sustainability

MDPI

Article

An Ensemble Approach to Predict a Sustainable Energy Plan for London Households

Niraj Buyo , **Akbar Sheikh-Akbari** * and **Farrukh Saleem**

School of Built Environment, Engineering and Computing, Leeds Beckett University, Leeds LS6 3QR, UK; n.buyo2974@student.leedsbeckett.ac.uk (N.B.); f.saleem@leedsbeckett.ac.uk (F.S.)
* Correspondence: a.sheikh-akbari@leedsbeckett.ac.uk

Abstract: The energy sector plays a vital role in driving environmental and social advancements. Accurately predicting energy demand across various time frames offers numerous benefits, such as facilitating a sustainable transition and planning of energy resources. This research focuses on predicting energy consumption using three individual models: Prophet, eXtreme Gradient Boosting (XGBoost), and long short-term memory (LSTM). Additionally, it proposes an ensemble model that combines the predictions from all three to enhance overall accuracy. This approach aims to leverage the strengths of each model for better prediction performance. We examine the accuracy of an ensemble model using Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE) through means of resource allocation. The research investigates the use of real data from smart meters gathered from 5567 London residences as part of the UK Power Networks-led Low Carbon London project from the London Datastore. The performance of each individual model was recorded as follows: 62.96% for the Prophet model, 70.37% for LSTM, and 66.66% for XGBoost. In contrast, the proposed ensemble model, which combines LSTM, Prophet, and XGBoost, achieved an impressive accuracy of 81.48%, surpassing the individual models. The findings of this study indicate that the proposed model enhances energy efficiency and supports the transition towards a sustainable energy future. Consequently, it can accurately forecast the maximum loads of distribution networks for London households. In addition, this work contributes to the improvement of load forecasting for distribution networks, which can guide higher authorities in developing sustainable energy consumption plans.

Keywords: ensemble model; LSTM; Prophet; XGBoost; energy load forecasting; time series analysis; sustainable energy plan

Academic Editor: Antonio Messineo

Received: 22 November 2024
Revised: 29 December 2024
Accepted: 7 January 2025
Published: 10 January 2025

Citation: Buyo, N.; Sheikh-Akbari, A.; Saleem, F. An Ensemble Approach to Predict a Sustainable Energy Plan for London Households. *Sustainability* **2025**, *17*, 500. https://doi.org/10.3390/su17020500

1. Introduction

Precisely predicting the highly variable data of home energy consumption (EC) is essential for modern smart cities. The rapid advancement of artificial intelligence has resulted in the widespread use of deep learning technologies, such as long short-term memory (LSTM) neural networks, to tackle the energy consumption forecasting (ECF) challenge for individual households [1]. It helps the government to develop more sensible and sustainable growth strategies while also responding quickly to fluctuations in energy demand. Furthermore, smart grids assist the effective control of power supply and demand, ensuring that the regular electricity requirements of businesses, residences, and other organizations are met [2].

Traditional electricity consumption (EC) research approaches require the adoption of separate models for diverse datasets. Conventional techniques for EC forecasting usually need direct data access, which may not be ideal for developing smart cities in real-world scenarios. Additionally, the EC dataset is a typical non-stationary time series. Significant shifts and apparent mutations are two of its distinguishing qualities. It is not immediately obvious whether the data are periodic. One of the trickiest aspects of time series forecasting studies nowadays is predicting non-stationary data. Moreover, one of the current issues is understanding the long-term dependence on sequences with exact temporal properties [3].

Forecast models are distinguished by numerous sets of variables, including those related to expected power consumption. Demand is influenced by many factors, such as the length of the forecast (long-, medium-, and short-term), degree of load aggregation, temperature, and socioeconomic activity. Considering many historical periods helps to effectively manage demand forecasting activities. However, there are still areas where investigation is insufficient. The traditional approach is praised for its practicality, while nonlinear and non-stationary sequences show inferior performance. The ARIMA model, while effective for non-stationary sequences, is ineffective for nonlinear ones due to its difficulty in adjusting its parameters. The Back Propagation (BP) neural network model faces challenges in incorporating temporal information into artificial intelligence models, despite its ability to accumulate time domain information [4]. These issues include gradient vanishing and gradient exploding, which limit the ability to accurately predict multivariate time series. Recent findings on attention mechanisms, which calculate variable correlations, may address input/output dependency and provide accurate long-term predictions, but they may be inaccurate due to not including short-term sequence components. Although each model has its limits, ensemble models may overcome these restrictions by using the unique abilities of several models.

Prior research has mostly neglected the benefits of incorporating optimization models, instead placing emphasis on the use of a single model. Ensemble approaches may improve forecast accuracy by leveraging the capabilities of multiple models. In addition, the attention mechanism, which has the capability to address the relationship between input and output, has not been thoroughly investigated in combination with LSTM for the specific task of short-term load forecasting [5]. This research specifically attempts to address an information gap by investigating the potential of ensemble models for predicting short-term energy consumption. The purpose of this technique is to enhance the accuracy and efficiency of short-term demand forecasting, resulting in valuable data for energy management and grid optimization.

Short-term forecasting often predicts power usage over the next several hours, days, or weeks. Its applications in the energy market include demand-side management and daily supply planning [6]. Petropoulos et al. [7] highlight statistical models and hybrid methods that combine custom-designed vector machines or deep learning models (DLMs). This paper also describes how to employ these strategies to infer objects. It offers numerous techniques for short-term demand forecasting, using deep learning approaches [7]. These models usually include climatic and temporal variables connected with energy use. Despite the literature on this topic, Hong and Wang, P. [8] stressed the significance of analysing such procedures, studying preprocessing methods, and conducting extra research to discover how each variable affects output [8].

Accuracy in estimating home energy consumption (EC) is vital for the development of smart cities, as it enables effective regulation of energy demand and optimization of power supply via smart networks [9]. Traditional investigation techniques require distinct models for each dataset, thereby hindering reusability and necessitating frequent retraining. In addition, conventional forecasting techniques often need direct access to data, which may not be ideal for practical applications in smart cities. Furthermore, the data from EC exhibit non-stationary features, making it difficult to predict quick changes and unclear periodicity.

Deep learning models, particularly LSTM networks, have shown potential in addressing these issues [10]. By deploying these algorithms centrally to analyse data from several houses, the need for retraining with each new dataset is avoided. In addition to managing non-stationary data, deep learning offers a more robust predictive tool.

Through the comparison of various machine learning and deep learning models, we can gain insights into their effectiveness in handling complex time series data with multiple dimensions and identifying relevant characteristics [11]. This knowledge can then be used to estimate energy usage in smart cities. LSTM was chosen for its superior performance in time series forecasting, particularly in managing sequential data and capturing temporal connections. Although attention-based models exhibit greater performance across several applications, LSTM remains a strong choice due to its ability to handle temporal dependencies effectively [12]. LSTMs are a reliable option for this project, owing to their proven efficacy in identical areas and reduced interpreting requirements [13,14]. LSTMs were chosen due to their proven reliability and efficiency in time series modelling.

This paper evaluates the efficacy, comprehensibility, and significance of three widely used models for predicting short-term energy demand: Recurrent Neural Networks (LSTM), automated ARIMA regression (XGBoost Model), and additive model (Prophet model). It then proposes an ensemble model using these three models, which significantly outperforms the individual models in terms of accuracy in predicting short-term household energy demand. The model's efficacy was assessed using metrics like MAE, RMSE, and MAPE to measure its performance in predicting outcomes. The model's comprehensibility was enhanced by filtering features based on correlation thresholds, assigning equal importance to all features, and simplifying the interpretation of model predictions. The author evaluated the model's reliability by comparing predicted values to actual test data, using scikit-learn to calculate the accuracy score, which quantifies the proportion of correct predictions made by the model [15]. The author utilized an ensemble method, combining predictions from LSTM, XGBoost, and Prophet, to enhance predictive performance. LSTM was chosen for its ability to capture temporal dependencies in sequential data, XGBoost for its strong performance with structured data, and Prophet for its effectiveness in handling seasonality and trends. These models were chosen over more sophisticated deep learning approaches, such as attention-based models, due to their proven efficacy in similar circumstances, reduced computational demands, and alignment with the study's objectives [16].

This research attempts to compare three distinct types of models, namely the additive model, the automated Autoregressive Moving Average (ARIMA) regression model, and the Recurrent Neural Network model, to estimate residential energy consumption. These models are often used in literature reviews. Recurrent-based models, such as LSTM, are deep learning algorithms. By comparing these models, this study aims to offer new insights into measuring performance, interpretability, and the importance of feature engineering in applying various forecasting models for short-term energy demand.

The main contribution of this research is listed below:

- Assessing the efficacy, comprehensibility, and significance of three widely used models for predicting short-term energy demand: Recurrent Neural Networks—LSTM, automated ARIMA regression—XGBoost model, and additive model—Prophet model.
- Proposing an ensemble approach using the three abovementioned state-of-the-art time series specialist models to improve accuracy of their short-term energy demand prediction.
- Various case studies are employed to assess the performance of models throughout both training and testing phases, utilizing varying sample proportions and models. This study presents multiple perspectives on the specific qualities and consequences of low readings during training.

- Preparing a final dataset comprises energy consumption, temperature, wind speed, humidity, cloud cover, and other weather factors, making it a comprehensive resource for constructing a predictive mode.
- Evaluate the model's performance during the assessment phase by using characteristics that consider the model's specificity and dataset. This study simplifies the value of all factors related to forecasting the research item.
- To evaluate the final performance of energy consumption forecasting, consider the kind of machine learning (ML) models, the number of characteristics, the selection method after training, and ensemble approach. Within the framework of short-term energy demand forecasting, the comparison helps to clarify the advantages and disadvantages of certain approaches and models.

The main goal of the investigation is to assess ML models for energy consumption estimates. The prediction approaches used include recurrent models such as LSTM, along with XGBoost and the Prophet model. Simulation results highlight the lack of information in predicting short-term energy demand for low-voltage systems. They offer new perspectives on the effectiveness and interpretability of additive models, automated ARIMA regression models, and recurrence-based models. The comparison enables handling multidimensional series in time frames, explaining the preprocessing technique, and ascertaining the relative importance of every feature. The research proposed in this paper can be helpful for future studies on DL architecture design, preprocessing, and model training. This research makes significant contributions to the field. The research's scientific contribution lies in its ability to do the following:

- Advance the state of the art: The ensemble approach and the insights gained from model comparison contribute to the advancement of techniques for short-term energy demand forecasting.
- Provide practical guidance: The comprehensive evaluation and analysis of different models and features offer practical guidance for researchers and practitioners working on energy consumption prediction.
- Address a specific research gap: By focusing on low-voltage systems, the research fills a void in the existing literature. It addresses a specific gap by concentrating on short-term energy demand prediction for low-voltage systems, where information is often limited.

Overall, the research provides valuable insights into model performance and feature importance through its comprehensive dataset, rigorous evaluation, novel ensemble approach, and potential replication of existing methodologies. The rest of the paper is structured as follows. Section 2 delves into the research background, offering a comprehensive examination of three major models: the additive model—Prophet, the automated ARIMA regression model—XGBoost, and the Recurrent Neural Network model—LSTM. Section 3 outlines the methods used for data processing and analysis, as well as the assessment criteria for the additive model—Prophet, the Recurrent Neural Network model—LSTM, and the automated ARIMA regression model—XGBoost. Moreover, it provides an in-depth discussion of the proposed method for forecasting household energy usage. It covers several aspects such as dataset description, preprocessing, feature selection, model training (Prophet, LSTM, and XGBoost), ensemble learning, and model evaluation using MAE, RMSE, and MAPE. Section 4 discusses experimental results. This phase also assesses several forecasting models, such as Prophet, LSTM, and XGBoost, to predict energy usage. The assessment criteria, which include MAE, RMSE, and MAPE, are explored to determine the performance of different models. Additionally, it examines the potential of combining these models into an ensemble technique to improve forecast accuracy. The ensemble model, which integrates multiple models, exhibited superior accuracy compared to the individual models. Finally, Section 5 concludes with some final thoughts on the research and provides recommendations for future research.

2. Research Background

The necessity of resource analysis when examining multivariable time series consumption is highlighted in the work of Forootani et al. [15]. Authors provided case studies that demonstrate how decreasing features improves predictive model accuracy. The author examines the recognition of outliers, machine and deep learning model comparisons, and the use of DL techniques to analyse short-term residential load estimates. It would be noteworthy to compare and evaluate the precision of ML models in the context of consumption prediction applications, despite their classification as evolutionary architectures [15].

Traditional forecasting techniques mainly depend on historical time series data and use primitive algorithms that may rapidly compute outcomes. Regression equations are frequently used to forecast future loads by analysing load data and the factors that influence the prediction. This approach treats load data as independent and dependent variables. Each technique uses historical data to generate forecasts; yet, they all employ overly simplistic frameworks, fail to explore data connections, and consider past and future data merely as mathematical equations, resulting in inaccurate predictions [16]. The rapid advancement of machine learning has led to several accomplishments in the domain of health [17], automatic detection [18], security [19], and energy load forecasting [20,21]. Typical machine learning techniques include decision trees [22], image classification, and support vector machines [23], highlighting the capabilities of ML algorithms in proposing an appropriate solution for different problems.

The Prophet LSTM model aims to improve existing methods by achieving high data trend fitting and accurate predictions [16,24,25]. It optimizes data trend matching and enhances accuracy simultaneously. The Prophet algorithm is useful for aligning recurring patterns in power load data, making it a valuable tool for achieving high-level data trend fitting and accurate predictions. Facebook introduced the Prophet algorithm (PA), which implements an additive/multiplicative structure for predicting time series data. By utilizing a combination of yearly, weekly, daily, seasonal, holiday, and external features, this algorithm makes it possible to model both linear and nonlinear trends. A hybrid ensemble model using a stacking approach is presented [24]. This model uses multiple attributes to increase prediction accuracy for short-term load estimate. The research used multiple attributes for improving accuracy such as temperature, rainfall, and daily electricity prices.

Recurrent Neural Networks (RNNs) are a category of artificial neural networks capable of efficiently processing sequential input by utilizing feedback connections. Still, a common RNN concern is the vanishing or growing gradient problem. This happens when the gradients rise or shrink exponentially, making it more difficult for the network to figure out long-term relationships. Bouktif et al. [25] use a hybrid model that integrates LSTM networks with genetic algorithms to forecast the electricity requirements of metropolitan regions in France. Feature selection and hyperparameter optimization are two examples of this approach.

The development of XGBoost was a significant advancement in machine learning. XGBoost incorporates parallel tree learning with efficient proposal computing and storage, making it highly beneficial for energy systems research. XGBoost has been applied several times on electricity consumption and forecasting domains. For instance, a study developed seven models for predicting energy usage in three metropolitan cities of the USA. The results explored the weather and temperature factors that are mainly contributing to the electricity consumption rate [26]. Another hybrid approach proposed using XGBoost and CatBoost applied on the daily consumption dataset collected from Turkey residents was reported in [27]. Their findings suggested a superior performance achieved by XGBoost, which can be helpful for planning and building a sustainable energy consumption plan.

To anticipate the hourly load of the regional transmission organization, a mutual information feature selection is used on lagged load, lagged temperature, and calendar data in a novel wavelet-based ensemble technique for short-term load forecasting (STLF) of North American load using a hybrid neural network. It has been demonstrated that feature selection may increase prediction accuracy by 22.4% [28]. Feature selection is used to choose informative meteorological features using the entropy measure to forecast the Korean power system's holiday demand [29]. The holiday demand is then predicted using polynomial regression. For the North Carolina electricity system, probabilistic load forecasting utilizing feature selection is carried out, accounting for past load and temperature data [30].

Another study was conducted on electricity load forecasting applied on Belgium-based Elia Grid data [31]. The suggested integrated model outperformed the single LSTM, single ARIMA, and single Prophet models over all three time horizons, according to a comparison of the MAPE, MAE, and RMSE values produced by various forecasting techniques. Table 1 compares the effectiveness of the single ARIMA model, single Prophet model, single LSTM model, and suggested hybrid model applied on a similar research domain.

Table 1. Comparison of RMSE, MAPE, and MAE in related work [31].

Model	MAE (kWh)	RMSE (kWh)	MAPE (%)
ARIMA	0.39698	0.59252	5.19
LSTM	0.19825	0.33543	3.72
Prophet	0.39157	0.47104	5.21
Hybrid ARIMA SVM	0.18494	0.20458	2.47

Although each model has its limits, ensemble models can overcome these restrictions by leveraging the unique abilities of several models. Prior research has discussed the benefits of incorporating optimization models, which requires further development by applying them to different datasets and combinations of models. Research has shown that LSTM might serve as a valuable tool for load forecasting, especially when combined with other methods such as XGBoost. Although LSTM models are effective, there is still room for improvement. Current research often focuses on individual models without considering the benefits of optimization approaches and ensemble methods. Due to their ability to tackle the problem of vanishing gradients and capture long-term dependencies, LSTM models are suitable for this purpose. Ensemble approaches may improve forecast accuracy by leveraging the capabilities of multiple models. This research specifically attempts to address an information gap by investigating the potential of ensemble models for predicting short-term energy consumption. The purpose of this technique is to enhance the accuracy and efficiency of short-term demand forecasting, resulting in valuable data for energy management and grid optimization.

3. Research Methodology

This paper aims at predicting individual household energy usage. This involves preprocessing the data and model training as well as determining the relative relevance of each attribute. Consequently, the paper's methodology includes the following:

- Preprocessing of the dataset including feature importance with the correlation-based feature selection (CFS) method [32].
- Training the models utilizing datasets comprising all the features processed during the feature engineering step and CFS-determined essential features during feature selection, with the MAE indicator used to evaluate loss in validation and evaluation.

- Finally, the behaviour of the trained models is evaluated using performance evaluation metrics such as MAE, MAPE, RMSE, and visual analysis.

A block diagram of the proposed ensemble energy consumption prediction model is shown in Figure 1. The recommended strategy is divided into several phases: data preprocessing, feature selection, and post-training model analysis. The subsequent subsections will provide detailed explanation of the decision-making process involved in data preparation, training the rolling window model, and other components of the technique.

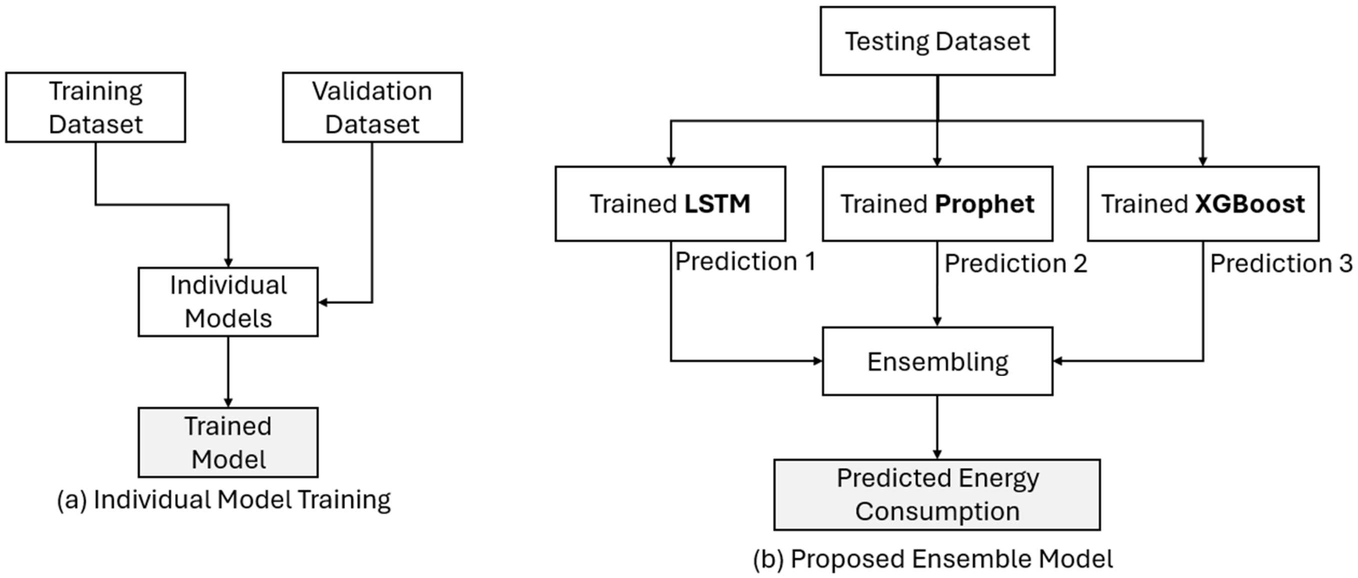

Figure 1. Proposed ensemble energy consumption prediction model: (**a**) individual model training (LSTM, Prophet, XGBoost); (**b**) ensemble model testing.

3.1. Dataset Description

The data used for this research were gathered from the London datastore that participated in the UK Power Network-led Low Carbon London. Energy measurements in kilowatt-hour (kWh) intervals were on a half-hourly basis of 5567 houses, along with relevant dates and times. Furthermore, this includes details on the CACI Acorn group and distinct identities assigned to each family. In addition, weather data were collected from the Dark Sky API (latest version), along with residence information comprising ACORN group, tariff details, and UK holiday data. Raw data typically comprise numerical, categorical, and time series information in its unprocessed form. The decompressed CSV file, which contains around 167 million entries, is roughly 10 terabytes in size [33].

The training dataset comprises actual weather and energy data captured on a half-hourly basis. The Prophet, XGBoost, and LSTM models are implemented using real-time electrical load data obtained from London Datastore—UK Power Networks' Smart Meter Energy Consumption Data in London Households from November 2011 to February 2014. The dataset consists of two separate client types. A sample of 1100 clients were assigned power tariffs that fell into three categories: high (67.20 p/kWh), low (3.99 p/kWh), or normal (11.76 p/kWh), based on certain time periods. The term "DTUs" is an abbreviation for Dynamic Time of Use, which is used to refer to such customers. The remaining 4500 units in the sample were billed at a fixed rate of 14.228p per kilowatt-hour. Table 2 presents the characteristics and descriptions of the dataset [34].

Table 2. Dataset attributes and description [34].

Input Attributes	Output Attribute	Format	Description
LCLid		String	Consumer unique ID
stdorToU		String	These are the existing tariff plans: dToU tariff = 1100 customers Non-ToU tariff = 4500 customers
DateTime		Datetime	Contains the consumer's half-hourly measurements
	kWh (per half hour)	Float	kWh (kilowatt-hour) reflects the total amount of energy used half-hourly, i.e., every 30 min, from smart meter measurements that are sent to energy supplier
Date		Date	The date of consumption
Hour		Numeric	The hour of consumption
Day		Numeric	The day of consumption
Month		Numeric	The month of consumption
Year		Numeric	The year of consumption
Period		Numeric	The period of the day, i.e., morning, noon, evening, night
Electricity Price		Float	p/kWh
Temperature		Numeric	Celsius
Cloud Cover		Numeric	Oktas
UV index		Float	Standard measure of UV radiation
Visibility		Float	mi
Wind Speed		Numeric	mph
Dew Point		Numeric	Celsius
Humidity		Numeric	Percentage
Pressure		Numeric	Hg
Holiday		Date	UK Bank Holidays
Acorn		String	ACORN (A Classification of Residential Neighbourhoods) was developed by CACI as a consumption classification of the UK population into demographic types comprising of 6 categories, 18 groups, and 62 types
Acorn_Grouped		String	ACORN categories are divided further into 18 groups, for instance, Affluent, Comfortable, Adversity

3.2. Data Preprocessing

Figure 2 shows a block diagram of the preprocessing steps used to prepare the dataset, which include data cleaning, computation of data statistics, data presentation, and Data Frame (DF) Correlation. This correlation analysis examines the connection between datasets and power load forecasting in a specific area by employing the Pearson correlation coefficient.

The initial stage of the investigation involves gathering descriptive dataset. The dataset should comprise weather data or other relevant sets to accurately estimate the association between energy use and connected variables. The dataset contains several characteristics with missing or incomplete information. The raw data undergo cleaning to address missing values and outliers during this phase. The data are then subjected to transformations, such as normalization and encoding, to prepare them for model training. The final stage involves partitioning the data into two distinct sets: one for training the models and the other for evaluating their performance.

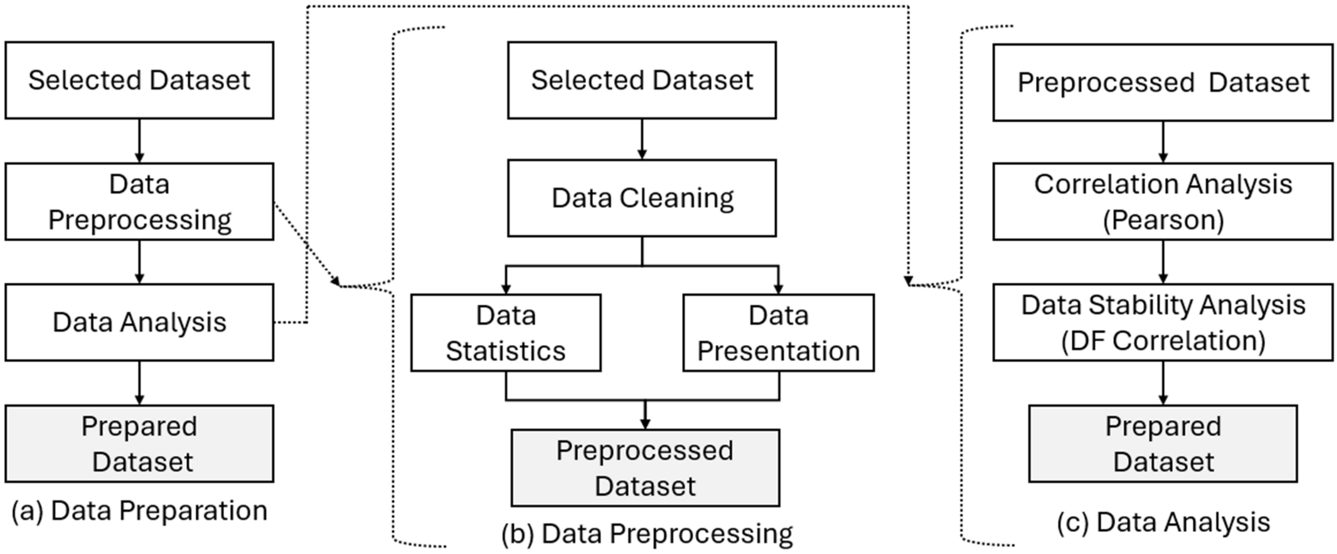

Figure 2. Data preparation steps: (**a**) data preparation, (**b**) data preprocessing, and (**c**) data analysis.

Consequently, feature engineering techniques are used to prepare the dataset. The data are arranged according to the time of sampling. Any readings that do not have any value are dealt with, and new characteristics are obtained by making conclusions using a multivariate method. If the selected dataset does not include distinct meteorological information, supplementary external datasets are included in the descriptive consumption set, given that they exhibit correlation in terms of location and time of gathering. The preparation stage yields several datasets that characterize the consumption patterns of an individual, property, or geographical area.

CFS can determine the significance of each connected property to the object of inference. Therefore, CFS is used to generate a new dataset that concentrates on the most significant characteristics of the original dataset. This is accomplished by reducing the number of dimensions in the dataset while maintaining its representativeness. The original preprocessed dataset and the CFS-significant characteristic dataset are obtained. These datasets comprise 80%, 10%, and 10% of all samples that are accessible for training, validation, and assessment, respectively. To prepare, divide the data into rolling window-formatted subsets for training, validation, and evaluation. The CFS dataset, split into the same subgroups, provides a superior model training analysis by incorporating all previously processed characteristics and those the CFS technique considers most relevant. The analysis, including model training and evaluation, is detailed in the model implementation stage (Section 3.4).

Forecasting future energy usage involves carefully combining data regarding energy and weather. The process includes data cleansing, merging datasets based on dates, and incorporating pertinent information such as holidays and household data. The final dataset comprises energy consumption, temperature, wind speed, humidity, cloud cover, and other weather factors, making it a comprehensive resource for constructing a predictive model.

3.3. Feature Selection

Feature extraction and feature selection are two commonly employed and highly effective techniques for reducing dimensionality. A common practice in machine learning is feature extraction. One major limitation of feature extraction is that the features obtained are different from the original ones. As a result, feature extraction does not provide tangible insights into informative aspects. Beyond dimension reduction, feature selection identifies the dominant features that offer valuable knowledge for household load forecasting. Feature selection is a critical step in an ML approach, as it enables the identification of the most influential factors affecting a model's performance. Additionally, it is a technique that reduces overfitting, improves model accuracy, and reduces the number of variables [35]. In this paper, the CFS technique, which is used for extracting features, is used.

Correlation analysis is used to determine the most relevant features that demonstrate a strong correlation with the target variable. This procedure helps reduce the model's complexity, improve accuracy, and mitigate multicollinearity issues. The CFS technique initially determines the correlation between the target variable and each individual feature. It then calculates the correlation between each pair of features. Subsequently, the features are arranged in order of their lowest correlation with each other and highest correlation with the target variable. CFS addresses classification and regression issues. Pearson's correlation coefficient is used to determine the relationship between each feature and the target variable in regression problems in this paper. The symmetric uncertainty measure determines classification correlation based on the target variable and features' mutual information.

The heatmap analysis of the data indicates that temperature has the highest impact on average energy consumption, which is consistent with the literature [24]. Typically, energy consumption decreases as temperature increases due to reduced heating requirements. To use the model in other places, it must be retrained using local temperature and energy data to account for regional variations in consumption patterns. For instance, in warmer areas, the consumption of energy for air conditioning may increase because of higher temperatures, whereas in cold regions, the consumption of energy for heating may increase. Incorporating local meteorological factors, such as temperature and humidity, enables the model to more accurately represent the distinct energy consumption patterns of each region.

Moreover, the results show a weak positive correlation between cloud cover and energy consumption, indicating that a marginal rise in cloud cover could lead to a marginal increase in energy consumption. The remaining weather variables, such as dew point, wind speed, pressure, visibility, humidity, UV index, and others, exhibit either minimal or no correlation with average energy consumption. This highlights that in comparison to temperature and cloud cover, these factors have a lesser impact on overall energy consumption. Further investigation, such as conducting time series analysis and integrating other variables, would enhance the understanding of the correlation between energy consumption and weather factors. Figure 3 below illustrates the relationship between average load and other characteristics in the heatmap.

This study uses the Pearson coefficient to examine the correlation between the dataset and the power load forecasting that requires prediction. The Pearson correlation coefficient (PCC) is a statistical measure that assesses the degree of the nonlinear correlation and correlation between two variables [36]. Therefore, the Pearson coefficient may be used to calculate the connection between the power load of certain areas and other characteristics. The coefficient equation $\rho_{X,Y}$ is derived from the power load X and other factors Y of the spot using Equation (1):

$$\rho_{X,Y} = \frac{E(XY) - E(X)E(Y)}{\sqrt{E(X^2) - E^2(X)}\,\sqrt{E(Y^2) - E^2(Y)}} \tag{1}$$

where $\rho_{X,Y} \in (-1, 1)$ represents the correlation strength of X and Y. A positive value denotes a positive correlation between this feature and the power load, with a stronger correlation as the value approaches 1. A negative value indicates a negative correlation between the feature and the power load. Demonstrating the fact that the selected eigenvalue remains unaffected by the power load, its magnitude approaches zero. The correlation coefficient, $\rho_{X,Y}$, measures the strength and direction of the linear relationship between two variables. It ranges from −1 to 1, where −1 indicates a perfect negative correlation, 1 indicates a perfect positive correlation, and 0 indicates no correlation. To detect strong positive and negative correlations, look for squares with colours closer to the extremes of the colour scale in Figure 3, with dark blue for strong negative and dark yellow for strong positive.

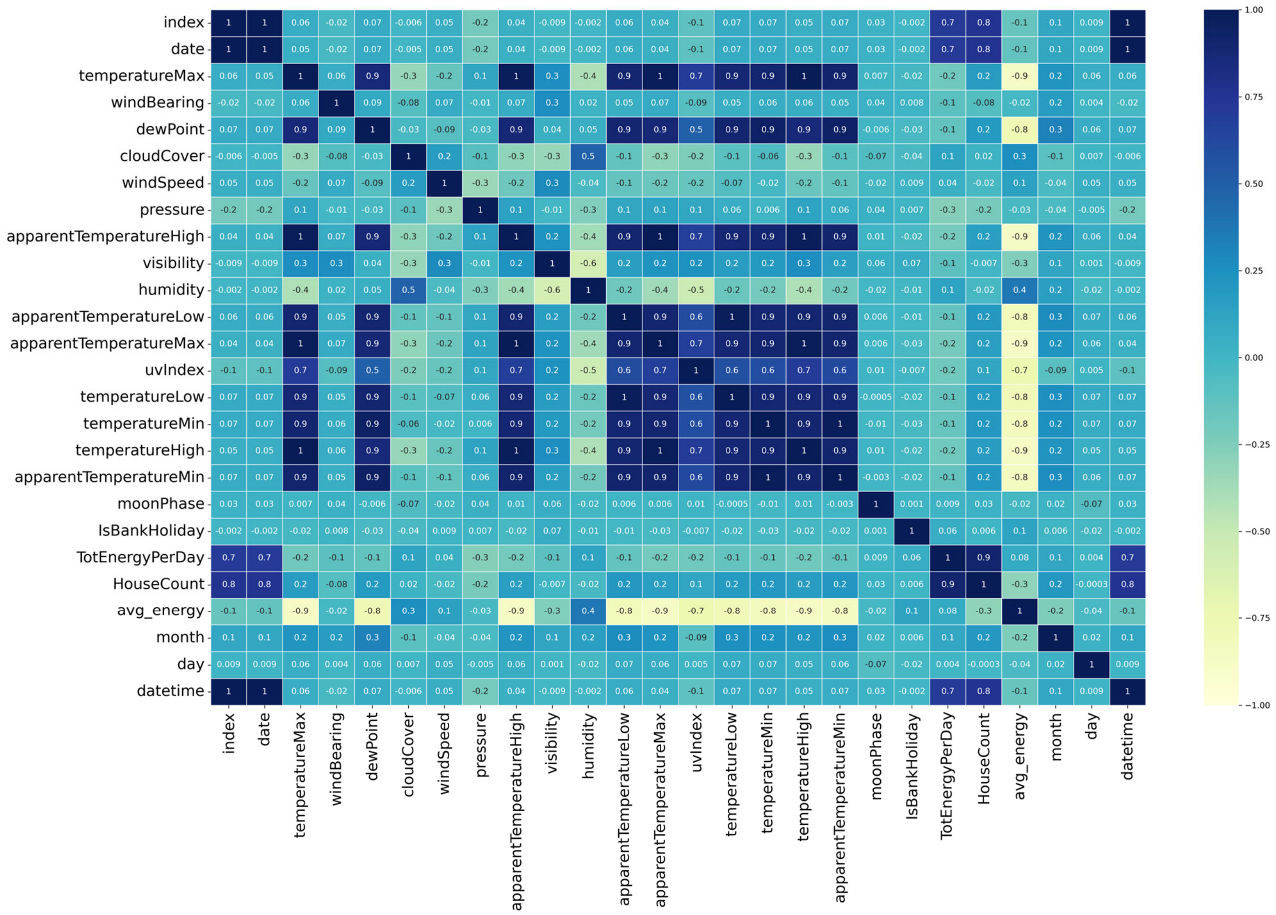

Figure 3. Heatmap feature selection.

The present study specified a minimum correlation criterion of 0.3 and included characteristics for determining average energy. It removed attributes having a Pearson correlation of 0.8 or higher to lessen multicollinearity. Features with high correlation, such as temperatureMax, temperatureLow, and temperatureHigh, were kept. To account for seasonal fluctuations in energy demand, temporal parameters such as date and month were introduced. Among the above features, those features larger than 0.3 have been given importance, as they highly influence load forecasting. The heatmap displays nine significant parameters, including datetime, temperatureMax, dewPoint, temperatureMin, temperatureLow, month, apparentTemperatureHigh, temperatureHigh and uvIndex. Amongst these nine important parameters, the uvIndex has the least significance, while the datetime has the most significance. The F-score, or feature importance score, in XGBoost indicates the significance of each feature in predicting outcomes. The calculation involves evaluating the frequency of a feature's use in a model's decision-making process and its impact on its accuracy. The F-score integrates two components: gain, reflecting the extent to which a feature enhances accuracy at each division, and coverage, representing the number of data points affected by that feature. A higher F-score indicates that the characteristic has more significance for the model's predictions, aiding in the identification of the most valuable features. The model's feature importance was assessed using the F-score from XGBoost, which measures how frequently each feature was used to make splits across all trees in the ensemble. Figure 4 below shows the features' contribution statistics with their F-scores.

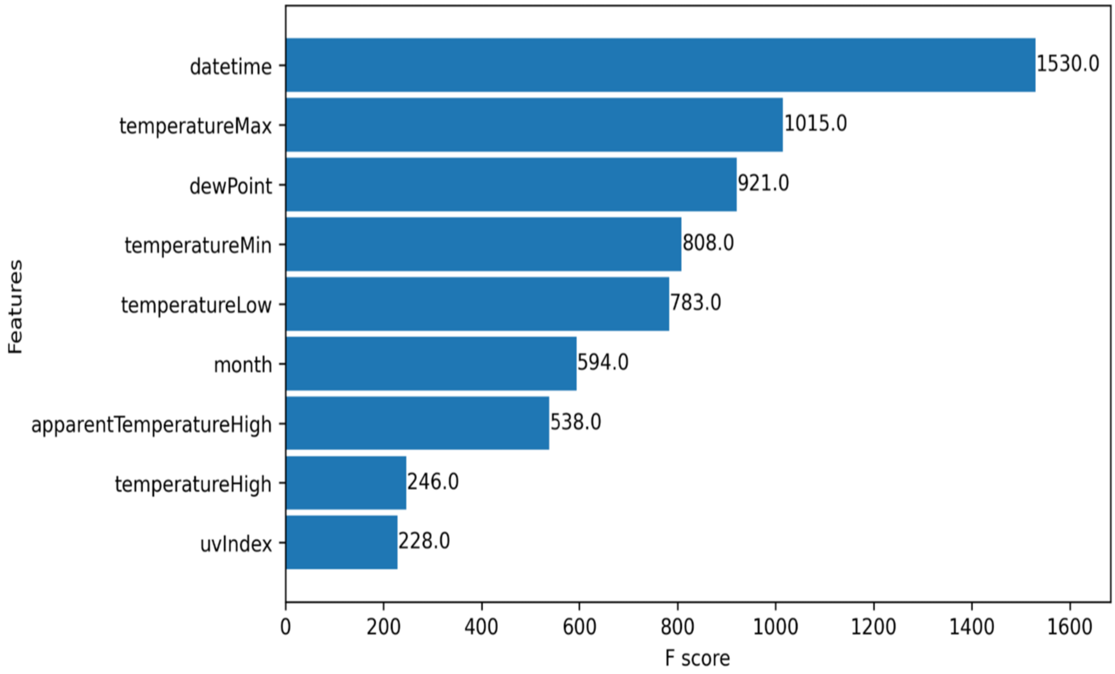

Figure 4. Feature contribution statistics.

3.4. Model Implementation Stage

Several steps are included in the model implementation stage, which are described in the subsequent sections.

3.4.1. Training, Validation, and Testing Datasets

After preprocessing and feature selection, the data are divided into training, validation, and testing sets, with the ratios of 80%, 10%, and 10% of all available samples, respectively, where the training dataset is used to train machine learning algorithms, allowing them to detect underlying patterns in the data. The test dataset is used to evaluate the effectiveness of the trained models. It serves as an unbiased evaluation of the models' ability to apply to unseen data.

3.4.2. Model Training (Prophet, LSTM, XGBoost)

This section examines the three widely recognized ML approaches: XGBoost, Prophet, and LSTM, and their respective training procedures. These models were selected based on their proven capability in predicting time series data, as demonstrated in the literature. For instance, time series analysis was performed for rainfall prediction in India using XGBoost [37]; the Prophet model has been applied to forecast stock market scenarios [38]. In addition, LSTM also outperformed other models and was applied on time series data for predicting consolidation settlement [39] and for human body energy expenditure prediction [40]. The details of these applications are discussed in the following subsection.

I. LSTM

LSTMs are the type of RNN with a more complex structure that can better retain long-term dependence on the data. They belong to a type of deep learning, sequential neural network that can learn order dependence in sequence prediction problems [41]. Unlike RNN, LSTM cells use distinct cell states at different time steps to recover historical data through forgetting gates, input gates, and output gates, therefore avoiding long-term dependencies and difficulties such as gradient vanishing or explosion. Whereas the output layer generates the ultimate result, the input layer of the neural network stores the initial set of data [42]. The layer in between is the hidden one that trains the LSTM model on the training set using a suitable optimizer—Adam optimizer, and a loss function, such as Mean Absolute Error. After that, an experiment with different LSTM architectures and hyperparameters was performed to determine the optimal configuration. For the data preprocessing, MinMaxScaler was used for training and testing datasets. The experiment was conducted using 80% of training data, then tested on 10% of unexplored data and validated with 10%.

LSTM networks are a specialized type of Recurrent Neural Network (RNN) known for their exceptional ability to retain and recall long sequences of data, such as electrical load data. To understand LSTM, it is helpful to first define RNNs briefly. RNNs are sequence-based models designed to handle dependencies within sequences. The mathematical dynamics of RNNs are as follows:

$$h(t) = \sigma(Ux(t) + V_h h(t-1)) \tag{2}$$

$$y(t) = W \cdot h(t) \tag{3}$$

In these equations, $x(t)$ represents the sequential input to the model; U is the weight associated with the input $x(t)$; $h(t-1)$ is the model's internal short-term memory; V_h is the weight associated with the short-term memory $h(t-1)$; σ is the sigmoid activation function; and W is the weight for the output.

A significant limitation of traditional RNNs is the vanishing gradient problem, which hinders their ability to update weights effectively during training over long sequences. LSTMs were developed to overcome this issue. LSTMs feature an internal cell structure composed of three gates: the input gate, forget gate, and output gate. The mathematical dynamics of these gates are as follows:

$$i(t) = \sigma(U_i x(t) + h(t-1) + b_i) \tag{4}$$

$$f(t) = \sigma\left(U_f x(t) + V_f h(t-1) + b_f\right) \tag{5}$$

$$o(t) = \sigma(U_o x(t) + V_o h(t-1) + b_o) \tag{6}$$

$$c(t) = f(t) \odot c(t-1) + i(t) \odot \tanh(U_{ia} x(t) + V_{ia} h(t-1) + b_{ia}) \tag{7}$$

$$h(t) = o(t) \odot \tanh(c(t)) \tag{8}$$

In these equations, $U,\ V$ represents the weight matrices, b denotes the bias values for the corresponding gates, and the σ symbol indicates the Hadamard product. The σ symbol represents the sigmoid activation function, which controls the opening and closing of the gates and enables the LSTM's nonlinear capabilities. The hyperbolic tangent function, tanh, regulates the outputs between -1 and 1 for the input activation and output gates. In the initial step of LSTM operation, the forget gate determines whether to retain or discard the previous cell state's contents, guided by the sigmoid activation function. Following this, the internal cell state is updated by combining the product of the forget gate's output and the previous cell state with the product of the input gate's output and the input activation. In the final step, the output gate decides whether to maintain or pass the output to the next cell, based on its sigmoid activation function. Consequently, the output gate's result is multiplied by the regulated output of the updated cell state, using the hyperbolic tangent activation function. The resulting cell state, $c(t)$, and output, $h(t)$, are then forwarded to the next stage, where the LSTM operation cycle repeats. For a detailed explanation of the LSTM model, refer to [31].

II. Prophet

Prophet is an open-source library for univariate time series forecasting developed by Facebook [43]. It implements an additive time series forecasting model that supports trends, seasonality, and holidays [44]. Prophet is a scalable forecasting tool based on a generalized additive model. Furthermore, it effectively manages outliers and missing data, thereby nearly automating the matching process and surpassing alternative methods. Previously, it has been effectively applied for network load prediction [16] and bitcoin prices [45]. Application of Prophet model for residential load forecasting is investigated in this paper. Multiple adjustments to hyperparameters such as the changepoint prior scale and the seasonality model in finding out optimal solution were applied.

The Prophet model is an enhanced decomposition-based time series model that includes three main components: trend, seasonality, and holiday effects. These elements are integrated into the following equation:

$$y^P(t) = g(t) + s(t) + h(t) + \varepsilon_t \tag{9}$$

Here, the trend function, $g(t)$, captures non-periodic variations in the time series, while the seasonality component, $s(t)$, accounts for regular, periodic fluctuations. The holiday term, $h(t)$, models the impact of holidays or events that occur on irregular schedules over one or more days. The error term, ε_t, represents any unique changes not captured by the model and is assumed to follow a normal distribution with a mean of zero. Detailed explanation of the Prophet model can be found in [46].

III. XGBoost

XGBoost is an ML approach that has substantially helped predictive modelling, data science, and machine learning alike. It is a sophisticated ML algorithm based on trees and is noted for its efficiency, speed, and accuracy [47]. It is part of the boosting algorithm family, which is an ensemble learning technique that combines the predictions of numerous weak learners. XGBoost begins as a weak learner but gradually increases its performance by learning from each training iteration and modifying the residuals from previous rounds [48]. Recently, it has been applied for predicting gold prices [49] and the compressive strength of concrete [50]. XGBoost has shown superior performance over other models. In this paper, the experiment was conducted using the XGBoost model on the dataset using relevant hyperparameters such as learning rate, number of boosting rounds, and maximum tree depth. The XGBoost model was configured using empirical tuning, trial-and-error approach, and n_estimators = 1000, learning_rate = 0.02, and max_depth = 3, without extensive optimization techniques like grid search or Bayesian optimization.

XGBoost employs a classification and regression tree (CART) as its base learner, refining its predictions in each training iteration to better fit the residuals from previous iterations, thereby creating a robust model. For a dataset with n samples, the final predicted value is given by the following:

$$\hat{y}_i^M = \sum_{m=1}^{M} f_m(x_i) \tag{10}$$

Here, $\hat{y}_i^M$, represents the predicted value for the i-th sample, M is the total number of CARTs, and $f_m(x_i)$ denotes the prediction for the i-th sample in the $(m-1)$-th tree. XGBoost incorporates a regularized objective function to the loss function to control the complexity of the CARTs and mitigate overfitting. The regularized objective for optimizing M iterations is expressed as follows:

$$Obj = \sum_{i=1}^{n} l\left(y_i,\ \ \hat{y}_i^M\right) + \sum_{m=1}^{M} \Omega(f_m) \tag{11}$$

In this equation, $l(\cdot)$ is a second-order differentiable loss function that quantifies the difference between the actual value y and the predicted value $\hat{y}_i^M$, while $\Omega(\cdot)$ represents the regularization term. The prediction $\hat{y}_i^m$ at the m-th iteration can be formulated as follows:

$$\hat{y}_i^m = \hat{y}_i^{m-1} + f_m(x_i) \tag{12}$$

A comprehensive explanation of the XGBoost model can be found in [46].

IV. Ensemble Learning

Ensemble learning improves prediction accuracy and reduces model errors by combining the benefits of individual models. The combination of multiple models helps to enhance the overall performance and overcome several issues such as overfitting, bias, and data imbalance [51]. The collective strategy of integrating multiple models has the advantages of combining the efficacy of multiple models in one framework. In this paper, Prophet, LSTM, and XGBoost models were integrated to develop an ensemble forecasting approach, which showed a significant improvement, as discussed in Section 4.

The proposed combination of LSTM, Prophet, and XGBoost introduces several innovative features to emphasize the novelty of our ensemble approach. The ensemble comprises three models, each specialized in tackling distinct challenges in energy forecasting. LSTM, Prophet, and XGBoost are used to analyse data, with LSTM capturing temporal dependencies and patterns, Prophet modelling seasonality and trends, and XGBoost handling structured data. The ensemble's unique feature is its ability to adapt to diverse datasets and conditions, balancing different modelling approaches, making it more robust to anomalies and variations. The paper introduces a novel combination of ensemble methods: LSTM, Prophet, and XGBoost for short-term residential energy forecasting, demonstrating its effectiveness on a real-world dataset and filling existing literature gaps. The proposed method enhances accuracy, facilitating more reliable load forecasting, a crucial aspect for energy management and grid optimization in smart cities.

The proposed Weighted Average Ensemble (WAE) model for energy consumption prediction integrates the forecasts from the LSTM, Prophet, and XGBoost models using the following equation:

$$\hat{y}_{WAE}(t) = k_p \hat{y}^p(\mathrm{t}) + k_{LSTM} \hat{y}^{LSTM}(t) + k_{XGBoost} \hat{y}^{XGBoost}(t) \tag{13}$$

In this equation, $\hat{y}^p(\mathrm{t})$, $\hat{y}^{LSTM}(t)$, and $\hat{y}^{XGBoost}(t)$ represent the energy consumption predictions from the Prophet, LSTM, and XGBoost models, respectively. The constants k_p, k_{LSTM}, and $k_{XGBoost}$ denote the contribution weights of the Prophet, LSTM, and XGBoost models, respectively, in the ensemble prediction at time t. The experimental results presented in this paper indicate that $k_p = k_{LSTM} = k_{XGBoost} = \frac{1}{3}$.

3.4.3. Data Split and Model Initialization

To optimize the performance of the LSTM, Prophet, and XGBoost models, appropriate hyperparameters were initialized based on the specific use case and the characteristics of the data. Typically, it is recommended to use a ratio of 70/30 or 80/20 for splitting the data into training, testing, and validation sets.

3.4.4. System Configuration

The proposed approach for residential load forecasting comprises an individual's prediction of LSTM, Prophet, and XGBoost models, as well as an ensemble model. The codes for all models are executed using Python 3.11.5. With an Intel(R) Core (TM) i5-5200U CPU @ 2.20 GHz and 8 GB of RAM, the operating system is a 64-bit version of Windows 10 Pro.

3.5. Evaluation Metrics

The performance of the trained LSTM, Prophet, and XGBoost models is evaluated using three commonly used regression metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). MAE quantifies the average absolute difference between actual and predicted values. RMSE is determined by averaging the squared differences between actual and predicted values. MAPE, expressed as a percentage, represents the average percentage difference between actual and predicted values. Equations (2)–(4) are used to calculate MAE, MAPE, and RMSE.

$$MAE = \frac{1}{N}\sum_{L=1}^{N}|F_L - O_L| \tag{14}$$

$$MAPE = \frac{\sum_{L=1}^{N}\left|\frac{F_L - O_L}{O_L}\right|}{N} \times 100\% \tag{15}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{L=1}^{N}(F_L - O_L)^2} \tag{16}$$

Consider that N represents the number of data values in the projected load and F_L represents the forecasted load, whereas O_L represents the original load at any given instance.

This step relies on the optimal epochs of the training models, which allow for the consideration of the most effective weights for each layer of the ML and DL models. The average contribution of each attribute is visually confirmed using the same test set. The performance of the LSTM, Prophet, and XGBoost models is assessed in terms of accuracy in forecasting load using the three assessment measures. MAE is used to identify significant discrepancies between the observed value of the original load O_L and the forecasted load F_L. RMSE amplifies the forecasting error of inaccurate numbers due to the use of squared calculations. Instances of relative deviation are displayed using MAPE. As a proportional measure, MAPE is more capable of capturing the full effect of outliers. Smaller values from these evaluation parameters indicate higher forecasting performance.

3.6. Ensemble Model Evaluation

Ensemble learning strategies are utilized for combined predictions for the minimization of errors. The implementation of the proposed framework utilized a combination of ML algorithms to understand energy usage patterns, with feature selection being employed to improve the effectiveness and precision of ML algorithms. It enhances the development of more precise and efficient energy forecasting models, hence enabling enhanced energy management and resource allocation. The proposed ensemble model increases the prediction accuracy by combining LSTM, Prophet, and XGBoost models' prediction, and its performance is assessed using the training dataset and MAE, MAPE, and RMSE assessment criteria. Basic mean of all output of individual models was utilized for the precise output of ensemble model prediction.

4. Experimental Results and Analysis

This section analyses weekly, monthly, quarterly and half-hourly trends using half-hourly smart meter datasets. It incorporates tariff plans and tariff groups derived from half-hourly datasets, specifically focussing on blocks 1/3/20/48/59/70/78/90/109. There are several blocks of data accessible; the ones that are presented are blocks used only for exploratory data analysis.

Similarly, it also examines the impact of temperature, humidity, cloud cover, and UV index on energy consumption among families and ACORN groups, utilizing all accessible blocks of daily smart meter datasets derived from half-hourly datasets, employing forecasting models (Prophet, LSTM, XGBoost) and ensemble methods for enhanced prediction accuracy.

4.1. Exploratory Data Analysis

4.1.1. Weekly Energy Consumption Analysis

Figure 5 illustrates a weekly energy consumption graph of single household from 7 March 2012 to 14 March 2012 using a half-hourly smart meter dataset with days labelled on the x-axis and kWh consumption on the y-axis. The line plot shows the fluctuating energy consumption patterns, with peaks and troughs, indicating varying usage patterns on different days. Although energy consumption was at its lowest every night, the highest energy use utilized was on the Tuesday of the week compared to the other days.

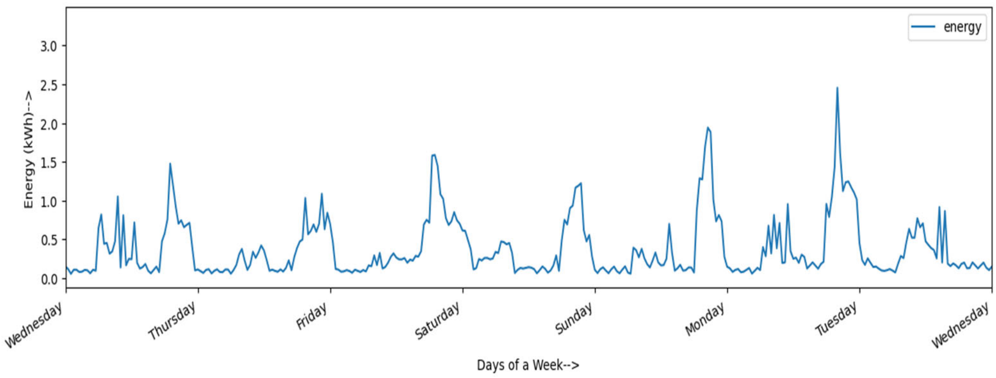

Figure 5. Energy consumption of a single house in a week.

Figure 6 depicts the mean energy use over the course of one week in the year 2013. The analysis of a half-hourly dataset from 2013 revealed that energy consumption was higher on weekends than weekdays due to increased home appliance usage and household activities. The analysis was conducted using specific blocks of the half-hourly dataset.

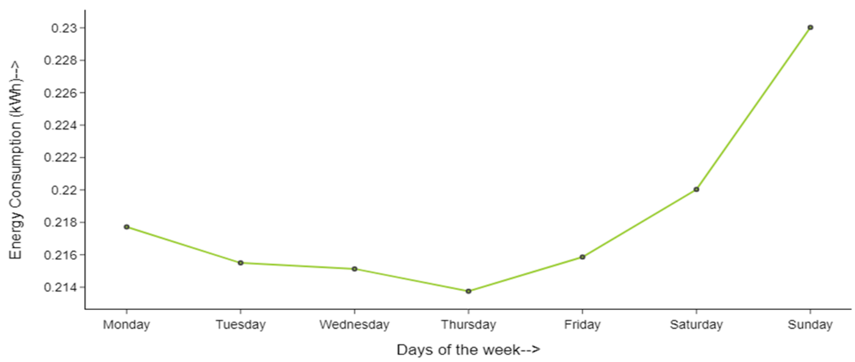

Figure 6. Average energy usage of multiple households for an entire week in 2013.

4.1.2. Monthly Energy Consumption Analysis

Figure 7 below displays the average energy use among all ACORN categories for the whole year of 2013. The analysis utilized specific blocks of a half-hourly dataset from 2013 to analyse energy consumption at specific times. Energy consumption rises from December to March for all ACORN groups due to increased heating needs in colder winter seasons, while consumption decreased in the summer and autumn seasons when heating is not needed and cooling demands are moderate.

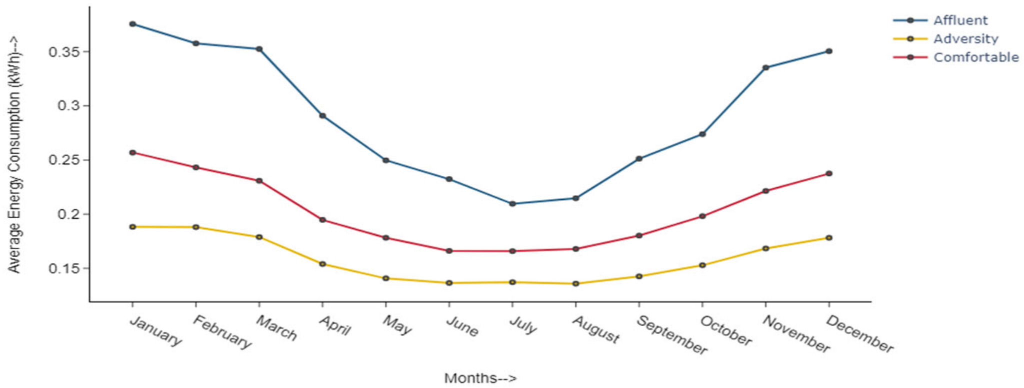

Figure 7. Average energy consumption per ACORN group for the year 2013.

4.1.3. Average Energy Consumption by Different Tariff Groups

In the United Kingdom, various tariff systems are employed to reduce overall energy consumption through the Dynamic Time of Use (DToU) plan. This plan offers three dis-tinct tariff rates: high, normal, and low. By considering their DToU plan, consumers can effectively manage their energy usage and reduce their total energy costs. Figure 8 illustrates the average energy consumption for each ACORN category across three different tariffs: Affluent, Adversity, and Comfortable groups, over a two-year period. The consumption trends are evaluated across different tariff plans: Standard and Dynamic Time of Use (DToU), spanning from 2011 to 2014 using half-hourly datasets. The data is illustrated in three line charts, each corresponding to a group, with quarterly intervals on the *x*-axis and average energy usage measured in kilowatt-hours (kWh) on the *y*-axis. The lines depict fluctuations in energy consumption over time for each group under both tariff plans. The plot in Figure 8 reveals that DToU members in the Comfortable ACORN group had higher average energy consumption compared to the Affluent group, while the Affluent group had lower consumption, and the Adversity group showed nearly the same consumption levels.

4.1.4. Half-Hourly Energy Consumption by Tariff Rates

Figure 9 depicts the energy use at half-hour intervals categorized by tariff rates using half-hourly smart meter data. The plots in Figure 9 show that the total energy consumption fluctuates throughout the day on an hourly basis. Energy consumption is lowest during the day and peaks in the evening. As expected, average energy consumption is at its lowest during late-night hours. Regardless of the tariff rates, low, normal, or high, the patterns of energy consumption are very similar. During high tariff periods, customers tend to reduce their electricity use to lower their overall costs, while they consume more energy when rates are lower.

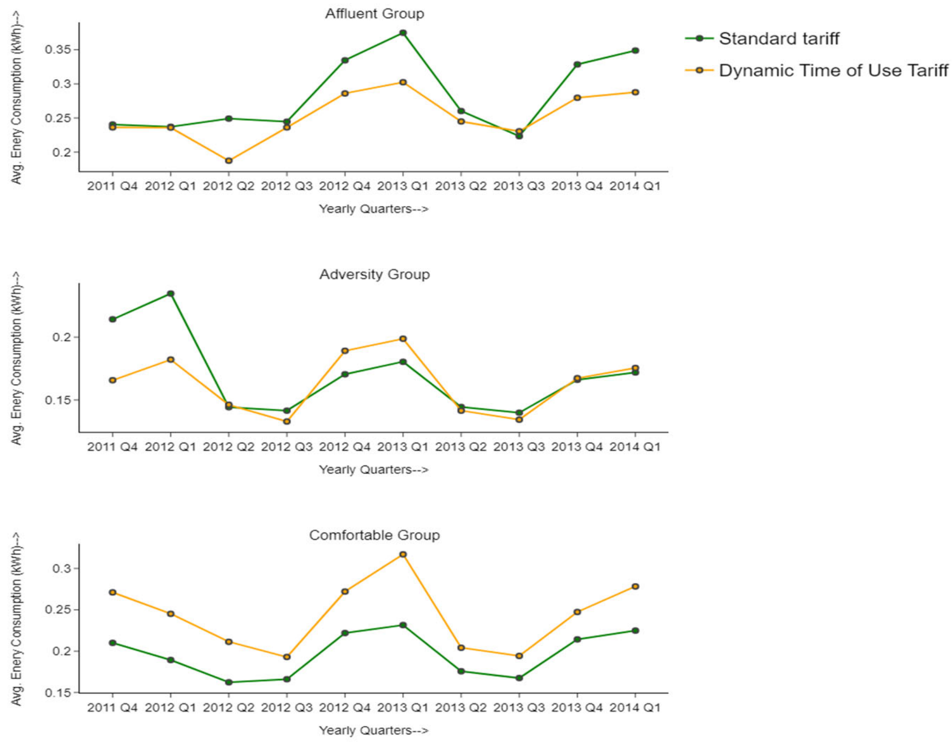

Figure 8. Average energy consumption by Standard tariff and DToU tariff further categorized in three groups: Affluent, Adversity, and Comfortable.

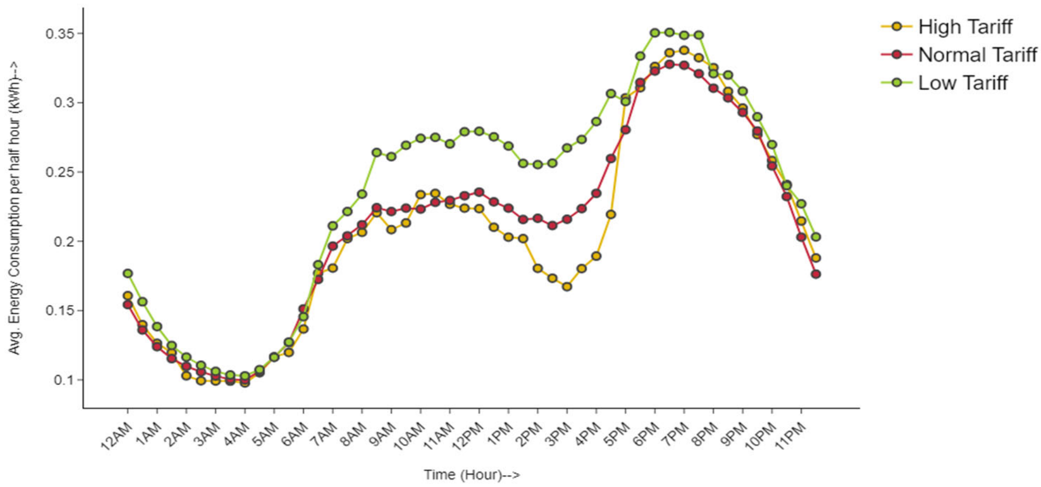

Figure 9. Half-hourly energy consumption by tariff rates (high, normal, and low).

4.1.5. Temperature vs. Mean Energy Consumption per Acorn Groups

Figure 10 illustrates the mean monthly energy consumption of the ACORN group in relation to weather temperature, a direct relationship between high energy use and low temperature. For this analysis, half-hourly smart meter energy consumption data were utilized. The plots in Figure 10 indicate that energy consumption decreases as weather conditions improve. During January, February, and March, the average temperature remains consistently below 5 °C. As a result, a significant amount of energy is consumed during these months due to the active use of heating systems.

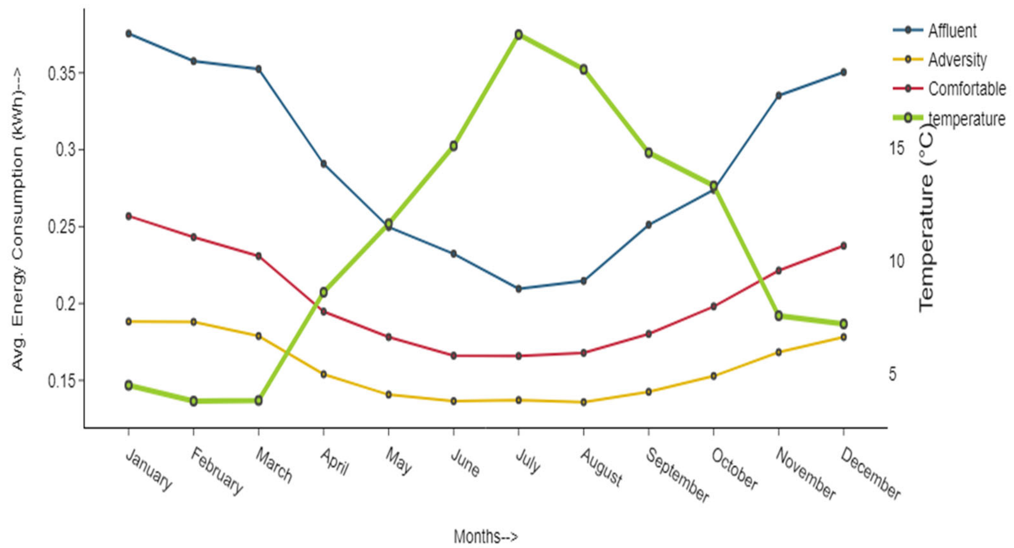

Figure 10. Temperature and mean energy consumption per ACORN group (Affluent, Adversity, Comfortable) of the year 2013.

4.1.6. Energy Consumption and Environmental Temperature

Figure 11 displays the ACORN groups' energy use in relation to the environmental temperature using a daily smart meter dataset. Environmental temperature and energy consumption are indirectly associated, as energy consumption increases with lower temperatures, and vice versa. The orange graph depicts the maximum recorded temperature, the blue graph represents the lowest recorded temperature, and the green graph indicates the average amount of energy utilized. When the outside temperature is low, people use heaters, air conditioners, and other devices to warm up their homes.

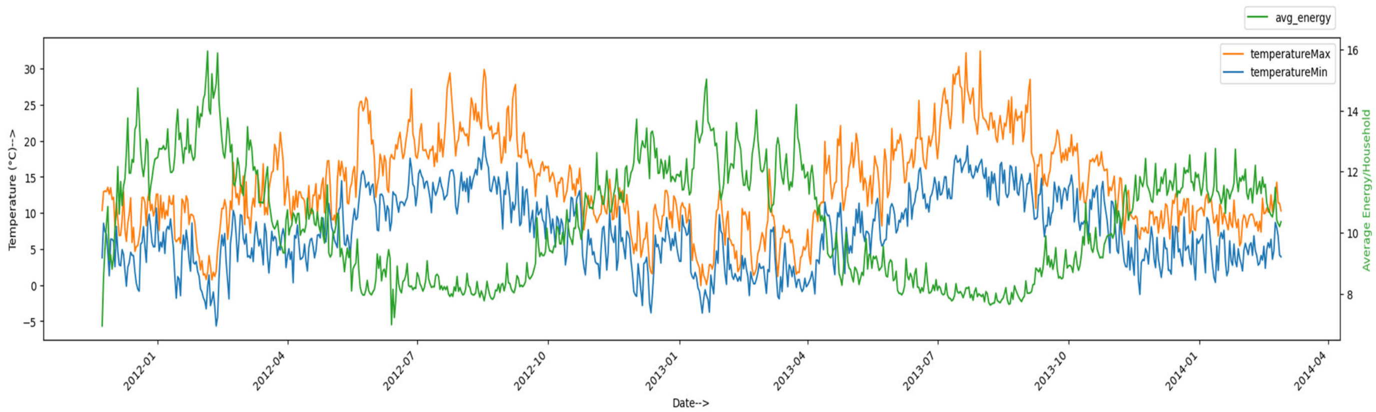

Figure 11. Average energy consumption and maximum and minimum temperature plots from January 2012 to April 2014.

4.1.7. Energy Consumption and Environmental Humidity

Figure 12 shows the link between energy use and humidity in the environment using a daily smart meter dataset. This graphic shows a clear correlation: greater humidity levels result in increased energy use, and vice versa. The average energy usage strongly correlates with the average humidity levels. The blue bar reflects the amount of humidity, while the green graph shows the average energy use.

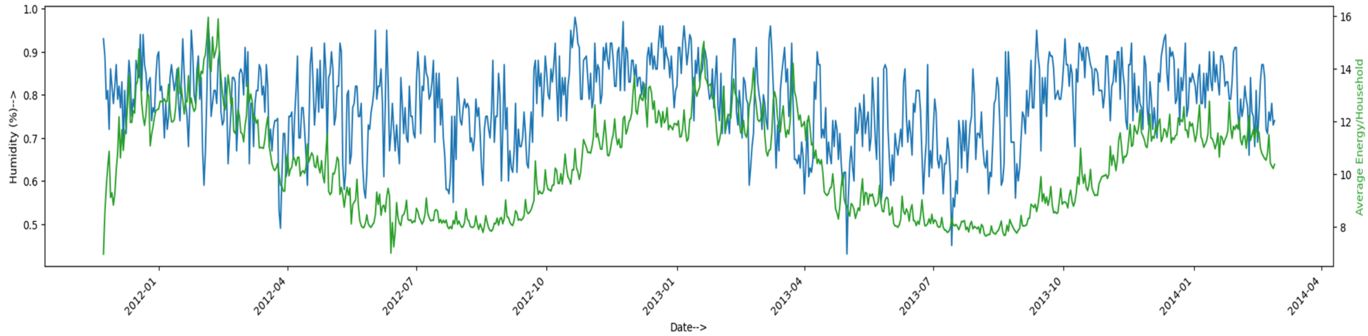

Figure 12. Energy consumption (plot in green) and humidity (plot in blue) during the 1st quarter of the year 2012.

4.1.8. Energy Consumption and Environmental Cloud Cover

Figure 13 displays the link between energy use and environmental cloud cover using a daily smart meter dataset. The blue graph depicts the degree of environmental cloud cover, whilst the green graph reflects the average energy use. This graphic shows a clear correlation: more cloud cover leads to increased energy use, and vice versa. Cloud cover is the proportion or percentage of the sky covered by clouds at a given location and time. Cloud cover is traditionally measured in oktas (eighths of the sky covered), using a scale of 0 (clear sky) to 8 (completely overcast). The investigation was based on daily energy use data from smart meters.

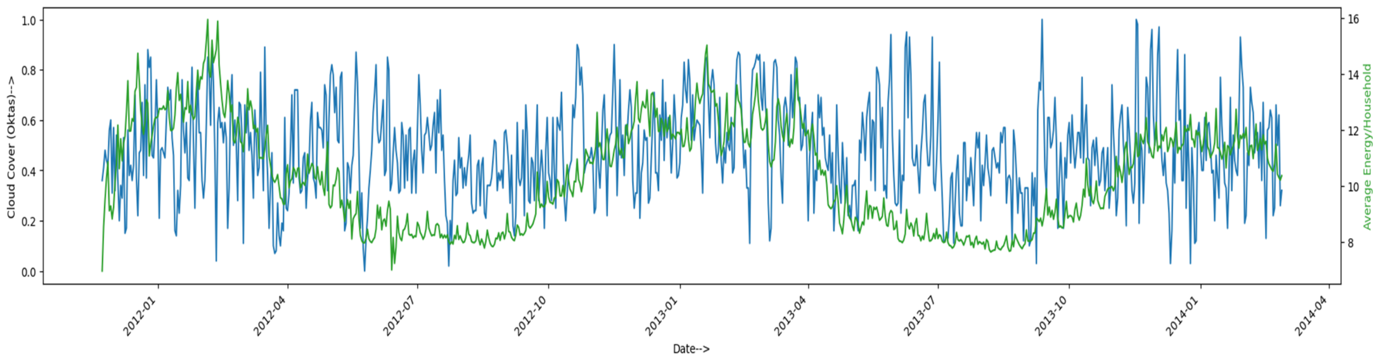

Figure 13. Energy consumption (plot in green) and cloud cover (plot in blue) during January 2012 to April 2014.

4.1.9. Energy Consumption and Environmental UV Index

Figure 14 shows the relationship between energy use and the environmental UV index using a daily smart meter dataset. There is an inverse link between energy consumption and the environmental UV index, which means that as the UV index drops, energy consumption rises, and vice versa. Therefore, environmental UV index and energy usage are indirectly associated. The UV index, which ranges from 0 to 11+ utilizing UV radiometers that detect ultraviolet radiation intensity, shows the danger of injury from unprotected sun exposure. The UV radiometer is used to monitor the UV index, and the study examines the previously measured value.

4.2. Energy Demand Prediction

Finally, energy demand prediction by utilizing the dataset selected in this study is applied by implementing three prevalent models: Prophet, LSTM, and XGBoost. Additionally, the study proposed an ensemble approach that utilizes the three aforementioned state-of-the-art time series models to enhance the accuracy of short-term energy demand prediction.

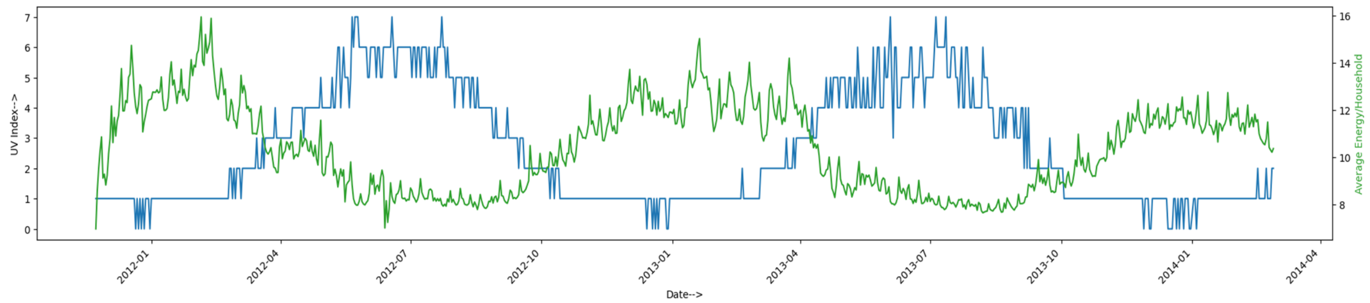

Figure 14. Average energy consumption (plot in green) and UV index (plot in blue) during January 2012 to April 2014.

4.2.1. Prophet Model Prediction

- Trend of Forecasted Data

Figure 15 illustrates the overall pattern of energy consumption during holidays, including the seasonal, weekly, annual, and monthly variations. The date is limited to a specific part due to the Prophet model forecast based on test data from February 2014. Throughout the course of a week, the highest amount of electricity consumption occurred on Saturday and Sunday. In contrast, the months of January and February exhibit the highest levels of energy consumption, while energy usage is at its lowest during June and July. During public holidays such as Boxing Day and Christmas, the volume of energy consumption was significantly reduced.

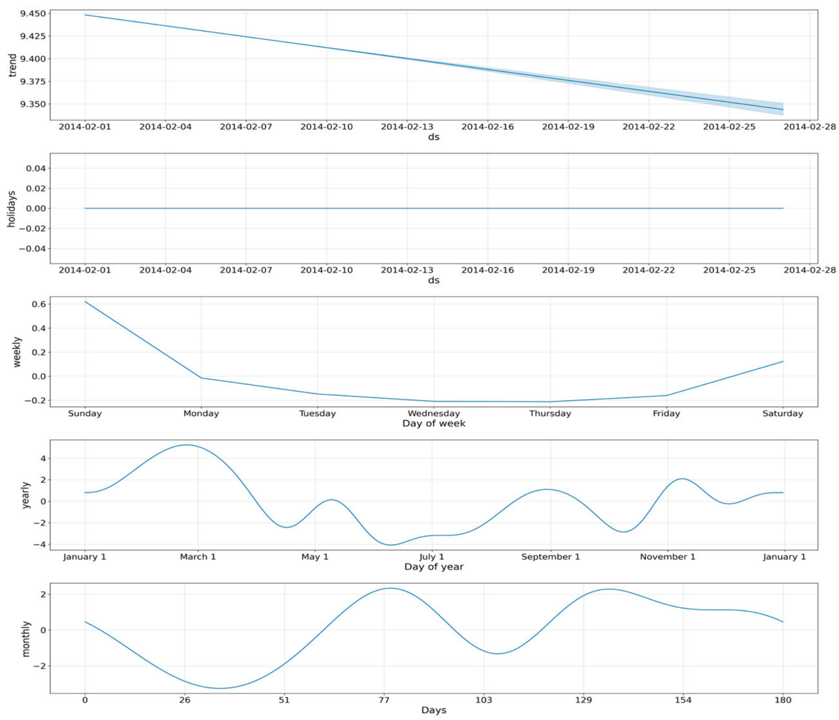

Figure 15. Prophet model components.

- Real vs. Forecasted Prophet

The Prophet model is a time series forecasting tool that predicts data with upper and lower bounds, indicating that the forecasted values are within these limits. Figure 16 illustrates the daily energy consumption prediction forecast using the Prophet forecast method alongside the actual energy consumption from 1 February to 26 February 2014. The forecast data are denoted by green lines, while the red lines represent the actual data.

The forecasted values closely resembled the actual data. The use of Prophet for predicting electrical load consumption resulted in a 62.963% accuracy score.

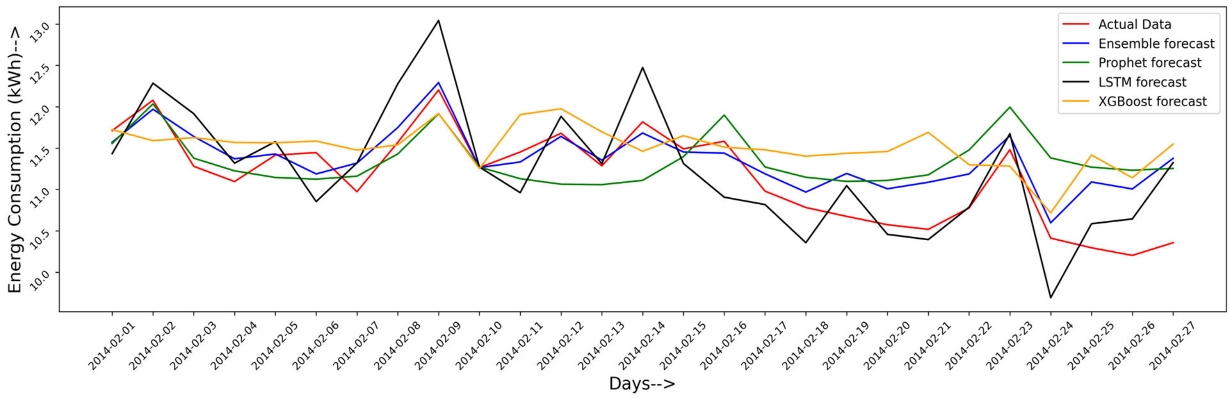

Figure 16. Comparison between individual and ensemble model predictions (Prophet, LSTM, XGBoost).

4.2.2. LSTM Model Prediction

Both training and validation losses occurred during the testing and training of the dataset. The LSTM model is designed to capture both short-term and long-term interdependence within time series data. Dropout is used to mitigate overfitting, while the model forecasts via a terminal output layer that produces a singular value. MAE is chosen as the loss function for its robustness to outliers, with the Adam optimizer ensuring efficient learning. Early stopping is utilized to terminate training when validation loss fails to improve, hence enhancing the model's generalization capabilities. The model is trained for a maximum of 1000 epochs, with early stopping implemented to prevent overfitting.

Figure 16 displays the actual and projected outcomes obtained by utilizing the LSTM model. The predicted data are represented by the black lines, while the real data are represented by the red line. It depicts nearly identical lines that are both real and projected. The model accurately predicts energy usage trends and is influenced by factors like data quality, model complexity, and external influences, despite potential for accuracy. Overall, the model provides a reasonable forecast based on the available data. The use of LSTM for predicting electrical load consumption resulted in a 70.370% accuracy score.

4.2.3. XGBoost Model Prediction

The prediction process utilized an XGBoost regressor. Figure 16 illustrates the actual and predicted trends for February 2014 using the XGBoost model. The red lines depict the actual data, while the yellow lines illustrate the predicted data. The model usually reflects energy use patterns and variances. Several variables, including model complexity, data quality, and external factors, might impact prediction accuracy. Overall, the model provides a reasonable forecast based on the available smart meter data. The actual and expected data trends were nearly the same. The accuracy score for electrical load consumption forecasts using XGBoost was 66.667%.

4.2.4. Ensemble Prediction

This study presented a novel framework for time series forecasting by combining the Prophet, XGBoost, and LSTM models to create an optimal prediction model. The input of the expected values of these three models results in a progressive decrease in errors and an improvement in accuracy. Aggregate prediction performance metrics were utilized to assess the accuracy of errors. Figure 16 illustrates the performance of the individual and proposed ensemble models. In the figure, the red line represents the actual data, while the orange, black, and green lines depict the performances of the XGBoost, LSTM, and Prophet models, respectively. This visual comparison highlights how each model performed against the actual energy consumption data. It can be seen in the diagram that the ensemble model forecast (blue line) that is generated by integrating the Prophet, LSTM, and XGBoost algorithms is very close to the actual data (red line). This highlights the importance and capability of ensemble modelling to achieve better forecasting results. The ensemble prediction used the Prophet, LSTM and XGBoost's fundamental mean. The accuracy study was conducted using both the test data and the projected mean data, resulting in a low error and an accuracy of 81.48%.

Once the final predicted outputs were obtained from each of the three individual models, they were evaluated using the output dataset derived from both the training and test datasets. Subsequently, evaluation metrics such as MAE, RMSE, and MAPE were employed for additional assessment. These findings demonstrate that the collective prediction yields a lower number of errors compared to individual predictions. The findings highlight the importance of the ensemble approach, which significantly outperforms the individual models in terms of accuracy in predicting short-term household energy demand.

The ensemble model can be applied to any dataset from any country or region. The model is scalable, generic, and robust due to its capability to be implemented on different datasets. Three separate models, namely, LSTM, Prophet, and XGBoost, are combined under the ensemble model. These models were selected based on their complementary strengths: LSTM effectively captures long-term dependencies in sequential data, Prophet handles seasonality and trend components, and XGBoost is robust for structured data and outlier detection. The ensemble integrates these models by averaging their predictions, utilizing their unique capabilities to minimize errors and improve accuracy. The ensemble approach offers improved accuracy through superior metrics like MAE, RMSE, and MAPE and ensures robust performance across varied conditions, reducing the risk of over-reliance on a single model's assumptions.

- Pseudocode for Ensemble Model (LSTM, XGBoost, Prophet)

 # 1. Initialize the three models: LSTM, XGBoost, Prophet
LSTM Model
initialize LSTM_model with architecture
train LSTM_model on training data (X_train, y_train)
predict LSTM_predictions on test data (X_test)
 # XGBoost Model
initialize XGBoost_model with hyperparameters
train XGBoost_model on training data (X_train, y_train)
predict XGBoost_predictions on test data (X_test)
 # Prophet Model
initialize Prophet_model with configuration
train Prophet_model on training data (date, y_train)
predict Prophet_predictions on test data (date, X_test)
 # 2. Ensemble Method: Take the average of the three model predictions

$$Ensemble = \frac{LSTM_{predictions} + XGBoost_{predictions} + Prophet_{predictions}}{3}$$

 # 3. Evaluate the performance of the ensemble model
calculate accuracy using actual y_test and ensemble_predictions
calculate other metrics (MAE, RMSE, MAPE) as needed
 # 4. Output ensemble predictions and performance metrics
print(ensemble_predictions)
print ("Model performance metrics: Accuracy, MAE, RMSE, MAPE")

4.2.5. Model Evaluation

The Prophet, XGBoost, and LSTM forecasting models were evaluated using performance metrics including MAE, RMSE, and MAPE, as shown in Table 3. This table depicts the performance of three machine learning models, namely, Prophet, LSTM, and XGBoost, and an ensemble model, using three widely used assessment metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). Compared to the other two models, LSTM exhibits a lower level of error. When employing the evaluation metrics MAE, RMSE, and MAPE, LSTM exhibits errors of 0.372300, 0.456462, and 3.3374%, respectively. The implementation of the ensemble method led to a substantial decrease in errors. The ensemble forecasted model achieves a minimum error of 0.282133, 0.381712, and 2.6229%, respectively, as measured by MAE, RMSE, and MAPE.

Table 3. Energy consumption prediction error in terms of evaluation metrics using Prophet, LSTM, XGBoost, and ensemble models.

Model	MAE (kWh)	RMSE (kWh)	MAPE (%)
Prophet	0.417952	0.514210	3.8420
LSTM	0.372300	0.456462	3.3374
XGBoost	0.459328	0.574614	4.2408
Ensemble	0.282133	0.381712	2.6229

- LSTM: LSTM seems to yield the most accurate point predictions, as it achieves the lowest MAE, RMSE, and MAPE scores compared with the remaining models.
- XGBoost: The MAE, RMSE, and MAPE findings of XGBoost were very competitive, suggesting excellent performance.
- Prophet: Prophet seems to be less reliable for point forecasts due to its highest values of MAE, RMSE, and MAPE.
- Ensemble: Overall, the ensemble approach, which consists of Prophet, LSTM, and XGBoost, outperformed each of the individual models in all aspects. This emphasizes the benefits of combining multiple models to enhance forecasting performance.

Based on these criteria, the ensemble model is considered the most accurate due to its reduced MAE, RMSE, and MAPE values. This supports the idea that the combination of models may improve predictive accuracy by using the unique capabilities of each model.

4.2.6. Accuracy Score

Accuracy score is an evaluation metric in machine learning that measures the number of correct predictions made by a model in relation to the total number of predictions made. In the context of this study, a "correct estimate" refers to the number of instances where the model's predicted values fall within a specified tolerance range of the actual values. For this analysis, predictions are considered correct if they fall within ± 0.5 of the actual values. This approach accommodates minor deviations, recognizing that small fluctuations around the true values may not significantly impact the overall performance of the model. In this analysis, the *accuracy score* was calculated using a custom function that determines correctness based on the defined tolerance range, resulting in an accuracy of 81.48%. The *accuracy score* is calculated by dividing the *number of correct estimates* by the *total number of forecasts*. Mathematically, the *accuracy score* can be expressed as follows:

$$Accuracy\ Score = \frac{Number\ of\ Correct\ Estimates}{Total\ Number\ of\ Forecasts} \tag{17}$$

This metric gives a simple but relevant evaluation of the model's functionality. For balanced datasets with evenly distributed classes, accuracy is quite useful. The accuracy of three machine learning models—Prophet, LSTM, and XGBoost—and an ensemble model is detailed in Table 4 presented.

Table 4. Model accuracy scores in percentage.

Model	Accuracy (%)
Prophet	62.96
LSTM	70.37
XGBoost	66.66
Ensemble	81.48

- LSTM model: among the individual models, LSTM had the highest accuracy, indicating its proficiency in detecting complex patterns within the data.
- XGBoost model: XGBoost demonstrated superior performance, exceeding Prophet.
- Prophet model: despite Prophet's poor accuracy, it nevertheless offers valuable insights and may be used with other models to improve overall performance.
- Ensemble model: The ensemble model, which included Prophet, LSTM, and XGBoost, obtained an impressive overall accuracy of 81.48%, outperforming the individual models. This accuracy was determined using a custom accuracy metric derived from scikit-learn's accuracy_score. This approach considers forecasts within ± 0.5 of actual values as accurate, rather than a perfect match, to accommodate for minor differences. This verifies the idea that the ensemble technique improves prediction accuracy by exploiting each model's unique skills.
- The learning process of the proposed ensemble model is highly influenced by feature selection strategies. Specifically, when dealing with a large number of features, ensemble models combine multiple individual models to make predictions. This helps to improve overall performance by reducing overfitting and increasing accuracy. However, the features selected during training phase greatly impacted the overall performance.
- The findings demonstrate that the ensemble model is superior in terms of accuracy for the problem undertaken in this study. This illustrates the potential benefits of merging multiple models to enhance the precision of forecasts.

Hence, this section analyses the influence of environmental conditions, periodic patterns, and tariff systems on the amount of energy used by households. The investigation also assesses the efficacy of an ensemble approach and three distinct forecasting models (Prophet, LSTM, and XGBoost) in correctly predicting future energy consumption. The analysis revealed substantial correlations between energy use and variables such as temperature, humidity, cloud cover, and UV index. Furthermore, energy use displayed regular patterns on a weekly, monthly, and half-hourly timescale. The quantity of energy utilized was also affected by the different pricing choices. The forecasting systems demonstrated varying levels of accuracy in predicting energy usage. The ensemble strategy, which combines Prophet, LSTM, and XGBoost, demonstrated superior performance compared to individual models in terms of error metrics. This suggests that the ensemble approach is beneficial in capturing complicated patterns and improving prediction accuracy. The findings of this research provide significant knowledge on the elements that impact household energy usage and the capability of machine learning to produce accurate forecasts. These observations have the potential to improve the efficiency of energy management techniques and policies.

5. Conclusions and Future Work

This research conducted a thorough performance comparison for energy demand forecasting in various scenarios. This paper assesses the effectiveness, clarity, and importance of three commonly used models for predicting short-term energy demand: Recurrent Neural Networks (LSTM), automated ARIMA regression (XGBoost), and the additive model (Prophet). Furthermore, the paper proposed an ensemble model that combines these three approaches. The study analyses real-world data collected from smart meters installed in 5567 residences across London, sourced from the London Datastore as part of the UK Power Networks-led Low Carbon London initiative. A comprehensive approach was applied to ensure a systematic analysis for data preparation, including feature selection using the CFS algorithm, model training using the Prophet, LSTM, and XGBoost techniques, and model evaluation using MAE, MAPE, and RMSE. The data underwent transformations like normalization and encoding to prepare them for model training. In the final stage, the data were divided into two separate sets: one for training the models and the other for evaluating their performance. The LSTM model was trained using the Mean Absolute Error loss function and Adam optimizer. The optimal configuration was determined by thoroughly examining LSTM architectures and hyperparameters, using MinMaxScaler from Scikit-learn (version 1.5.2) for data normalisation. The training data for the Prophet model were used to adjust the hyperparameters, which included the seasonality model and the changepoint prior scale. The XGBoost model underwent optimization on the training set to fine-tune hyperparameters such as learning rate, maximum tree depth, and number of boosting rounds. The performance metrics for individual models revealed accuracies of 62.96% for the Prophet model, 70.37% for LSTM, and 66.66% for XGBoost. Notably, the proposed ensemble model, which integrates LSTM, Prophet, and XGBoost, demonstrated a significantly higher accuracy of 81.48%, outperforming each individual model. The research aims to enhance energy forecasting models for more accurate and reliable outcomes, which are crucial for energy management, grid optimization, and demand response programs in smart cities. The model can be trained and verified with the latest observational data. The ensemble model can be applied to any dataset from any country or region. The model is scalable, generic, and robust due to its capability to be implemented on different datasets.

In summary, this study illustrates the effectiveness of the ensemble approach, which demonstrated a significant improvement over the performance of individual models in accurately forecasting short-term residential energy demand. However, the reliability and applicability of these models in real-world scenarios could be enhanced by addressing the identified limitations and pursuing further research opportunities. In the future, the proposed framework can be enhanced by incorporating hybrid models, transfer learning, explainable AI, and real-time forecasting. Combining traditional time series forecasting techniques with deep learning models maximizes their strengths, emphasizing transfer learning's role in addressing data limitations and enhancing model application.

Author Contributions: Conceptualization, N.B. and A.S.-A.; methodology, N.B. and A.S.-A.; software, N.B. and A.S.-A.; validation, N.B. and F.S.; formal analysis, A.S.-A. and F.S.; investigation, N.B.; resources, A.S.-A.; data curation, N.B., A.S.-A. and F.S.; writing—original draft preparation, N.B., F.S. and A.S.-A.; writing—review and editing, N.B., A.S.-A. and F.S.; visualization, F.S., A.S.-A. and N.B.; supervision, A.S.-A. and F.S.; project administration, A.S.-A. and F.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This project has received research ethical approval with no ethical issues with the Research Ethics Policy and Procedures of Leeds Beckett University.

Data Availability Statement: The dataset is publicly available online [33].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Jin, N.; Yang, F.; Mo, Y.; Zeng, Y.; Zhou, X.; Yan, K.; Ma, X. Highly Accurate Energy Consumption Forecasting Model Based on Parallel LSTM Neural Networks. *Adv. Eng. Inform.* **2022**, *51*, 101442. [CrossRef]
2. Taïk, A.; Cherkaoui, S. Electrical Load Forecasting Using Edge Computing and Federated Learning. In Proceedings of the ICC 2020—2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–6.
3. Ma, C.; Dai, G.; Zhou, J. Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM_BILSTM Method. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 5615–5624. [CrossRef]
4. Wang, K.; Zhang, J.; Li, X.; Zhang, Y. Long-Term Power Load Forecasting Using LSTM-Informer with Ensemble Learning. *Electronics* **2023**, *12*, 2175. [CrossRef]
5. Haben, S.; Arora, S.; Giasemidis, G.; Voss, M.; Greetham, D.V. Review of Low Voltage Load Forecasting: Methods, Applications, and Recommendations. *Appl. Energy* **2021**, *304*, 117798. [CrossRef]
6. Mir, A.A.; Alghassab, M.; Ullah, K.; Khan, Z.A.; Lu, Y.; Imran, M. A Review of Electricity Demand Forecasting in Low and Middle Income Countries: The Demand Determinants and Horizons. *Sustainability* **2020**, *12*, 5931. [CrossRef]
7. Petropoulos, F.; Apiletti, D.; Assimakopoulos, V.; Babai, M.Z.; Barrow, D.K.; Taieb, S.B.; Bergmeir, C.; Bessa, R.J.; Bijak, J.; Boylan, J.E. Forecasting: Theory and Practice. *Int. J. Forecast.* **2022**, *38*, 705–871.
8. Hong, T.; Wang, P. Artificial Intelligence for Load Forecasting: History, Illusions, and Opportunities. *IEEE Power Energy Mag.* **2022**, *20*, 14–23. [CrossRef]
9. Suanpang, P.; Jamjuntr, P. Machine Learning Models for Solar Power Generation Forecasting in Microgrid Application Implications for Smart Cities. *Sustainability* **2024**, *16*, 6087. [CrossRef]
10. Buratto, W.G.; Muniz, R.N.; Nied, A.; Gonzalez, G.V. Seq2Seq-LSTM with Attention for Electricity Load Forecasting in Brazil. *IEEE Access* **2024**, *12*, 30020–30029. [CrossRef]
11. Sherstinsky, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. *Phys. D* **2020**, *404*, 132306. [CrossRef]
12. Vaswani, A. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
13. Akber, M.Z.; Chan, W.-K.; Lee, H.-H.; Anwar, G.A. TPE-Optimized DNN with Attention Mechanism for Prediction of Tower Crane Payload Moving Conditions. *Mathematics* **2024**, *12*, 3006. [CrossRef]
14. Moussavou Boussougou, M.K.; Park, D.-J. Attention-Based 1D CNN-BILSTM Hybrid Model Enhanced with FastText Word Embedding for Korean Voice Phishing Detection. *Mathematics* **2023**, *11*, 3217. [CrossRef]
15. Forootani, A.; Rastegar, M.; Sami, A. Short-Term Individual Residential Load Forecasting Using an Enhanced Machine Learning-Based Approach Based on a Feature Engineering Framework: A Comparative Study with Deep Learning Methods. *Electr. Power Syst. Res.* **2022**, *210*, 108119. [CrossRef]
16. Chen, Z.; Wang, C.; Lv, L.; Fan, L.; Wen, S.; Xiang, Z. Research on Peak Load Prediction of Distribution Network Lines Based on Prophet-LSTM Model. *Sustainability* **2023**, *15*, 11667. [CrossRef]
17. Akbari, A.S.; Kumar, A.; Reddy, B.R.; Singh, K.K.; Takei, M. Vision Transformer Based Automated Model for Enhancing Lung Cancer Classification. In Proceedings of the 2024 IEEE International Conference on Imaging Systems and Techniques (IST), Tokyo, Japan, 14–16 October 2024; pp. 1–6.
18. Ullah, Z.; Jamjoom, M.; Manikandan, T.; Alajmani, S.; Saleem, F.; Sheikh-Akbari, A.; Ali Khan, U. A Deep Learning Based Intelligent Decision Support System for Automatic Detection of Brain Tumor. *Biomed. Eng. Comput. Biol.* **2024**, *15*, 11795972241277322. [CrossRef]
19. Mehta, R.; Sheikh-Akbari, A.; Singh, K.K. A Noble Approach to 2D Ear Recognition System Using Hybrid Transfer Learning. In Proceedings of the 2023 12th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, 6–10 June 2023; pp. 1–5.
20. Baur, L.; Ditschuneit, K.; Schambach, M.; Kaymakci, C.; Wollmann, T.; Sauer, A. Explainability and Interpretability in Electric Load Forecasting Using Machine Learning Techniques—A Review. *Energy AI* **2024**, *16*, 100358. [CrossRef]
21. Tziolis, G.; Lopez-Lorente, J.; Baka, M.-I.; Koumis, A.; Livera, A.; Theocharides, S.; Makrides, G.; Georghiou, G.E. Direct Short-Term Net Load Forecasting in Renewable Integrated Microgrids Using Machine Learning: A Comparative Assessment. *Sustain. Energy Grids Netw.* **2024**, *37*, 101256. [CrossRef]
22. Alsolami, F.J.; Saleem, F.; Abdullah, A.L. Predicting the Accuracy for Telemarketing Process in Banks Using Data Mining. *J. King Abdulaziz Univ. Comput. Inf. Technol. Sci.* **2020**, *9*, 69–83.
23. Saleem, F.; Al-Ghamdi, A.S.A.-M.; Alassafi, M.O.; AlGhamdi, S.A. Machine Learning, Deep Learning, and Mathematical Models to Analyze Forecasting and Epidemiology of COVID-19: A Systematic Literature Review. *Int. J. Environ. Res. Public Health* **2022**, *19*, 5099. [CrossRef]

24. Guo, F.; Mo, H.; Wu, J.; Pan, L.; Zhou, H.; Zhang, Z.; Li, L.; Huang, F. A Hybrid Stacking Model for Enhanced Short-Term Load Forecasting. *Electronics* **2024**, *13*, 2719. [CrossRef]
25. Bouktif, S.; Fiaz, A.; Ouni, A.; Serhani, M.A. Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting. *Energies* **2020**, *13*, 391. [CrossRef]
26. Wang, Z.; Hong, T.; Li, H.; Piette, M.A. Predicting City-Scale Daily Electricity Consumption Using Data-Driven Models. *Adv. Appl. Energy* **2021**, *2*, 100025. [CrossRef]
27. Li, X.; Wang, Z.; Yang, C.; Bozkurt, A. An Advanced Framework for Net Electricity Consumption Prediction: Incorporating Novel Machine Learning Models and Optimization Algorithms. *Energy* **2024**, *296*, 131259. [CrossRef]
28. Li, S.; Wang, P.; Goel, L. A Novel Wavelet-Based Ensemble Method for Short-Term Load Forecasting with Hybrid Neural Networks and Feature Selection. *IEEE Trans. Power Syst.* **2015**, *31*, 1788–1798. [CrossRef]
29. Wi, Y.-M.; Joo, S.-K.; Song, K.-B. Holiday Load Forecasting Using Fuzzy Polynomial Regression with Weather Feature Selection and Adjustment. *IEEE Trans. Power Syst.* **2011**, *27*, 596–603. [CrossRef]
30. Wang, Y.; Gan, D.; Zhang, N.; Xie, L.; Kang, C. Feature Selection for Probabilistic Load Forecasting via Sparse Penalized Quantile Regression. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 1200–1209. [CrossRef]
31. Bashir, T.; Chen, H.; Tahir, M.F.; Zhu, L. Short Term Electricity Load Forecasting Using Hybrid Prophet-LSTM Model Optimized by BPNN. *Energy Rep.* **2022**, *8*, 1678–1686. [CrossRef]
32. Rotondo, A.; Quilligan, F. Evolution Paths for Knowledge Discovery and Data Mining Process Models. *SN Comput. Sci.* **2020**, *1*, 109. [CrossRef]
33. Jean-Michel, D. Smart Meters in London—Dataset. Available online: https://www.kaggle.com/datasets/jeanmidev/smart-meters-in-london (accessed on 2 November 2024).
34. Okereke, G.E.; Bali, M.C.; Okwueze, C.N.; Ukekwe, E.C.; Echezona, S.C.; Ugwu, C.I. K-Means Clustering of Electricity Consumers Using Time-Domain Features from Smart Meter Data. *J. Electr. Syst. Inf. Technol.* **2023**, *10*, 2. [CrossRef]
35. Theng, D.; Bhoyar, K.K. Feature Selection Techniques for Machine Learning: A Survey of More than Two Decades of Research. *Knowl. Inf. Syst.* **2024**, *66*, 1575–1637. [CrossRef]
36. Akber, M.Z. Improving the Experience of Machine Learning in Compressive Strength Prediction of Industrial Concrete Considering Mixing Proportions, Engineered Ratios and Atmospheric Features. *Constr. Build. Mater.* **2024**, *444*, 137884. [CrossRef]
37. Mishra, P.; Al Khatib, A.M.G.; Yadav, S.; Ray, S.; Lama, A.; Kumari, B.; Sharma, D.; Yadav, R. Modeling and Forecasting Rainfall Patterns in India: A Time Series Analysis with XGBoost Algorithm. *Environ. Earth Sci.* **2024**, *83*, 163. [CrossRef]
38. Sunki, A.; SatyaKumar, C.; Narayana, G.S.; Koppera, V.; Hakeem, M. Time Series Forecasting of Stock Market Using ARIMA, LSTM and FB Prophet. In Proceedings of the MATEC Web of Conferences, Kuala Lumpur, Malaysia, 6–8 November 2024; EDP Sciences: Les Ulis, France, 2024; Volume 392, p. 01163.
39. Hong, S.; Ko, S.-J.; Woo, S.I.; Kwak, T.-Y.; Kim, S.-R. Time-Series Forecasting of Consolidation Settlement Using LSTM Network. *Appl. Intell.* **2024**, *54*, 1386–1404. [CrossRef]
40. Cortes, V.M.P.; Chatterjee, A.; Khovalyg, D. Dynamic Personalized Human Body Energy Expenditure: Prediction Using Time Series Forecasting LSTM Models. *Biomed. Signal Process. Control* **2024**, *87*, 105381.
41. Al-Selwi, S.M.; Hassan, M.F.; Abdulkadir, S.J.; Muneer, A.; Sumiea, E.H.; Alqushaibi, A.; Ragab, M.G. RNN-LSTM: From Applications to Modeling Techniques and beyond—Systematic Review. *J. King Saud. Univ. Comput. Inf. Sci.* **2024**, *36*, 102068. [CrossRef]
42. Wang, J.; Hong, S.; Dong, Y.; Li, Z.; Hu, J. Predicting Stock Market Trends Using LSTM Networks: Overcoming RNN Limitations for Improved Financial Forecasting. *J. Comput. Sci. Softw. Appl.* **2024**, *4*, 1–7.
43. Elseidi, M. A Hybrid Facebook Prophet-ARIMA Framework for Forecasting High-Frequency Temperature Data. *Model. Earth Syst. Environ.* **2024**, *10*, 1855–1867. [CrossRef]
44. Nizam, N.N.S.; Jasin, A.M.; Asmat, A.; Abdullah, N.A.; Abdul-Rahman, S. Predictive Analytics of Train Delays Using Facebook Prophet. *AIP Conf. Proc.* **2024**, *3128*, 030007.
45. Cheng, J.; Tiwari, S.; Khaled, D.; Mahendru, M.; Shahzad, U. Forecasting Bitcoin Prices Using Artificial Intelligence: Combination of ML, SARIMA, and Facebook Prophet Models. *Technol. Forecast. Soc. Chang.* **2024**, *198*, 122938. [CrossRef]
46. Wang, J.; Du, X.; Qi, X. Strain Prediction for Historical Timber Buildings with a Hybrid Prophet-XGBoost Model. *Mech. Syst. Signal Process.* **2022**, *179*, 109316. [CrossRef]
47. Banik, R.; Biswas, A. Enhanced Renewable Power and Load Forecasting Using RF-XGBoost Stacked Ensemble. *Electr. Eng.* **2024**, *106*, 1–21. [CrossRef]
48. Wang, Z.H.; Liu, Y.F.; Wang, T.; Wang, J.G.; Liu, Y.M.; Huang, Q.X. Intelligent Prediction Model of Mechanical Properties of Ultrathin Niobium Strips Based on XGBoost Ensemble Learning Algorithm. *Comput. Mater. Sci.* **2024**, *231*, 112579. [CrossRef]
49. Jabeur, S.B.; Mefteh-Wali, S.; Viviani, J.-L. Forecasting Gold Price with the XGBoost Algorithm and SHAP Interaction Values. *Ann. Oper. Res.* **2024**, *334*, 679–699. [CrossRef]

50. Gogineni, A.; Panday, I.K.; Kumar, P.; Paswan, R.K. Predicting Compressive Strength of Concrete with Fly Ash and Admixture Using XGBoost: A Comparative Study of Machine Learning Algorithms. *Asian J. Civil. Eng.* **2024**, *25*, 685–698. [CrossRef]
51. Ganaie, M.A.; Hu, M.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble Deep Learning: A Review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [CrossRef]