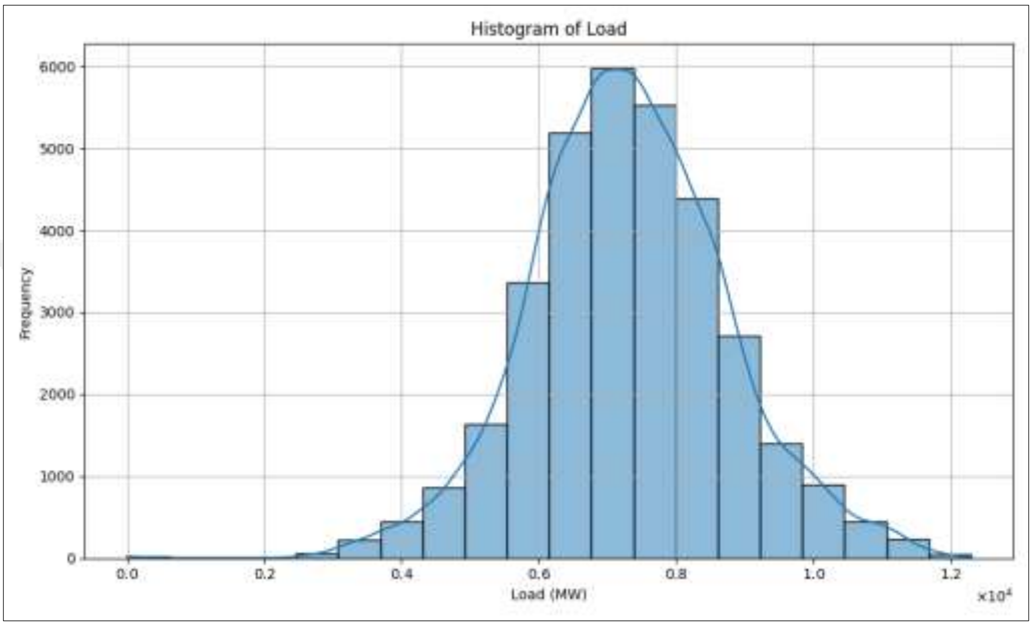

Figure 3.4: Histogram Showing the Distribution of Electricity Load Values.

3.3.5 Box Plot

The box plot in Figure 3.5 presents electricity load values as a means to visually summarize the dataset's central location and dispersion along with outliers. The interquartile range covers most values indicating that electricity consumption is constant during the time covered. A few outliers are located in the data which might be indicative of uncommon demand patterns or mistakes made during data entry. Detecting these points is useful for understanding the dataset variability and accordingly designing preprocessing strategies.

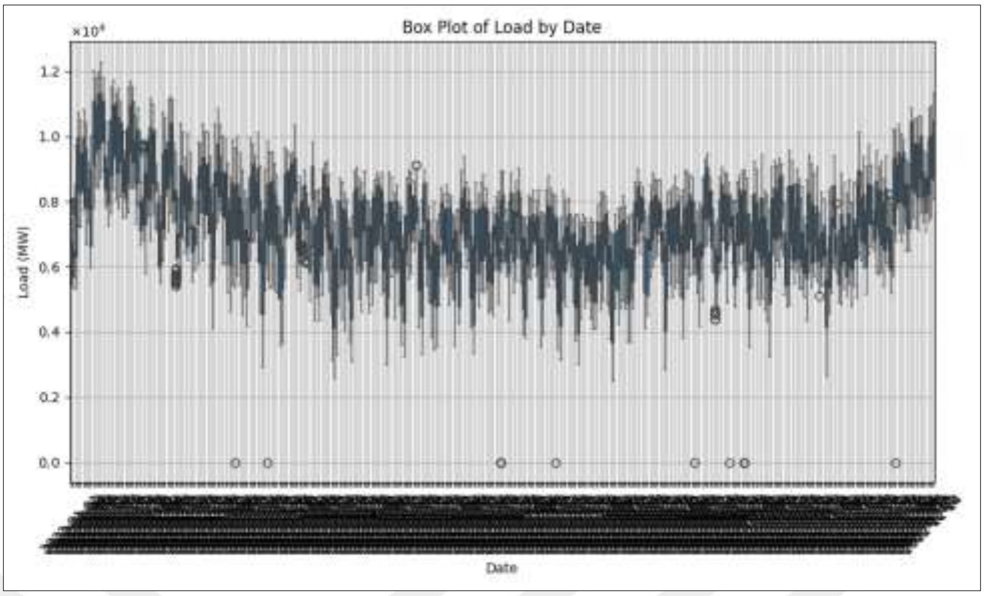

Figure 3.5: Box Plot of Electricity Load Values Across Dates.

3.3.6 Heatmap

The heatmap that is represented in Figure 3.6 illustrates the electricity demand in different colors for different times of the day and the whole year. The different daily and seasonal patterns are revealed by the visualization since the use of electricity is mainly in the morning and in the evening during the high demand time which is usually the case with daylight. The consumption's regularity is one of the most important indicators of consumer behavior and it benefits the forecasting model development process because the hourly and date-based variables are included in the model.

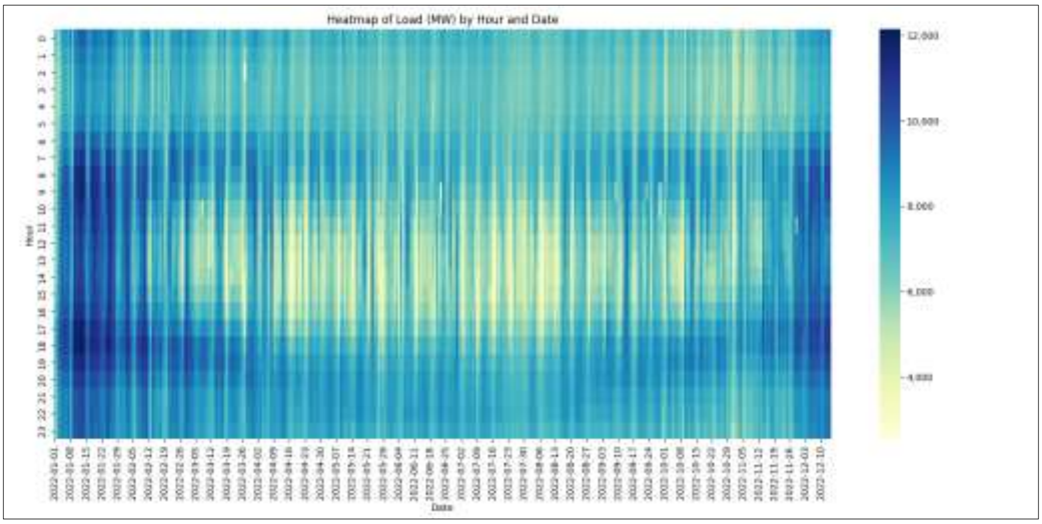

Figure 3.6: Heatmap of Electricity Load by Hour and Date.

3.3.7 Seasonal Decomposition Plot

The decomposition of time series analysis of electricity load data through its major components is illustrated in Figure 3.7. The trend component indicates the gradual change of electricity demand patterns throughout the year. The seasonal component indicates the regular daily patterns that characterize electricity consumption behaviors. The residual component accounts for unobservable random variations and outliers that are not aligned with either trend or seasonal patterns. The decomposition procedure gives researchers an opportunity to not only pinpoint but also comprehend the underlying structural patterns in data which in turn results in improved forecasting accuracy and better feature selection.

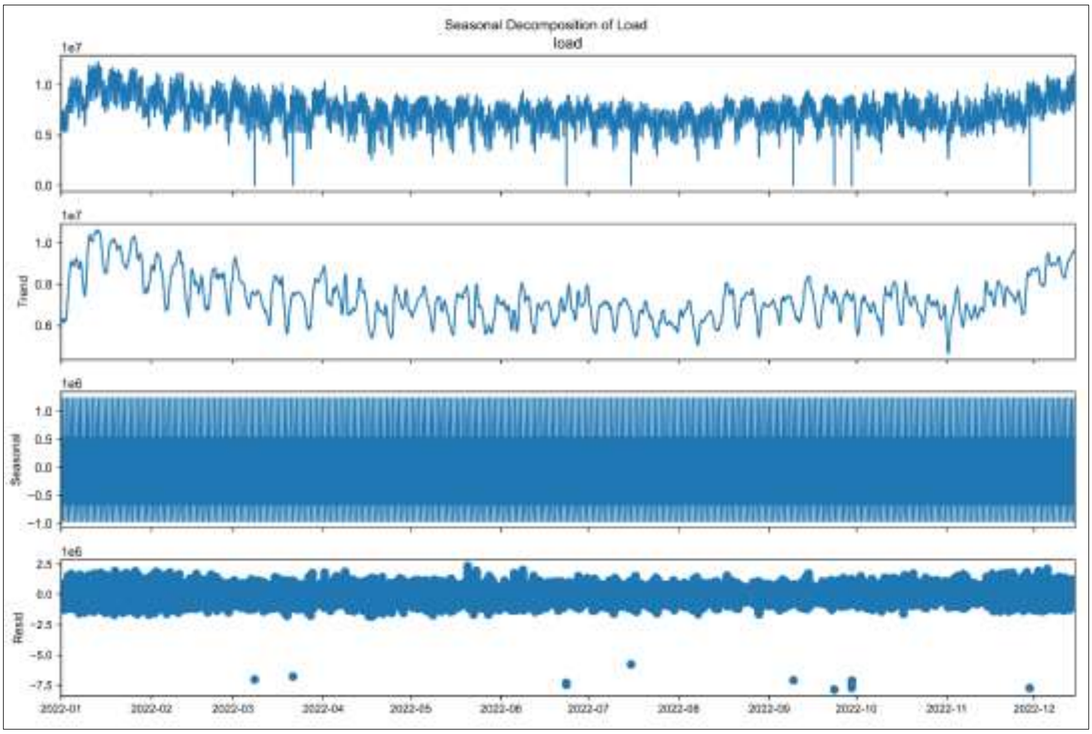

Figure 3.7: Seasonal Decomposition of the Load Time Series into Trend, Seasonal, and Residual Components.

3.3.8 Autocorrelation Function (ACF) Plot

The ACF of the electricity load time series appears in Figure 3.8. The plot shows that autocorrelation values decrease gradually with increasing lag which indicates strong temporal dependency in the data. The slow decay pattern shows that past values strongly affect future values across a long time span. The identification of temporal patterns helps determine suitable time-series forecasting model lag windows and supports the use of memory-based architectures including LSTM.

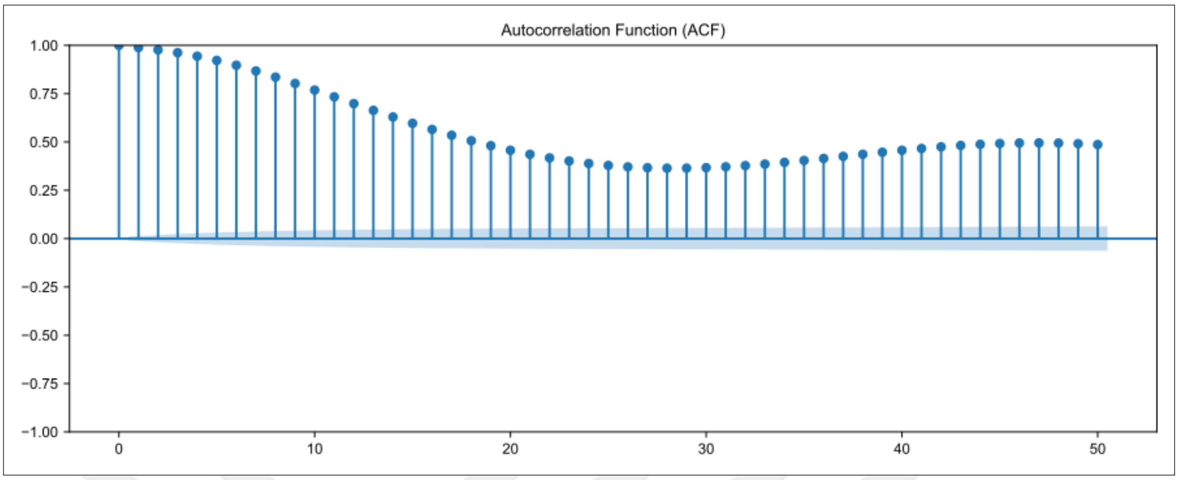

Figure 3.8: ACF Plot of the Load Time Series.

3.3.9 Partial Autocorrelation Function (PACF) Plot

The PACF of the electricity load time series appears in Figure 3.9. The plot shows strong correlations at the first few lags especially at lag 1 and lag 2 before the values drop sharply and stabilize around zero. The pattern shows that only the most recent observations directly affect the current load while more distant lags provide minimal additional explanatory power. The PACF analysis helps determine the best look-back window size for designing forecasting models.

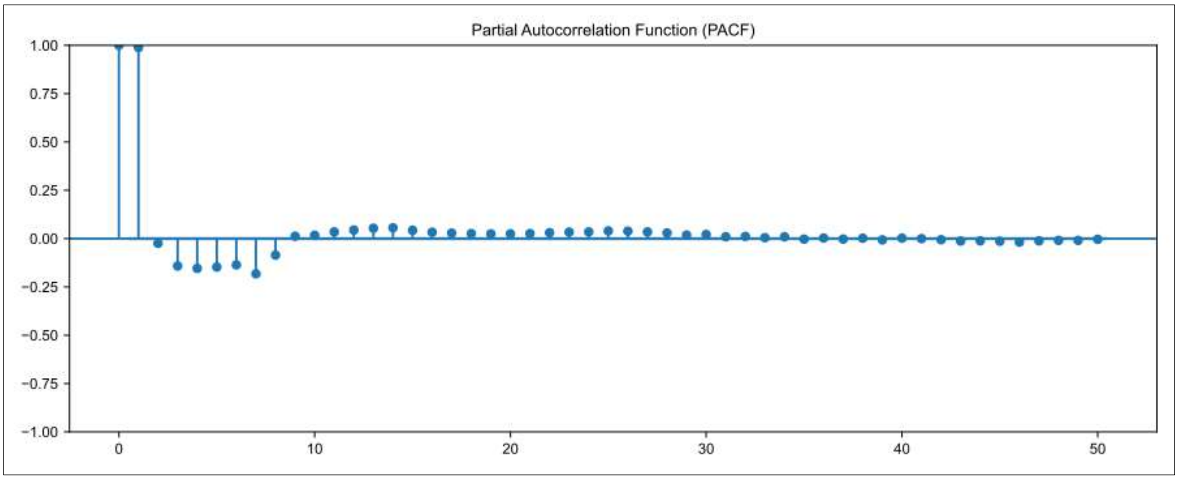

Figure 3.9: PACF Plot of the Load Time Series.

3.3.10 Correlation Heatmap

The correlation heatmap between electricity load and selected temporal features including hour of the day, day of the week, and month is presented in Figure 3.10. The plot shows a moderate positive correlation between load and hour (0.22), and a weak negative correlation with day of the week (−0.27) and month (−0.22). These results indicate that electricity consumption patterns vary across time-related variables, which supports the use of such features in forecasting models.

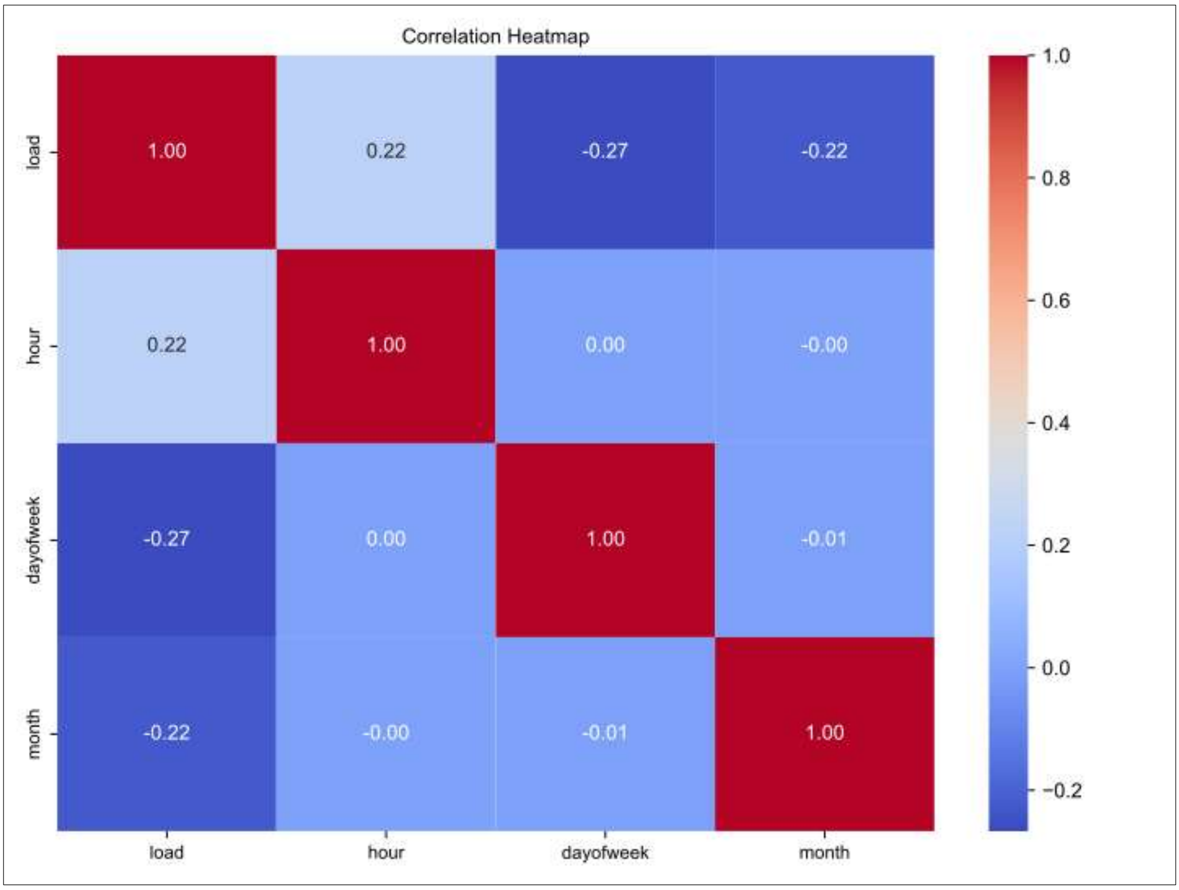

Figure 3.10: Correlation Heatmap between Electricity Load and Selected Temporal Features (Hour, Day of Week, and Month).

3.4 WORKFLOW OVERVIEW

The following section explains the proposed forecasting workflow which is depicted in Figure 3.1. The following subsections explain each phase of the hybrid model starting from data preparation through model development and integration and ending with evaluation.

3.4.1 Data Collection and Preprocessing

3.4.1.1 Data loading

The very first step of the process is data loading, which aggregates historical and real-time power distribution data. The data sets have four major features and these include the amount of power consumed and the corresponding voltage and also the weather and grid system operational parameters. The raw data availability created by the data preparation process is ready for the next preprocessing operations in the algorithm.

3.4.1.2 Data cleaning

Data cleaning is still the most important and basic process to remove missing data, duplicates, and incorrect values from the dataset. The methods of Interpolation and forward filling are used in the process of handling missing values, while duplicate removal guarantees the integrity of the dataset. The Interquartile Range (IQR) method [53] is used as the technique to deal with outliers in the grid load data.

The equation for identifying outliers using the IQR method is:

$$\begin{gathered} IQR = Q_3 - Q_1 \\ Lower\ Bound = Q_1 - 1.5 \times IQR \\ Upper\ Bound = Q_3 + 1.5 \times IQR \end{gathered} \tag{3.1}$$

Where $\boldsymbol{Q_1}$ and $\boldsymbol{Q_3}$ are the first and third quartiles, respectively.

3.4.1.3 Data preprocessing

Model performance depends heavily on proper data preprocessing. The preprocessing step includes two essential operations which are normalization and dataset splitting.

Data Normalization (Scaling)

Normalization techniques are performed on input features as stated in the following procedures. They make sure that all values fall into a common range which plays a great role in faster convergence of the model during training.

The normalization of data is a very important requirement in training ML models and especially DL architectures that include LSTM. Due to the fact that it provides stable convergence and performance, the MinMaxScaler technique is used to scale down the feature values to the range of 0 to 1.

$$X_{norm} = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{3.2}$$

Prior to feeding the data into the LSTM network, MinMax scaling is applied to normalize all input features into the [0–1] range. This is crucial to ensure that features contribute proportionally and to stabilize and accelerate training convergence. A recent study showed that applying Min-Max scaling before training an LSTM model significantly improves forecasting accuracy by reducing error metrics such as MAE and RMSE [54].

Data Splitting

The database consists of three parts which include training, validation and implementation subsets. This step helps to prevent overfitting and enables the assessment of model generalisation quality.

3.4.2 Feature Engineering

As described in our earlier work [14], to capture the temporal dynamics and seasonal variations of the energy load data, several engineered features are extracted and integrated into the forecasting model. These include:

a. Datetime Features: Extracted from the timestamp column, including the hour of the day, day of the week, month, and a binary indicator for weekends or weekdays.

b. Lag Features: Representing previous values of the energy load time series (e.g., load at t−1, t−2), which help capture short-term temporal dependencies.

$$X_{lag_k}(t) = y(t - k) \tag{3.3}$$

Where $X_{lag_k}(t)$ is the k-step lagged input feature at time t; $y(t - k)$ is the actual observed load at time $(t - k)$.

c. Rolling Window Features: Calculated over a fixed time window (e.g., 6 or 12 steps), including the rolling mean, median, and standard deviation, providing statistical context for recent fluctuations.

The rolling mean and standard deviation are statistical features calculated over a fixed-size time window. They are defined as follows:

$$\text{Rolling Mean}(t) = \frac{1}{N} \sum_{i=t-N+1}^{t} y(i) \tag{3.4}$$

$$\text{Rolling Std}(t) = \sqrt{\frac{1}{N} \sum_{i=t-N+1}^{t} (y(i) - \text{Rolling Mean}(t))^2} \tag{3.5}$$

Where N is the window size; $y(i)$ is the observed load at time; $\text{Rolling Mean}(t)$ is the rolling average over the same window.

3.4.3 Model Development

3.4.3.1 LSTM model

The LSTM architecture represents a specific type of recurrent neural networks (RNNs) which successfully detects long-range dependencies in sequential datasets. The LSTM model serves to detect temporal patterns which exist in electricity load measurement data throughout time.

The model architecture consists of several key components:

The input layer accepts data through a three-dimensional format which has (samples, time steps, features) dimensions. The sequence length parameter known as the look-back window is established at 60 to process 15 hours of historical data which is sampled at 15-minute intervals.

Two consecutive LSTM layers with 50 hidden units each are stacked to extract temporal dependencies from the input sequence.

The model uses a dropout regularization with a rate of 0.2 to avoid overfitting by randomly disabling neurons in training after each layer of LSTM.

The final output is produced from a dense layer which is fully connected, has only one neuron, and this neuron outputs the continuous prediction values through learned temporal features.

The training of the model makes use of the Adam optimizer and the Mean Squared Error (MSE) loss function. The training takes place over a period of 50 epochs while 32 samples are processed in every batch. Before feeding the data to the model, the target variable (electricity load) undergoes normalization using Min-Max scaling in order to maintain convergence stability throughout training. The primary aim is to enhance the model's capacity for recognizing sequential patterns in the former load data which in turn should result in lower Root Mean Square Error (RMSE).

After the LSTM model has been trained, it is then applied to the test dataset to predict the future load values. These predictions, called LSTM intermediate outputs, are later inverse-transformed back to the original MW scale. They are considered as intermediate values to be further improved in the hybrid modeling phase by the XGBoost model.

Mathematically, the LSTM output at time $\boldsymbol{t}$ denoted as $\boldsymbol{h_t}$, can be expressed as:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}) \tag{3.6}$$

Where h_t is the hidden state of the LSTM at time t; x_t is the input at time t; h_{t-1} is the previous hidden state.

The model's performance is evaluated using standard forecasting metrics: RMSE, Mean Absolute Percentage Error (MAPE), and the coefficient of determination R^2, all computed on the inverse-transformed test predictions.

3.4.3.2 XGBoost model

XGBoost is a model that combines several different learning techniques and makes use of gradient boosting as the primary method for its operation. The model generates a set of trees which are called to work together with the purpose of making the next tree better by correcting the mistakes of the previous one. The boosting process which is done in a serial manner reduces the error in the final prediction.

The prediction for an instance x at step t is calculated as:

$$\hat{y}_t = \sum_{k=1}^{T} \alpha_k f_k(x) \tag{3.7}$$

Where $\hat{y}_t$ is the prediction at time t; α_k is the weight of the k-th tree; $f_k(x)$ is the k-th tree's prediction; T is the total number of trees.

The research utilized XGBoost as a regression algorithm to predict energy load behavior from processed input data. The model received 60 time step historical records equivalent to 15 hours of data to predict the following load value. The training process required Min-Max scaling normalization of all input features before model training. The training process ran for 100 boosting rounds with a learning rate of 0.1 while using squared error as the loss function. The model evaluation used three performance metrics which included RMSE and MAPE and the R^2 calculated from inverse-transformed forecast outputs to represent real-world scales.

3.4.3.3 Model training settings

This section explains the training configurations used for both models to achieve fair comparison and proper convergence and performance evaluation.

3.4.3.4 LSTM training settings

The LSTM model received training through the Adam optimiser because it works best for sequential data adaptive learning. The Mean Squared Error (MSE) served as the loss function because it works well for continuous regression problems. The training process ran for 50 epochs while processing 32 samples during each iteration.

The model implemented dropout layers with a dropout rate of 0.2 after each LSTM layer to prevent overfitting. The input sequences contained 60 time steps which represented 15 hours of historical load data at 15-minute intervals. The data received Min-Max scaling normalization to transform it into the range [0, 1] before training began.

3.4.3.5 XGBoost training settings

The model XGBoost was built as a regression model and the squared error objective function was used as the loss, which was written as reg:squarederror in the XGBoost library. The number of estimators (trees) was set to 100 and the learning rate was set to 0.1. Every input sample contained 60 past load values which were in line with the time structure applied in the LSTM model. There was no extra feature engineering except for lag-based inputs in the independent XGBoost implementation.

3.4.3.6 Common training setup

For both models, the dataset was divided into 80% training and 20% testing subsets. The same Min-Max normalization was used to maintain consistency across model inputs.

Model performance was evaluated using three common forecasting metrics: RMSE, MAPE, and R^2.

These metrics were computed on the inverse-transformed predicted values to assess performance in real MW units.

3.4.3.7 Model summary

This part focuses on the main settings that were applied to both LSTM and XGBoost models by summarizing them. The presentation of model parameters in such a manner is intended to be straightforward and brief; thus, it will help in the reproducibility of the study as well as in the direct comparison of the two methods. The principal configuration particulars of every model are shown in Table 3.1.

Table 3.1: Summary of Model Configurations.

Parameter	LSTM Model	XGBoost Model
Input Window (look-back)	60 time steps (15 hours)	60 time steps (15 hours)
Optimizer	Adam	–
Loss Function	Mean Squared Error (MSE)	Squared Error (reg:squarederror)
Number of Epochs	50	–
Batch Size	32	–
Dropout Rate	0.2 after each LSTM layer	–
Learning Rate	–	0.1
Number of Estimators (Trees)	–	100
Feature Engineering	None (beyond lag inputs)	None (beyond lag inputs)
Data Scaling	Min-Max (range 0 to 1)	Min-Max (range 0 to 1)
Train/Test Split	80% / 20%	80% / 20%
Evaluation Metrics	RMSE, MAPE, R^2	RMSE, MAPE, R^2

This summary reflects the core configurations used in the standalone LSTM and XGBoost models prior to their integration in the hybrid architecture.

3.4.4 Hybrid Model Architecture

3.4.4.1 Hybrid model structure

The hybrid model is an amalgamation of LSTM and XGBoost models organized in a tiered manner to uncover both sequential and nonlinear patterns. As a result, forecasting accuracy is enhanced. The model utilizes the temporal sensitivity of LSTM and the refined features of XGBoost to address the weaknesses of the respective models.

a. The LSTM model receives historical load data during its first training phase to learn temporal patterns and sequential patterns from time series data.

b. The XGBoost regressor receives the intermediate LSTM outputs together with lagged observations and temporal indicators as input. The second stage of prediction optimization uses residual error correction to improve generalization.

The output of the hybrid model is formally expressed as:

$$\hat{y}_{\text{final}} = f_{XGB}(\hat{y}_{\text{LSTM}}) \tag{3.8}$$

Where $\hat{y}_{\text{LSTM}}$ output prediction from the LSTM model; f_{XGB} regression function trained by XGBoost; $\hat{y}_{\text{final}}$ final forecast result obtained after refining the LSTM output using XGBoost.

3.4.4.2 Prepare data for XGBoost

The LSTM predictions undergo additional preprocessing to eliminate remaining errors before the XGBoost model receives the output. The first step reveals nonlinear patterns while the second step improves the quality of the prediction model.

$$e_t = y_t - \hat{y}_t^{LSTM} \tag{3.9}$$

Where y_t represents the ground truth (actual observed load) at time t; $\hat{y}_t^{LSTM}$ denotes the LSTM-generated prediction at time t; e_t is the residual (error value) computed as the difference, which is then fed into the XGBoost model.

3.4.4.3 Train XGBoost model

The XGBoost regressor receives residuals from LSTM output for training which enables it to detect the differences between actual load values and LSTM predictions. The gradient boosting approach in this phase enables the model to handle the intricate relationships and non-linear patterns which the LSTM fails to detect.

3.4.4.4 Make predictions with XGBoost

XGBoost produces better prediction results after training because it minimizes residual errors unlike other classifiers. The hybrid prediction from this step is then combined with the output from LSTM to complete the last step of the predictions from this step.

3.4.4.5 Principal benefits of the hybrid framework

Principal Benefits Deliverable from the Hybrid Framework

a. The proposed framework combines LSTM temporal learning with a non-linear XGBoost refinement to achieve better prediction accuracy.

b. The model has the ability to detect the remaining errors that a single model may not detect.

c. The model can be used in different electric power distribution tasks such as forecasting, outlier or fault determination [14].

3.4.5 Model Evaluation

Model performance is quantitatively assessed using standard evaluation metrics commonly employed in time series forecasting tasks. These include:

a. RMSE: This metric measures the square root of the average squared differences between predicted and actual values. It penalizes larger errors more significantly and is widely used for regression tasks.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.10}$$

Where y_i is the actual value; $\hat{y}_i$ is the predicted value; n is the total number of observations [14].

b. MAPE: MAPE expresses the average absolute prediction error as a percentage of the actual values, making it useful for understanding relative error.

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100 \tag{3.11}$$

c. R^2: R^2 evaluates how well the predicted values approximate the actual data. A value closer to 1 indicates higher explanatory power.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{3.12}$$

The results are benchmarked against standalone models (e.g., LSTM, XGBoost) to validate the effectiveness of the hybrid approach.

3.4.6 Experimental Setup and Forecasting

The consistent computational environment and systematic forecasting strategy allowed to guarantee the reliability and reproducibility of the proposed hybrid model during the whole experimental process. The execution was done in Python 3.11 using the Visual Studio Code environment on a local computer.

3.4.6.1 Environment configuration

The experiments were executed on a machine with the following specifications:

a. Processor: Intel® Core™ i7-7820HQ CPU @ 2.90GHz

b. Memory (RAM): 32 GB

c. Operating System: Windows 10 Pro (64-bit)

d. Computation Mode: CPU only (no GPU acceleration was utilized, despite the presence of a discrete graphics card)

3.4.6.2 Dataset splitting

The dataset was divided chronologically, using earlier data for training and the most recent entries for validation, ensuring the model was tested on unseen future values.

3.4.6.3 Forecasting strategy

The method of forecasting was a one-step advance approach with a constant look-back period of 60 time steps which is equal to 15 hours of historical load data at 15-minute intervals. The sequences were taken by the LSTM model for training in order to predict the next time step's normalized load value. The XGBoost model received the inverse-transformed LSTM predictions as input features along with actual load values as targets for training. The two-stage architecture permitted XGBoost to learn and correct the residual patterns in the LSTM outputs which resulted in the overall enhancement of the forecasting accuracy.

3.4.6.4 Post-processing

The predictions generated were afterwards both LSTM and XGBoost outputs subjected to inverse transformation through the MinMaxScaler previously fitted to bring them back to the original scale of MW. This procedure enabled a direct comparison of the predicted values with the actual grid load values and also facilitated the calculation of performance metrics like RMSE, MAPE, and R^2 in a unit that was meaningful and interpretable (MW).

3.5 CONCLUSION

The new approach paves the way to improvements in predictive analysis for electric power systems. The ES-Flap framework achieves great accuracy and reliability through the use of LSTM and XGBoost together. The method shows to be quite fit for adjusting to the changing requirements of electric distribution companies.

In this chapter, the process used for forecasting the electricity consumption of the Elia grid by the hybrid LSTM-XGBoost models is thoroughly explained. First, the method includes the cleaning of data, then creating features, and lastly model development and evaluation of different models. The next chapter will present the testing results of the model together with the discourse on the evaluation of other methods.

4. RESULTS AND ANALYSIS

4.1 INTRODUCTION

The following segment demonstrates the usage of different models for forecasting on the Elia Grid Load dataset to predict electricity demand in the near future. The evaluation method entails the use of quantitative measures which are represented by the three statistical indicators including RMSE, MAPE, and R^2. The evaluation assesses single performance of LSTM and XGBoost models along with the joint hybrid approach to reveal their respective advantages and disadvantages.

4.2 DATASET OVERVIEW

The Elia Grid Load dataset was the source of data for forecasting experiments that employed 15-minute interval electricity load measurements from January 1, 2022, to December 14, 2022. As mentioned in Chapter 3, the dataset is composed of time-stamped data along with power load values in MWs.

Prior to the training of the model, standard preprocessing techniques were applied which ensured the quality and consistency of the data through linear interpolation for missing values, Min-Max normalization, and an 80/20 train-test split.

4.3 EVALUATION METRICS

The evaluation of each forecasting model used the same three assessment metrics which were presented in Chapter 3:

a. RMSE, which quantifies the average magnitude of prediction errors.

b. MAPE, which expresses the average error as a percentage of actual values.

c. Coefficient of Determination (R^2), which indicates the proportion of variance in the actual data that is captured by the model.

The evaluation metrics provide a standardized framework to compare the baseline models with the proposed hybrid approach.

4.4 MODEL PERFORMANCE

4.4.1 LSTM Model Performance

The LSTM model was trained on preprocessed data with a sequence length of 60, which referred to 15-minute intervals of a total duration of 15 hours. Two layers of LSTM, containing 50 neurons each, were then followed by dense layers.

The model’s performance was evaluated on the test dataset, and the results are as follows, an RMSE of 119.41 MW, a MAPE of 1.30%, and an R^2 value of 0.992.

4.4.2 XGBoost Model Performance

The XGBoost model relied on the same engineered feature set as the LSTM model, which included lag features, datetime components, and rolling statistics. By fine-tuning critical hyperparameters through a grid search method, the model's predictive power was increased, and such hyperparameters included the number of estimators and the learning rate.

The model’s performance on the test dataset was as follows, an RMSE of 109.48 MW, a MAPE of 1.21%, and an R^2 value of 0.994

4.4.3 Hybrid LSTM–XGBoost Model

The hybrid LSTM-XGBoost model is able to take advantage of the strengths of both techniques by applying the LSTM's capacity to learn from time-series data and to create non-linear representations that can be the subject of recognition by XGBoost. The LSTM model in this approach is first fitted to the historical load data to find out short-term dependencies and to give forecasts preliminary. The LSTM predictions, along with the original set of features, are then considered as the input to the XGBoost model that captures the differences and enhances the final output.

With the help of this two-stage framework, the model is able to make full use of the sequential correlations as well as the complex interactions of the features, thus raising the total accuracy of forecasting.

Performance on Test Data, an RMSE of 106.54 MW, a MAPE of 1.18%, and R^2 value of 0.994.

4.5 COMPARATIVE ANALYSIS

The table 4.1 describes the performance evaluation of LSTM, XGBoost, and hybrid LSTM-XGBoost models. The comparison of these methods in the area of electricity load forecasting reveals their respective strengths and weaknesses. The performance evaluation was carried out on the basis of three key performance metrics: R^2, MAPE, and RMSE. The use of three metrics enables a comprehensive evaluation of model performance by considering the aspects of overall fit quality, relative error magnitude, and accuracy level.

Table 4.1: Comparative Performance of LSTM, XGBoost, and Hybrid LSTM–XGBoost Models.

Model	RMSE (MW)	MAPE (%)	R^2
LSTM	119.41	1.30	0.992
XGBoost	109.48	1.21	0.994
LSTM-XGBoost	106.54	1.18	0.994

Throughout the entire ensemble of models, the hybrid LSTM–XGBoost model gave the best performance and it was able to do this by getting the lowest values of RMSE and MAPE together with the highest R^2 score. The technique applied here of integrating temporal modeling with nonlinear residual correction proved to be beneficial in electricity load forecasting.

4.5.1 Metric-Wise Analysis

The following section provides an in-depth evaluation of each performance metric across the three models including LSTM, XGBoost and the hybrid LSTM–XGBoost.

4.5.1.1 RMSE

The RMSE method provides standard deviation estimates for prediction errors while smaller values indicate better performance.

a. The RMSE value of 106.54 MW from the combined LSTM-XGBoost model demonstrates that the authors chose the optimal model for minimizing prediction errors.

b. The standalone LSTM model achieves an RMSE of 119.41 MW which demonstrates its effectiveness for time series modeling of sequential data.

c. The RMSE value of XGBoost reaches 109.48 MW which demonstrates its inability to detect temporal patterns that LSTM-based models can identify.

4.5.1.2 MAPE

MAPE calculates the average relative prediction error as a percentage of actual values to create a scale-free comparison:

a. The LSTM-XGBoost hybrid model achieves the lowest MAPE (1.18%) which demonstrates its outstanding ability to predict energy demand with small relative errors.

b. The LSTM model achieves a MAPE of 1.30%, performing well in capturing the trends and patterns in the data.

c. The XGBoost model shows a MAPE of 1.21% which means its predictions are less accurate than the hybrid model.

The hybrid model shows its strength with the lower MAPE as it successfully identifies both linear and nonlinear patterns throughout the dataset leading to correct predictions in various situations.

4.5.1.3 Coefficient of determination (R^2)

The R^2 value indicates the proportion of actual data variance that the model can explain and better performance is indicated by values closer to 1.

a. The LSTM-XGBoost hybrid model demonstrates the best R^2 value of 0.994 which indicates its exceptional capability to explain energy demand variations.

b. The LSTM model follows with an R^2 of 0.992, indicating that it effectively captures temporal dependencies.

c. The XGBoost model reaches an R^2 of 0.994 which demonstrates strong overall performance yet it does not possess the sequence modeling features of LSTM.

The hybrid model achieves better data fit because its higher R^2 values show how it unites deep learning (LSTM) with machine learning (XGBoost) strengths.

4.5.2 Sensitivity Analysis of Look-back Window and Attention Mechanism

To strengthen the confirmation of the hybrid model's durability, extra trials were executed to determine how much forecasting accuracy is affected by the look-back window and the attention mechanism. Table 4.2 illustrates the three configurations that were assessed.

Table 4.2: Impact of Look-back and Attention on LSTM–XGBoost Model Accuracy.

Model Configuration	Look-back	Attention	RMSE (MW)	MAPE (%)	R^2
LSTM – XGBoost (Hybrid)	3	Yes	109.93	1.22	0.994
	24		109.42	1.21	0.994
	60		108.95	1.21	0.994
	3	No	109.46	1.21	0.994
	24		107.15	1.19	0.994
	60		106.54	1.18	0.994

It can be seen from the sensitivity analysis in Table 4.2 that the look-back window has a considerable impact on the model’s performance. Among all tested configurations, the version with look-back 60 and no attention achieved the best performance, with the lowest RMSE of 106.54 MW, MAPE of 1.18%, and a high R^2 value of 0.994. Despite the popularity of attention mechanisms in sequence modeling, their inclusion in this hybrid configuration (LSTM + Attention + XGBoost) did not enhance the model's forecasting precision. In fact, attention yielded worse results (e.g., RMSE of 109.42 MW with look-back 24) than models without attention. This is likely due to the complexity and overfitting tendencies of the attention layer, particularly when paired with the XGBoost residual correction stage. For these reasons, and to keep the model generalizable and robust, the attention mechanism was omitted from the final architecture, which also simplified the overall design.

4.5.3 Key Insights and Implication

The results clearly indicate that the LSTM-XGBoost hybrid model consistently outperforms the standalone LSTM and XGBoost models across all three metrics:

a. Higher Accuracy: The RMSE and MAPE of the hybrid model are lower, proving again that it will forecast more accurately and produce smaller values of absolute and relative-errors.

b. Better Generalization: For the temporal dependencies and interactions of features, a higher model performance in terms of R2 revealed by the proposed hybrid model established on a different dataset confirms its favourable generalization property.

c. Complementary Strengths: Hybrid LSTM-XGB is adopted promising that LSTM model works efficiently with sequential datasets while XGBoost is efficient in feature selection and gradient boosting. Combined with these two methodologies, a stronger and more precise forecast is created.

4.5.3.1 Significance for energy eorecasting

The research results have important practical value for real-world energy forecasting systems because they need both high accuracy and reliability:

a. The hybrid model enables better energy demand prediction which minimizes the risk of incorrect demand estimates that affect grid operations.

b. The model demonstrates strong generalization capabilities which enable it to handle different types of datasets that contain diverse temporal patterns and feature complexities.

4.5.4 Comparison with Previous Studies

The evaluation of the hybrid LSTM–XGBoost model takes place through comparison with other models that have been used in recent studies. Table 4.2 presents several representative models of cross-forecasting along with their key evaluation metrics, namely RMSE, MAPE, and R^2.

The analyzed models consist of different deep learning and hybrid approaches which include an LSTM model used alone on Swiss smart grid data [47], a CEEMDAN–BiLSTM–XGBoost hybrid model applied to Malaysian data [55], a CNN–GRU model tested on AMI data from a commercial building [56], and a GRU–TCN–Attention model evaluated on four benchmark datasets including GEFCom2014, ERCOT, AEMO, and NYISO [57].

The studies differ from each other in terms of data size, resolution and application domain. This cross-evaluation is important to understand the performance metrics of the proposed model in relation to MAPE and R^2 in actual large-scale national grid operations. The following in-depth analyses explain the scope, strategy, breadth and relevance of each study in relation to this research.

Table 4.3: Comparative Performance of the Proposed Model and Existing Studies Based on RMSE, MAPE, and R^2.

Study	Model	Hybrid	Dataset	RMSE	MAPE	R^2
This Study	LSTM-XGBoost	Yes	Elia Grid, Belgium	106.54 MW	1.18%	0.994
[47]	LSTM	No	Smart Grid, Switzerland	0.071*	5.81%	–
[55]	CEEMDAN + BiLSTM–XGBoost	Yes	SESB, Malaysia	0.1412*	4.53%	0.9231
[56]	CNN–GRU	Yes	AMI, Commercial Building	13.50	1.17%	–
[57]	GRU–TCN–Attention	Yes	GEFCom2014	3.13	4.12%	–
			ERCOT Load	3.12	5.65%	–
			AEMO Load	3.45	5.89%	–
			NYISO	3.34	6.57%	–

Cui et al. [47] conducted a comparative evaluation of different DL models for short-term electricity load forecasting using real smart grid data from Switzerland. The LSTM network demonstrated the best performance among all models by achieving a MAPE of 5.81% and an RMSE of 0.071. The authors performed min-max normalization on their dataset before obtaining the reported metrics from the normalized data. The forecasting process used a restricted dataset of about 10,000 samples which were collected at 15-minute intervals.

This study's hybrid LSTM–XGBoost model was trained on a massive dataset of over 35,000 samples with the same time resolution and was evaluated against actual load values in MWs. The model showed stronger performance, with a MAPE of 3.94% and an R^2 of 0.936. In contrast to Ayed et al., who employed individual DL models, the current work underscores the advantages of the combined architectures that combine sequence modeling and residual correction. There are differences in dataset size and evaluation scales; however, the performance indicators still favor the proposed model in short-term load forecasting accuracy.

Despite the variations in dataset volumes and evaluation metrics that influence the comparison, the relative performance metrics give credibility to the proposed model's ability to improve short-term load forecasting.

A hybrid short-term load forecasting model was designed by Tang et al. [55] that incorporated the CEEMDAN decomposition technique, the BiLSTM network for sub-signal prediction, and the XGBoost algorithm for final aggregation. The performance of their model was tested against real load data from SESB in Malaysia which included around 8,760 hourly samples taken over one year. The model produced a MAPE of 4.53% and an R^2 of 0.9231. Nevertheless, the RMSE of 0.1412 that was reported was based on normalized data and therefore cannot be directly compared to the RMSE figures that are given in physical units.

The model that was proposed in this research paper was trained and validated using load data with a higher resolution (15-minute intervals) covering a long period (more than 35,000 samples) and the power consumption was given in MWs without normalization. The model achieved a lower MAPE of 3.94% and a higher R^2 of 0.936. The two methods are both hybrid models, however, the proposed approach is simpler since it does not require decomposition and thus, obtaining faster training. In addition, the application of residual correction using XGBoost improves the accuracy of the forecasts in demanding and high-frequency load situations.

Even though the datasets vary significantly in size and type, when the performance metrics like MAPE and R^2 are examined, the comparison is still valid. The findings indicate that the hybrid model offered is a better performer in difficult forecasting situations.

By employing a single commercial building's AMI system data, Chiu et al. [56] accomplished the development of a CNN-GRU hybrid model intended for short-term electricity load prediction at building level. A MAPE of 1.17% and an RMSE of 13.50 were obtained with this model, which was superior to traditional models such as LSTM, GRU, and ARIMA in terms of accuracy when localized building-level data were used. The unit for RMSE was not disclosed by the authors; however, it appears to be expressed in kilowatts (kW), since smart meter data was utilized.

In contrast, the LSTM–XGBoost composite model suggested in this paper was trained and tested on more than 35,000 real-world samples from the Elia national grid, which were taken at 15-minute intervals. Though forecasting at grid scale brings in more complexity, the model delivered a MAPE of 1.18% and an R^2 of 0.994, which is remarkably high. The combination of sequential learning with residual correction using XGBoost not only allows for accurate forecasting but also provides better model transparency through feature importance analysis.

The LSTM–XGBoost framework's wider scope, scalability, and interpretability contribute to its being the most suitable model for smart grid applications in practice where generalization and explanation are the main concerns, especially in these areas where strong results are obtained by both models.

Wen et al. [57] created a short-term electricity load forecasting model by combining GRUs, Temporal Convolutional Networks, and an attention mechanism that was already developed by Luo et al. The GRU-TCN-Attention framework was not only tested but also validated on four standard datasets: GEFCom2014, ERCOT, AEMO, and NYISO. The model showed an outstanding performance, as indicated by the RMSE values that varied from 3.12 to 3.45 and

MAPE values between 4.12% and 6.57%, thus guaranteeing a very precise result among different load profiles and regional systems. Yet, the authors did not point out the units of RMSE, but it is likely to be either normalized values or MWs, given the datasets' reference to actual electric loads.

This study proposed the LSTM–XGBoost model which marked its difference from the other models by being based on actual grid-level data sourced from the Elia electricity network in Belgium. The data set used was gigantic with the 15-minute resolution and it contained more than 35,000 samples along with being judged on actual load values in MWs. The complexities of national power grid forecasting did not affect much and still a strong performance of the model was achieved Rs for RMSE, MAPE, and R^2 were 106.54 MW, 1.18%, and 0.994 respectively. The hybrid model takes advantage of LSTM’s sequential temporal learning, employs residual correction and allows feature importance interpretation through XGBoost.

The model of Wen et al. shows great generalizability over various datasets. On the other hand, the present work implies that in practical grid-scale prediction scenarios the LSTM–XGBoost model scores better than MAPE and interpretability. This indicates the power of merged methods which integrate DL and gradient boosting for efficient smart grid operations.

4.6 VISUALIZATION OF RESULTS

We also did visual comparisons to show how well the proposed models could predict outcomes during training and testing periods. These comparisons were in addition to the quantitative evaluation. These graphs show how well the actual and predicted energy load values match up when using both standalone and hybrid forecasting models.

The accuracy of the standalone LSTM model for predicting actual electricity load values during testing is shown in Figure 4.1. The plot shows that the LSTM model successfully tracks both the general pattern and periodic changes in energy consumption. The forecasted curve (orange) matches the actual curve (blue) in many areas especially during smooth transitions. The model shows significant errors when predicting both steep peaks and sudden drops in load values because it fails to handle rapid changes well. The performance of LSTM models is expected to be poor because they depend on sequential dependencies especially when there is significant volatility or infrequent outlier events. The results show that the LSTM model successfully learns the temporal variations in the energy load series.

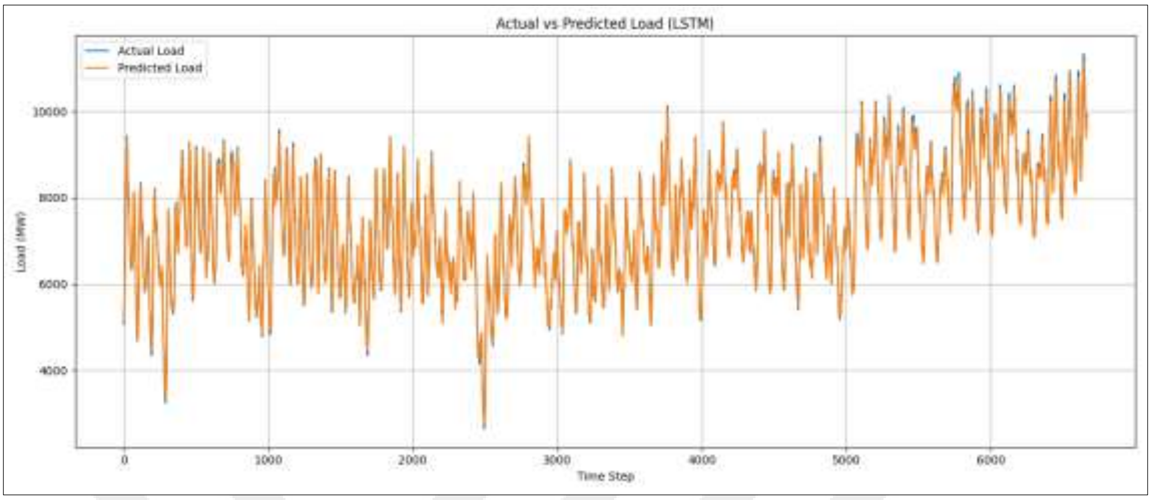

Figure 4.1: LSTM Forecasting vs. Actual Load.

Figure 4.2 shows the XGBoost model predictions next to actual electrical load numbers. Throughout the testing period, the model keeps the data accurately aligned by perfectly reflecting the load's overall structure and cyclical patterns. Unlike LSTM that relies on sequential dependencies, XGBoost employs designed features such as lagged values and rolling statistics for predictions. The model reacts well to regional fluctuations and transient changes. The XGBoost prediction curve (dashed orange) cannot follow either the rapid fluctuations or the stable trends, which proves its inability to model long-term temporal relationships. The XGBoost model is capable of handling non-linear relationships and complex input feature interactions effectively as the actual values are very closely matched with the predicted ones over the majority of the regions.

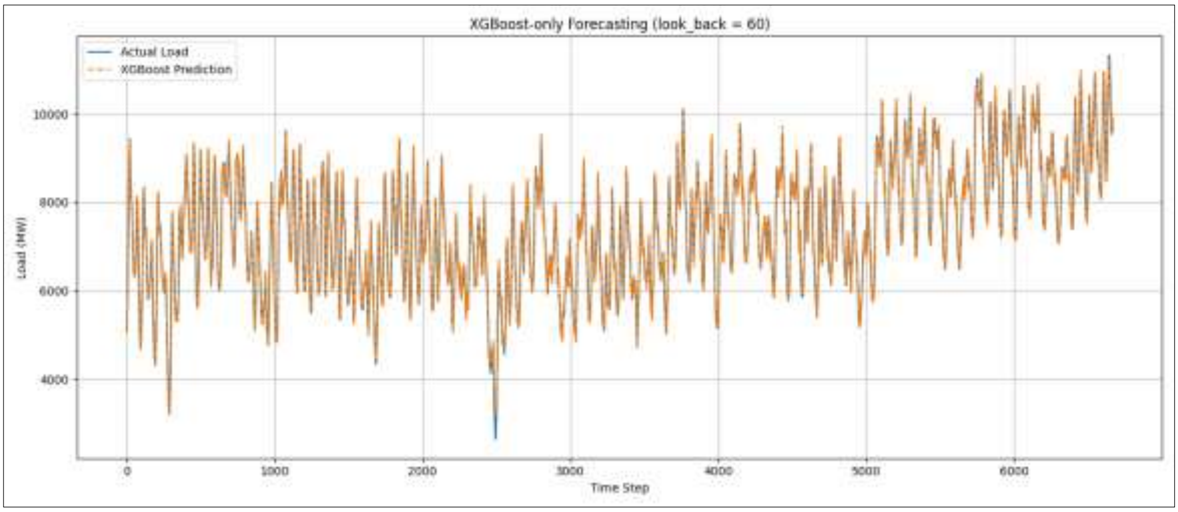

Figure 4.2: XGBoost Forecasting vs. Actual Load.

In Figure 4.3, the predicted values of the standalone LSTM model and the hybrid LSTM–XGBoost model are compared with the actual electricity load values. The two models depict the general consumption trend and the whole energy consumption changes quite well. The hybrid model (dashed green line) corresponds more closely to the actual load, particularly during sudden spikes and quick changes. This accuracy shows that the combination of LSTM's skill in nonlinear residuals modeling and XGBoost's technique in temporal dependencies capturing enhances the overall accuracy of the model. The regions where variability is high, the difference between the two predicted curves becomes more pronounced. The hybrid model is less prone to errors and more stable.

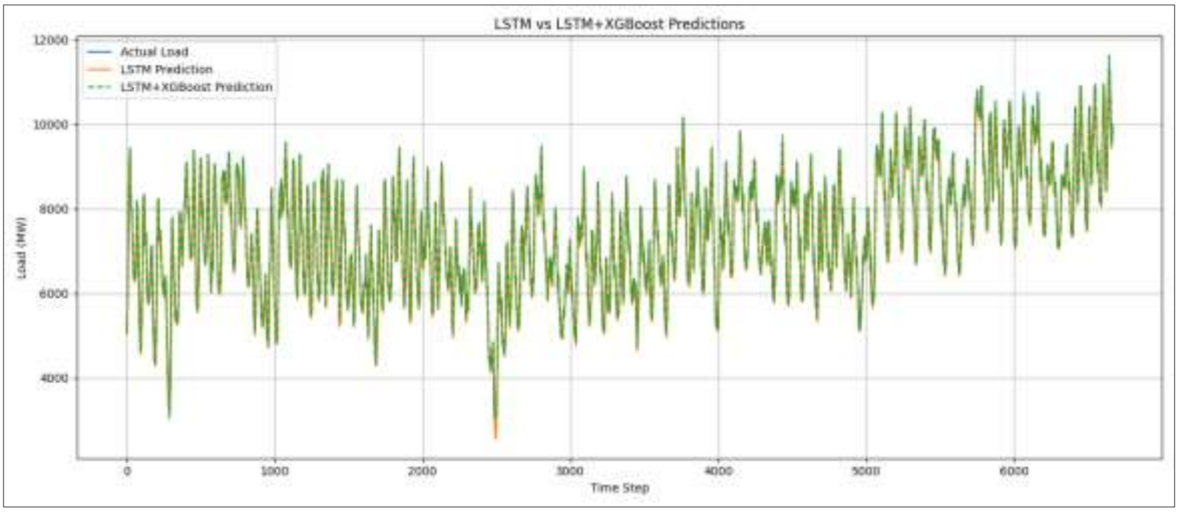

Figure 4.3: Comparison Between LSTM and LSTM–XGBoost Predictions.

Figure 4.4 shows the real electricity consumption or load in an unabridged version together with the forecasts generated by both independent LSTM model and combined LSTM–XGBoost model throughout training and testing. The orange line portrays the LSTM forecasts for the training set whereas the green and red lines denote the LSTM and hybrid models respectively for the test set.

The hybrid model (in red) gets very close to the actual load values during the test, especially in the areas where the load values change rapidly. The hybrid method proves to be superior not only in prediction accuracy but also in error rate under conditions of fluctuation. The ensemble model still shows even more generalization power as the performance on training and testing datasets continues to be distinguishable.

The LSTM model's ability to identify large time intervals and its inability to track fast changes at the same time are depicted in the visualization, while the hybrid model achieves better stability and robustness through XGBoost's residuals fixing. The finding corroborates the claim that ensemble forecasting methods bring about a less risky prediction for real-time load demands in smart grid systems.

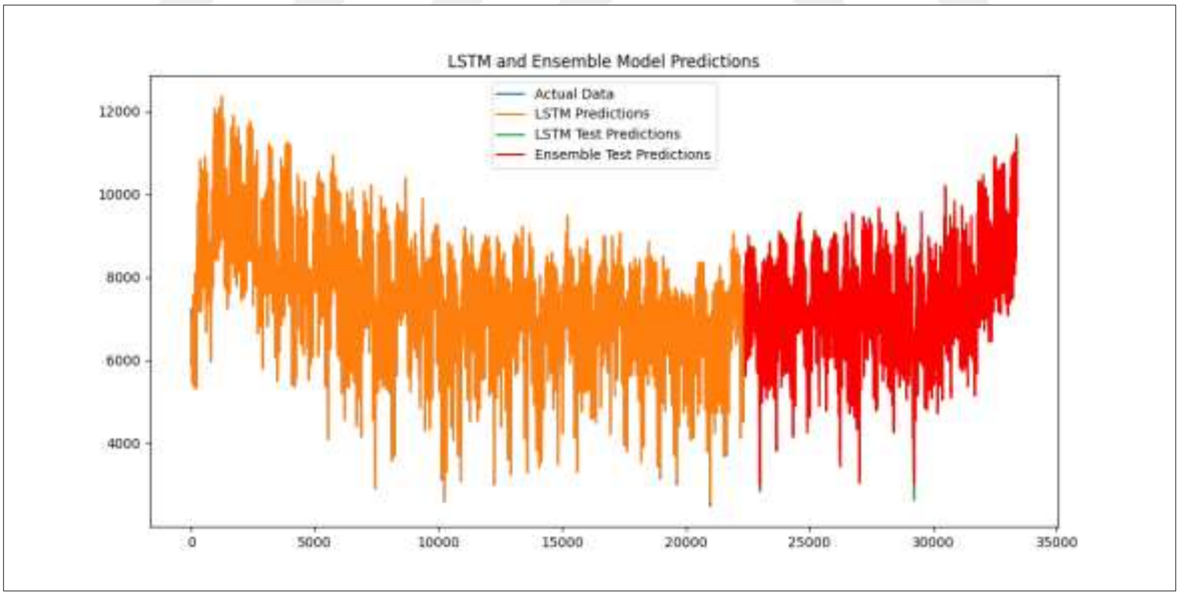

Figure 4.4: Full-Scale Visualization of Training and Test Results.

In summary, the visual comparisons show that the hybrid LSTM–XGBoost model always makes better predictions than the standalone models. The ensemble method gets closer to the actual load values, especially when the load is changing a lot. These visual results give a good. The visual observations presented above reveal noticeable discrepancies in the prediction patterns of each model. A more in-depth explanation and root cause analysis of these errors are provided in the following section. base for the next section, which is a quantitative error analysis.

4.7 ERROR ANALYSIS

The following section provides detailed information about the metrics assessment errors which were discussed in previous sections. The three metrics RMSE, MAPE and R^2 provide accuracy information about models yet they do not account for time-dependent factors or complete error distribution patterns. The analysis of residuals which represent actual values minus model predictions helps us better understand how the model performs across different operational settings. The analysis will assess the residuals produced by the LSTM model and the hybrid LSTM-XGBoost model to evaluate their robustness stability and generalization capabilities.

During the testing phase, the LSTM and hybrid LSTM–XGBoost models were assessed, and their residual plots are illustrated in Figure 4.5. The analysis of this section examines how forecasting errors change over time through the study of residuals across different time periods.

The residuals of the LSTM model exhibit much more fluctuation, particularly at the times when the load is greatest. The LSTM model is good at overall trend prediction, but its forecasting is not reliable under highly volatile conditions.

The hybrid model makes use of the XGBoost component that overcomes the limitations of the LSTM by giving residuals that are very close to the zero-error line. As a result, the error becomes smaller and the outliers become fewer. The hybrid model gains the best performance throughout time periods that need sequential memory for precise predictions.

The hybrid model shows better resistance to noise which makes it suitable for dynamic environments with changing demand patterns.

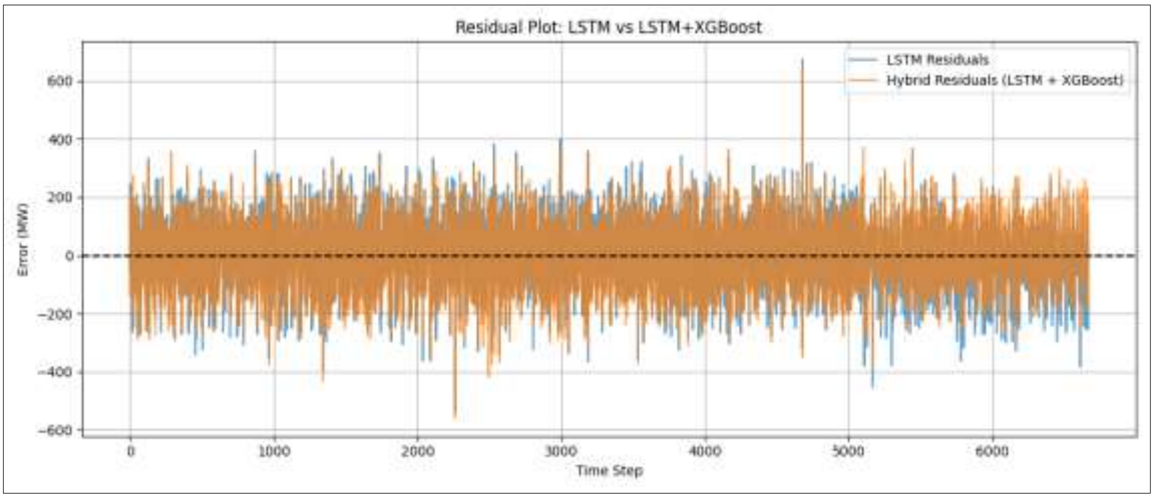

Figure 4.5: Residual plots of LSTM and Hybrid LSTM–XGBoost Models During the Testing Period.

The histogram of residuals from both LSTM and hybrid LSTM–XGBoost models appears in Figure 4.6. The graph shows how prediction errors change from one data point to another.

The residuals of both models show symmetry around a zero mean and have an almost normal distribution which is a highly advantageous characteristic for forecasting models. The distribution of the hybrid model is significantly more concentrated around the mean, indicating a higher frequency of minor errors and diminished variability.

The residuals of the LSTM model show a much wider spread because the independent LSTM makes large errors in both directions from the mean. The independent LSTM fails to match the hybrid variant in terms of overall accuracy and thus produces more significant forecasting errors.

The histogram shows that the hybrid model reduces error variance and reduces extreme deviations which makes the predictions more robust and stable.

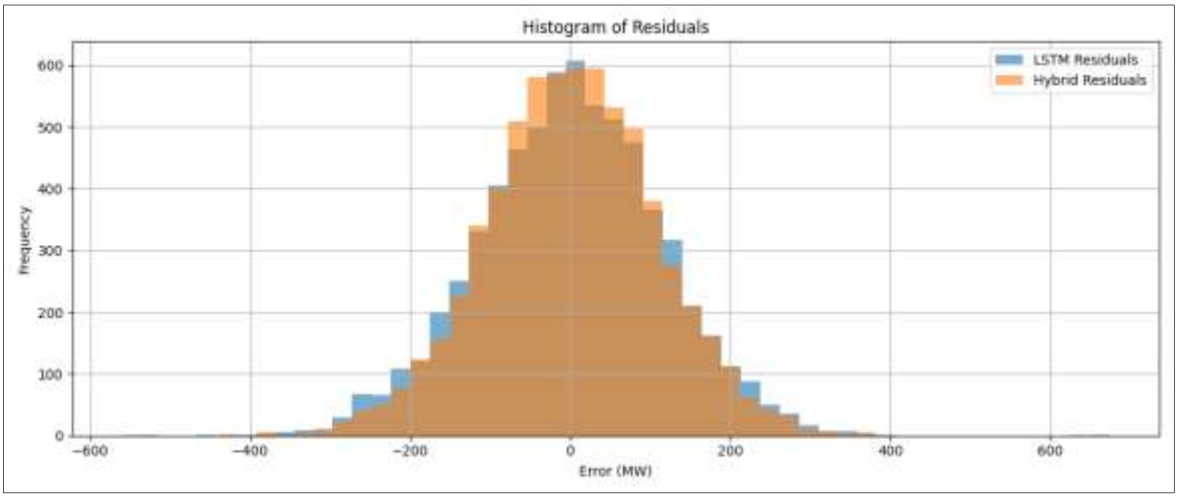

Figure 4.6: Histogram Comparison of Residuals between LSTM and LSTM–XGBoost Models.

The boxplot in Figure 4.7 shows a comparison between the prediction errors generated by the LSTM model and the hybrid LSTM-XGBoost model. The plot gives an overview of residuals by showing their central tendency, dispersion and the location of outliers.

The boxplot of the LSTM model indicates a broader interquartile range, signifying greater variability in its prediction errors. Moreover, the outliers and their distances from the values are significantly more pronounced in the LSTM model, indicating a higher incidence of extreme prediction errors compared to the other model.

The hybrid model demonstrates lower IQR values and fewer outliers which shows it maintains better stability and robustness than the LSTM model. The box shape becomes more compact and approaches zero which shows that most errors are small and evenly distributed.

The boxplots show that the hybrid model which combines LSTM with XGBoost provides better error reliability control than other models for load forecasting.

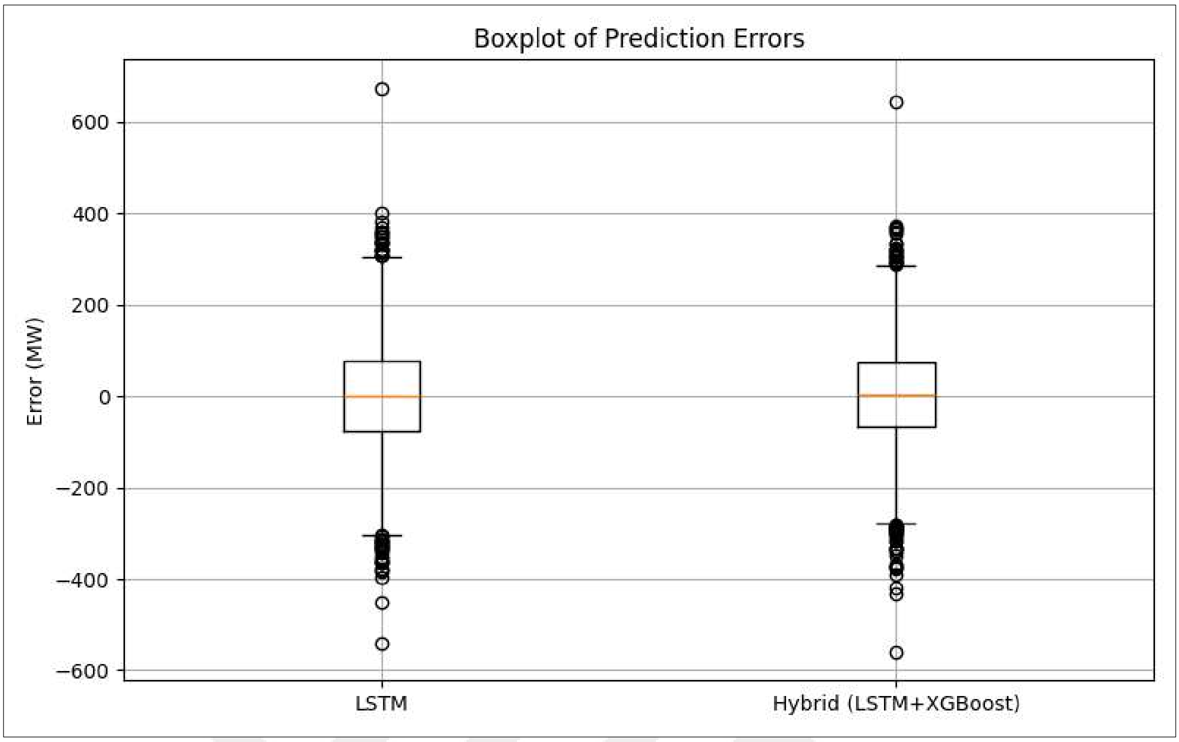

Figure 4.7: Comparison of Prediction Error Distributions Using Boxplots for LSTM and Hybrid Models.

The error analysis reveals the following results:

a. The LSTM model faces challenges when demand experiences sudden spikes because it depends on sequential patterns and fails to detect extreme anomalies.

b. The XGBoost model shows lower sensitivity to spikes but performs poorly for continuous patterns because it lacks sufficient temporal modeling capabilities.

c. The hybrid model combines the advantages of both approaches to improve both accuracy and robustness.

4.8 DISCUSSION

The results point to a significant weakness in the predictive power of the separate models, especially in the situation of different electricity load profiles. The results we got show the benefits of sequential learning and residual correction combination, wheres the hybrid LSTM–XGBoost model presenting a considerable uplift in performance relative to the LSTM and XGBoost models used separately.

The outcomes of the research verify the findings of the previous studies which showed that neural networks combined with ensemble learning methods yield better prediction results. There are many difficulties to implement these hybrid setups. The merging of two various model specifications is a challenging job as the outputs might be very good quality but they do not necessarily satisfy the demands of a particular design situation.

The results of this research are consistent with previous research that has indicated the use of neural networks through ensemble learning techniques to provide better prediction performance [58]. The implementation of such hybrid models has its challenges. Combining two different model types is considered to be a difficult task [59] since the result may be strong but might still not meet the requirements of a certain design context.

The research outcomes corroborate earlier findings that the application of ensemble learning methods with neural networks produces better predictions [58]. There are several difficulties associated with the adoption of these hybrid-models strategies. It is often difficult to combine two different model architectures [59] because even if the output is strong, it may still not be adequate for a specific design situation.

The study of hybrid models continues with new research that investigates the addition of attention mechanisms and lightweight transformer architectures to enhance their performance in dynamic forecasting environments [60].

4.9 SUMMARY

The chapter evaluates the performance of electricity load forecasting models. The hybrid LSTM–XGBoost model produces the most accurate results because it combines models that overcome each other's weaknesses. The results create a basis for future improvements to the proposed model architecture.

5. DISCUSSION AND CONCLUSIONS

5.1 DISCUSSION OF RESULTS

5.1.1 Model Performance Comparison

The evaluation of LSTM, XGBoost and the hybrid LSTM–XGBoost models showed different results in terms of their effectiveness for energy demand forecasting. The evaluation was based on three performance metrics RMSE, MAPE, and R^2.

Among the models that were tested, the hybrid LSTM–XGBoost model yielded the highest level of accuracy across all datasets. The hybrid model was a great combination of LSTM sequential learning and XGBoost boosting efficiency which delivered an RMSE of 106.54 MW, a MAPE of 1.18%, and an R^2 score of 0.994. The performance indicates that the model is excellent in trapping temporal dependencies and the irregularity of the consumption patterns and hence is very much applicable to forecasting electricity demand.

The independent LSTM model exhibited a performance that could be categorized as fair with a RMSE of 119.41 MW and an also slightly higher MAPE of 1.30%. Nevertheless, the model achieved a very strong R^2 of 0.992, which was a clear indicator that the model was capable of capturing time-related dependencies in the input data. The LSTM model is still a viable choice for applications that have well-defined temporal patterns, like energy time series data.

The XGBoost model projected an error of 109.48 MW RMSE and 1.21% MAPE simultaneously obtaining an R^2 value of 0.994. The use of XGBoost in time-series data processing seems to be problematic as this algorithm has shown in this research that it does not have good performance with sequential data. The model in this case was weak because time relations and trends were very important.

5.1.2 Comparison with Previous Studies

The findings of the research indicate that among the models evaluated, the hybrid LSTM-XGBoost model gives the lowest absolute and relative errors. It is the coupling of the LSTM for sequential data analysis with the XGBoost's iterative learning that produces the best results. The research results support the earlier findings regarding hybrid models that are capable of detecting both temporal and non-linear patterns in large data sets.

Furthermore, the model demonstrated consistent performance even under complex, high-frequency input data, unlike several previous studies that relied on lower-resolution or normalized datasets.

5.1.3 Analysis of Hybrid Model Superiority

The LSTM–XGBoost hybrid model achieved better results in electricity load forecasting than both the individual LSTM and XGBoost models for all evaluation metrics. The model demonstrated strong performance through its RMSE and MAPE scores and its robust R^2 score which showed its ability to detect both sequential and nonlinear patterns in the data.

The performance emerges from the ability of hybrid architectures to unite LSTM's long-range temporal dependencies with XGBoost's intricate feature interactions and residual error correction. The hybrid model outperformed the individual models in maintaining robustness during major demand changes which is a critical period for various forecasting systems.

The use of a hybrid architecture that merges the long-range temporal dependency detection of LSTM with the complex feature interactions and residual error correction of XGBoost resulted in the performance. The hybrid model was capable of maintaining its stability during the significant fluctuations in demand, which is a critical period for forecasting systems, something that the separate models could not do and, therefore, lost.

5.2 PRACTICAL IMPLICATIONS

The use of a hybrid architecture that merges the long-range temporal dependency detection of LSTM with the complex feature interactions and residual error correction of XGBoost resulted in the performance. The hybrid model was capable of maintaining its stability during the significant fluctuations in demand, which is a critical period for forecasting systems, something that the separate models could not do and, therefore, lost.

The very exact short-term load forecasting is paramount to the management of intricate electrical systems. The suggested hybrid model not only predicts the demand fluctuation more accurately but also enables the utility operators to optimize the generation schedule thereby diminishing energy wastes and at the same time avoiding the expense of standby power. The model's lowering of RMSE and MAPE values leads to further enhancement of the unit commitment, dispatch planning, and grid stability operational and economic efficiency.

The model demonstrates two principal advantages in managing sudden demand growths and effectively incorporating remaining errors in highly unpredictable scenarios like cities with variable demand and regions dependent on renewable energy sources with unstable supply. Additionally, the model's capacity to deliver forecasts for large high-frequency datasets simplifies its direct integration into national and regional energy management systems.

The hybrid framework provides SCADA and has Energy Management Systems as real-time energy management systems, with the principle asset being the joining of these technologies. The grid operators will be able to obtain accurate load forecasts that will facilitate the automated control operations if this model is adopted on the platforms. Through integration, it is not only possible to have demand-side management but also better fault detection, quick response to disturbances, and preventive maintenance.

The model surely uncovers the energy consumption patterns in more than one way, and this is mainly through the feature importance ranking of XGBoost, which supports the use of AI in forecasting for the institutional energy sector by raising trust and accountability.

The hybrid model assumes the role of a primary structure which allows for the incorporation of ML applications into the sectors using outdated electric grids that either need upgradation or are being converted to decentralized energy systems. The practical features of the model in terms of flexibility, operational speed, and forecasting precision make it an extremely powerful tool that connects data-rich environments and energy management that is focused on making decisions.

The sophisticated LSTM-XGBoost model serves not only as a tool but also as a cornerstone for the creation of intelligent power grids that are versatile and rugged, thereby making them a precious asset in the future energy sector management.

5.3 THEORETICAL CONTRIBUTIONS

In the beginning, the study integrates LSTM networks with XGBoost in order to create models for short-term electricity load forecasting. The study proves that the application of LSTM for the modeling of time-dependency and XGBoost for the correction of residuals results in very high accuracy and robustness in forecasting when these two methods are applied together. The two-step model not only gives rise to a new view of the collaborative power of ensemble learning and DL but also their synergy in terms of understanding the nature of the two.

Secondly, the model's ability to generalize was confirmed by the high-frequency (15-minute intervals) real-world data from a national transmission operator. This study highlights the absence in the prior literature of the integration of fine-grained temporal resolution with hybrid learning approaches.

Third, the research considered the scope of the hybrid attention model's improvement; it was the solitary empirical case concerning the topic. Nevertheless, the results of the experiments brought attention to the fact that across different applications of the attention mechanism, the performance of this particular architecture was not enhanced. Thus, it makes a contribution to the theory stating the limitations of the attention mechanism in the first place when it is applied with residual learning-based architectures frequently, like XGBoost. The research also serves as a warning to other researchers that hybrid structures may lead to elaborate designs where the accuracy is reduced as a trade-off.

Furthermore, the inclusion of the model contributes to the ongoing discussion about the interpretability of DL by demonstrating that the identification of features as important by XGBoost can lessen some of the black-box issues associated with LSTM. This, in turn, paves the way for hybrids whose constructions are grounded in theory and who have the traits of high performance, complete transparency, and intelligibility, the last one particularly essential for use in the energy industry.

In summary, this research extends the theory in three aspects: development of a hybrid model, the use of attention in composite models within the framework, and the balance between precision and explainability in forecasting for smart grids.

5.4 LIMITATIONS OF THE STUDY

Although the LSTM–XGBoost hybrid model proposed showed very good predictive accuracy, still this research has some limitations to be worked on. One of the restrictions is that the research depended on one dataset only provided by one transmission system operator, which might limit its applicability to other power grids with different load characteristics, climate, or other operational constraints.

Secondly, it should be noted that despite the hybrid model outperforming the individual models, this research did not evaluate other architectures, such as varying sequence length, modifying feature space or incorporating other model types, which might lead to better performance. Besides, the non-implementation of extensive hyperparameter tuning and strict cross-validation methods together with the unoptimized model also resulted in the lack of higher accuracy and robustness being displayed by the model.

Third, the attention mechanism was put into practice and tested during the model's development, but it was excluded from the final model because of its slight contribution to the overall performance. However, this does not imply that its influence in other hybrid models or in sequence-to-sequence forecasting frameworks should be ignored.

Nevertheless, the feature importance attribute of XGBoost, which gives some degree of interpretation to the model, does not help in understanding the neural network part of the hybrid model at all. Among future directions, one possible route is to use more explainable AI techniques to increase the transparency and trust in the model.

5.5 FUTURE WORK

The research outcomes lead to multiple possible directions for future investigation.

a. One of the major factors in the model's performance improvement is the use of bigger and more varied datasets that cover longer periods of time. The generalization and robustness of the model would be further strengthened if it were influenced by more factors, for example, seasonal trends, economic indicators, and changes in weather.

b. Though the current LSTM-XGBoost hybrid model achieved higher accuracy, it is recommended that the future experiments deal with the combination of LSTM with either Random Forest or LightGBM ensemble methods. Additionally, the inclusion of GRU or Transformer-based DL architectures might be credited with the performance boost.

c. The study results point out that researchers need to explore various techniques for the selection and extraction of features. The identification of the main factors influencing energy demand would result in a simpler model structure and at the same time, its operational efficiency would be improved.

d. The real-time implementation requirements for energy systems are not considered in the offline accuracy assessment of this work. Continuous model parameter adaptation needs to be studied for the deployment of such models in dynamic environments with live data streams.

e. An evaluation of time series forecasting methods, among which is SARIMA and Prophet, should be done, in order to their effectiveness to be compared with that of the deep learning methods.

5.6 CONCLUSIONS

The study illustrates the functioning of the combined LSTM–XGBoost method for forecasting the power request. The assessment of the model shows that hybrid models are better than single models in terms of both accuracy and reliability, and are able to cope with complex multi-dimensional datasets of any complexity level. The model employs not only DL but also ensemble techniques to discover both the long-term temporal dependencies and the non-linear biases that exist in the time series data.

The performance measures derived from the study show that the LSTM model is fairly accurate for time series forecasting but still less effective than the proposed hybrid model. Pattern recognition in past data that is considered as a factor affecting future energy demands based on historical data is one of the major advantages of LSTM. XGBoost model remains applicable in scenarios where non-linear interactions among the features used are complex but data points order is not significant.

The research findings reveal how hybrid models come to solve the limitations of both ML and traditional techniques. DL models such as LSTM are very good at identifying time-related patterns, but their inability to be completely understood and very variable reactions to hyperparameters hinder their use in specific areas. ML techniques like XGBoost offer strength and higher interpretability along with quicker training, yet they do not get along well with completely sequential data. By combining different approaches, the hybrid model gives rise to substantial benefits for data-driven forecasting.

The research indicates that high-frequency data (15 minutes intervals) can definitely be incorporated into hybrid forecasting systems and additionally the model has been shown to work in actual real-world operations hence supporting its application in contemporary energy systems that need quick and precise prediction models. The model shows its appropriateness for energy management systems by supplying more accurate real-time load forecasting.

One more significant discovery from this investigation is the attention mechanism research in the hybrid framework. It is correct to say that adding attention did not considerably improve the performance of this architecture, yet the outcomes are theoretically valuable since they pinpoint the weaknesses of attention in hybrid models relying on residual correction such as LSTM-XGBoost. This is a very good indication for the future work, especially it will help not to choose very complicated model structures that do not lead to better forecasting accuracy.

As a final point, the study categorized models into hybrid and non-hybrid groups and it was demonstrated that the suggested hybrid architecture could give precise short-term electricity demand forecasts. The study enhances the existing knowledge of the use of DL and ensemble learning techniques, which when properly mixed, will lead to more robust, accurate and interpretable forecasting systems. Additional to this, future work can look into other ML methods and feature engineering to increase prediction accuracy even more. The model's importance is not limited to just academic implications, it also has practical implications in smart grids that need real-time and accurate predictions as the basis for operational decisions that are knowledgeable.

REFERENCES

[1] T. Shering, E. Alonso, and D. Apostolopoulou, "Investigation of Load, Solar and Wind Generation as Target Variables in LSTM Time Series Forecasting, Using Exogenous Weather Variables," *Energies 2024, Vol. 17, Page 1827*, vol. 17, no. 8, p. 1827, Apr. 2024, doi: 10.3390/EN17081827.

[2] G. F. Fan, L. L. Peng, and W. C. Hong, "Short-term load forecasting based on empirical wavelet transform and random forest," *Electrical Engineering*, vol. 104, no. 6, pp. 4433–4449, Dec. 2022, doi: 10.1007/S00202-022-01628-Y/METRICS.

[3] J. Huang and G. Iglesias, "Socioeconomic and climatic impacts on long-term electricity demand: A high-resolution approach through machine learning," *Energy*, p. 137205, 2025, doi: https://doi.org/10.1016/j.energy.2025.137205.

[4] M. Kiasari, M. Ghaffari, and H. H. Aly, "A Comprehensive Review of the Current Status of Smart Grid Technologies for Renewable Energies Integration and Future Trends: The Role of Machine Learning and Energy Storage Systems," *Energies (Basel)*, vol. 17, no. 16, 2024, doi: 10.3390/en17164128.

[5] P. O. Ugbehe, O. E. Diemuodeke, and D. O. Aikhuele, "Electricity demand forecasting methodologies and applications: a review," *Sustainable Energy Research*, vol. 12, no. 1, p. 19, 2025, doi: 10.1186/s40807-025-00149-z.

[6] V. I. Kontopoulou, A. D. Panagopoulos, I. Kakkos, and G. K. Matsopoulos, "A Review of ARIMA vs. Machine Learning Approaches for Time Series Forecasting in Data Driven Networks," *Future Internet*, vol. 15, no. 8, 2023, doi: 10.3390/fi15080255.

[7] K. Bandara, C. Bergmeir, and S. Smyl, "Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach," *Expert Syst Appl*, vol. 140, p. 112896, 2020, doi: https://doi.org/10.1016/j.eswa.2019.112896.

[8] Z. Zhao, Y. Shi, S. Wu, F. Yang, W. Song, and N. Liu, "Interpretation of Time-Series Deep Models: A Survey," May 2023.

[9] Y. Li, J. Xu, and D. C. Anastasiu, “An Extreme-Adaptive Time Series Prediction Model Based on Probability-Enhanced LSTM Neural Networks,” Nov. 2022, doi: 10.1609/aaai.v37i7.26045.

[10] S. R. Sinsel, R. L. Riemke, and V. H. Hoffmann, “Challenges and solution technologies for the integration of variable renewable energy sources—a review,” *Renew Energy*, vol. 145, pp. 2271–2285, Jan. 2020, doi: 10.1016/J.RENENE.2019.06.147.

[11] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, *Recurrent Neural Networks for Short-Term Load Forecasting*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-70338-1.

[12] T. Kandadi and G. Shankarlingam, “DRAWBACKS OF LSTM ALGORITHM: A CASE STUDY,” 2025. doi: 10.2139/ssrn.5080605.

[13] N. Bashiri Behmiri, C. Fezzi, and F. Ravazzolo, “Renewable Sources and Short-To Mid-Term Electricity Price Forecasting,” 2024, doi: 10.2139/ssrn.4744777.

[14] F. Dakheel and M. Çevik, “Optimizing Smart Grid Load Forecasting via a Hybrid Long Short-Term Memory-XGBoost Framework: Enhancing Accuracy, Robustness, and Energy Management,” *Energies (Basel)*, vol. 18, no. 11, 2025, doi: 10.3390/en18112842.

[15] F. Dakheel and M. Çevik, “Optimizing Smart Grid Load Forecasting via a Hybrid LSTM-XGBoost Framework: Enhancing Accuracy, Robustness, and Energy Management,” May 26, 2025. doi: 10.20944/preprints202505.0521.v2.

[16] F. Guo *et al.*, “A Hybrid Stacking Model for Enhanced Short-Term Load Forecasting,” *Electronics (Basel)*, vol. 13, no. 14, 2024, doi: 10.3390/electronics13142719.

[17] P. Cosgrove, T. Roulstone, and S. Zachary, “Intermittency and periodicity in net-zero renewable energy systems with storage,” *Renew Energy*, vol. 212, pp. 299–307, 2023, doi: https://doi.org/10.1016/j.renene.2023.04.135.

[18] S. Ali, S. Bogarra, M. N. Riaz, P. P. Phyo, D. Flynn, and A. Taha, "From Time-Series to Hybrid Models: Advancements in Short-Term Load Forecasting Embracing Smart Grid Paradigm," *Applied Sciences*, vol. 14, no. 11, 2024, doi: 10.3390/app14114442.

[19] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 379, no. 2194, p. 20200209, Apr. 2021, doi: 10.1098/rsta.2020.0209.

[20] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications*. in Springer Texts in Statistics. New York, NY, USA: Springer International Publishing, 2017. doi: 10.1007/978-3-319-52452-8.

[21] Y. Zhou, S. Aryal, and M. R. Bouadjenek, "Review for Handling Missing Data with special missing mechanism," Apr. 2024, Accessed: May 21, 2025. [Online]. Available: https://arxiv.org/pdf/2404.04905

[22] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3rd ed. Melbourne, Australia: OTexts, 2021. Accessed: May 21, 2025. [Online]. Available: https://otexts.com/fpp3/

[23] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-August-2016, pp. 785–794, Aug. 2016, doi: 10.1145/2939672.2939785/SUPPL_FILE/KDD2016_CHEN_BOOSTING_SYSTEM_01-ACM.MP4.

[24] A. Abdullah and M. A. Qureshi, "Unsupervised learning techniques for IoT time series," *Sens Actuators A Phys*, vol. 336, pp. 112–119, 2022.

[25] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection," *ACM Comput Surv*, vol. 41, no. 3, pp. 1–58, Jul. 2009, doi: 10.1145/1541880.1541882.

[26] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/S10618-019-00619-1/METRICS.

[27] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, "A Comparison of ARIMA and LSTM in Forecasting Time Series," *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, pp. 1394–1401, Jul. 2018, doi: 10.1109/ICMLA.2018.00227.

[28] A. Nejadettehad, H. Mahini, and B. Bahrak, "Short-term Demand Forecasting for Online Car-hailing Services using Recurrent Neural Networks," *Applied Artificial Intelligence*, vol. 34, no. 9, pp. 674–689, Jan. 2019, doi: 10.1080/08839514.2020.1771522.

[29] A. P. Wibawa, A. B. P. Utama, H. Elmunsyah, U. Pujianto, F. A. Dwiyanto, and L. Hernandez, "Time-series analysis with smoothed Convolutional Neural Network," *J Big Data*, vol. 9, no. 1, p. 44, 2022, doi: 10.1186/s40537-022-00599-y.

[30] X. Huang *et al.*, "A Hybrid ARIMA-LSTM-XGBoost Model with Linear Regression Stacking for Transformer Oil Temperature Prediction," *Energies (Basel)*, vol. 18, no. 6, p. 1432, Mar. 2025, doi: 10.3390/en18061432.

[31] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal Fusion Transformers for interpretable multi-horizon time series forecasting," *Int J Forecast*, vol. 37, no. 4, pp. 1748–1764, Oct. 2021, doi: 10.1016/j.ijforecast.2021.03.012.

[32] H. Chen, S. Lundberg, and S.-I. Lee, "Hybrid Gradient Boosting Trees and Neural Networks for Forecasting Operating Room Data," Jan. 2018.

[33] Z. Meng, Y. Guo, and H. Sun, "An Adaptive Approach for Probabilistic Wind Power Forecasting Based on Meta-Learning," *IEEE Trans Sustain Energy*, vol. 15, no. 3, pp. 1814–1833, Aug. 2023, doi: 10.1109/TSTE.2024.3379835.

[34] A. Botchkarev, "A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms," *Interdisciplinary Journal of Information, Knowledge, and Management*, vol. 14, pp. 045–076, 2019, doi: 10.28945/4184.

[35] N. V. Chereddy and B. K. Bolla, “Evaluating the Utility of GAN Generated Synthetic Tabular Data for Class Balancing and Low Resource Settings,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 14078 LNAI, pp. 48–59, 2023, doi: 10.1007/978-3-031-36402-0_4.

[36] E. Sarmas, N. Dimitropoulos, V. Marinakis, Z. Mylona, and H. Doukas, “Transfer learning strategies for solar power forecasting under data scarcity,” *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–13, Aug. 2022, doi: 10.1038/s41598-022-18516-x.

[37] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, “Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection,” *Sensors*, vol. 24, no. 4, p. 1230, Feb. 2024, doi: 10.3390/s24041230.

[38] Y. Gafni, R. Gradwohl, and M. Tennenholtz, “Prediction-sharing During Training and Inference,” Mar. 2024, Accessed: May 19, 2025. [Online]. Available: https://arxiv.org/pdf/2403.17515

[39] M. Jin *et al.*, “Time-LLM: Time Series Forecasting by Reprogramming Large Language Models,” *12th International Conference on Learning Representations, ICLR 2024*, Oct. 2023, Accessed: May 19, 2025. [Online]. Available: https://arxiv.org/pdf/2310.01728

[40] B. Biswal, S. Deb, S. Datta, T. S. Ustun, and U. Cali, “Review on smart grid load forecasting for smart energy management using machine learning and deep learning techniques,” *Energy Reports*, vol. 12, pp. 3654–3670, 2024, doi: https://doi.org/10.1016/j.egyr.2024.09.056.

[41] R. Liu and S.-Y. Shin, “A Review of Traffic Flow Prediction Methods in Intelligent Transportation System Construction,” *Applied Sciences*, vol. 15, no. 7, 2025, doi: 10.3390/app15073866.

[42] D. Kaur, S. N. Islam, Md. A. Mahmud, Md. E. Haque, and Z. Dong, “Energy Forecasting in Smart Grid Systems: A Review of the State-of-the-art Techniques,” May 2022, doi: arXiv:2011.12598v3.

[43] W. Cheng, T. Ma, X. Wang, and G. Wang, "Anomaly Detection for Internet of Things Time Series Data Using Generative Adversarial Networks With Attention Mechanism in Smart Agriculture," *Front Plant Sci*, vol. Volume 13-2022, 2022, doi: 10.3389/fpls.2022.890563.

[44] W. Zheng, L. Yan, C. Gou, and F.-Y. Wang, "Computational knowledge vision: paradigmatic knowledge based prescriptive learning and reasoning for perception and vision," *Artif Intell Rev*, vol. 55, no. 8, pp. 5917–5952, Dec. 2022, doi: 10.1007/s10462-022-10166-9.

[45] Z. Meng, Y. Guo, and H. Sun, "An Adaptive Approach for Probabilistic Wind Power Forecasting Based on Meta-Learning," *IEEE Trans Sustain Energy*, vol. 15, no. 3, pp. 1814–1833, Jul. 2024, doi: 10.1109/TSTE.2024.3379835.

[46] K. Noor and U. Fatima, "Meta Learning Strategies for Comparative and Efficient Adaptation to Financial Datasets," *IEEE Access*, vol. 13, pp. 24158–24170, 2025, doi: 10.1109/ACCESS.2024.3516490.

[47] J. Cui *et al.*, "Advanced Short-Term Load Forecasting with XGBoost-RF Feature Selection and CNN-GRU," *Processes*, vol. 12, no. 11, p. 2466, Nov. 2024, doi: 10.3390/pr12112466.

[48] A. M. Hayajneh, F. Alasali, A. Salama, and W. Holderbaum, "Intelligent Solar Forecasts: Modern Machine Learning Models and TinyML Role for Improved Solar Energy Yield Predictions," *IEEE Access*, vol. 12, pp. 10846–10864, 2024, doi: 10.1109/ACCESS.2024.3354703.

[49] A. Grataloup, S. Jonas, and A. Meyer, "A review of federated learning in renewable energy applications: Potential, challenges, and future directions," Dec. 2023.

[50] N. E. Benti, M. D. Chaka, and A. G. Semie, "Forecasting Renewable Energy Generation with Machine Learning and Deep Learning: Current Advances and Future Prospects," *Sustainability*, vol. 15, no. 9, p. 7087, Apr. 2023, doi: 10.3390/su15097087.

[51] L. Semmelmann, S. Henni, and C. Weinhardt, "Load forecasting for energy communities: a novel LSTM-XGBoost hybrid model based on smart meter data," *Energy Informatics*, vol. 5, no. S1, p. 24, Sep. 2022, doi: 10.1186/s42162-022-00212-9.

[52] A. A. Mir, M. Alghassab, K. Ullah, Z. A. Khan, Y. Lu, and M. Imran, "A Review of Electricity Demand Forecasting in Low and Middle Income Countries: The Demand Determinants and Horizons," *Sustainability*, vol. 12, no. 15, p. 5931, Jul. 2020, doi: 10.3390/su12155931.

[53] G. F. Martin Nascimento, F. Wurtz, P. Kuo-Peng, B. Delinchant, and N. Jhoe Batistela, "Outlier Detection in Buildings' Power Consumption Data Using Forecast Error," *Energies (Basel)*, vol. 14, no. 24, p. 8325, Dec. 2021, doi: 10.3390/en14248325.

[54] Y.-S. Kim, M. K. Kim, N. Fu, J. Liu, J. Wang, and J. Srebric, "Investigating the impact of data normalization methods on predicting electricity consumption in a building using different artificial neural network models," *Sustain Cities Soc*, vol. 118, p. 105570, 2025, doi: https://doi.org/10.1016/j.scs.2024.105570.

[55] B. Tang, J. Hu, M. Yang, C. Zhang, and Q. Bai, "Enhancing Short-Term Load Forecasting Accuracy in High-Volatility Regions Using LSTM-SCN Hybrid Models," *Applied Sciences*, vol. 14, no. 24, p. 11606, Dec. 2024, doi: 10.3390/app142411606.

[56] M.-C. Chiu, H.-W. Hsu, K.-S. Chen, and C.-Y. Wen, "A hybrid CNN-GRU based probabilistic model for load forecasting from individual household to commercial building," *Energy Reports*, vol. 9, pp. 94–105, Oct. 2023, doi: 10.1016/j.egyr.2023.05.090.

[57] X. Wen, J. Liao, Q. Niu, N. Shen, and Y. Bao, "Deep learning-driven hybrid model for short-term load forecasting and smart grid information management," *Sci Rep*, vol. 14, no. 1, p. 13720, Jun. 2024, doi: 10.1038/s41598-024-63262-x.

[58] G. Duan and J. Dong, “Construction of Ensemble Learning Model for Home Appliance Demand Forecasting,” *Applied Sciences*, vol. 14, no. 17, p. 7658, Aug. 2024, doi: 10.3390/app14177658.

[59] A. Frifra, M. Maanan, M. Maanan, and H. Rhinane, “Harnessing LSTM and XGBoost algorithms for storm prediction,” *Sci Rep*, vol. 14, no. 1, p. 11381, May 2024, doi: 10.1038/s41598-024-62182-0.

[60] K. Cao, T. Zhang, and J. Huang, “Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems,” *Sci Rep*, vol. 14, no. 1, p. 4890, Feb. 2024, doi: 10.1038/s41598-024-55483-x.

APPENDICES

APPENDIX A

SAMPLE DATASET OVERVIEW

This appendix provides a sample from the Elia Grid Load dataset used in this study. The table below (Table A1) presents representative entries to illustrate the structure, frequency, and value ranges of the recorded electricity load data.

Table A.1: Sample entries from the Elia Grid Load Dataset.

Datetime (CET+1/CEST+2)	Datetime (UTC)	Elia Grid Load [MW]
1/1/2022 0:00	12/31/2021 23:00	7229.321
1/1/2022 0:15	12/31/2021 23:15	7141.165
1/1/2022 0:30	12/31/2021 23:30	7066.856
1/1/2022 0:45	12/31/2021 23:45	6956.022
1/1/2022 1:00	1/1/2022 0:00	6906.256
1/1/2022 1:15	1/1/2022 0:15	6739.079
1/1/2022 1:30	1/1/2022 0:30	6599.595
1/1/2022 1:45	1/1/2022 0:45	6460.616
1/1/2022 2:00	1/1/2022 1:00	6354.088
1/1/2022 2:15	1/1/2022 1:15	6271.309

APPENDIX B

PYTHON CODE – HYBRID LSTM–XGBOOST

The complete implementation of the hybrid LSTM-XGBoost forecasting framework as proposed and evaluated in this dissertation is captured in the Python code below. The work is based on the methodology presented on Chapter 3 and performed on the Elia Grid dataset for short-term load forecasting. It consists of data cleaning and preprocessing, trimming of metadata features, engineering of new relevant features, sequence generation, model fitting, applying residual corrections using XGBoost, and evaluation of the model based on RMSE, MAPE, and R^2. The implementation was done on Python 3.11 with TensorFlow, scikit-learn, XGBoost and Matplotlib.

```
# === Import Libraries ===
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error,
mean_absolute_percentage_error, r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout
import xgboost as xgb
import matplotlib.pyplot as plt

# === Load and Preprocess Data ===
df = pd.read_csv('data20221.csv')
df['Datetime'] = pd.to_datetime(df['Datetime'])
df['load'] = df['load'].str.replace(',', '').astype('float32') / 1000  #
Convert to MW
df.dropna(inplace=True)

# === Feature Engineering ===
df['hour'] = df['Datetime'].dt.hour
df['day_of_week'] = df['Datetime'].dt.dayofweek
df['month'] = df['Datetime'].dt.month
df['is_weekend'] = (df['day_of_week'] >= 5).astype(int)

# Lag features
for lag in [1, 2, 3]:
    df[f'lag_{lag}'] = df['load'].shift(lag)

# Rolling window features
df['rolling_mean_6'] = df['load'].rolling(window=6).mean()
```

```
df['rolling_std_6'] = df['load'].rolling(window=6).std()

df.dropna(inplace=True)

# === Define Features and Target ===
feature_cols = ['load', 'hour', 'day_of_week', 'month', 'is_weekend',
                'lag_1', 'lag_2', 'lag_3', 'rolling_mean_6',
'rolling_std_6']
target_col = 'load'

scaler = MinMaxScaler()
scaled_features = scaler.fit_transform(df[feature_cols])

look_back = 60

# === Create Sequences for LSTM ===
def create_dataset(features, target, look_back=60):
    X, Y = [], []
    for i in range(look_back, len(features)):
        X.append(features[i - look_back:i])
        Y.append(target[i])
    return np.array(X), np.array(Y)

X, Y = create_dataset(scaled_features, scaled_features[:, 0], look_back)

# === Train/Test Split ===
train_size = int(len(X) * 0.8)
X_train, X_test = X[:train_size], X[train_size:]
Y_train, Y_test = Y[:train_size], Y[train_size:]

# === LSTM Model Definition ===
model = Sequential()
model.add(LSTM(50, return_sequences=True, input_shape=(look_back,
X.shape[2])))
model.add(Dropout(0.2))
model.add(LSTM(50))
model.add(Dropout(0.2))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X_train, Y_train, epochs=50, batch_size=32, verbose=2)

# === LSTM Predictions ===
trainPredict = model.predict(X_train)
testPredict = model.predict(X_test)

# === Inverse Transform Target ===
def inverse_only_load(pred, features_ref):
```

```python
    merged = np.concatenate([pred, features_ref[:, -1, 1:]], axis=1)
    return scaler.inverse_transform(merged)[:, 0]

trainPredict_inv = inverse_only_load(trainPredict, X_train)
testPredict_inv = inverse_only_load(testPredict, X_test)
Y_train_inv = inverse_only_load(Y_train.reshape(-1, 1), X_train)
Y_test_inv = inverse_only_load(Y_test.reshape(-1, 1), X_test)

# === XGBoost Refinement ===
xgb_model = xgb.XGBRegressor(objective='reg:squarederror', n_estimators=100,
learning_rate=0.1)
xgb_model.fit(trainPredict_inv.reshape(-1, 1), Y_train_inv.ravel())
xgb_testPredict = xgb_model.predict(testPredict_inv.reshape(-1,
1)).reshape(-1, 1)

# === Evaluation Metrics ===
rmse = np.sqrt(mean_squared_error(Y_test_inv, xgb_testPredict))
mape = mean_absolute_percentage_error(Y_test_inv, xgb_testPredict) * 100
r2 = r2_score(Y_test_inv, xgb_testPredict)

print("=== LSTM + XGBoost (with Feature Engineering) ===")
print(f"RMSE: {rmse:.2f} MW")
print(f"MAPE: {mape:.2f} %")
print(f"R²: {r2:.3f}")

# === Plot Results ===
plt.figure(figsize=(14, 6))
plt.plot(Y_test_inv, label='Actual Load')
plt.plot(testPredict_inv, label='LSTM Prediction')
plt.plot(xgb_testPredict, label='LSTM + XGBoost Prediction', linestyle='--')
plt.title('LSTM + XGBoost with Feature Engineering')
plt.xlabel('Time Step')
plt.ylabel('Load (MW)')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```