Edelweiss Applied Science and Technology
ISSN: 2576-8484
Vol. 8, No. 5, 317-327
2024
Publisher: Learning Gate
DOI: 10.55214/25768484.v8i5.1689

A comparative analysis of single learning and ensemble learning approaches for the forecasting of electricity prices

Fatma Yaprakdal
Information System Engineering Department, Bucak Computer and Informatics Faculty, Burdur Mehmet Akif Ersoy University; fatmaya-prakdal@mehmetakif.edu.tr (F.Y.).

Abstract: Forecasting electricity prices accurately is imperative in deregulated power markets. Nonetheless, the intricate nature of electricity prices, characterized by high frequency and volatility, poses a challenge in building an effective forecasting model for policymakers and scientists. Precision in the electricity price prediction is crucial for providing valuable guidance to market participants, helping them maximize their benefits. In prior studies, various methods, including statistical models and artificial neural network models, have been used to forecast electricity prices. This study proposes three ensemble learning approaches - AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU - which combine the AdaBoost algorithm with LSTM, Bi-LSTM, and GRU networks, respectively, to enhance the accuracy of electricity price predictions. The study aims to assess the effectiveness of the proposed models by comparing their predictive performance with that of single RNN-based models (LSTM, Bi-LSTM, GRU) using daily maximum electricity prices from 2004 to 2008. Notably, there has been no existing research that compares the effectiveness of these single and hybrid models. In the current literature, these single models are widely acknowledged as potent tools for improving forecasting accuracy. On the other hand, although the proposed ensemble learning approaches obtained using the AdaBoost boosting technique have been used in areas such as financial forecasting so far, they have never been used in electricity price forecasting. Accuracy assessment utilizing R-squared and MAPE clearly demonstrates that the AdaBoost-BLSTM approach performs very closely to, but better than, other boosting ensemble approaches, and significantly better than single models.
Keywords: *AdaBoost, Bi-LSTM, Electricity price forecasting, Ensemble learning, GRU, LSTM.*

1. Introduction

The proficient management of power markets plays a critical role in advancing environmentally friendly production and fostering sustainable economic growth. With the ongoing deregulation of the power industry, the intricacies and uncertainties within power market trading present prime opportunities for fostering innovation and driving growth. In this environment, in order to optimize resource allocation, mitigate risks, and maximize economic benefits, accurate forecasting of electricity prices becomes a critical tool for stakeholders, policymakers, and suppliers. Understanding spot market volatility provides financial benefits for generators while maintaining grid stability. Consumers are also able to use forecasting to make informed choices and avoid loss due to high pricing. As a result, ongoing electricity market reforms in various countries highlight the growing importance of accurate electricity price forecasting.

Electricity has unique characteristics set it apart from other commodities, such as oil and gas. Unlike these commodities, it has no natural storage and requires a constant balance between generation and demand. Electricity prices have intrinsic characteristics such as high frequency, unstable mean and variance, seasonality, and non-linear behavior because of these unique characteristics. This makes it

* Correspondence: fatmaya-prakdal@mehmetakif.edu.tr

challenging to forecast electricity prices accurately. Therefore, to cope with these complexities, there is an urgent need to develop a high-quality and efficient electricity price forecasting model [1]. Electricity price forecasting has been the subject of various statistical and econometric analyses, including Vector Autoregression, Error Correction, and Autoregressive Integrated Moving Average models. However, conventional methods encounter difficulties capturing electricity price data's complex, non-linear nature, leading to less precise forecasting. Therefore, exploring more effective forecasting methods with greater learning capacity for this data type is crucial. The forecasting accuracy of non-linear AI methods typically surpasses traditional econometric and statistical models despite facing challenges such as parameter overfitting and optimization. Advanced artificial intelligence (AI) methods such as Artificial Neural Networks (ANNs) [2], Support Vector Regression (SVR) [3, 4], and Deep Learning (DL) techniques [5] have been developed and implemented to enhance the reliability of electricity price forecasts. ANNs are commonly employed for time series forecasting, underscoring their suitability for this task. Combining the strengths of different ANN methods can lead to improved forecasting performance. The long short-term memory (LSTM) method, one of the variations of the recurrent neural network (RNN), is considered the preferred option for electricity price forecasting [6]. For an in-depth analysis of the recent advancements in electricity price forecasting over the last two decades, please refer to the comprehensive reviews in [7-9].

Moreover, several hybrid prediction methods have been developed to enhance prediction performance. Ensemble learning, a technique that involves training multiple models and combining their outputs, has been widely applied in forecasting time series in various areas, including financial time series prediction [10, 11], crude oil price prediction [12-14], energy production and consumption prediction [15, 16], electricity demand prediction [17-21], and electricity price forecasting [22-24]. Ensemble learning models like AdaBoost, Gradient Boosting, and Bagging leverage the strengths of multiple base learners to enhance forecast accuracy. These approaches are especially beneficial for reducing overfitting and enhancing the reliability of predictions. The AdaBoost algorithm has been thoroughly researched in the literature because it can enhance the accuracy of weak classifiers by amalgamating them into a robust predictive model. This technique is especially effective when individual models have difficulty performing well, as AdaBoost adjusts the weights of misclassified instances iteratively, guaranteeing that subsequent models concentrate more on these challenging cases [25]. The AdaBoost algorithm integrates machine learning (ML) techniques into a unified forecasting model, effectively mitigating bias. Throughout the boosting process, observations are weighted to ensure each contributes to multiple combinations. This approach enables subsequent learners to concentrate on and learn from challenging cases during their training. A common practice in the ensemble learning technique is using the Adaptive Boosting algorithm (AdaBoost) to combine various RNN-based predictors. The ensemble learning models AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU harness the power of two robust machine learning techniques: AdaBoost, an ensemble method celebrated for its boosting capabilities, and LSTM, BiLSTM, and GRU types of RNNs which are specifically designed for time-series forecasting [26-29]. These innovative hybrid approaches have been increasingly recognized in academic literature for their efficacy in addressing intricate, nonlinear patterns in sequential data, particularly in domains like financial markets, energy prices, and other forecasting fields.

This research delves into three ensemble approaches - AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU models - to enhance the accuracy of electricity price predictions. Currently, there is a lack of studies in the literature that make use of the proposed ensemble models for forecasting electricity prices. This creates an opportunity to contribute new insights to this area of research. These models leverage the AdaBoost algorithm in conjunction with LSTM, Bi-LSTM, and GRU networks, respectively. The study assesses these new models' predictive capabilities compared to individual RNN-based models (LSTM, Bi-LSTM, GRU) using daily maximum electricity prices from January 1, 2004, to December 31, 2008. It's noteworthy that no existing research has undertaken a comparative analysis of the efficacy of these single and ensemble models. While the individual models are well-regarded for

their ability to improve forecasting accuracy, the proposed boosting ensemble methods (AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU) have not been widely applied in the realm of electricity price forecasting, despite their utilization in other domains such as financial forecasting.
The paper is structured as follows: First, the relevant methodology for both single and proposed ensemble prediction methods is presented. Section 2 includes data analysis, providing detailed information about the dataset and the preprocessing steps. Section 4 details the experimental settings of all the prediction methods. Section 5 explains the evaluation metrics used to assess electricity price prediction results. Section 6 presents the prediction performance results of the models and discusses the findings. The final section outlines conclusions and future work.

2. Methodology

2.1. LSTM

In traditional neural networks, the assumption of independent inputs limits their effectiveness with sequential data and varied input and output sizes. However, the RNN is particularly suited to sequential data using memory loops and recurrent hidden states to accommodate inputs of varying lengths. Nevertheless, RNNs face the vanishing gradient problem, which hampers their learning ability as relevant information diminishes during back-propagation. Due to the tendency of gradients to either disappear or explode as they move in and out of feedback loops, training deep RNNs to retain information across multiple time steps is challenging In response to this challenge, Schmidhuber and Hochreiter proposed LSTM, a model that effectively solves the problem of vanishing gradients by using memory cells.

LSTM is an innovative variation of the RNN approach, purposefully crafted to handle long-term dependency issues effectively. While both LSTM and RNN share a similar structure, a basic LSTM comprises three gates, a single cell, block input, peephole connections, and an output activation function. The LSTM introduced a groundbreaking solution to the issue of vanishing and exploding gradients within RNN architecture In LSTM, the forgetting gate (f_t) decides which information to keep or discard in the cell state, the input gate (i_t) decides which new information to store in the cell state, and the output gate (o_t) controls the output of each cell. Additionally, these decisions are informed by the current state of the cell, as well as by the filtered and recently input data. During network training, each gate is assigned a learnable weight matrix. The equations defining an LSTM unit are:

$$\tilde{c_t} = \tanh\,(W_c[h_{t-1}, x_t] + b_c), \tag{1}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \tag{2}$$
$$f_t = \sigma\left(W_f[h_{t-1}, x_t] + b_f\right), \tag{3}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \tag{4}$$
$$c_t = (i_t \times \tilde{c_t}) + (f_t \times c_{t-1}), \tag{5}$$
$$h_t = o_t \times \tanh\,(c_t), \tag{6}$$

where W_i, W_f, and W_o, and b_i, b_f, and b_o are the weights and biases that drive the behavior of i_t, f_t and o_t gates, respectively. W_c and b_c are the weight and bias of the $\tilde{c_t}$ memory cell candidate, respectively. c_t, update the cell state by combining the previous state and the new candidate values. 'tanh' is the hyperbolic tangent activation function [30].

2.2. Bi-LSTM

To maximize the efficiency of model computation and minimize the potential for overfitting, the Bi-LSTM model consists of two opposing LSTM arrays. Output comes from both directions, with the information from the time t=1 up to T entering the forward layer, and the information from the time t=T up to 1 passing into the LSTM backward layer. For each time step t, the computation of the Bi-LSTM layers as follows.

Forward LSTM:

$$f_t = LSTM(x_t, f_{t-1}, (C_{t-1})^f), \quad (7)$$

Backward LSTM:

$$b_t = LSTM(x_t, b_{t-1}, (C_{t-1})^b), \quad (8)$$

Here, f_t represents the forward hidden state, and b_t represents the backward hidden state.

Concatenation:

$$h_t = [f_t; b_t], \quad (8)$$

The final output h_t at each time step is obtained by concatenating the hidden states from both directions [31].

2.3. GRUs

The GRU model is a more efficient mechanism compared to the LSTM model. GRU combines the input gate and the forget gate, resulting in a model with fewer parameters. GRU excels at collecting current information, making it well-suited for tasks where current data points are better predictors. Its reset gate (R_t) and update gate (Z_t) enable effective processing of sequential data while optimizing memory usage. GRU is recommended for small datasets, while LSTM tends to perform better with larger datasets. Here are the key equations that define the operations within a GRU.

Update gate:

$$z_t = \sigma(W_z.[h_{t-1}, x_t], + b_z), \quad (9)$$

Here, z_t is the update gate vector, h_{t-1} is the previous hidden state, x_t is the current input, σ is the sigmoid activation function, W_z and b_z are the weight matrix and bias for the update gate, respectively.

Candidate activation*:*

$$\widetilde{h_t} = \tanh\left((W_h.[r_t * h_{t-1}, x_t], + \ b_r\right), \quad (10)$$

The candidate activation vector, $\widetilde{h_t}$ is the new hidden state computed using the current input and the reset hidden state. It is a candidate for updating the hidden state. Here, $\widetilde{h_t}$ is the candidate hidden state, W_h and b_h are the weight matrix and bias for the candidate hidden state, respectively. '*' denotes element-wise multiplication.

Hidden state update:

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t}, \quad (11)$$

Finally, the hidden state h_t is updated using the update gate and the candidate hidden state [30].

2.2. Ensemble Learning

Ensemble learning involves training several models. Their outputs are then combined to produce the final prediction. This approach can improve prediction accuracy and consistency. There are three common ensemble methods for combining the predictions of multiple classifiers: Weighted Average (WA), Max Voting (MV), and Averaging. MV is commonly used for classification tasks. In this technique, multiple individual classifier models are used to make predictions for each data point. The output of each classifier is regarded as a vote, and the final output is based on the majority of votes. Averaging works in a similar way to MV, but the final output is an average of all individual or single classifier outputs. In contrast to MV, averaging is applicable to regression as well as to classification machine learning. WA is an advanced version of the averaging technique. The WA method assigns each model a different weight, indicating its importance in making predictions.

Advanced combination techniques, on the other hand, include stacking, blending, bagging, and boosting.

Stacking refers to using predictions from multiple models (m_1, m_2, ..., m_n) to create another model, which is then utilized to make predictions on the test dataset. The objective of stacking is to enhance the accuracy of a classifier. The fundamental concept behind stacking involves combining the forecasts of the models (m_1, m_2, ..., m_n) using a linear combination of weights (a_i) learned by a meta-learner, as shown in the following equation.

$$f_{STK}(x) = \sum_{i=1}^{n} a_i f_i(x),\ (\mathrm{i} = 1, \ldots, \mathrm{n}), \tag{12}$$

The blending ensemble technique works in a similar way to a stacking technique. However, the main distinction between stacking and blending is that while stacking relies on a separate test dataset to make the prediction, blending utilizes a validation set of the training data for the prediction. In other words, the prediction is based solely on the validation data set of the training data set. The output of the predicted data set and the validation data set is used to construct the final model for prediction of the test data set.

We can use bagging, also known as bootstrap aggregation, to construct the ultimate predictive model for the test data set. Bagging combines the results of several models, such as N number of k-nets, to produce a more general output. In this technique, many subsets (bags) of the original training data set are generated with replacement using bootstrapping sampling methods. The bags produced by the bagging process give the model unbiased insight into the whole data set. Bagging meta-estimators and random forests are some ML algorithms that employ bagging techniques. Overall, bagging aims to reduce model variance.

Boost, sometimes called a "meta-algorithm", is an iterative process in which every subsequent model attempts to improve on the errors of the previous model. Each successive model depends upon the previous one. A boosting algorithm is used to reduce the model's bias. Boosting techniques combine multiple weak learners into a single strong learner. Individual models perform well on specific subsets of the dataset, although they may not perform better on the whole dataset. Each model, therefore, represents a significant improvement in the overall performance capability of the Ensemble. Popular boosting methods include the AdaBoost, the GBM, the XGBM, the Light GBM, and the CatBoost [32]. This study employs the AdaBoost algorithm as an ensemble learning technique for LSTM, Bi-LSTM, and GRU models, enhancing their predictive capabilities.

2.2.1. Adaboost Algorithm

In 1997, Freund and Schapire introduced the AdaBoost algorithm. This innovative algorithm amalgamates various ML techniques into a unified forecasting model, effectively reducing bias. Throughout the boosting process, observations are meticulously weighted to ensure meaningful contributions to multiple combinations. Additionally, during the training process, each classifier takes into account the performance of previous classifiers. After each training step, the weights are adjusted and redistributed. Significantly, misclassified data is allocated greater weight, highlighting the most challenging cases. By utilizing this approach, subsequent learners can specifically focus on and learn from these demanding cases during their training [33].

2.2.1. Adaboost-LSTM, Adaboost-BLSTM and AdaBoost-GRU Approaches

To begin, it is necessary to input a series of samples $(y_1, \hat{y}_1)$, ...$(y_1, \hat{y}_n)$, where the output "$\hat{y}_i$" represents a real number. Following this, a model is constructed using a base weak learner such as LSTM, Bi-LSTM, or GRU. Subsequently, the training sample set's sampling weights (Wi) are initialized. The created models are then encapsulated in scikit-learn and finally boosted. It is important to start by assigning equal weight to all the samples as in the following equation.

$$W_i = \frac{1}{N}, \tag{13}$$

The variable N represents the number of samples $(i = 1, 2, 3, \ldots, N)$ for the LSTM, Bi-LSTM, and GRU models during training. These models are provided with training samples, and the error is calculated using the second step of the Adaboost Algorithm. The foresting error 'err_m' and ensemble weight 'α_m' of the model based on its error is calculated as follows.

$$err_m = \sum_{i=1}^{N} W_i^{\ m}\ \mathbb{I}(y_i \neq \hat{y}_i), \tag{14}$$

$$\alpha_m = \frac{1}{2} \ln \left(\frac{1 - err_m}{err_m}\right) \tag{15}$$

ISSN: 2576-8484
Vol. 8, No. 5: 317-327, 2024
DOI: 10.55214/25768484.v8i5.1689

where $\amalg(y_i \neq \hat{y}_i)$ is 1 if the prediction $\hat{y}_i$ is incorrect, and otherwise, it is zero. Next, the sampling weights are updated for the next iteration.

$$W_i^{(m+1)} = W_i^{m}\, e^{\alpha_m \cdot \amalg(y_i \neq \hat{y}_i)}, \tag{16}$$

The final ensemble prediction is obtained by calculating the weighted sum of the predictions from each model, as seen in the following equation [32].

$$\hat{y}_i = sign(\textstyle\sum_{m=1}^{M} \alpha_m \cdot y_m), \tag{17}$$

3. Data Analysis

3.1. Collection of the Data

Artificial neural network (ANN) algorithms require a substantial amount of data to achieve accurate results. Therefore, it is better to use intraday or daily data, as it allows for a significant number of observations to be collected within a shorter timeframe. This research used five-year data in daily resolution to forecast electricity prices.

The dataset used is a spreadsheet of historical hourly load data, natural gas and electricity prices, and temperature data collected by the New England ISO from 2004 until 2008. The weather information specifies the dry bulb temperature, which measures the ambient air temperature, and the dew point, which indicates the temperature at which air becomes saturated with water vapor. This valuable dataset is sourced from the MathWorks website and can be easily accessed in reference [34]. For this study, the original dataset was reorganized to a daily maximum resolution instead of its original hourly resolution.

3.2. Processing of the Data

When conducting data analysis, it is imperative to preprocess the data before applying any forecasting models. Applying a min-max scaler to normalize the data is essential in the preprocessing step. As described in equation 5, this technique will scale the data to a range between 0 and 1, maintaining the proportionality of the original values. Normalizing the data in this way is essential for ensuring that the input features contribute equally to the analysis and modeling processes, thus improving the overall effectiveness of the data processing methodology. Specifically, we set the min-max scaler range as [0, 1] for our analysis. It is important to note that at the output layer of both the AdaBoost LSTM and AdaBoost GRU, the input and output variables are scaled from 0 to 1 to match the activation function scale [23].

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{18}$$

We implemented a forward-moving window approach with a size of 7 to forecast the eighth data point, using the initial 7 data points (x_1 to x_7) as input. Afterward, we modified the window to cover the second to eighth data points (x_2 to x_8) to forecast the ninth data point and repeated this process. We used the Pandas shift function to move the whole column by the number given to keep the window size at seven. The dataset encompasses a total of 1820 rows and five columns. The columns contain information related to historical daily maximum loads, natural gas prices, electricity prices, the dew point, and dry bulb temperature. We partitioned the data into 80% for training and 20% for testing. We partitioned the dataset into two sets. The first set comprised 1456 observations and was assigned to the training set. The second set consisted of the remaining 364 observations and was designated as the test set. Throughout this process, we maintained a fixed window size of 7. Suppose that the set of observations is represented by the symbol K,

$$K = (x_i, y_i)_{i=1}^{N}, \tag{19}$$

whereas $(x_i)_{i=1}^{N}$ are the inputs to the model, and $(y_i)_{i=1}^{N}$ are the labels [27]. In our research, we are working with the dimensional variables y and x_i. x_i is taken from the set of real numbers raised to the power of D. D is the number of variables (historical daily maximum loads, natural gas prices, electricity prices, the dew point, and dry bulb temperature), 5 for us. Additionally, 'y_i' belongs to the set of real numbers.

4. Experimental Setting

To effectively use the LSTM, Bi-LSTM, GRU, Adaboost-LSTM, Adaboost-BiLSTM, and AdaBoost-GRU models in the Python Keras library, the input and target data must conform to a specified 3-dimensional format that includes the number of observed data, the hidden state length, and the predictor number.

When configuring the LSTM, Bi-LSTM, and GRU models, we included two hidden layers of 300 neurons each and used a single output layer neuron. The Rectified Linear Unit (ReLU) activation function was used in dense layers. This choice was deliberate as ReLU is known for its resilience against the gradient vanishing problem. Researchers have widely embraced it to enhance the network's trainability effectively. The mean squared error was used to determine the cost functions.

The Adaptive Moment Estimation (Adam) optimizer, which has a learning rate of 0.001, was used to train for 100 epochs.

The batch size was set to 32, emphasizing the importance of preserving data order during our analysis by setting the shuffle parameter to 'false'. In addition, an early stopping patience of 5 was applied by using EarlyStopping callback in the Keras library, which is one of the most widely used regularization techniques to combat the overfitting issue. Then, we trained all the networks using Keras in Python through the Google collaborative application. The summarized parameters for all models can be found in Table 1.

Table 1.
The parameter informations of all models.

The parameters	Values/Types
The batch size	32
Maximum number of epochs	100
Early stopping patience	5
Hidden layer number	1
Optimizer	Adam
Activation function	ReLU
Number of neurons	300
Learning rate	0.001

Furthermore, the AdaBoost algorithm was seamlessly integrated into the LSTM, Bi-LSTM, and GRU methods using the sklearn wrapper and was subsequently boosted. The different stages of data processing are effectively outlined, and the exact application of the AdaBoost algorithm is explained in the visualization in Figure 1.

Implementing AdaBoost on any model involves wrapping the defined model and boosting it with either AdaBoost classifier or AdaBoost Regressor as the case may be.

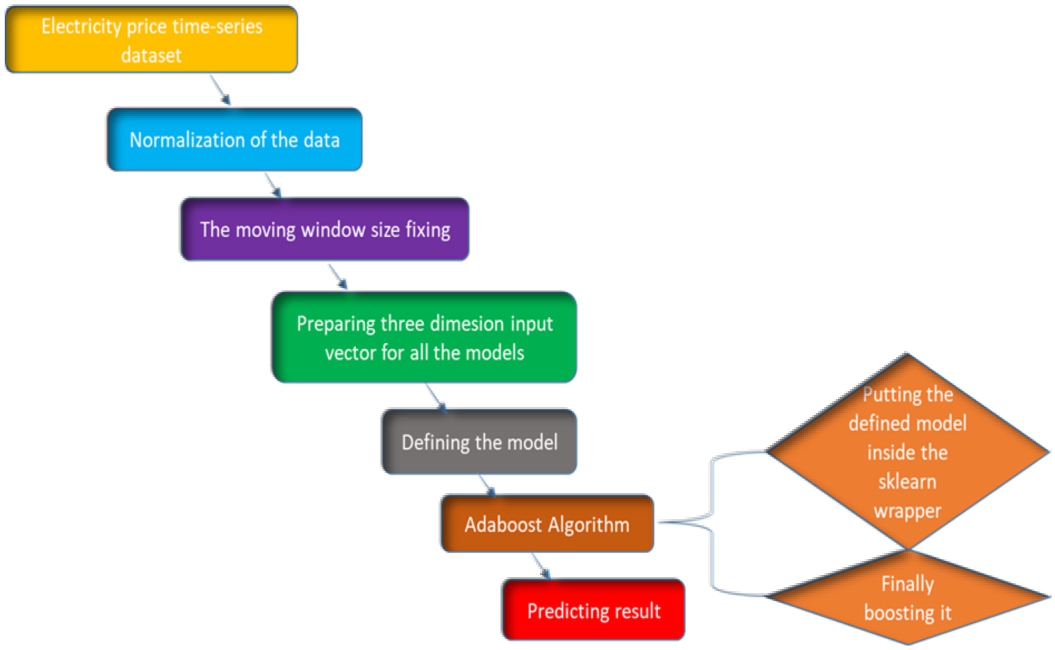


Figure 1.
Study framework.

5. Evaluation Metrics

Numerous methods are available for evaluating predictive performance, each with distinct strengths and weaknesses. Three different statistics were used to assess the predictive effectiveness of the proposed AdaBoost-LSTM, AdaBoost-GRU, and AdaBoost-LSTM models against the other benchmark methods: the Coefficient of Determination (R-squared) and the Mean Absolute Percentage Error (MAPE). Evaluation metrics play a crucial role in research papers by providing insight into the effectiveness of the methods. In this paper, MAPE and R-squared, both independent of scale, have been used to compare prediction values with different units. MAPE is a common metric for assessing model prediction accuracy. It measures the average size of errors produced by a model and indicates the average level of predictive accuracy. R-squared is a statistical metric of the amount of variation in the dependent variable that is explainable by the independent variable of a regression analysis. Ranges from 0 to 1, where 1 is a perfect fit. Eqs. (20) and Eqs. (21) represent the mathematical expressions for the MAPE and R-squared metrics, respectively.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{A_i - F_i}{A_i}\right|, \quad (20)$$

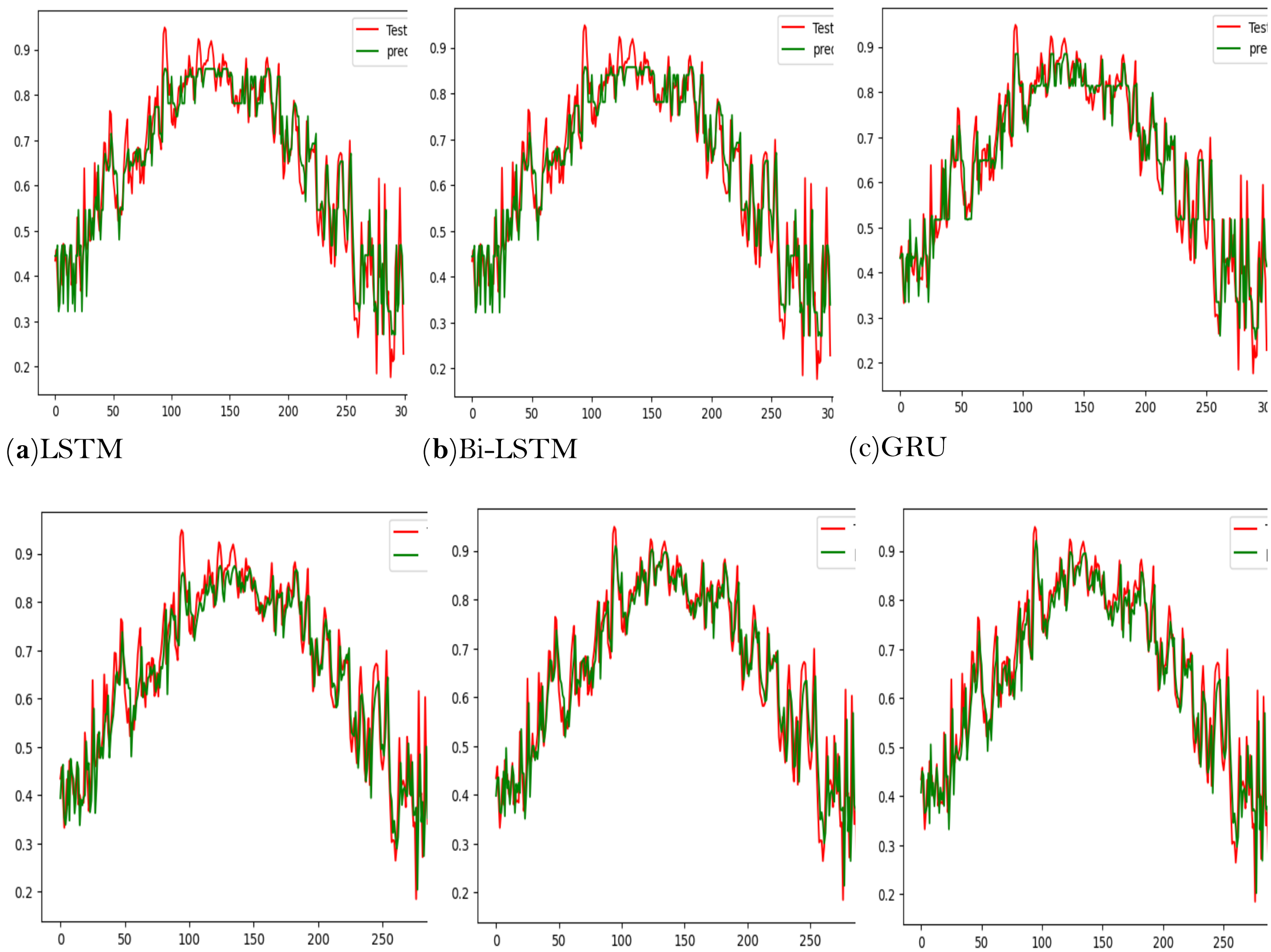
$$R^2 = \frac{\sum_{i=1}^{n}(F_i - A_i)^2}{\sum_{i=1}^{n}(A_i - A_m)^2}, \quad (21)$$

In the formula provided, N represents the total number of data points used in the calculation. The variable F_i represents the prediction, whereas Ai represents the actual observation. The symbol A_m denotes the data mean or average, calculated by summing all data points and dividing by the total number of data points.

6. Empirical Results and Discussion

The LSTM, Bi-LSTM, and GRU models allow the prediction accuracy results of the present time step to be examined using data from previous time steps. For comparison, the models AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU were benchmarked against the single LSTM, Bi-LSTM, and GRU models.

Differences between actual and predicted values are shown in Figure 2, with actual values represented by green lines and predicted values by red lines. The predictive trends of all models closely align with the trend of the real values line, indicating their strong prediction accuracy relative to the actual data. This can be attributed to the inherent ability of RNN variants to forecast smooth regression trends and generate accurate predictions from sequential data. The ensemble learning approach of AdaBoost-BLSTM, AdaBoost-LSTM, and AdaBoost-GRU outperformed the individual models, as seen in Figure 2.



(**a**)LSTM (**b**)Bi-LSTM (c)GRU

(**d**)Adaboost-LSTM (**e**) Adaboost-BLSTM (f) Adaboost-GRU

Figure 2.
The actual and predicted values of single models: (a) LSTM, (b) Bi-LSTM, (c) GRU; and ensemble learning models: (d) Adaboost-LSTM, (e) Adaboost-BLSTM, (f) Adaboost-GRU.

Using electricity price data, various models, including LSTM, Bi-LSTM, and GRU, as well as three ensemble forecasting models—AdaBoost-LSTM, AdaBoost-BLSTM, and AdaBoost-GRU—were analyzed. The AdaBoost-BLSTM model displayed superior predictive performance, substantiated by metrics such as R-squared error and MAPE, as shown in Table 2. Both AdaBoost-LSTM and AdaBoost-GRU also showed improved performance. These findings suggest that ensemble methods offer enhanced forecasting performance compared to individual models.

Table 2.
Performance metric results of all models.

Forecasting approaches	Models	R-squared	MAPE(%)
Single Forecasts	GRU	0.8990	10.8750
	LSTM	0.8963	11.2533
	Bi-LSTM	0.8922	11.6561
Ensemble Forecasts	AdaBoost-LSTM	0.9034	10.8120
	Adaboost_GRU	0.9120	10.4017
	Adaboost_BLSTM	0.9125	10.4001

7. Conclusion

This paper explored an exciting opportunity to improve forecasting performance by comparing individual LSTM, Bi-LSTM, and GRU models with their ensemble counterparts using the AdaBoost algorithm. It also highlighted the benefits of selecting features from a dataset without prior knowledge of the predictor variables. A robust dataset of five years of electricity prices was used to rigorously test the predictive ability of the ensemble models against the individual models. The study used scale-independent evaluation metrics such as R-squared and MAPE to assess the predictive performance. The results highlighted a consistent trend in the performance of the models, showing that the ensemble models consistently outperformed their single counterparts. Despite the superior performance of a single GRU compared to a single LSTM and Bi-LSTM, the AdaBoost-BLSTM ensemble model consistently showed the smallest prediction errors, outperforming both the AdaBoost-LSTM and AdaBoost-GRU models. As a result, the study identifies the AdaBoost-BLSTM ensemble learning model as a particularly promising approach to electricity price forecasting.

The model is expected to play a crucial role in conducting electricity load and price forecasting studies. It will be further developed by incorporating advanced ensemble techniques to enhance its accuracy and effectiveness in predicting electricity load and price trends.

References

[1] Mohammadreza, H.; Farhad, H.;, Meysam, F. Daily electricity price forecasting using artificial intelligence models in the Iranian electricity market. *Energy* **2023**, 263, 126011. https://doi.org/10.1016/j.energy.2022.126011

[2] Ioannis, P. P.; Athanasios, S. D. Day-ahead electricity price forecasting via the application of artificial neural network based models. *Applied Energy* **2016**, 172, 132-151. https://doi.org/10.1016/j.apenergy.2016.03.089

[3] Zahid, M.; Ahmed, F.; Javaid, N.; Abbasi, R.A.; Zainab Kazmi, H.S.; Javaid, A.; Bilal, M.; Akbar, M.; Ilahi, M. Electricity Price and Load Forecasting using Enhanced Convolutional Neural Network and Enhanced Support Vector Regression in Smart Grids. *Electronics* **2019**, *8*, 122. https://doi.org/10.3390/electronics8020122

[4] Golia, S.; Grossi, L.; Pelagatti, M. Machine Learning Models and Intra-Daily Market Information for the Prediction of Italian Electricity Prices. *Forecasting* **2023**, *5*, 81-101. https://doi.org/10.3390/forecast5010003

[5] Vega-Márquez, B.; Rubio-Escudero, C.; Nepomuceno-Chamorro, I.A.; Arcos-Vargas, Á. Use of Deep Learning Architectures for Day-Ahead Electricity Price Forecasting over Different Time Periods in the Spanish Electricity Market. *Appl. Sci.* **2021**, *11*, 6097. https://doi.org/10.3390/app11136097

[6] Manfre Jaimes, D.; Zamudio López, M.; Zareipour, H.; Quashie, M. A Hybrid Model for Multi-Day-Ahead Electricity Price Forecasting considering Price Spikes. *Forecasting* **2023**, *5*, 499-521. https://doi.org/10.3390/forecast5030028

[7] Weron, R. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *Int J Forecast* **2014**, 30(4), 1030–81.

[8] Hong, T.; Pinson, P.; Wang, Y.; Weron, R.; Yang, D; Zareipour, H. Energy forecasting: A review and outlook. *IEEE Open Access J Power Energy* **2020**, 7, 376–88.

[9] Lago, J.; Marcjasz, G.; De Schutter, B.; Weron, R. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Appl Energy* **2021**, 293, 116983.

[10] Busari, G.A.; Kwak, N. W.; Lim, D. H. An Application of AdaBoost-GRU Ensemble Model to Economic Time Series Prediction. *Indian Journal of Science and Technology* **2021**, 14(31), 2557-2566. https://doi.org/10.17485/IJST/v14i31.1204
[11] Wang, S.; Guotai, Chi. Cost-sensitive stacking ensemble learning for company financial distress prediction. *Expert Systems with Applications* **2024**, 255, 124525. https://doi.org/10.1016/j.eswa.2024.124525Get rights and content
[12] Ganiyu, A.B.; Dong, H. L. Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Computers & Chemical Engineering* **2021**, 155, 107513. https://doi.org/10.1016/j.compchemeng.2021.107513
[13] Hasan, M.; Abedin, M.Z.; Hajek, P. et al. A blending ensemble learning model for crude oil price forecasting. *Ann Oper Res* **2024**, 1-31. https://doi.org/10.1007/s10479-023-05810-8
[14] Yu, L.; Wu, Y.; Tang, L. et al. Investigation of diversity strategies in RVFL network ensemble learning for crude oil price forecasting. *Soft Comput* **2021**, 25, 3609–3622. https://doi.org/10.1007/s00500-020-05390-w
[15] Chakraborty, D. et al. Computational solar energy–Ensemble learning methods for prediction of solar power generation based on meteorological parameters in Eastern India. *Renewable energy focus* **2023**, 44, 277-294. https://doi.org/10.1007/s10479-023-05810-8
[16] Tina, G.M.; Ventura, C.; Ferlito, S.; De Vito, S. A State-of-Art-Review on Machine-Learning Based Methods for PV. *Appl. Sci.* **2021**, 11, 7550. https://doi.org/10.3390/app11167550
[17] Burger, E. M.; Scott, J. M. Gated ensemble learning method for demand-side electricity load forecasting. *Energy and Buildings* **2015**, 109, 23-34.
[18] Baker, M.R. Uncertainty management in electricity demand forecasting with machine learning and ensemble learning: case studies of COVID-19 in the US metropolitans. *Engineering Applications of Artificial Intelligence* **2023**, 123, 106350.
[19] Chen, K. et al. A novel data-driven approach for residential electricity consumption prediction based on ensemble learning. *Energy* **2018**, 150, 49-60.
[20] Wang, Z.; Yueren, W.; Ravi, S. Srinivasan. A novel ensemble learning approach to support building energy use prediction. *Energy and Buildings* **2018**, 159, 109-122.
[21] Weeraddana, D.; Nguyen, L. D. K.; Nariman, M. Machine learning based novel ensemble learning framework for electricity operational forecasting. *Electric Power Systems Research* **2021**, 201, 107477.
[22] Manfre Jaimes, D.; Zamudio López, M.; Zareipour, H.; Quashie, M. A Hybrid Model for Multi-Day-Ahead Electricity Price Forecasting considering Price Spikes. *Forecasting* **2023**, 5, 499-521. https://doi.org/10.3390/forecast5030028
[23] Rahul, K. A.; Frankle, M.; Madan M. T. Ensemble of relevance vector machines and boosted trees for electricity price forecasting. *Applied Energy* **2019**, 250, 540-548. https://doi.org/10.1016/j.apenergy.2019.05.062
[24] Ushagra, B.; Rajat, M.; Jyothi V.; Tripathi, M.M. An ensemble approach for electricity price forecasting in markets with renewable energy resources. *Utilities Policy* **2021**, Volume 70, 101185. https://doi.org/10.1016/j.jup.2021.101185
[25] Sewell, M. Ensemble learning. *RN* **2008**, 11.02, 1-34.
[26] Sun, S.; Yunjie W.; Shouyang W. AdaBoost-LSTM ensemble learning for financial time series forecasting. Computational Science–ICCS 2018: 18th International Conference, Wuxi, China, June 11–13, 2018.
[27] Wu, Y.; Jianwei, G. AdaBoost-based long short-term memory ensemble learning approach for financial time series forecasting. *Current Science* **2018**, 115.1, 159-165.
[28] Busari, G. A.; Nae, W. K.; Dong, H. L. An application of AdaBoost-GRU ensemble model to economic time series prediction. *Indian Journal of Science and Technology* **2021**, 14.31, 2557-2566.
[29] Kyeremeh, F. et al. Solar PV power forecasting with a hybrid LSTM-AdaBoost ensemble. IEEE/IET International Utility Conference and Exposition (IUCE). IEEE, 2022.
[30] Aurav, K.; Nuttanan, W. Electricity price forecasting in New Zealand: A comparative analysis of statistical and machine learning models with feature selection. *Applied Energy* **2023**, 347, 121446. https://doi.org/10.1016/j.apenergy.2023.121446
[31] Mounir, N.; Hamid, O.; Ismael, J. Short-term electric load forecasting using an EMD-BI-LSTM approach for smart grid energy management system. *Energy and Buildings* **2023**, 288, 113022.
[32] Nti, I.K.; Adekoya, A.F.; Weyori, B.A. A comprehensive evaluation of ensemble learning for stock-market prediction. *J Big Data* **2020**, 7, 20. https://doi.org/10.1186/s40537-020-00299-5
[33] Kankanala, P.; Das, S.; Pahwa, A. AdaBoost+: An Ensemble Learning Approach for Estimating Weather-Related Outages in Distribution Systems, *IEEE Transactions on Power Systems* **2014**, 29.1, 359-367.
[34] Deoras, A. Electricity Load and Price Forecasting with MATLAB. Mathwork's Webinar Case Study. Available online: https://www.mathworks.com/matlabcentral/fileexchange/28684-electricity-load-and-price-forecasting-webinar-case-study. (accessed on 1 March 2024).