

Article

Hybrid Long Short-Term Memory Wavelet Transform Models for Short-Term Electricity Load Forecasting

Agbassou Guenoukpati ^{1,2,*}, Akuété Pierre Agbessi ^{1,2}, Adekunlé Akim Salami ^{1,2} and Yawo Amen Bakpo ¹ 

¹ Centre d'Excellence Régional pour la Maîtrise de l'Électricité (CERME), Université de Lomé, Lome P.O. Box 1515, Togo; pierreagbessi2@gmail.com (A.P.A.); akim_salami@yahoo.fr (A.A.S.); amenbakpo@gmail.com (Y.A.B.)

² Laboratoire de Recherche en Sciences de l'Ingénieur (LARSI), Department of Electrical Engineering, Ecole Polytechnique de Lomé (EPL), Université de Lomé, Lome P.O. Box 1515, Togo

* Correspondence: guenoukpatib@gmail.com; Tel.: +228-91-48-55-83

Abstract: To ensure the constant availability of electrical energy, power companies must consistently maintain a balance between supply and demand. However, electrical load is influenced by a variety of factors, necessitating the development of robust forecasting models. This study seeks to enhance electricity load forecasting by proposing a hybrid model that combines Sorted Coefficient Wavelet Decomposition with Long Short-Term Memory (LSTM) networks. This approach offers significant advantages in reducing algorithmic complexity and effectively processing patterns within the same class of data. Various models, including Stacked LSTM, Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network—Long Short-Term Memory (CNN-LSTM), and Convolutional Long Short-Term Memory (ConvLSTM), were compared and optimized using grid search with cross-validation on consumption data from Lome, a city in Togo. The results indicate that the ConvLSTM model outperforms its counterparts based on Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and correlation coefficient (R^2) metrics. The ConvLSTM model was further refined using wavelet decomposition with coefficient sorting, resulting in the WT+ConvLSTM model. This proposed approach significantly narrows the gap between actual and predicted loads, reducing discrepancies from 10–50 MW to 0.5–3 MW. In comparison, the WT+ConvLSTM model surpasses Autoregressive Integrated Moving Average (ARIMA) models and Multilayer Perceptron (MLP) type artificial neural networks, achieving a MAPE of 0.485%, an RMSE of 0.61 MW, and an R^2 of 0.99. This approach demonstrates substantial robustness in electricity load forecasting, aiding stakeholders in the energy sector to make more informed decisions.



Citation: Guenoukpati, A.; Agbessi, A.P.; Salami, A.A.; Bakpo, Y.A. Hybrid Long Short-Term Memory Wavelet Transform Models for Short-Term Electricity Load Forecasting. *Energies* **2024**, *17*, 4914. <https://doi.org/10.3390/en17194914>

Academic Editor: Ahmed Abu-Siada

Received: 15 August 2024

Revised: 11 September 2024

Accepted: 15 September 2024

Published: 30 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Over the past few decades, electrical load forecasting has gained increasing importance in the electricity sector. This topic has become particularly crucial due to the growing demand for electricity and the need to ensure efficient management of power grids. This ever-evolving technique allows us to anticipate variations in energy consumption, a critical issue for infrastructure adaptation, and short-, medium-, and long-term planning.

The hierarchical analysis in [1] underscores the importance of proper preparation in electricity exchanges, ensuring efficient and sustainable management of energy resources in a global context. Non-linear and hybrid methods, such as those developed in [2,3], emphasize the significance of combining different approaches to improve forecasting accuracy. Furthermore, the integration of artificial intelligence technologies has significantly enhanced the ability of forecasting systems to handle uncertainty [4,5]. Additionally, specific techniques, such as residual modification in Autoregressive Integrated Moving Average (ARIMA) models for seasonal forecasts, have also shown promising results [6]. Electricity network planners rely on these forecasting methods to maintain grid stability

and optimize operating costs. Anticipating demand more accurately not only helps prevent overloads and blackouts but also reduces costs associated with power generation and distribution. The hybrid approaches and adaptive models proposed by [7,8] demonstrate the effectiveness of mixed methods in multi-stage forecasting. Moreover, these strategic forecasts help mitigate the risk of outages and ensure a reliable electricity supply, as demonstrated by studies in [9,10].

Studies have shown that a 1% prediction error can cost millions of US dollars for power utilities [1,11–15]. In the literature, machine learning methods such as artificial neural networks, support vector machines, and recurrent neural networks are commonly used to address this issue [16–19]. Unfortunately, artificial neural networks suffer from generalization problems [19]. Support vector machines are inefficient with few training datasets and present algorithmic complexity that increases rapidly with the size of the data [16–18]. With a long-sequence dataset, recurrent networks suffer from explosion and vanishing gradients [20,21]. Long Short-Term Memory (LSTM) networks offer an alternative to this problem [22], and perform well with large datasets. However, they face the issue of overfitting due to sensitivity to missing data and the training data size, which sometimes leads to costly computation times and significant resource consumption during hyperparameter optimization. Alternatively, wavelet transforms can make LSTMs more suitable for compression and feature extraction from nonstationary data. Recent contributions have been made in this context for forecasting purposes in the electricity sector [16,23,24].

In our case, the aim is to propose new approaches that combine Deep Learning and data preprocessing techniques to develop robust models. The objective is to integrate wavelet coefficients with sorting, combined with Stacked LSTM, Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM), and Convolutional Long Short-Term Memory (ConvLSTM), to predict electricity demand. The main reasons for selecting these models are (i) their ability to learn complex relationships, (ii) their flexibility in predicting spatiotemporal data, and (iii) their performance with large amounts of training data. The use of wavelet transforms with sorted coefficients offers significant advantages: (iv) extraction of relevant information by denoising, (v) reduction in bias and variance errors, and (vi) improved processing of patterns within the same data class.

Although wavelet decomposition and LSTM networks have been explored in the literature [25], the specific approach described here, which integrates wavelet decomposition with coefficient sorting and the ConvLSTM model, provides novel elements. The promising results and improved performance compared to existing models indicate a significant contribution and open up prospects for future research and practical applications.

The rest of the paper is organized as follows: Section 2 presents study contributions; Section 3 discusses the methodology of the proposed approach in detail and outlines the mathematical formalisms. The results and discussion of the proposed model are presented in Section 4, followed by conclusions, recommendations, and policy perspectives in Section 5.

2. Contribution of the Study

Electrical load is influenced by a multitude of non-linear and interdependent factors, such as weather conditions, economic cycles, and special events. This complexity necessitates models that can capture these intricate and dynamic relationships, thereby justifying the use of sophisticated methods like recurrent neural networks (RNN) and LSTM. These methods are specifically designed to handle non-linear time series and to capture long-term dependencies between variables. Indeed, machine learning methods, such as LSTMs, have demonstrated superior accuracy compared to traditional time series methods (ARIMA, ARMAX, ...) and artificial neural networks (ANN) as evidenced by several comparative studies [26–29].

Furthermore, improved accuracy is crucial to avoid forecasting errors that could lead to critical imbalances in the power grid. Machine learning models, while demanding in terms of data and computation, offer robustness in the face of volatile and noisy data, which is essential for reliable forecasts in the dynamic environment of electrical load management. Additionally, the ability of these models to adapt to increasingly large datasets and incorporate new variables without requiring a complete redesign is a key selection criterion. However, methods such as Support Vector Machines (SVMs), although less complex than LSTMs, can quickly become impractical with large volumes of data due to their exponential training time.

Although sophisticated models like LSTMs offer high accuracy, they also introduce significant computational complexity. It is therefore essential to strike a balance between the need for accuracy and the complexity of implementation. Forecast optimality, however, is not solely about accuracy; it also includes the ability to generate forecasts within the reasonable timeframes required for real-time decision-making. As a result, it may be necessary to select methods that, while potentially less accurate, allow for faster and more economical implementation, ensuring an optimal compromise between accuracy and complexity. In this context, wavelets, with their efficiency in signal processing and mathematical analysis, can become powerful tools for enhancing LSTM networks. Consequently, the impact of this study on electrical load management decision-making is manifested in a significant improvement in network stability and efficiency.

This study aimed to improve the electric load prediction methods identified above. The use of sorted wavelet coefficients combined with LSTM variants was explored to forecast electric load. Four models, Stacked LSTM, BiLSTM, CNN-LSTM, and Conv-LSTM, were employed with hyperparameter optimization. The selected optimal model was then combined with the wavelet transform using sorted coefficients to predict the electrical load of the city of Lome. This approach offers significant advantages, such as extracting relevant features per data class, dimensionality reduction, robustness, and algorithmic complexity reduction, to better understand and model the complex variations in electrical loads, enabling more reliable and efficient predictions. This involves the introduction of a novel approach that has not yet been applied to electrical load forecasting, with the contributions outlined below.

- Use of Multiple LSTM Architectures: The research explores four variants of LSTM neural networks, namely Stacked LSTM, BiLSTM, CNN-LSTM, and ConvLSTM.
- Hyperparameter Optimization: An optimization process was implemented to identify the best-performing model among the variants tested.
- Integration of Sorted Wavelet Coefficients: The approach combines sorted wavelet coefficients with the optimal LSTM model to enhance the accuracy of electrical load forecasts.
- Application to the City of Lomé: The model was applied to predict the electrical load of the city of Lomé, demonstrating its effectiveness in a real-world context.

3. Materials and Methods

The diagram in Figure 1 illustrates the complete methodology of the proposed approach. The preprocessing consists of eliminating missing values, smoothing out spurious data, and handling outliers. In our approach, inconsistent values were replaced with the mean of the dataset. This is followed by the optimization step, where the defined hyperparameters are optimized using a grid search algorithm to identify the best configuration. The mean squared error (MSE), as defined in Equation (1), was chosen as the loss function during the training process. Next, the entire dataset is divided into training and test data. The training data are subject to wavelet decomposition. This extracts various sorted components, such as the approximate component (CA_Tri) and the detailed components (CD1_Tri, ..., CDn_Tri). These components are then used to train the optimal model. For the prediction phase, the test data are also decomposed into wavelets to extract the same types of components with the same level of decomposition and the same parent wavelet.

The trained model is then used to make predictions on these decomposed components. The resulting forecasts are recombined using the inverse wavelet transform to reconstruct the predicted time series. Finally, the forecast results are compared with the actual data to make a final assessment of the model's performance.

$$MSE = \frac{1}{N} \sum_{j=1}^N (Y_{j,p} - Y_{j,r})^2 \quad (1)$$

where N is the number of training samples, $Y_{j,p}$ and $Y_{j,r}$ are the predicted and target values, respectively, for each sample. To optimize this loss function, we used the Adam optimizer [30] to train various LSTM models. The Adam (Adaptive Moment Estimation) optimizer is widely used in deep learning for several reasons: the Adam optimizer is chosen for its ability to combine the advantages of different optimization methods, its ability to automatically adjust learning rates, and its stability when training complex models. For an in-depth understanding and details of the optimizer are presented [30]. In this paper, the authors present the theory behind Adam, details of the algorithm, and its effectiveness compared to other optimizers.

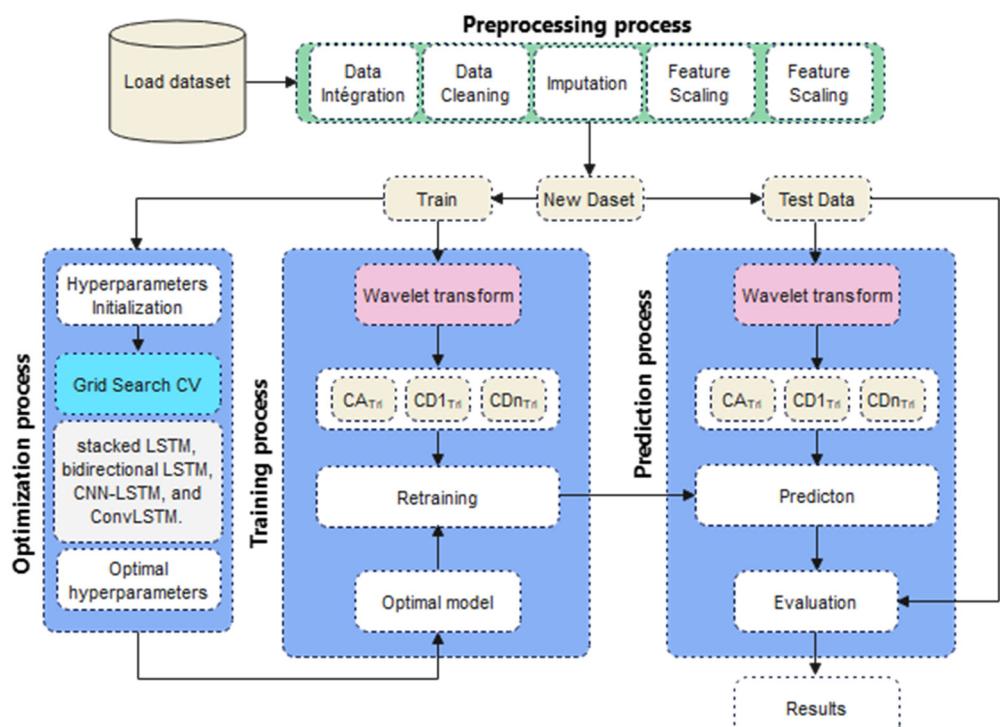


Figure 1. Methodology of the proposed approach.

To avoid long discussions, only the wavelet transform is detailed. For the mathematical formalisms of the other four models, the reader can consult the documentation [31–37].

3.1. Wavelet Decomposition Analyses

This technique decomposes a signal into its various components using the mother wavelet and the decomposition level. Equation (2) is used to represent a function $f(t)$ in terms of basic functions called wavelets [38]. In this relationship, the first term represents the sum of the low scales, and the second is the sum of the high scales. The coefficients of

detail and approximation were obtained using Mallat's algorithm. The original signal was obtained by applying inverse decomposition.

$$f(t) = \sum_{k \in \mathbb{Z}} C_{j_0,k} \phi_{j_0,k}(t) + \sum_{k \in \mathbb{Z}} \sum_{j=0}^{j_1} d_{j,k} \psi_{j,k}(t) \quad (2)$$

where $C_{j_0,k}$ are the scaling coefficients for level j_0 , $\phi_{j_0,k}(t)$ are the scaling functions, often called scale basis functions, which correspond to the low-frequency components of the function $f(t)$; and $d_{j,k}$ are the wavelet coefficients for level j , $\psi_{j,k}(t)$ are the wavelet functions, which represent detail at different scales or resolutions, j is the scale index or decomposition level, and $k:j$ the translation index.

3.2. Models Architectures

LSTM networks are recurrent neural networks [31], used on long data sequences [22]. Variants of LSTM networks exist in the literature, including those with and without a forgetting gate and LSTM with a peephole connection. This article uses LSTM networks with a peephole connection to develop four load prediction models: stacked LSTM, bidirectional LSTM, CNN-LSTM, and ConvLSTM. Indeed, LSTMs with forgetting gates offer the advantage of removing unnecessary information and retaining only essential data during training. By modifying the original LSTM cell and introducing a forgetting gate into the cell [32], this process determines whether information should be retained or eliminated from the cell state. A value of the forgetting gate $f(t)$ close to 1 indicates the retention of information, while a value close to 0 means discarding all information. For stacked LSTMs, LSTM cells are arranged in layers to form a deep network. In this configuration, the number of LSTM cells per layer remains constant. Any deviation from this configuration results in a misconfigured network, which leads to information loss. If N represents the number of LSTM cells in the first layer, this layer outputs N values. If the subsequent layer has a smaller number of cells than the preceding layer, some values are inevitably lost. This contrasts with the role of the forgetting gate, which removes information no longer required by the cell. It is essential to note that the decision to delete or retain information lies in the LSTM cell itself, not the network configuration. The fully connected layer allows adjustment of the number of network outputs. Therefore, with this final layer, it is possible to have $1, 2, 3, \dots, n$ output(s). The Bidirectional LSTM network introduced in [33], operates in two opposite directions. This bidirectional nature enables it to learn data sequences in both directions during training at the network input. Consider a data sequence $[x_{t-1}, x_t, x_{t+1}, \dots, x_T]$; to learn this sequence, the first layer of the BiLSTM network will use the data sequence from x_T to x_{t-1} . The second layer will use the data sequence from x_T to x_{t-1} . The first layer stores past information by executing the data sequence from x_{t-1} to x_T , and the second layer stores future information by executing the data sequence from x_{t-1} vers x_T . The BiLSTM network therefore can store both past and future information, which is impossible for the stacked LSTM network. Figure 2 presents the architecture of the stacked LSTM and BiLSTM, and the input data structure of these models. The input data are initially prepared as a matrix of dimensions $(n, 3)$, and then resized in 3d with a final dimension of $(n, 3, 1)$.

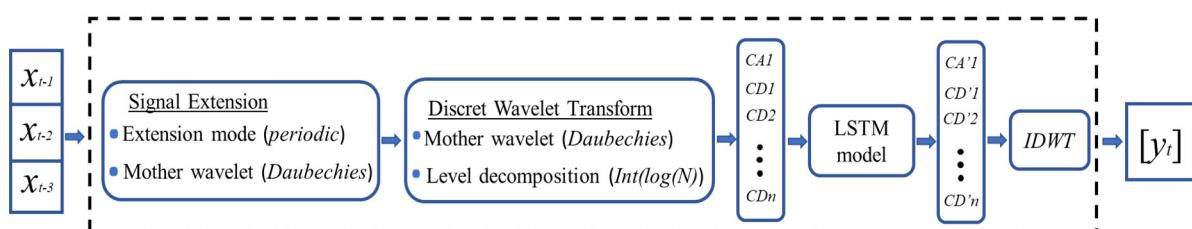


Figure 2. LSTM Stacked and BiLSTM process diagram.

Convolutional neural networks (CNNs) are widely used in the field of computer vision [34] to solve problems such as object identification, image recognition, image classification, image segmentation, and so on. A CNN consists mainly of three layers, namely the convolution layer, the pooling layer, and the fully connected layer. These three layers perform mathematical operations that are necessary for the functioning of a CNN. The ConvLSTM network [35], was designed for nowcasting precipitation and is particularly well-suited for handling spatiotemporal data like images and videos [36,37]. Each input is a three-dimensional spatiotemporal tensor, with the first two dimensions representing spatial dimensions. The matrix product between the input data and the weight matrices in a standard LSTM is replaced by a convolution product. The CNN-LSTM models utilize input data arranged as a matrix of dimensions (n, 4), which is then reshaped into 3d with a final dimension of (n, 2, 2, 1). Figure 3 illustrates the data shape for the CNN-LSTM and ConvLSTM. For the ConvLSTM model, the input data have a dimension of (n, 4), which is then reshaped into 3D with a final dimension of (n, 2, 2).

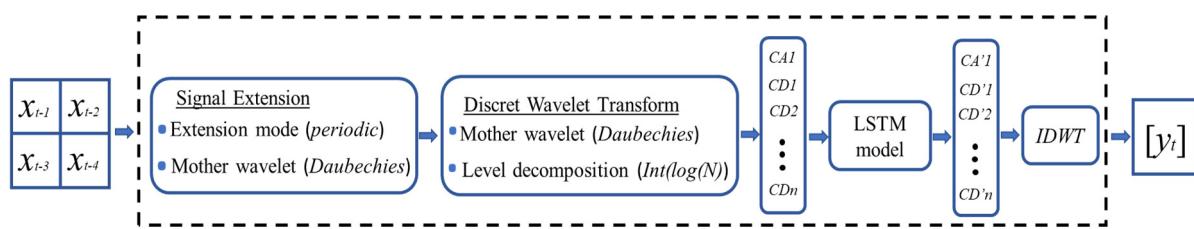


Figure 3. CNN-LSTM and ConvLSTM process diagram.

3.3. Model Evaluation

For the performance evaluation indices like Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Correlation coefficient (R^2). The RMSE criterion measures accuracy by comparing the deviation between the estimated values and the measured data. The RMSE is always positive and is calculated using Equation (3).

$$RMSE = \sqrt{\frac{1}{N} \sum_{j=1}^N (Y_{j,p} - Y_{j,r})^2} \quad (3)$$

where $Y_{j,p}$ represents the estimated values and $Y_{j,r}$ represents the measured values

The MAPE criterion shows the average percentage difference between the estimated values and the measured values. The MAPE is calculated using Equation (4).

$$MAPE = \frac{1}{N} \sum_{j=1}^N \left| \frac{Y_{j,p} - Y_{j,r}}{Y_{j,r}} \right| \quad (4)$$

The correlation coefficient (R^2) indicates the strength of the linear relationship between the predicted values and the observed values. R^2 is given by Equation (5).

$$R^2 = \frac{\sum_{j=1}^N (Y_{j,p} - Y_{p,avg}) \times (Y_{j,r} - Y_{r,avg})}{\sqrt{\left[\sum_{j=1}^N (Y_{j,p} - Y_{p,avg})^2 \right] \times \left[\sum_{j=1}^N (Y_{j,r} - Y_{r,avg})^2 \right]}} \quad (5)$$

where $Y_{p,avg}$ is the mean predicted value and $Y_{r,avg}$ is the mean observed value.

A calculation of signal-to-noise by the ratio between the variance of the signal values and the RMSE squared can be approximated by the inverse of the RMSE squared. The higher the ratio, the better the quality of the signal relative to the noise by Equation (6).

$$S/N = \frac{\sigma_s}{RMSE^2} \approx \frac{1}{RMSE^2} \quad (6)$$

3.4. Modelling Data and Materials

The data are provided by Compagnie Energie Electrique du Togo (CEET) and concerns electricity consumption in Lome. The database used covers a period from 1 January 2018 to 31 December 2023, i.e., six (06) years of data. For the remainder of the work, 80% of the data will be used for training our models and 20% for testing. The database is structured in Excel format, with workbooks comprising 12 sheets corresponding to the different months of the year. Each workbook includes details of electricity consumption. Each of these data points is collected at hourly intervals. In addition, the entire project runs on an Intel(R) Core (TM) i7-4720HQ @ 2.60 GHz processor (4 CPUs) with 16 GB RAM from LENOVO. Each of these models was developed using Python 3.12.

4. Results and Discussion

The key hyperparameters for LSTM networks include input data size, output data size, number of hidden layers, activation function, learning rate, optimizer, epoch, batch size, neuron deactivation, and the number of LSTM cells. Each of these hyperparameters significantly influences the performance of LSTM networks. Figures 4–7 depict the effects of hidden layers, activation function, epoch, and the number of LSTM cells on each model.

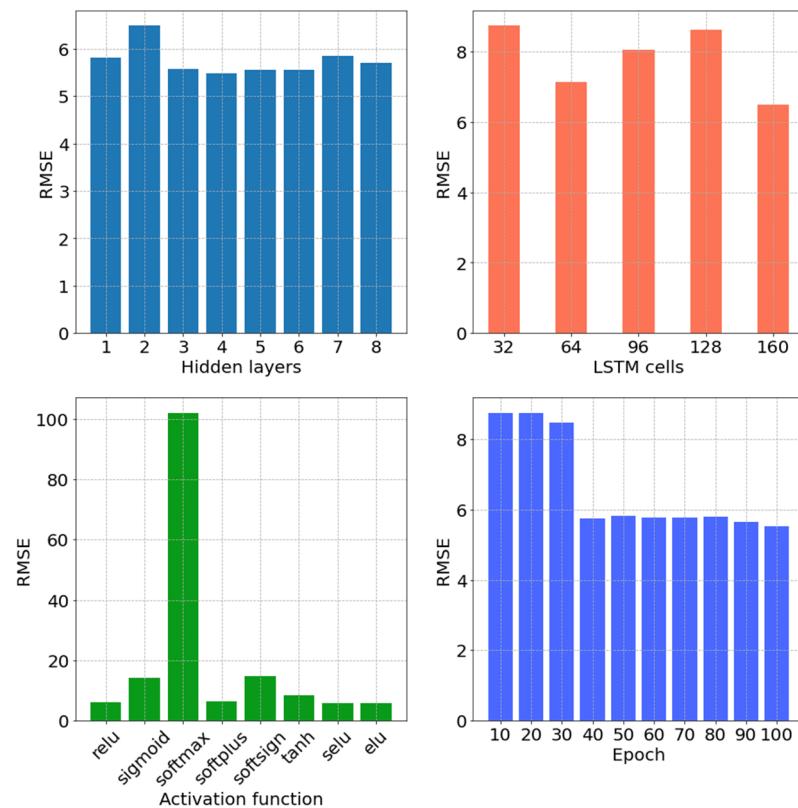


Figure 4. Hyperparamters effects on the Stacked LSTM model.

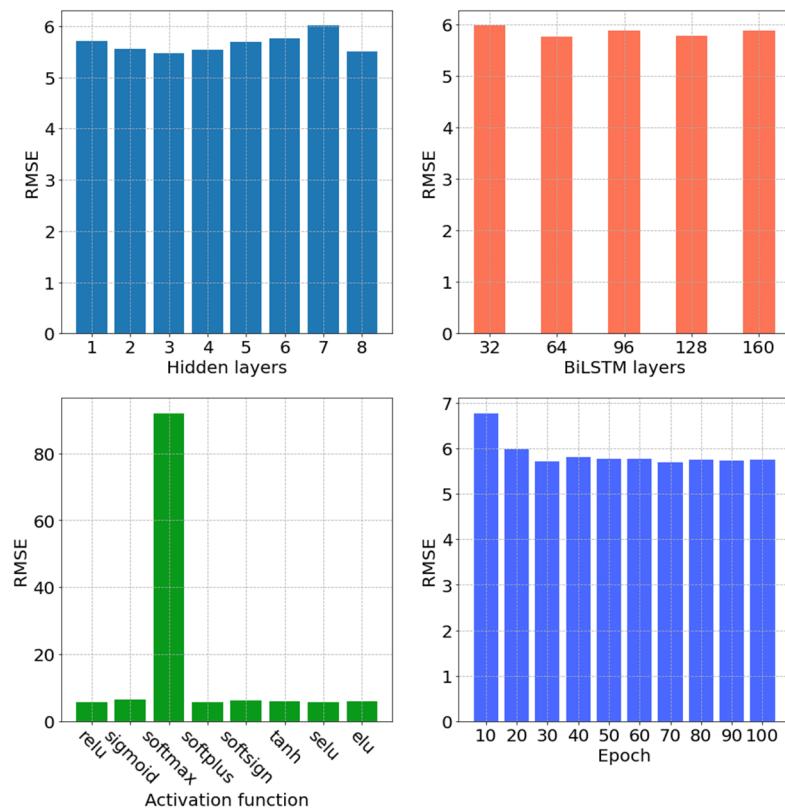


Figure 5. Hyperparamters effects on the BiLSTM model.

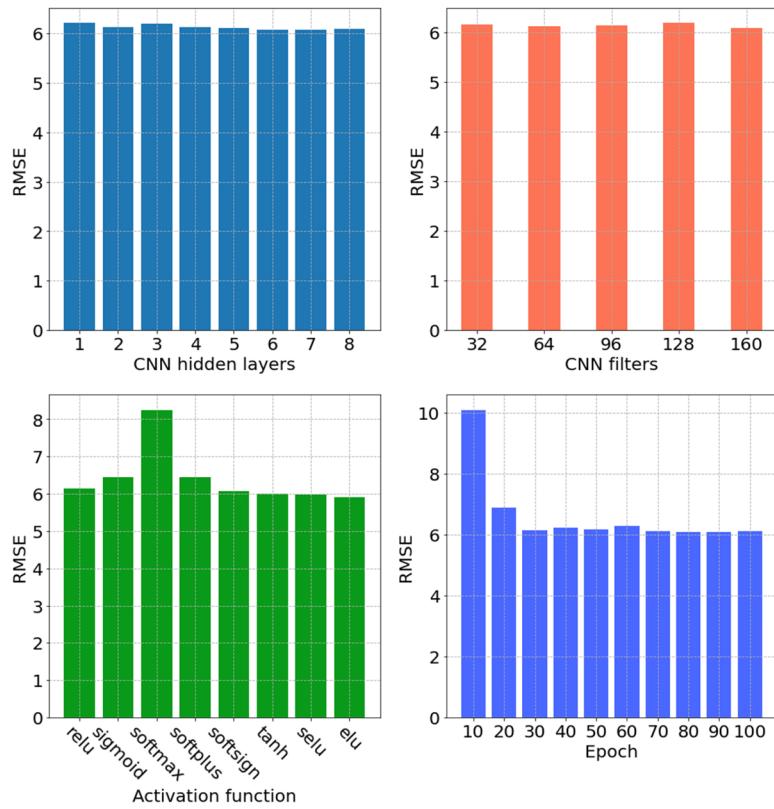


Figure 6. Hyperparamters effects on the CNN-LSTM model.

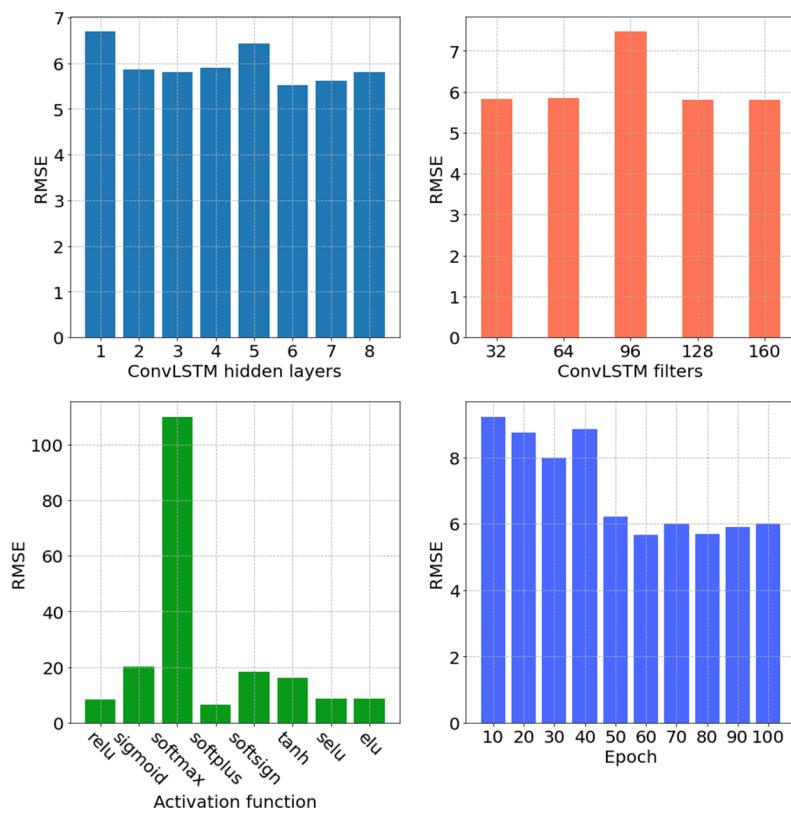


Figure 7. Hyperparameters effects on the ConvLSTM model.

Figure 4 reveals that the RMSE increases between the first and second hidden layers. It stabilizes from the third to sixth hidden layers and a slight increase for the seventh and eighth hidden layers. The number of LSTM cells that give the lowest RMSE value is 160 cells. The results of the effect of the activation function on RMSE show that only the “softmax” function gives a very high RMSE value compared with the other activation functions. The minimum RMSE values were obtained with the “relu, softplus, selu, and elu” functions. Finally, the RMSE of Stack LSTM decreases from 10 to 100 epochs. It is recommended to choose a training epoch greater than or equal to 100.

In Figure 5, it is evident that the RMSE value experiences an increase between the third and seventh hidden layers. Additionally, RMSE values exhibit slight variations with the number of BiLSTM layers, with the lowest RMSE observed at 64 BiLSTM cells. The impact of activation functions on RMSE reveals that only the “softmax” activation function results in a notably high RMSE compared to other activation functions. Conversely, minimum RMSE values were obtained with “relu, sigmoid, softplus, softsign, tanh, selu, and elu” activation functions. For the BiLSTM model, the RMSE decreased from 10 to 30 epochs and remained virtually constant from 40 to 100 epochs.

Figure 6 indicates that the RMSE value remained practically constant when the number of CNN layers varied from one to eight. Slight variations in the RMSE value were observed owing to the changes in the number of CNN filters. The analysis of the activation function’s effect on RMSE indicates that the softmax function leads to a significantly higher RMSE compared to other functions. Conversely, the minimum RMSE values were achieved with the “relu, softsign, tanh, selu, and elu” activation functions. For the CNN-LSTM model, the RMSE sharply decreased from 10 to 20 epochs and stabilized from 30 to 100 epochs.

Figure 7 reveals that the minimum RMSE occurs with six ConvLSTM layers. The RMSE remained constant for 32, 64, 128, and 160 filters, whereas only 96 ConvLSTM filters resulted in a notably high value compared to other filter numbers. Additionally, the impact of activation functions on RMSE highlights that only the “softmax” function yields a considerably higher RMSE, while minimum RMSE values are associated with the “relu,

softplus, selu, and elu” functions. Finally, the optimal minimum RMSE value was obtained using 60 epochs.

4.1. Hyperparameters Optimization

The algorithms commonly used to optimize hyperparameters LSTM networks include grid search and random search. Grid search involves testing all possible combinations of candidate hyperparameter values to determine the best combination. However, it faces challenges in high-dimensional search spaces, where the search space expands with the number of hyperparameters [39]. Nevertheless, parallelization can be applied to mitigate this issue. The random search involves randomly selecting hyperparameter values from a specified list and testing various combinations. According to [40], random search is more efficient than grid search when the search space is reduced. However, for high-dimensional search spaces, grid search is often preferred. In summary, the choice between grid search and random search depends on the complexity and dimensionality of the hyperparameter search space. In this study, grid search is employed for hyperparameter optimization. This algorithm, however, tends to be memory-intensive. To mitigate this, the study focuses on optimizing four fixed hyperparameters. The process is parallelized to enhance search time efficiency. For each network type (stacked LSTM, BiLSTM, CNN-LSTM, and Conv-LSTM), the “GridSearchCV” class from the “Scikit-learn” module was used. This class integrates grid search with cross-validation. Cross-validation involves dividing the training dataset into k folds, optimizing hyperparameters on one-fold, and training the network on the remaining $k - 1$ folds. The parameterization of the “GridSearchCV” class aligns with the specifications outlined in Equation (7).

$$\text{grid} = \text{GridSearchCV}(\text{model}, \text{parameters}, \text{cv} = 3, \text{n_jobs} = -1) \quad (7)$$

where: model defines the model to be optimized; parameters are the list of hyperparameters to be optimized; cv: number of folds fixed at three in this context; n_jobs are the number of processors used for the search ($n_{\text{jobs}} = -1$ allows all available processors to be used to speed up the search). The optimized hyperparameter results and their codifications used to elaborate some configurations are given in Table 1 and Figure 8, respectively.

Table 1. Optimized hyperparameters of the selected model.

Models	Stacked LSTM & BiLSTM	CNN-LSTM	Conv-LSTM
Number of hidden layer	[1, 2, 3]	[1, 2, 3] CNN + one (01) layer LSTM with 32 LSTM	[1, 2, 3]
Number of LSTM cell	[32, 64, 96]	Filter 1: [[96], [64], [32]] Filter 2: [[96, 64], [64, 32]] Filter 3: [[96, 64, 32]]	Filter 1: [[96], [64], [32]] Filter 2: [[96, 64], [64, 32]] Filter 3: [[96, 64, 32]]
Activation function Epoch	[relu, selu, elu] [50, 100]	[relu, selu, elu] [50, 100]	[relu, selu, elu] [50, 100]

The training curves of the different forecasting models are depicted in Figure 9. Analysis of these curves reveals a scenario of a good fit, characterized by a training loss (loss) and validation loss (val_loss) decreasing to a stable point with a minimal gap (almost equal to zero) between the two final loss values. These results indicate that the training can be extended further, as there is no sign of overfitting yet. Therefore, the training epoch is configured to 500 (epochs = 500), and early stopping is implemented to prevent overtraining.

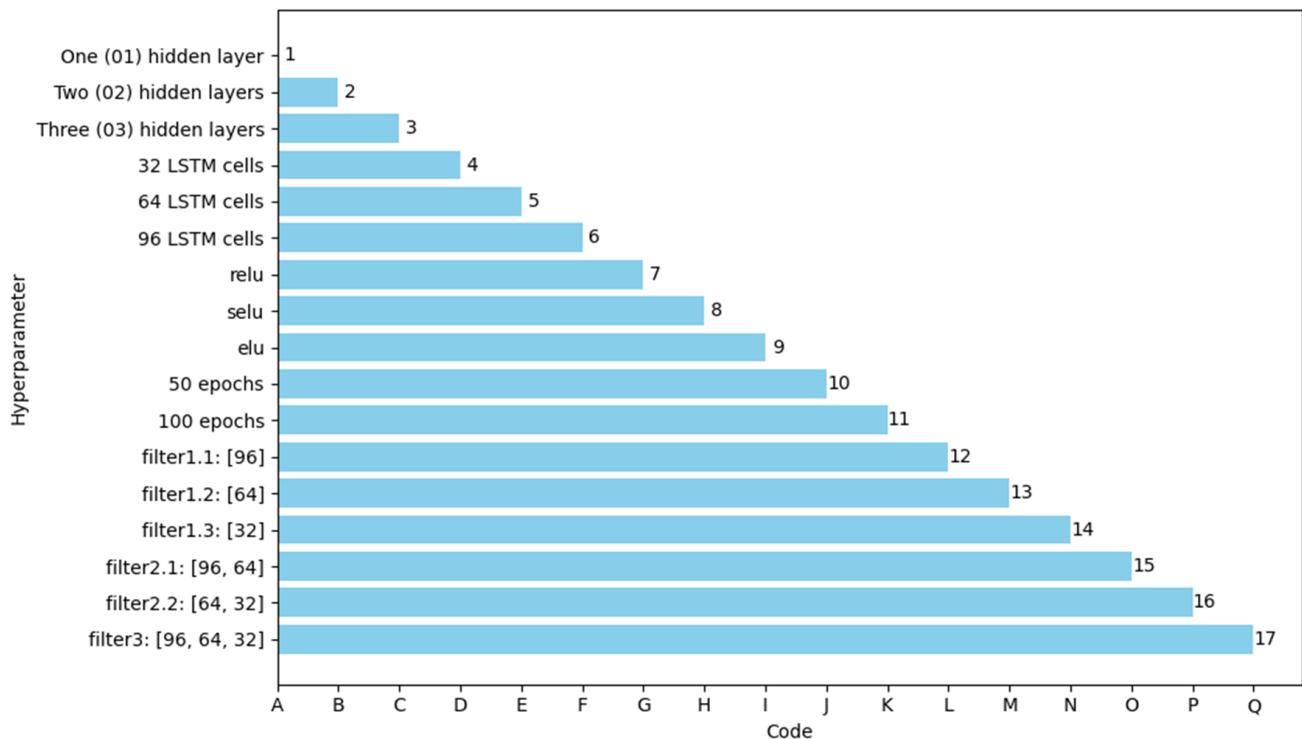


Figure 8. Hyperparameters coding.

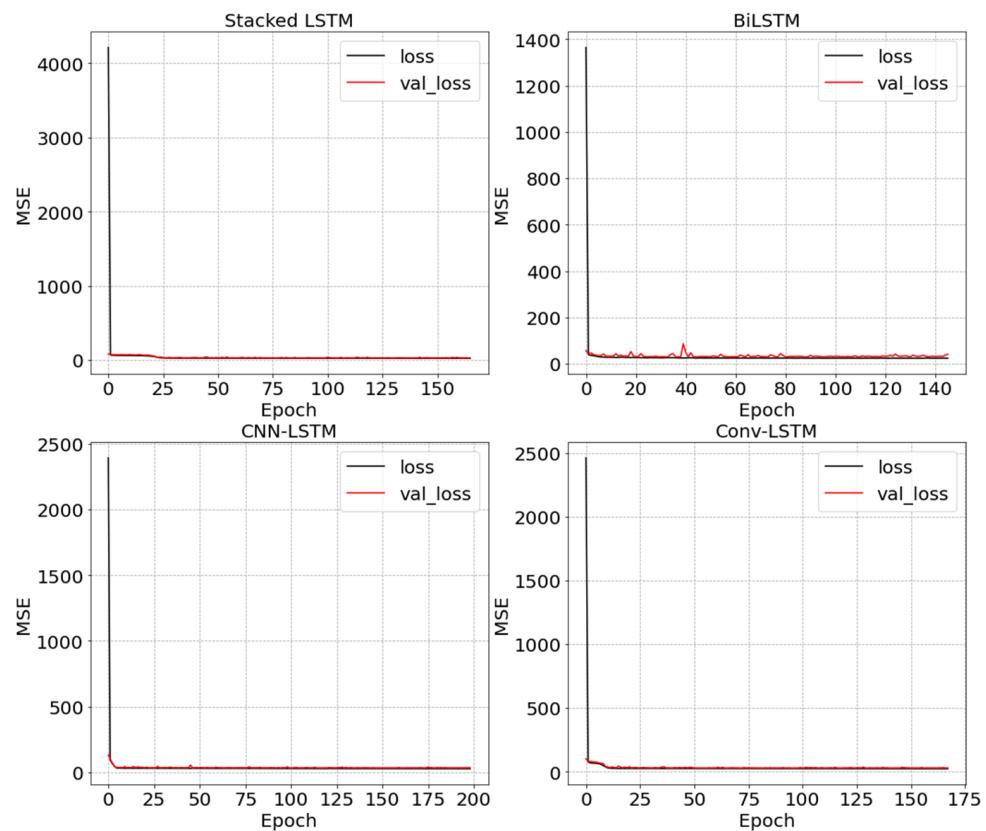


Figure 9. Training curves.

4.2. Optimal Models Training

For the Stacked LSTM, the optimal model was obtained with the configuration [C K I D] consisting of three (03) hidden layers, 100 training epochs, the activation function elu, and 32 LSTM cells. The optimal BiLSTM configuration was [C K H E], with three (03) hidden layers, 100 training epochs, activation function selu, and 64 LSTM cells. For the optimal CNN-LSTM configuration [B K H O], the model is obtained with two (02) hidden layers, 100 training epochs, the activation function selu, 96 filters on the first CNN layer, and 64 filters on the second CNN layer. The optimal ConvLSTM configuration is [C K H Q], obtained with three (03) hidden layers, 100 training epochs, the activation function relu, 96 filters on the first convolutional layer, 64 filters on the second convolutional layer, and 32 filters on the third convolutional layer. The various trained models are used to predict electricity consumption over a one-hour time horizon. The MSE for these models are presented in Table 2.

Table 2. Optimized models.

Models	Configuration	MSE
Stacked LSTM	[C K I D]	−25.039926
BiLSTM	[C K H E]	−24.380357
CNN-LSTM	[B K H O]	−28.554211
Conv-LSTM	[C K H Q]	−24.787564

The correlation between predicted and actual load is presented according to Figure 10. Figure 11 displays the forecast results for the first seven days of 2023. It shows an overview of the forecasting results achieved by all these models. The correlation results between actual and predicted load are depicted. The criteria used to evaluate model prediction results include the MAPE, the RMSE, and the coefficient of determination (R^2). Based on these criteria, the most accurate model will be the one with the lowest MAPE, lowest RMSE, and highest R^2 . Moreover, the model accuracy also depends on finding a compromise with the algorithms used. The complexity of each prediction model is evaluated by measuring the average time taken to predict an output value. This average prediction time is inherently linked to the mathematical operations executed within each model and is influenced by the characteristics of the computer used, such as the number and frequency of the CPU. The results presented in Figure 12, indicate that the fastest model is BiLSTM, with an average prediction time of 0.13 s for one output value. The slowest model is ConvLSTM, taking an average of 0.23 s. Despite the speed differences, all models yielded satisfactory results.

It is noteworthy that the ConvLSTM model while being the most accurate, is also the slowest. This highlights the trade-off between accuracy and prediction speed. In terms of model ranking, both Stacked LSTM and CNN-LSTM are notable for their accuracy and speed. It is worth mentioning that electric charge prediction models are at times deployed on the web for public access. Under these conditions, the Stacked LSTM and CNN-LSTM networks are suitable, as the calculation time of the forecasting model is very high when operating on remote servers. For electricity market applications, the ConvLSTM model is suitable, as at this level 1% forecast error can lead to increased operational costs.

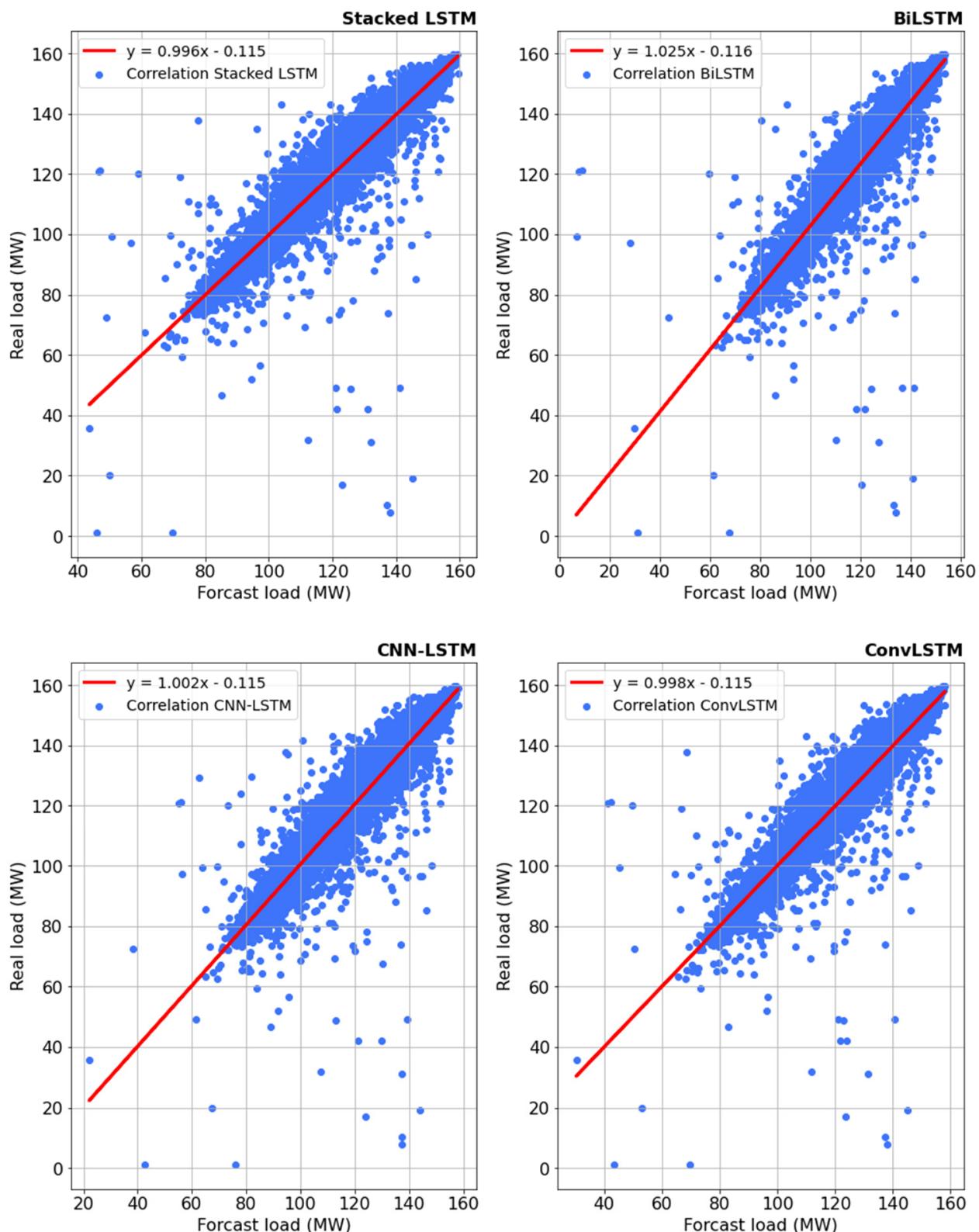


Figure 10. Correlation between the predicted and the actual load using different models.

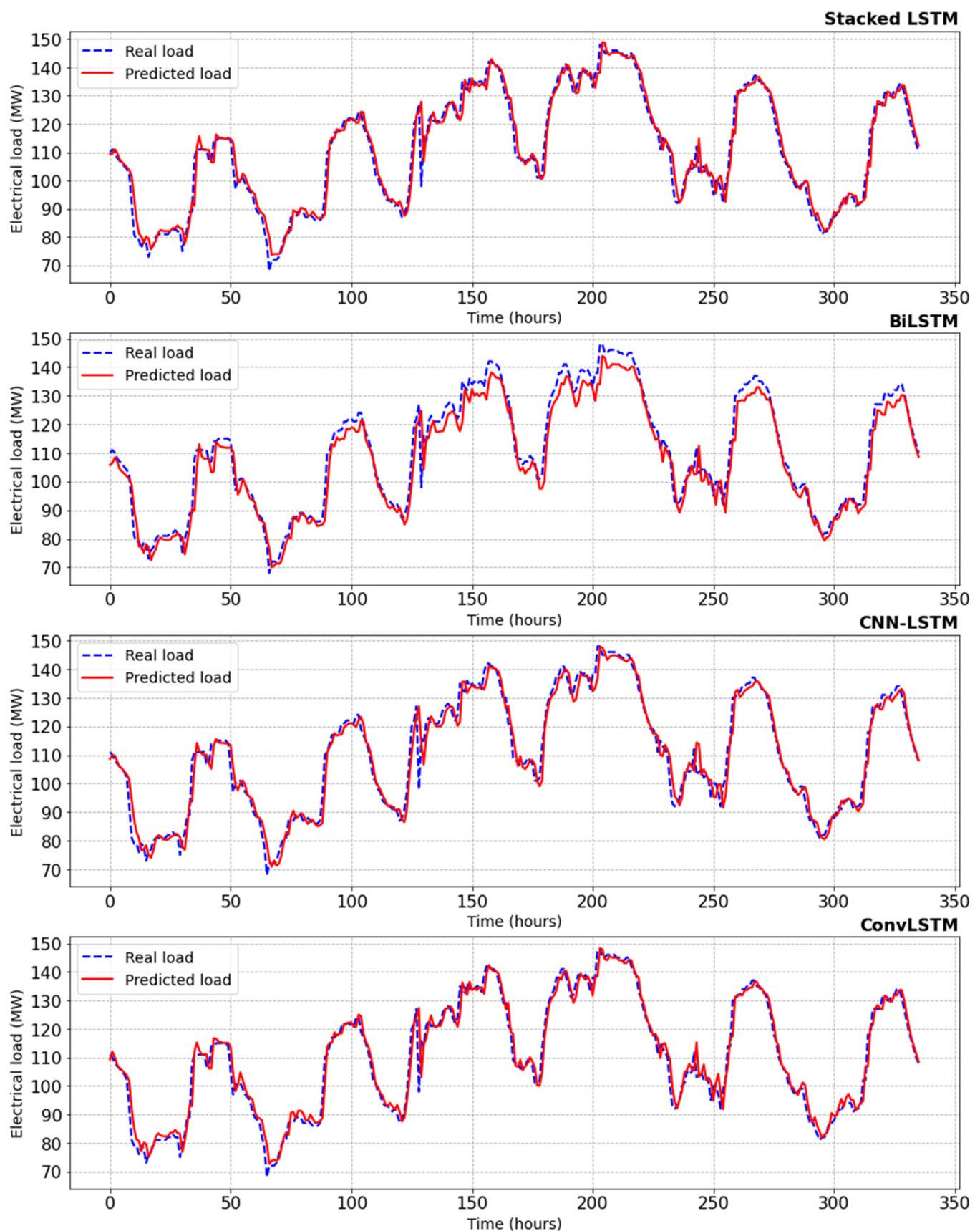


Figure 11. Electric load forecasting results using different models.

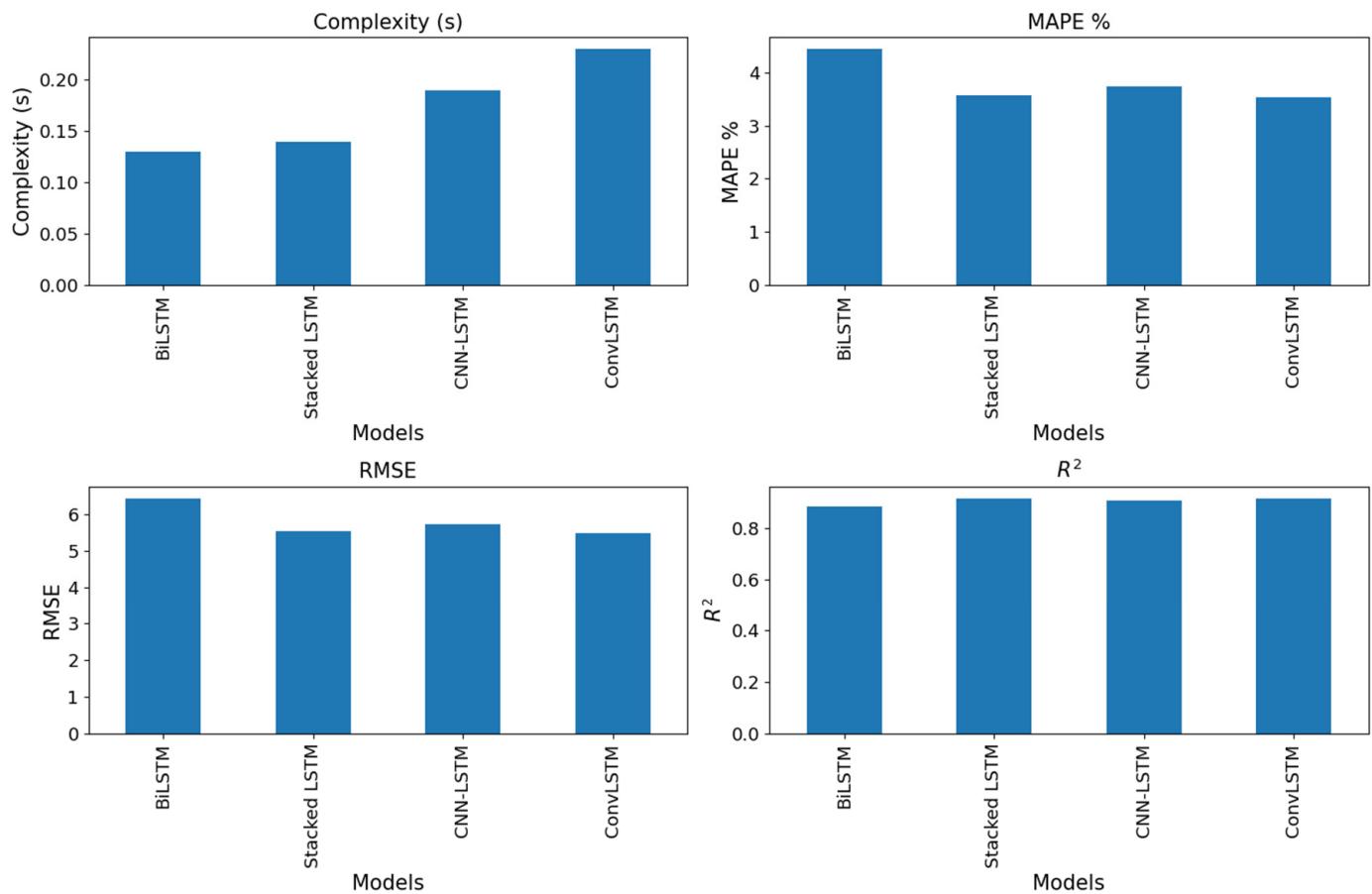


Figure 12. Comparison of forecasting models.

4.3. Effect of Wavelet Transform

We considered the best model ConvLSTM for the rest of the study. Various steps are described as follows: the electric charge is decomposed into approximation and detail coefficients using wavelet transform. The results are sorted in ascending or descending order. The sorted values are used to train the model. The final model is used to predict the approximation and detail coefficients. The sorted coefficients are then transformed into their initial states. The predicted load is finally used to reconstruct with these coefficients. This approach offers several advantages, such as reduced data dimensionality, highlighting the most significant signal components, potentially improving prediction performance, and leading to less costly computation time etc. In the literature, orthogonal wavelets are adapted for time series analysis [24,41–43]. The fourth-order Daubechies mother wavelet was employed. The maximum decomposition level is given by Equation (8), where N represents the number of samples of the time series [44–46]. In our context, the decomposition level is 4.

$$L = \text{int}[\log(N)] \quad (8)$$

The results of the decomposition as well as those with the unsorted coefficients are depicted in Figure 13. Training with the sorted coefficients of the multi-level decomposition led to an enhancement in the performance of the ConvLSTM network. The coefficients predicted by the ConvLSTM model in sorted form and subsequently reconstructed are illustrated in Figure 14.

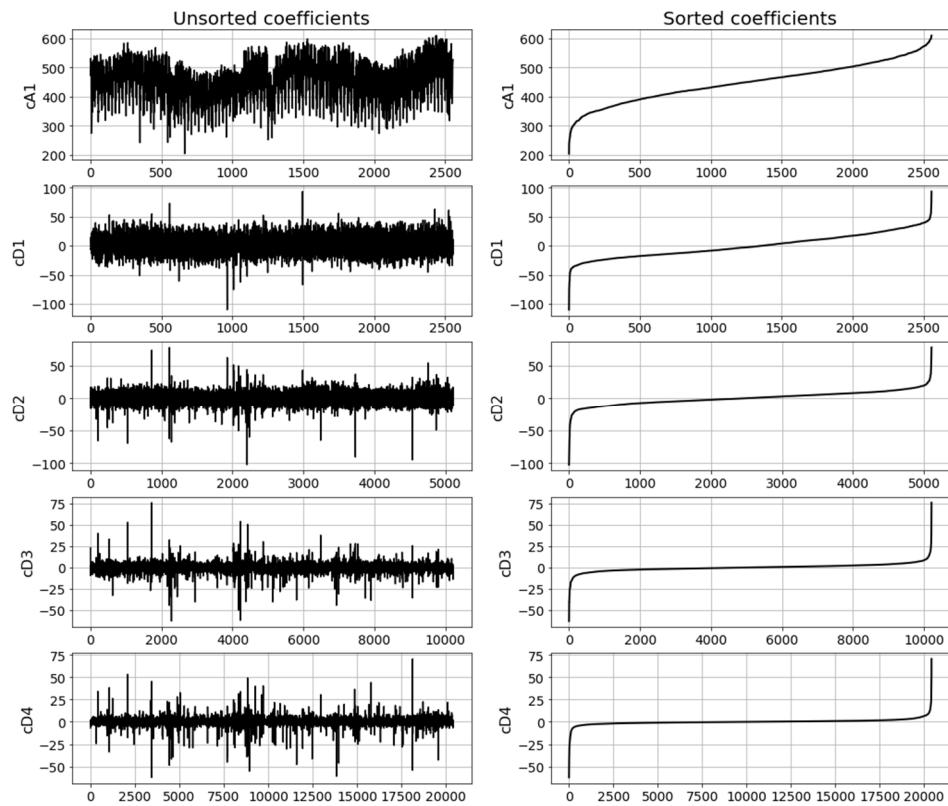


Figure 13. Wavelet transform coefficients.

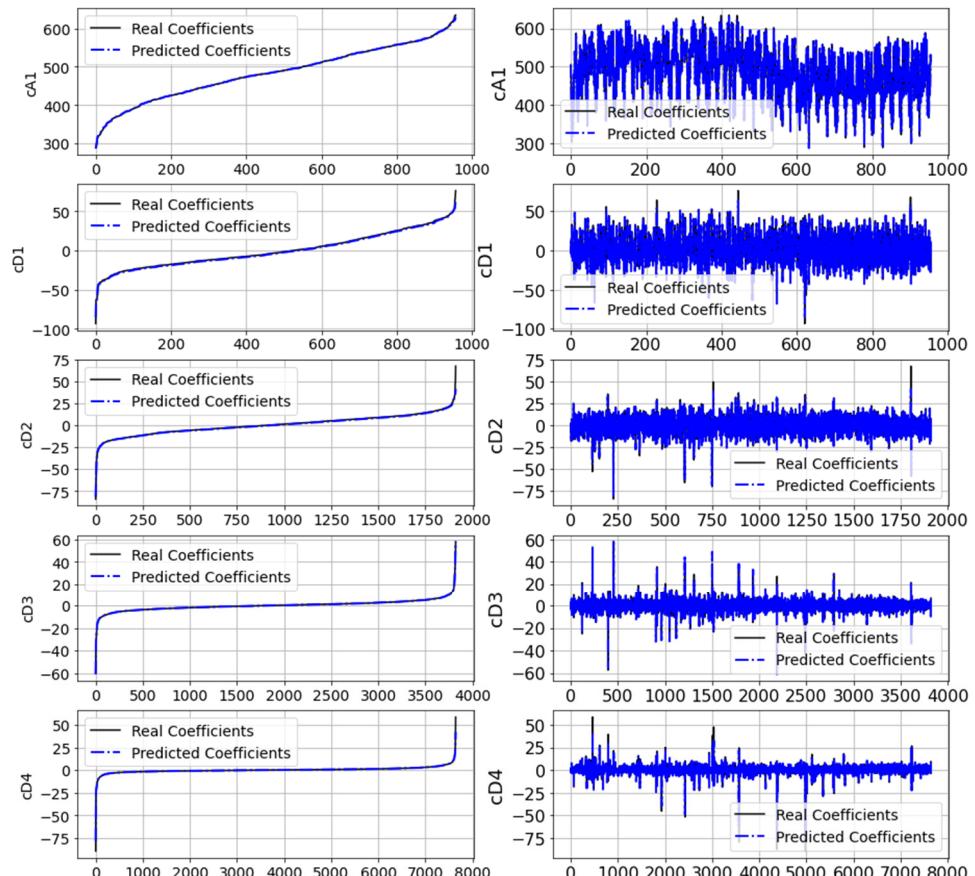


Figure 14. Wavelet transform sorted coefficients predicted.

Figure 15 illustrates the evolution of load profiles over time with unsorted coefficients. The analysis reveals a large discrepancy between actual load and estimated forecast load in absolute error values between 10 and 50 MW.

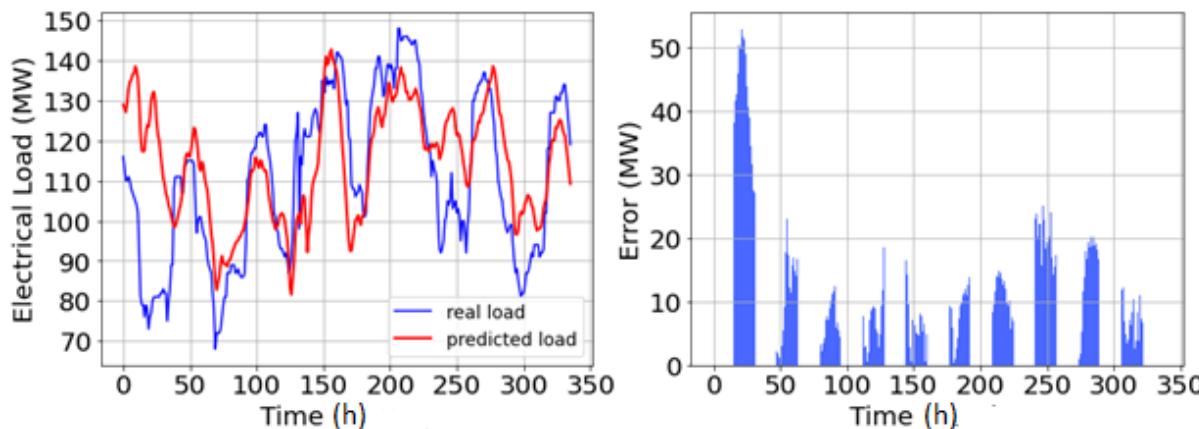


Figure 15. Actual and predicted loads without sorted wavelet coefficient.

Compared au model with wavelet coefficients sorting, the prediction results are shown in Figure 16. The analysis reveals that this technique improves the performance of the ConvLSTM model. The proposed model offers a better match with the CEET electrical load data with absolute error ranging between 0.5 and 3 MW.

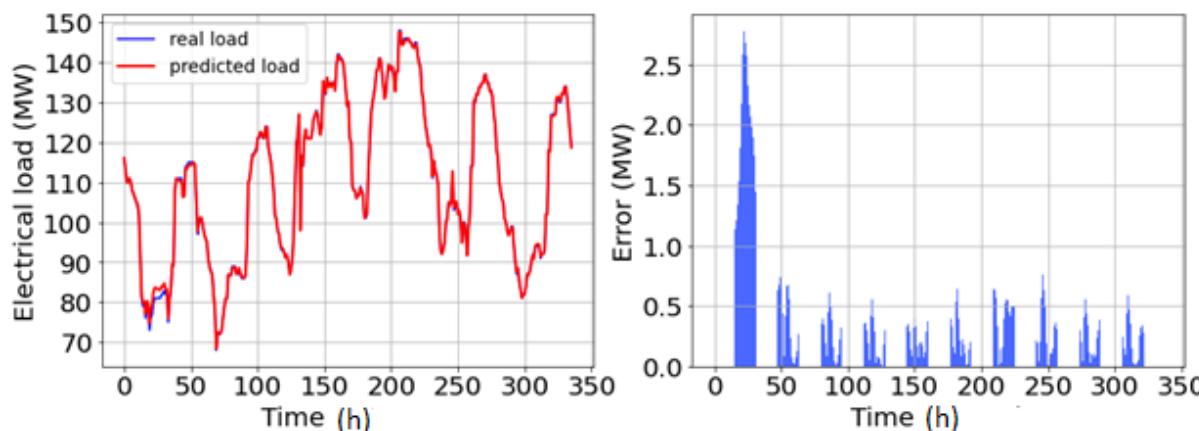


Figure 16. Actual and predicted loads with sorted wavelet coefficients.

4.4. Benchmark Models

To demonstrate the effectiveness of the proposed approach, a comparative study was carried out on reference models, including the ARIMA and ANNs of the MLP-type multilayer perceptron and ConvLSTM as reliable forecasting methods. The reasons for making these choices are enormous, some of which are mentioned in the introduction to the article; and others in [47,48]. The prediction results for these models are illustrated in Figure 17. All models underwent optimization using the same LSTM hyperparameter optimization algorithms (GridsearchCV). The optimized MLP model consists of three (03) hidden layers. The first hidden layer comprises 96 neurons. The second has 64 neurons and the third has 32 neurons—these values were obtained through optimization. The chosen activation function is “relu”. For the ARIMA model, the best hyperparameters are “p = 4. d = 1. q = 0”. corresponding to ARIMA (4. 1. 0). The performance of these four models is evaluated based on the MAPE, RMSE, and R² criteria. The results are presented in Table 3.

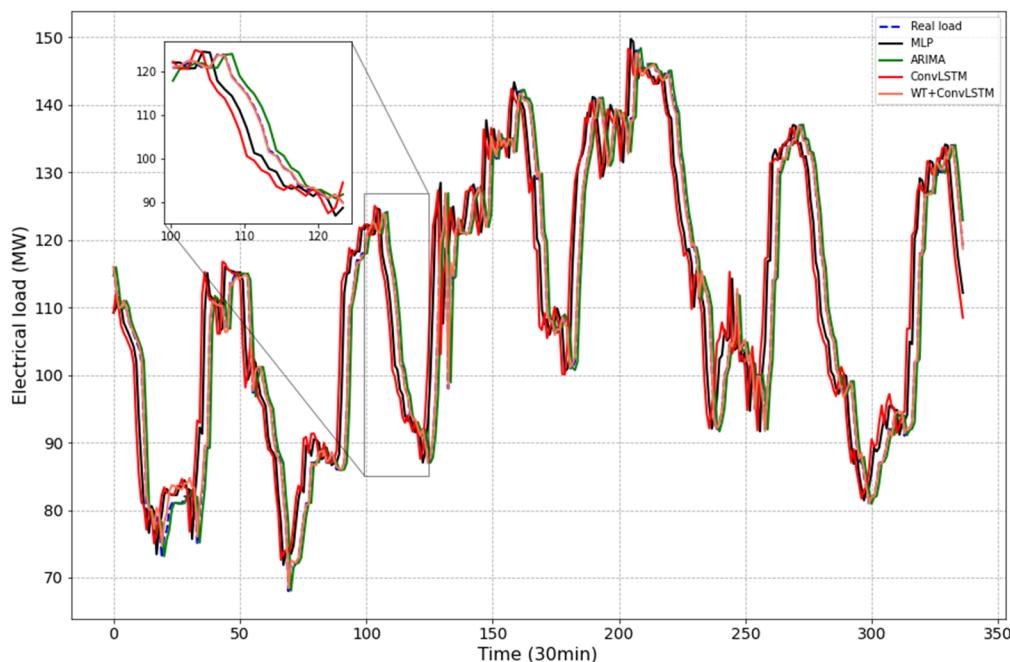


Figure 17. Forecast results for various models.

Table 3. Model evaluation criteria after wavelet decomposition.

Models	MAPE (%)	RMSE	R2
MLP	3.39	5.503	0.915
ARIMA	3.21	5.895	0.903
ConvLSTM	3.535	5.476	0.916
WT+ConvLSTM	0.485	0.61	0.999

The WT+ConvLSTM model has the highest SNR (1.6393), indicating the best performance in terms of noise robustness among the compared models. The other models, with SNRs around 0.17 to 0.18, show lower performance, with the ARIMA model having the lowest SNR. The results demonstrate the impact of sorting wavelet transform coefficients on electric load prediction. The relevance of using this approach often overlooked in the literature reveals better prospects in the electricity sector. The proposed method can be employed for short-term forecasts and assisting grid operators in integrating other sources more effectively and enhancing grid stability.

5. Conclusions

To address the challenges of electric load forecasting, an in-depth study was conducted, leading to the development of a robust model based on wavelet decomposition, both with and without coefficient sorting, combined with different types of LSTM networks. Four models were employed: Stacked LSTM, BiLSTM, CNN-LSTM, and ConvLSTM. The hyperparameters of each model were optimized using grid search cross-validation. Evaluation criteria, including MAPE, RMSE, and R², indicated that the ConvLSTM model outperformed the others.

Subsequently, a hybrid approach was proposed, combining ConvLSTM with wavelet decomposition using coefficient sorting, specifically employing a fourth-order Daubechies wavelet and a decomposition level of 4. This enhancement significantly improved accuracy, reducing the absolute error from 10–50 MW to 0.5–3 MW. Moreover, the proposed hybrid model demonstrated its effectiveness by surpassing traditional reference models such as ARIMA and MLP ANNs based on performance metrics.

This method can be utilized by power plant operators to address high prediction errors associated with the nature and quality of data, optimizing operational and maintenance activities through accurate load forecasts, and facilitating overall planning.

The study will have a significant impact on the decision-making process regarding the regulation of electricity prices in line with market demand. Additionally, it could have medium- and long-term repercussions by contributing to the development of policies aimed at improving universal access to electricity and reducing energy dependency.

Future work suggests increasing model complexity by incorporating additional variables, including social factors, to further improve forecast accuracy. Exploring combinations of these models with each other or with other machine learning techniques is also recommended. Additionally, it is advised to investigate other hyperparameter optimization algorithms for more comprehensive theoretical research in electric load prediction. Extending model validation to other cities or countries is essential to ensure robustness and generalizability.

Author Contributions: Conceptualization: A.G. and A.P.A.; Methodology: A.G.; Software: A.G. and Y.A.B.; Validation: A.G., A.P.A. and A.A.S.; Formal Analysis: A.G.; Investigation: A.G.; Resources: A.G.; Data Curation: A.G. and Y.A.B.; Writing—Original Draft Preparation: A.G.; Writing—Review and Editing: A.G. and A.A.S.; Visualization: A.G. and Y.A.B.; Supervision: A.G. and A.A.S.; Project Administration: A.G. and A.A.S.; Funding Acquisition: A.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this study come from the Compagnie Energie Electrique du Togo (CEET) and are subject to a strict confidentiality and non-disclosure policy due to their sensitive nature. Consequently, they cannot be made publicly available. However, interested parties may contact the authors for specific information, subject to the signing of confidentiality agreements and compliance with ethical requirements.

Acknowledgments: The authors want to thank the Centre d’Excellence Régional pour la Maîtrise de l’Électricité (CERME) of University of Lome for providing support and enabling environment during this research.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Tian, C.; Hao, Y. A novel nonlinear combined forecasting system for short-term load forecasting. *Energies* **2018**, *11*, 712. [[CrossRef](#)]
- Quan, H.; Srinivasan, D.; Khosravi, A. Uncertainty handling using neural network-based prediction intervals for electrical load forecasting. *Energy* **2014**, *73*, 916–925. [[CrossRef](#)]
- Fan, S.; Chen, L. Short-term load forecasting based on an adaptive hybrid method. *IEEE Trans. Power Syst.* **2006**, *21*, 392–401. [[CrossRef](#)]
- Du, P.; Wang, J.; Yang, W.; Niu, T. Multi-step ahead forecasting in electrical power system using a hybrid forecasting system. *Renew. Energy* **2018**, *122*, 533–550. [[CrossRef](#)]
- Yang, W.; Wang, J.; Wang, R. Research and application of a novel hybrid model based on data selection and artificial intelligence algorithm for short term load forecasting. *Entropy* **2017**, *19*, 52. [[CrossRef](#)]
- Zjavka, L.; Snášel, V. Short-term power load forecasting with ordinary differential equation substitutions of polynomial networks. *Electr. Power Syst. Res.* **2016**, *137*, 113–123. [[CrossRef](#)]
- Wang, Y.; Wang, J.; Zhao, G.; Dong, Y. Application of residual modification approach in seasonal ARIMA for electricity demand forecasting: A case study of China. *Energy Policy* **2012**, *48*, 284–294. [[CrossRef](#)]
- Blum, M.; Riedmiller, M. Electricity demand forecasting using gaussian processes. In Proceedings of the Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, WA, USA, 14–18 July 2013.
- Adjamagbo, C.; Ngae, P.; Vianou, A.; Vigneron, V. Modélisation de la demande en énergie électrique au Togo. *J. Renew. Energ.* **2011**, *14*, 67–83. [[CrossRef](#)]
- Regmi, S.; Rayamajhi, A.; Poudyal, R.; Adhikari, S. Evaluating Preparedness and Overcoming Challenges in Electricity Trading: An In-Depth Analysis Using the Analytic Hierarchy Process and a Case Study Exploration. *Electricity* **2024**, *5*, 271–294. [[CrossRef](#)]
- Hobbs, B.F.; Jitprapaikulsarn, S.; Konda, S.; Chankong, V.; Loparo, K.A.; Maratukulam, D.J. Analysis of the value for unit commitment of improved load forecasts. *IEEE Trans. Power Syst.* **1999**, *14*, 1342–1348. [[CrossRef](#)]

12. Lee, C.-W.; Lin, B.-Y. Application of hybrid quantum tabu search with support vector regression (SVR) for load forecasting. *Energies* **2016**, *9*, 873. [[CrossRef](#)]
13. Soares, L.J.; Medeiros, M.C. Modeling and forecasting short-term electricity load: A comparison of methods with an application to Brazilian data. *Int. J. Forecast.* **2008**, *24*, 630–644. [[CrossRef](#)]
14. Bunn, D.; Farmer, E.D. *Comparative Models for Electrical Load Forecasting*; Wiley: Hoboken, NJ, USA, 1985.
15. Tatsa, S.; en Économique, M. Modélisation et Prévision de la Consommation Horaire D'électricité au Québec. Master's Thesis, Université Laval, Quebec City, QC, Canada, 2013.
16. Pełka, P. Pattern-based forecasting of monthly electricity demand using support vector machine. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Virtual, 18–22 July 2021; pp. 1–8.
17. Pełka Paweł and Dudek, G. Pattern-based forecasting monthly electricity demand using multilayer perceptron. In Proceedings of the International Conference on Artificial Intelligence and Soft Computing, Zakopane, Poland, 16–20 June 2019; pp. 663–672.
18. Moghram, I.; Rahman, S. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans. Power Syst.* **1989**, *4*, 1484–1491. [[CrossRef](#)]
19. Dotche, K.A.; Salami, A.A.; Kodjo, K.M.; Blu, Y.P.C.D.; Diabo, Y.E.J. Evaluating Solar Energy Harvesting using Artificial Neural Networks: A Case study in Togo. In Proceedings of the 2019 II International Conference on High Technology for Sustainable Development (HiTech), Sofia, Bulgaria, 10–11 October 2019; pp. 1–5.
20. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
21. Pant, N. A guide for time series prediction using recurrent neural networks (LSTMS). *Stats Bots* **2017**, *7*, 9.
22. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
23. Salami, A.A.; Ajavon, A.S.A.; Dotche, K.A.; Bedja, K.-S. Electrical load forecasting using artificial neural network: The case study of the grid inter-connected network of benin electricity community (CEB). *Am. J. Eng. Appl. Sci.* **2018**, *11*, 471–481. [[CrossRef](#)]
24. Zhang, J.; Wei, Y.-M.; Li, D.; Tan, Z.; Zhou, J. Short term electricity load forecasting using a hybrid model. *Energy* **2018**, *158*, 774–781. [[CrossRef](#)]
25. He, W. Load forecasting via deep neural networks. *Procedia Comput. Sci.* **2017**, *122*, 308–314. [[CrossRef](#)]
26. Wentz, V.H.; Maciel, J.N.; Gimenez Ledesma, J.J.; Ando Junior, O.H. Solar Irradiance Forecasting to Short-Term PV Power: Accuracy Comparison of ANN and LSTM Models. *Energies* **2022**, *15*, 2457. [[CrossRef](#)]
27. Ma, Q. Comparison of ARIMA, ANN and LSTM for stock price prediction. *E3S Web Conf.* **2020**, *218*, 1026. [[CrossRef](#)]
28. Mao, G.; Wang, M.; Liu, J.; Wang, Z.; Wang, K.; Meng, Y.; Zhong, R.; Wang, H.; Li, Y. Comprehensive comparison of artificial neural networks and long short-term memory networks for rainfall-runoff simulation. *Phys. Chem. Earth Parts A/B/C* **2021**, *123*, 103026. [[CrossRef](#)]
29. Adam, K.; Smagulova, K.; Krestinskaya, O.; James, A.P. Wafer quality inspection using memristive LSTM, ANN, DNN and HTM. In Proceedings of the 2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS), Chandigarh, India, 16–18 December 2018; pp. 1–3.
30. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
31. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)] [[PubMed](#)]
32. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [[CrossRef](#)] [[PubMed](#)]
33. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
34. Khan, S.; Rahmani, H.; Shah, S.A.A.; Bennamoun, M.; Medioni, G.; Dickinson, S. *A Guide to Convolutional Neural Networks for Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2018; Volume 8.
35. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; Woo, W. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
36. Vrskova, R.; Hudec, R.; Kamencay, P.; Sykora, P. A new approach for abnormal human activities recognition based on ConvLSTM architecture. *Sensors* **2022**, *22*, 2946. [[CrossRef](#)]
37. Wang, Y.; Jiang, L.; Yang, M.-H.; Li, L.-J.; Long, M.; Fei-Fei, L. Eidetic 3D LSTM: A model for video prediction and beyond. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
38. Blanco-Velasco, M.; Cruz-Roldán, F.; Moreno-Martinez, E.; Godino-Llorente, J.-I.; Barner, K.E. Embedded filter bank-based algorithm for ECG compression. *Signal Process.* **2008**, *88*, 1402–1412. [[CrossRef](#)]
39. Liashchynskyi, P.; Liashchynskyi, P. Grid search, random search, genetic algorithm: A big comparison for NAS. *arXiv* **2019**, arXiv:1912.06059.
40. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
41. Cristea, P.; Tduce, R.; Cristea, A. Time series prediction with wavelet neural networks. In Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering, NEUREL 2000 (IEEE Cat. No. 00EX287), Belgrade, Yugoslavia, 27 September 2000; pp. 5–10.
42. Lau, K.-M.; Weng, H. Climate signal detection using wavelet transform: How to make a time series sing. *Bull. Am. Meteorol. Soc.* **1995**, *76*, 2391–2402. [[CrossRef](#)]

43. Sharma, J.K.; Gopal, Y.; Birla, D.; Lalwani, M. An Algorithm for Selecting Compatible Wavelet Function in Electrical Signals to Detect and Localize Disturbances. *Int. J. Appl. Eng. Res.* **2018**, *13*, 11440–11447.
44. Belayneh, A.; Adamowski, J.F.; Khalil, B. Long-term Drought Forecasting Using Wavelet-Neural Networks and Wavelet-Support Vector Regression. *Am. Soc. Agric. Biol. Eng.* **2012**.
45. Nourani, V.; Komasi, M.; Mano, A. A multivariate ANN-wavelet approach for rainfall–runoff modeling. *Water Resour. Manag.* **2009**, *23*, 2877–2894. [[CrossRef](#)]
46. Wang, W.; Ding, J. Wavelet network model and its application to the prediction of hydrology. *Nat. Sci.* **2003**, *1*, 67–71.
47. Hosseini, S.; Henao, N.; Kelouwani, S.; Agbossou, K.; Cardenas, A. A study on Markovian and deep learning based architectures for household appliance-level load modeling and recognition. In Proceedings of the 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE), Vancouver, BC, Canada, 12–14 June 2019; pp. 35–40.
48. Dab, K.; Agbossou, K.; Henao, N.; Dubé, Y.; Kelouwani, S.; Hosseini, S.S. A compositional kernel based gaussian process approach to day-ahead residential load forecasting. *Energy Build.* **2022**, *254*, 111459. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.