SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage: https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Electrical Load Forecasting using a Novel BI-GRU Encoder Decoder Model

Ahmed Gaber[1,*], Hesham H. Aly[1], Noha Ghatwary[2], Ahmed K. Abdelsalam[3]

[1] Electronics and Communication Engineering Department, Arab Academy for Science, Technology and Maritime Transport, Giza P.O. Box 2033, Egypt
[2] Computer Engineering Department, Arab Academy for Science, Technology and Maritime Transport, Giza P.O. Box 2033, Egypt
[3] Electrical Engineering Department, Arab Academy for Science, Technology, and Maritime Transport, Cairo P.O. Box 12577, Egypt

ARTICLE INFO

Article history:
Received 21 December 2023
Received in revised form 23 May 2024
Accepted 14 August 2024
Available online 2 September 2024

Keywords:
Electrical load forecasting; Encoder-decoder framework; Bi-directional gated recurrent units; LSTM; CNN

ABSTRACT

Precisely forecasting electrical load, especially through univariate time series analysis, is pivotal for effectively operating and planning power systems. This research introduces a hybrid model leveraging univariate time series and deep learning techniques. The model combines the Bidirectional Gated Recurrent Units (Bi-GRU) based encoder-decoder structure with the Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) architecture to enhance single-step short-term electrical demand forecasting. Integrating Bi-GRUs ensures adept capture of temporal dependencies, while CNNs meticulously extract spatial features. Concurrently, LSTMs provide a robust mechanism to memorize long-range dependencies. The model's competence was rigorously assessed through evaluations using the publicly available American Electric Power (AEP) dataset, which represents real-world electrical load patterns. Findings highlight that the proposed model outstrips competing models in algorithmic stability and prediction accuracy. With a Mean Absolute Percentage Error (MAPE) of 80.032, this research posits a promising avenue for utilizing deep learning in univariate time series power load prediction.

1. Introduction

The current enormous surge in energy consumption has been caused by several factors, including the quick development of motorized vehicles, complex machine systems, and global market commerce [1]. The Smart Meter Infrastructure enables the integration of smart grids and active power distribution systems, hence facilitating the construction of accurate and reliable short-term energy forecasting systems [2].

Power framework experts should develop and implement new techniques as the number of machines increases to manage power usage in consideration of customer interest effectively. Energy consumption analysis and estimations might be an approach for executives in the current energy sector to consider this issue. However, because energy consumption is nonlinearly unique and depends on various factors [3], developing an appropriate mathematical model of it is challenging.

* *Corresponding author.*
E-mail address: a.mohamed13701@student.aast.edu

https://doi.org/10.37934/araset.51.1.114

Time series data can be thought of as a collection of observations made in time. Its kind can be divided into univariate and multivariate time series, each with a lot of data, is highly dimensional, and constantly changing. Based on traditional machine learning models, a number of methods have been proposed in the literature to address time series prediction. Deep learning techniques, being a worldly methodology, have the potential to supplant statistical approaches owing to their significant impact on the processes of pattern recognition and prediction. Convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are used to illustrate this [4]. Even though RNN-based approaches have significantly improved power load forecasting, using other techniques, such as time series prediction using encoder-decoder networks, which resolve power forecasting and related management problems. However, in short-term load forecasting, encoder-decoder models encounter specific challenges. Firstly, encoder-decoder models may need help to capture the rapid and volatile patterns in short-term load forecasting, resulting in less accurate predictions for short-term horizons. The second challenge is that encoder-decoder models typically rely on historical information encoded by the encoder to make predictions.

In light of the aforementioned issues, a novel hybrid model is introduced for the first time. This model integrates the traditional Bi-GRU encoder-decoder structure with the CNN-LSTM prediction model. The objective is to effectively learn implicit temporal dependence features and improve the accuracy of predicting complexity performance and power demand. After conducting rigorous validation using a comprehensive dataset from American Electric Power, the proposed model demonstrates robustness and yields satisfactory predictive outcomes. The primary contribution of our model can be succinctly expressed as follows:

i. The Bi-GRU Encoder-Decoder structure is tailored to univariate time series data. This allows the accommodation of temporal dependencies present in the electrical load data. By harnessing information from both preceding and succeeding time frames, this integration becomes vital for accurate forecasting. Furthermore, the Encoder-Decoder paradigm facilitates the transformation of input sequences into enriched internal representations, embedding time-sensitive data. This functionality significantly contributes to the enhancement of forecasting precision.
ii. Incorporating CNNs in the model enhance its ability to detect and extract fine-grained patterns and refined features from the dataset. This is valuable for identifying cyclical trends in electrical consumption, such as daily or weekly patterns. Further, by incorporating LSTM layers, the model gains proficiency in recognizing and learning long-term dependencies, enabling it to consider short-term temporal patterns and broader trends such as seasonal fluctuations in electricity usage.
iii. The model is extensively evaluated on a publicly available dataset and compared to an alternative model, providing compelling evidence of its efficiency and superiority in addressing the challenges of single-step short-term electrical demand forecasting within the context of univariate time series data.

The present paper is organized in the following manner: Section 2 presents a survey of relevant literature on time series modelling and forecasting. In Section 3, the research motivations behind the proposed model are clarified, followed by an analysis of the Bi-GRU encoder-decoder framework's overall structure, accompanied by an explanation of the associated theories and procedural specifics. Section 4 encloses the analytical content of the experiments. Followed by an in-depth analysis of the experimental outcomes. The study concludes in Section 5 by providing a summary of the findings and potential possibilities for future research.

2. Literature Review

Depending on their predictive performance, power load forecasting models can be categorized based on the forecast time horizon, including short, medium, and long-term forecasting. Depending on their ability to anticipate the future, power load forecasting models can be divided into three categories: short-, medium-, and long-term forecasting. The primary uses of short-term load forecasting (STLF) models are production and delivery plans, which try to estimate changes in electricity within a week or even a day. While long-term forecasting models concentrate on power prediction for periods longer than one-year, medium-term forecasting models are developed to forecast electricity demand from one week to one year. In order to estimate electric load for effective operational planning, reduce power waste, and provide a solid foundation for economic management and the long-term sustainability of the power system, short-term load forecasting models are crucial [5].

Different methodologies have been categorized into two broad groups for efficient energy consumption forecasting in various domains: buildings, industries, institutes, and residential areas. These categories include conventional techniques and approaches based on artificial intelligence. Traditional techniques often rely on statistical methods for accurate predictions [6]. Various statistical modelling methods are commonly employed in data analysis. These methods include exponential smoothing [9], weighted moving average [10], multiple regression analysis [7], auto-regressive (AR) [11], Kalman filtering [8] and autoregressive moving average (ARMA) [12]. These models typically involve a comprehensive theoretical derivation process and modelling steps that rely on prior knowledge and empirical assumptions to extract information from the data and determine the model parameters [13], where predicting the power load outcome becomes challenging when dealing with complex nonlinear data or when there is a mismatch between the data distribution and the model Hypothesis. Due to the nonlinear characteristics of time series load data, conventional methods may only sometimes perform effectively in short-term load forecasting. Additionally, most of the methods mentioned earlier are commonly employed for forecasting demand at the aggregated system level. Initial research in short-term load forecasting at the household level involved using time series analysis and traditional statistical techniques [14].

In recent research, the emphasis has shifted towards artificial intelligence methods, leading to the application of various machine learning and deep learning techniques for predicting home energy use. Machine learning techniques include k-nearest neighbours (K-NN) [15], decision trees (DT) [16]. The performance of machine learning-based techniques in predicting non-linear load sequences has been found to be limited. These approaches also possess certain limitations. One of the problems lies in the necessity for manually built features, which entails a substantial degree of human involvement. The presence of this particular aspect poses a challenge in accurately capturing essential non-linear relationships and underlying temporal dependencies, particularly when taking into account the limited training data available for load forecasting. Deep learning techniques have been shown to be highly effective in addressing complex, non-linear, and dynamic problems across several domains, while also optimizing machine learning challenges [17]. In recent years, RNNs have been successfully applied to Short-Term Load Forecasting (STLF) due to their unique structure [21,22]. However, traditional RNNs encounter the issue of vanishing gradients, which can cause them to get stuck in local extreme values and need help to capture long-term dependencies [4]. To address this limitation and improve the accuracy of time-series prediction, Long Short-Term Memory (LSTM) units have been introduced. LSTM units incorporate input, forget, and output gates in their computational mechanism, allowing them to overcome the vanishing gradient problem and achieve significant success in various electricity load forecasting applications. Recently, another variant of

RNNs known as Gated Recurrent Units (GRU) has gained traction in power sequence prediction. The GRU model presents a neural architecture that is characterized by a simplified structure compared to LSTM. By combining the input and forgetting gates into a single update gate, this is accomplished. This design allows for faster computation and improved sequence-based electric load data expression capabilities.

A recent study introduced a hybrid sequential learning methodology that used a deep learning model. This strategy involved a two-phase solution, where a Convolutional Neural Network (CNN) was implemented in the initial phase to extract features from the energy consumption dataset. In the succeeding phase, the utilization of Gated Recurrent Unit (GRU) models was employed to capitalize on its efficient gated architecture for the purpose of making predictions. In contrast to LSTM-based models, GRU-based models tend to exhibit reduced volatility because to their simpler architecture and fewer gradient flow gates [20]. Examining alternate modern networks, such as encoder-decoder networks, is an appropriate plan of action within the framework of time series prediction. Because encoder-decoder designs are successful in machine translation, natural language processing, and other areas, they have become more common as sequence-to-sequence (seq2seq) models [21]. The encoder-decoder structure typically consists of multiple RNN layers, including an encoder and a decoder, to encode the source data into a fixed-length vector representation. The decoder then uses this encoded representation to provide a translation or prediction, successfully capturing the input data's time-series properties and transformation features. It has been shown that the GRU-based seq2seq model performs better at forecasting short, medium, and long-term power data [22]. The application of temporal attention-based encoder-decoder models has been observed in the context of multivariate time series multi-step forecasting issues [23].

In short-term load forecasting, encoder-decoder models encounter specific challenges. Firstly, encoder-decoder models may struggle to capture the rapid and volatile patterns in short-term load forecasting, resulting in less accurate predictions for short-term horizons. The second challenge is that encoder-decoder models typically rely on historical information encoded by the encoder to make predictions. However, short-term forecasting often requires a stronger emphasis on recent data points. The traditional encoder-decoder architecture may not effectively prioritize current information, reducing accuracy in short-term predictions. To address these challenges, bidirectional Gated Recurrent Units (GRUs) within the encoder are used in this paper.

3. Methodology

This section describes Our proposed model, which is depicted in Figure 1, consisting of four main components: an Encoder with Bi-GRU units, a Decoder with Bi-GRU units, a Convolutional Layer, and an LSTM Layer. The Bi-GRU Encoder ingests a sequence of historical electricity load data and effectively captures the temporal patterns by considering past and future contexts. The Bi-GRU Decoder takes the encoded sequence and reconstructs the target sequence. The Convolutional layer is employed to learn spatial features from the output of the Decoder. Convolutional networks are known for effectively learning hierarchical patterns in data, and we leverage this property to capture spatial correlations in electricity loads. The LSTM layer is integrated to refine further the temporal patterns extracted by the convolutional layer. An explanation of the possible structure of such a hybrid model is provided below:

3.1 Encoder-Decoder Structure

In the domain of short-term load forecasting (STLF) in the energy sector, the conventional Encoder-Decoder architecture can be employed to anticipate the power load for a brief timeframe in advance, relying on past load data. The Encoder-Decoder structure can be employed for this objective.

3.1.1 Encoder

The input sequence X could be historical load data at different time intervals e.g., X = [$load_{t-24}$,$load_{t-23}$,...,$load_t$] , where $load_t$ is the load at time *t*. The Encoder uses GRU cells to process this sequence. The GRU is a variation of RNN that is designed to remember long-term dependencies. The state of a GRU at time *t*, is given by h_t. GRU is a variant of LSTM with only two gates (the update gate and the reset gate). Because GRU has fewer training parameters than LSTM, it converges more quickly than LSTM during training [24]. The GRU structure is shown in Figure 2 (a), Where update Gate (z_t) determines how much of the previous hidden state should be preserved and how much of the new input should be incorporated and reset Gate (r_t) determines how much of the previous hidden state should be forgotten and how much of the new input should be focused on, the formulas of GRU [25] can be shown as:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \tag{1}$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \tag{2}$$

$$h'_t = tanh(W_n \cdot x_t + U_n \cdot (r_t \odot h_{t-1}) + b_n) \tag{3}$$

$$h_t = (1 - z_t) \odot h'_t + z_t \odot h_{t-1} \tag{4}$$

where (z_t) is the update gate, (r_t) is the reset gate, (h'_t) is the new memory content, (h_t) is the hidden state , W, U, and b are weight matrices and bias vectors, σ is the sigmoid function and ⊙ represents element-wise multiplication.

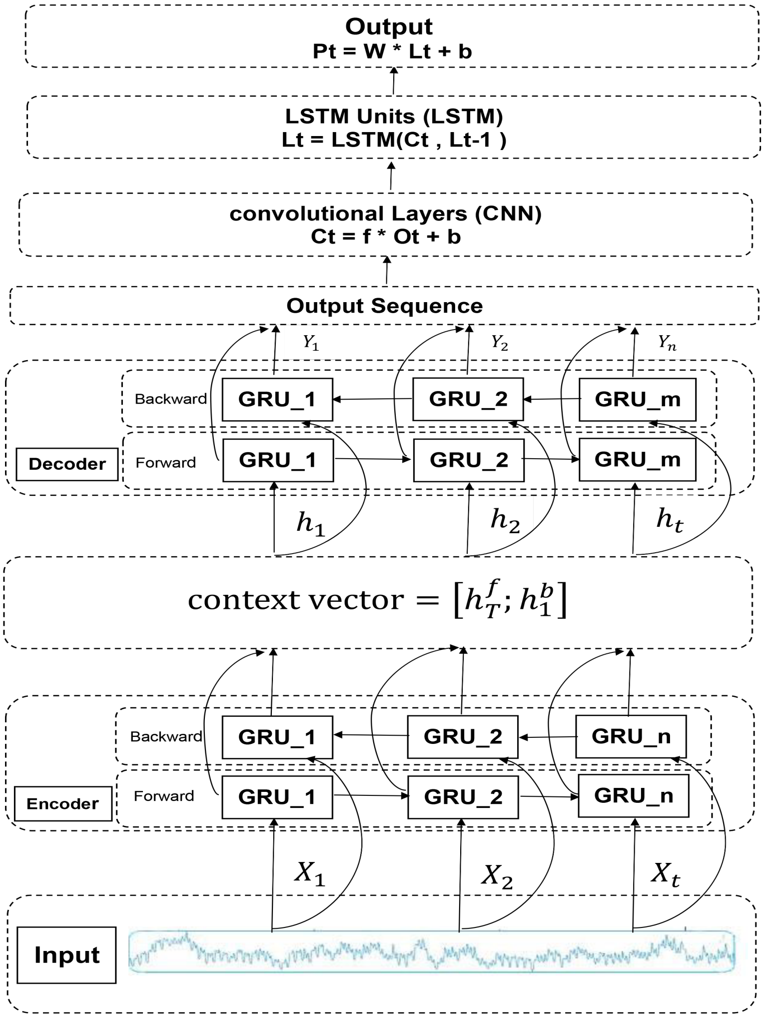


Fig. 1. The architecture of the proposed BI-GRU Encoder-Decoder prediction model

The GRU reads in the input sequence element by element, updating its hidden state(ht) at each step. Once the entire sequence is read, the final hidden state represents the encoded information of the input sequence. This final state (h_t) serves as the context vector. However, future events can sometimes be known in advance for short-term load forecasting, influencing the current forecast. For example, if we know a holiday in the next few days, this information could help forecast the load today. A unidirectional GRU cannot use this type of information because it processes the sequence in a single direction (past to future).

On the other hand, a Bidirectional GRU (Bi-GRU) can process information from both directions and, therefore, use future details when it's available. Bi-GRUs have two GRU cells for each time step; one processes the sequence from left to right (forward states) and the other from right to left (backward states). The forward GRU cell equations are the same as the regular GRU [26] and can be shown as:

$$z_t = \sigma\left(W_z \cdot x_t + U_z \cdot h_{t-1}^f + b_z\right) \tag{5}$$

$$r_t = \sigma\left(W_r \cdot x_t + U_r \cdot h_{t-1}^f + b_r\right) \tag{6}$$

$$h'_t = tanh(W_n \cdot x_t + U_n \cdot (r_t \odot h^f_{t-1}) + b_n) \quad (7)$$

$$h^f_t = (1 - z_t) \odot h'_t + z_t \odot h^f_{t-1} \quad (8)$$

The backward GRU [26] cell equations are similar, but it processes the sequence in reverse can be shown as:

$$z_t = \sigma(W_z \cdot x_t + U_z \cdot h^b_{t-1} + b_z) \quad (9)$$

$$r_t = \sigma(W_r \cdot x_t + U_r \cdot h^b_{t-1} + b_r) \quad (10)$$

$$h'_t = tanh(W_n \cdot x_t + U_n \cdot (r_t \odot h^b_{t-1}) + b_n) \quad (11)$$

$$h^b_t = (1 - z_t) \odot h'_t + z_t \odot h^b_{t-1} \quad (12)$$

the final context vector typically consists of the concatenation of the final hidden state of the forward GRU and the last hidden state of the backward GRU

$$context\ vector = [h^f_t \ ; \ h^b_1] \quad (13)$$

where h^f_t represents the last hidden state of the forward GRU after it has processed the entire input sequence from left to right. Here, T is the length of the input sequence, and f indicates that this is the hidden state from the forward GRU, h^b_1 represents the last hidden state of the backward GRU after it has processed the entire input sequence from right to left. The index 1 is used here because it refers to the position of the first element in the original sequence, and b indicates that this is the hidden state from the backward GRU.

3.1.2 Decoder

The context vector from the encoder is used to initialize the hidden state of the decoder. Similar to the encoder, the decoder also uses two GRUs for each time step. The decoder's forward and backward GRUs also similarly calculate hidden states to the encoder. Additionally, the decoder produces an output at each time step. This output is typically computed using an activation function over the hidden states to produce a probability distribution for the output tokens. The decoder generates the output sequence Y = (y_1 , y_2 , ..., y_N), where N is the length of the target sequence.

3.2 CNN-LSTM Layer

The Convolutional Neural Network (CNN) is a widely employed deep learning method utilized for various image processing and computer vision applications. Nevertheless, it is also possible to modify it for the purpose of short-term load electricity forecasting. A CNN is commonly employed for the purpose of short-term load electricity forecasting. This architecture typically comprises convolutional layers, pooling layers, fully connected layers, and an output layer.

LSTM (Long Short-Term Memory) networks are a specific variant of RNN that excel in capturing long-term dependencies within sequential data. This characteristic renders them particularly advantageous for time series forecasting tasks, such as the prediction of power loads [27]. In electricity load forecasting, the LSTM can model the temporal dependencies of the load data. It can learn patterns across different time scales, such as hourly, daily, and seasonal trends. The LSTM structure is shown in Figure 2 (b), the formulas of LSTM [28] can be shown as follows:

$$f_t = \sigma\left(W_f \cdot [h_t - 1, x_t] + b_f\right) \tag{14}$$

$$i_t = \sigma(W_i \cdot [h_t - 1, x_t] + b_i \tag{15}$$

$$C'_t = tanh(W_c \cdot [h_t - 1, x_t] + b_c) \tag{16}$$

$$C_t = f_t * C_{t-1} + i_t * C'_t \tag{17}$$

$$o_t = \sigma(W_o \cdot [h_t - 1, h_t - 1] + b_o) \tag{18}$$

$$h_t = o_t * tanh(C_t) \tag{19}$$

where (f_t) is the forget gate which decides what information to forget from the cell state, (i_t) is the input gate which decides what new information to store in the cell state, (C'_t) is the candidate memory cell which computes candidate values to be added to the cell state, (C_t) is the cell state which stores the long-term dependencies, (o_t) is the output gate which decides what parts of the cell state are going to be output, (h_t) is the hidden state which stores the short-term dependencies, (x_t) is the input at time step t, W and b are the weights and biases respectively, σ denotes the sigmoid activation function, whereas "tanh" is used to represent the hyperbolic tangent activation function. Additionally, the * is used to indicate element-wise multiplication.

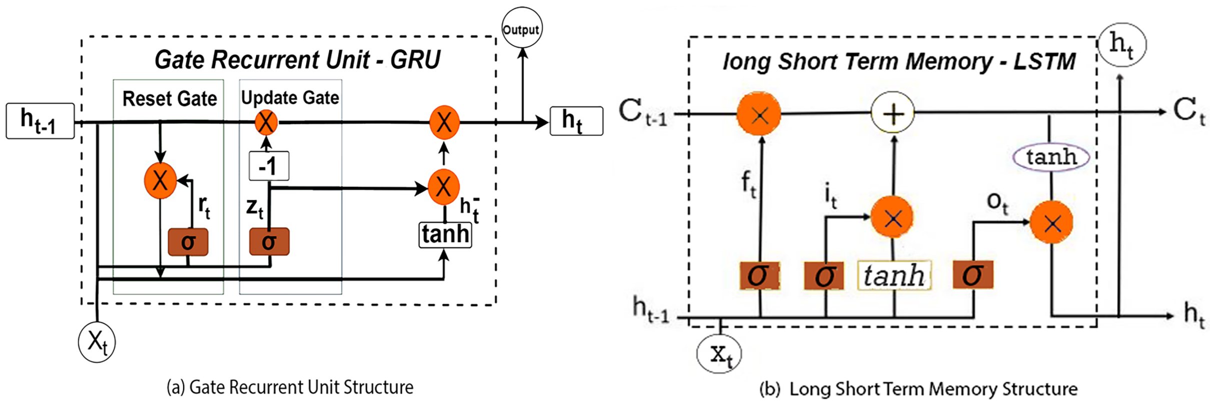


Fig. 2. The architecture of (a) Gate Recurrent Unit Structure (b) Long Short-Term Memory Structure [28]

In our proposed model, after the decoder has processed the output sequence, we employ a CNN-LSTM layer. Firstly, convolutional layers adeptly and autonomously learn the spatial hierarchies inherent in the decoder output. Subsequently, at time step t, the output of the CNN is calculated.

$$C_t = f . O_t + b \tag{20}$$

where O_t represents the output from the decoder at time step t, f represents filters, and b represents the bias term. Then, this output is passed through LSTM units. The LSTM layer captures the temporal dependencies of the features obtained by the convolutional layer. The output of the LSTM at time step t can be represented as:

$$L_t = LSTM(C_t, L_{t-1}) \tag{21}$$

Finally, the output from the LSTM layer is passed through a dense layer with a linear activation function to make the final predictions Pt.

$$P_t = W.L_t + b \tag{22}$$

where W represents the weight matrix and b represents the bias term.

4. Experiments

4.1 Datasets and Setup

The dataset used for this study consists of univariate time series data capturing the electric power load sourced from the American Electric Power Company (AEP) [29]. The dataset comprises 121,273 data points, covering the time frame from December 2004 to January 2018, with an hourly sampling frequency. Electricity consumption patterns versus month for 2004-2018 is shown in Figure 3 .

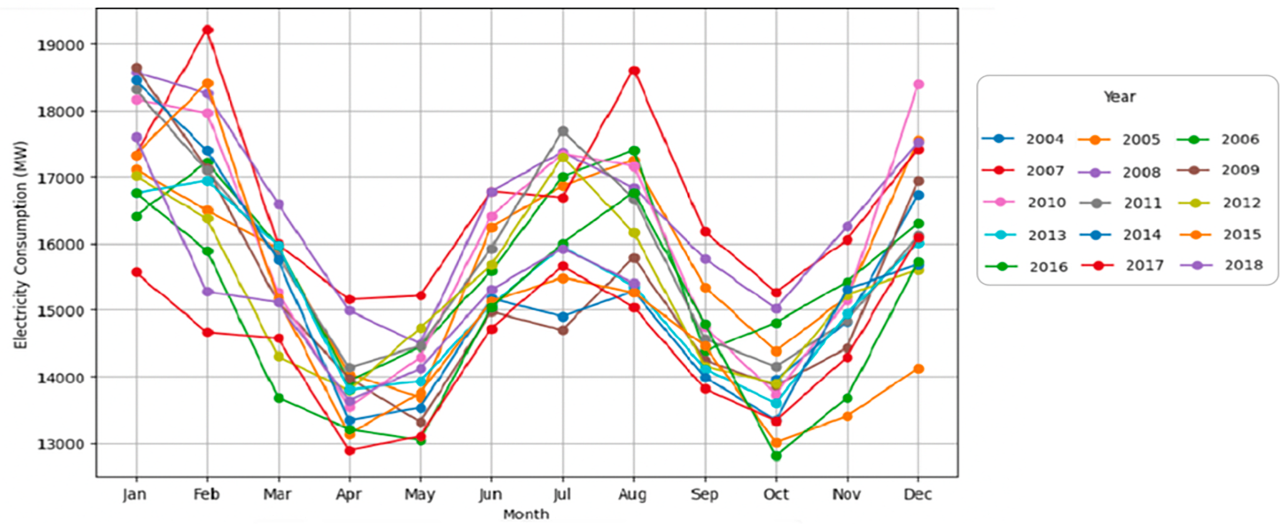
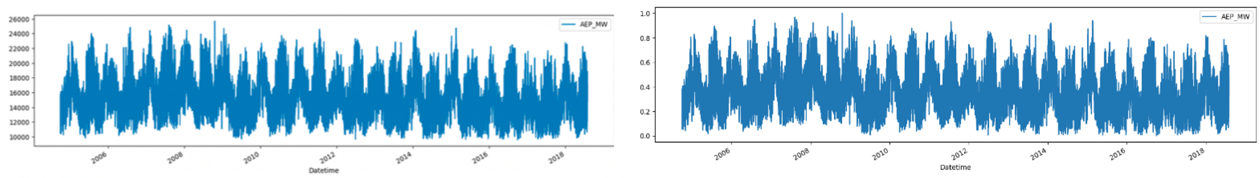


Fig. 3. The actual load of the dataset represented by years

Figures 4 (a) and (b) demonstrate the study of the AEP hourly power consumption data before and after normalization. Figure 4 (a) illustrates the raw AEP hourly power consumption data without normalization. The values represent the original power consumption measurements recorded over a specific period, capturing the actual load patterns and variations in the dataset. Figure 4 (b), on the other hand, depicts the AEP hourly power consumption data after normalization. The normalization process has been applied to scale the data within a predefined range or distribution, removing any inherent biases or absolute scales [30]. This transformation allows for fair comparisons and facilitates the modelling process by mitigating the influence of extreme values. By comparing Figures 4 (a) and (b), we can observe the effects of normalization on the power consumption data, providing insights into the significance of this pre-processing step in the context of short-term electrical load forecasting. The normalization procedure enhances the forecasting model's performance by ensuring that all features contribute equally and appropriately to the predictive process, leading to more accurate and reliable results. The dataset was partitioned into three subsets: 60% for training, 20% for validation, and 20% for testing. The training set is used for learning the model's parameters, the validation set is used for hyperparameter tuning and to prevent overfitting, and the test set is used for evaluating the model’s performance on unseen data, followed by normalization. Employing a supervised learning methodology, the model utilizes a sliding window technique to segment the electric load data into numerous input sets and corresponding target values, arranged sequentially in chronological order. The neural network can learn this temporally-ordered configuration. To enhance the model’s generalization capabilities, the data subsets generated by the sliding window are shuffled before being input into the neural network for training. In the research, several optimized parameter configurations were ascertained utilizing a randomized search cross-validation (CV) approach, specifically for the stacked deep learning layers present in the model under consideration.



(a) Before Normalization (b) After Normalization

Fig. 4. AEP hourly power consumption data (a) before (b) after Normalization

4.2 Evaluation Metrics

In the present study, four criteria were employed to assess the performance of the model. The indices encompassed in this set are Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Normalized Root Mean Square Error (NRMSE), and Mean Absolute Percentage Error (MAPE). The Root Mean Square Error (RMSE) is a commonly employed metric for quantifying the extent of discrepancy between the projected values generated by a model and the corresponding observed values. The system has a high degree of tolerance towards substantial errors. The Mean Absolute Error (MAE) is a metric used in academic research and statistical analysis to quantify the average magnitude of errors between predicted and actual values. The Mean Absolute Error (MAE) is a statistical metric used to quantify the average magnitude of errors within a given set of predictions, irrespective of their directions. The metric of interest exhibits a lower sensitivity to outliers in comparison to the root mean square error (RMSE). The Normalized Root Mean Square Error (NRMSE) is calculated by dividing the Root Mean Square Error (RMSE) by the range of the observed data. The purpose of this technique is to normalize the root mean square error (RMSE) and facilitate the comparison of model performance across different scales. The Mean Absolute Percentage Error (MAPE) is a metric utilized to represent the magnitude of prediction mistakes in terms of a percentage. This characteristic enables a clear and direct interpretation and facilitates comparisons across various scales.

4.3 Results and Discussion

To evaluate the applicability of our method for electric load prediction on the AEP dataset, we conducted a series of experiments aimed at comparing the performance of our proposed models. To ensure a comprehensive assessment, we introduced our novel models and integrated and evaluated well-established state-of-the-art deep learning architectures within our framework. These architectures represent cutting-edge approaches in the field of deep learning. We could perform direct head-to-head by incorporating these advanced architectures into our work. comparisons, providing valuable insights into the effectiveness and innovation of our proposed methods.

Table 1 presents the variations observed in the predictive outcomes of various models. providing the values for MAE, MAPE, MSE, and RMSE for the different models. Our model gives the lowest RMSE (0.01997261) , MAE (0.109563443), and NRMSE (0.182292656), while the (CNN GRU encoder-decoder) achieves the lowest MAPE (77.82%). For instance, the (LSTM encoder-decoder) exhibits a relatively high RMSE value, implying a larger deviation between predicted and actual consumption values. The (GRU encoder-decoder) and (encoder-decoder GRU) models demonstrate similar performance across most metrics, suggesting their comparable effectiveness in handling time series forecasting tasks. These results emphasize the importance of bidirectional recurrent connections in capturing temporal dependencies. The (encoder-decoder LSTM dilated causal CNN) and (encoder-decoder causal CNN LSTM) models demonstrate comparable performance in most metrics. This hints at the potential of combining dilated causal convolutional layers with LSTM structures for capturing local and global electricity consumption patterns. Overall, the (encoder-decoder CNN LSTM) model emerges as a notable candidate, delivering promising results across multiple error metrics. Its balanced performance suggests its effectiveness in capturing both spatial and temporal features in the data, making it a strong contender for electricity consumption forecasting applications.

Table 1
Errors of the prediction result in different models

Algorithm	RMSE	MAE	NRMSE	MAPE
GRU Encoder-Decoder	0.022510408	0.118326365	0.190240003	78.95109703
Encoder-Decoder GRU	0.021486559	0.115697263	0.185713634	79.0329015
CNN Encoder-Decoder GRU	0.023046887	0.12268567	0.187853119	80.7012239
Encoder-Decoder CNN GRU	0.021107744	0.11368845	0.185663051	79.5901271
CNN GRU Encoder-Decoder	0.02587543	0.13164793	0.196550217	77.8205254
GRU CNN Encoder-Decoder	0.021721118	0.1163123	0.186748255	80.3584475
LSTM Encoder-Decoder	0.026619573	0.1361029	0.195584177	81.7614897
Encoder-Decoder LSTM	0.020211411	0.11098989	0.182101364	80.2165176
CNN Encoder-Decoder LSTM	0.022869683	0.12343165	0.18528216	79.1671769
CNN LSTM Encoder-Decoder	0.02303152	0.12267128	0.187749889	80.5441498
LSTM CNN Encoder-Decoder	0.022938372	0.12211122	0.187848195	78.4883429
Encoder-Decoder LSTM dilated causal CNN	0.022037947	0.11603436	0.189926046	79.58317607
Encoder-Decoder causal CNN LSTM	0.022055267	0.12150262	0.181520917	79.6837816
Proposed Model	0.019972611	0.10956344	0.182292656	80.0328196

In Figure 5 present the results of single-step short-term electrical load forecasting using our proposed encoder-decoder CNN-LSTM model. The figure compares the actual power consumption data and the model’s predicted values over a specific period. The blue line in the plot represents the actual power consumption data, which serves as a reference for the ground truth values. On the other hand, the orange line depicts the model’s predicted power consumption values. By visually examining the figure, we can assess the model’s performance in capturing the dynamic behaviour of electricity demand. The encoder-decoder CNN-LSTM model demonstrates its efficacy in producing accurate and precise forecasts. The alignment between the predicted and actual lines indicates that the model successfully captures the underlying patterns in the data. This includes accurately predicting peak load periods, seasonal variations, and short-term fluctuations, which are essential for effective power system operation and planning. The proximity of the predicted and actual lines indicates the model’s ability to capture short-term dependencies and adapt to real-time variations in electricity consumption. This level of accuracy is crucial for electricity providers and energy management teams, as it enables them to make well-informed decisions in allocating resources, optimizing energy generation, and mitigating potential grid disruptions.

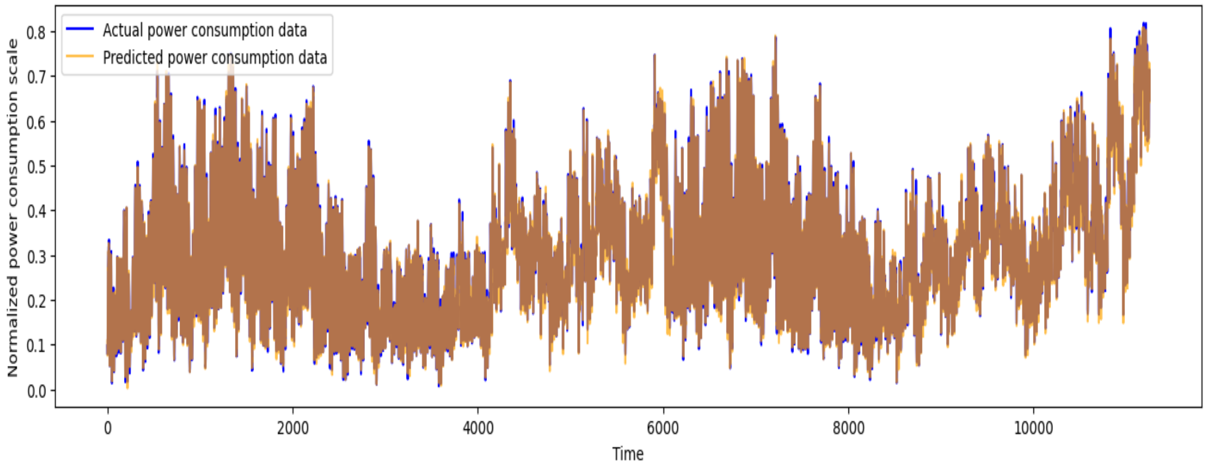


Fig. 5. Prediction values of the proposed model

5. Conclusions

Short-term electrical load forecasting ensures power production and scheduling processes' safety, stability, and sustainability. Improved prediction results have significant implications for electricity industries and power supply companies, enabling them to make reliable decisions regarding operation control, power system management, cost reduction, and pollution prevention. The present study provides a novel hybrid model that considerably improves the accuracy and dependability of single-step short-term electrical load forecasting. The proposed model successfully captures temporal dependencies, extracts spatial features, and retains long-range dependencies, yielding highly accurate load predictions. It does this by fusing the strengths of the conventional Bi-GRU based encoder-decoder structure with the CNN-LSTM architecture. The model's performance has been thoroughly assessed through comprehensive testing using the publicly available AEP dataset, which represents actual electrical load data. The results demonstrate that the presented hybrid model outperforms competing models regarding algorithm stability and prediction accuracy. This research contributes to advancing deep learning techniques for time-series power load prediction, offering a practical and reliable method for effectively operating and planning power systems. The precise forecasting capabilities of the proposed model can lead to optimized energy management, cost savings, and improved overall efficiency in the power industry. The hybrid model presented in this paper represents a significant step forward in electrical load forecasting, paving the way for more accurate and efficient power system management. Further investigations will explore the model's adaptation to other datasets and evaluate its performance under various load patterns and environmental circumstances. Incorporating cutting-edge methods and architectures may also significantly improve the model's forecasting abilities, given the rapid evolution of deep learning.

Acknowledgement
This research was not funded by any grant.

References

[1] Zhang, Zhenxing, Rennong Yang, and Yuhuan Fang. "LSTM network based on on antlion optimization and its application in flight trajectory prediction." In *2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, pp. 1658-1662. IEEE, 2018. https://doi.org/10.1109/IMCEC.2018.8469476

[2] Hajiaghayi, Mahdi, and Ehsan Vahedi. "Code failure prediction and pattern extraction using LSTM networks." In *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, pp. 55-62. IEEE, 2019. https://doi.org/10.1109/BigDataService.2019.00014

[3] Lu, An, Ling Yu, and Lihua Tan. "Apso-based optimization algorithm of lstm neural network model." In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 2194-2200. IEEE, 2021. https://doi.org/10.1109/IAEAC50856.2021.9390997

[4] Morchid, Mohamed. "Parsimonious memory unit for recurrent neural networks with application to natural language processing." *Neurocomputing* 314 (2018): 48-64. https://doi.org/10.1016/j.neucom.2018.05.081

[5] Pang, Yuexia, Yongxiu He, Jie Jiao, and Hua Cai. "Power load demand response potential of secondary sectors in China: The case of western Inner Mongolia." *Energy* 192 (2020): 116669. https://doi.org/10.1016/j.energy.2019.116669

[6] Jacob, Maria, Cláudia Neves, and Danica Vukadinović Greetham. *Forecasting and assessing risk of individual electricity peaks*. Springer Nature, 2020. https://doi.org/10.1007/978-3-030-28669-9

[7] González-Briones, Alfonso, Guillermo Hernández, Tiago Pinto, Zita Vale, and Juan M. Corchado. "A review of the main machine learning methods for predicting residential energy consumption." In *2019 16th International Conference on the European Energy Market (EEM)*, pp. 1-6. IEEE, 2019. https://doi.org/10.1109/EEM.2019.8916406

[8] Yi, Sheng-Lun, Xue-Bo Jin, Ting-Li Su, Zhen-Yun Tang, Fa-Fa Wang, Na Xiang, and Jian-Lei Kong. "Online denoising based on the second-order adaptive statistics model." *Sensors* 17, no. 7 (2017): 1668. https://doi.org/10.3390/s17071668

[9] Liu, Zhao, Xiaohong Wang, Qiang Zhang, and Chunying Huang. "Empirical mode decomposition based hybrid ensemble model for electrical energy consumption forecasting of the cement grinding process." *Measurement* 138 (2019): 314-324. https://doi.org/10.1016/j.measurement.2019.02.062

[10] Holt, Charles C. "Forecasting seasonals and trends by exponentially weighted moving averages." *International journal of forecasting* 20, no. 1 (2004): 5-10. https://doi.org/10.1016/j.ijforecast.2003.09.015

[11] Fan, Chengliang, Yunfei Ding, and Yundan Liao. "Analysis of hourly cooling load prediction accuracy with data-mining approaches on different training time scales." *Sustainable Cities and Society* 51 (2019): 101717. https://doi.org/10.1016/j.scs.2019.101717

[12] Liao, Zhijing, Nian Gai, Peter Stansby, and Guang Li. "Linear non-causal optimal control of an attenuator type wave energy converter m4." *IEEE Transactions on Sustainable Energy* 11, no. 3 (2019): 1278-1286. https://doi.org/10.1109/TSTE.2019.2922782

[13] Jin, Xue-Bo, Nian-Xiang Yang, Xiao-Yi Wang, Yu-Ting Bai, Ting-Li Su, and Jian-Lei Kong. "Hybrid deep learning predictor for smart agriculture sensing based on empirical mode decomposition and gated recurrent unit group model." *Sensors* 20, no. 5 (2020): 1334. https://doi.org/10.3390/s20051334

[14] Syed, Dabeeruddin, Haitham Abu-Rub, Ali Ghrayeb, and Shady S. Refaat. "Household-level energy forecasting in smart buildings using a novel hybrid deep learning model." *IEEE Access* 9 (2021): 33498-33511. https://doi.org/10.1109/ACCESS.2021.3061370

[15] Bian, Haihong, Yiqun Zhong, Jianshuo Sun, and Fangchu Shi. "Study on power consumption load forecast based on K-means clustering and FCM–BP model." *Energy Reports* 6 (2020): 693-700. https://doi.org/10.1016/j.egyr.2020.11.148

[16] Liu, Tao, Zehan Tan, Chengliang Xu, Huanxin Chen, and Zhengfei Li. "Study on deep reinforcement learning techniques for building energy consumption forecasting." *Energy and Buildings* 208 (2020): 109675. https://doi.org/10.1016/j.enbuild.2019.109675

[17] Tee, Chee Yin, Pay Jun Liew, Intan Sharhida Othman, Siang Yee Chang, Raja Izamshah Raja Abdullah, and Jingsi Wang. "Electrical Discharge Coating of Mild Steel with Fly Ash Suspension." *Journal of Advanced Research in Applied Mechanics* 112, no. 1 (2023): 1-9. https://doi.org/10.37934/aram.112.1.19

[18] Kang, Taehyung, Dae Yeong Lim, Hilal Tayara, and Kil To Chong. "Forecasting of power demands using deep learning." *Applied Sciences* 10, no. 20 (2020): 7241. https://doi.org/10.3390/app10207241

[19] Rahman, Aowabin, Vivek Srikumar, and Amanda D. Smith. "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks." *Applied energy* 212 (2018): 372-385. https://doi.org/10.1016/j.apenergy.2017.12.051

[20] Sajjad, Muhammad, Zulfiqar Ahmad Khan, Amin Ullah, Tanveer Hussain, Waseem Ullah, Mi Young Lee, and Sung Wook Baik. "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting." *Ieee Access* 8 (2020): 143759-143768. https://doi.org/10.1109/ACCESS.2020.3009537

[21] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014).

[22] Sehovac, Ljubisa, Cornelius Nesen, and Katarina Grolinger. "Forecasting building energy consumption with deep learning: A sequence to sequence approach." In *2019 IEEE International Congress on Internet of Things (ICIOT)*, pp. 108-116. IEEE, 2019. https://doi.org/10.1109/ICIOT.2019.00029

[23] Du, Shengdong, Tianrui Li, Yan Yang, and Shi-Jinn Horng. "Multivariate time series forecasting via attention-based encoder–decoder framework." *Neurocomputing* 388 (2020): 269-279. https://doi.org/10.1016/j.neucom.2019.12.118

[24] Gao, Xin, Xiaobing Li, Bing Zhao, Weijia Ji, Xiao Jing, and Yang He. "Short-term electricity load forecasting model based on EMD-GRU with feature selection." *Energies* 12, no. 6 (2019): 1140. https://doi.org/10.3390/en12061140

[25] Cho, Kyunghyun, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." *arXiv preprint arXiv:1406.1078* (2014). https://doi.org/10.3115/v1/D14-1179

[26] Hameduddin, Ismail, Charles Meneveau, Tamer A. Zaki, and Dennice F. Gayme. "Geometric decomposition of the conformation tensor in viscoelastic turbulence." *Journal of Fluid Mechanics* 842 (2018): 395-427. https://doi.org/10.1017/jfm.2018.118

[27] Zainab, Ameema, and Dabeeruddin Syed. "Deployment of deep learning models on resource-deficient devices for object detection." In *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 73-78. IEEE, 2020. https://doi.org/10.1109/ICIoT48696.2020.9089651

[28] Somu, Nivethitha, Gauthama Raman MR, and Krithi Ramamritham. "A deep learning framework for building energy consumption forecast." *Renewable and Sustainable Energy Reviews* 137 (2021): 110591. https://doi.org/10.1016/j.rser.2020.110591

[29] Mulla, Rob. "Hourly energy consumption." *Distributed by Kaggle: Public Dataset Storage* (2018).

[30] Khattak, Muhammad Adil, Tuan Mohamad Hakimi Tuan Yahya, Muhammad Wazif Mohd Sallehhudin, Muhammad Izzuan Mohd Ghazali, Nur Awal Abdullah, Nur Athirah Nordin, and Nur Ainida Mohd Mohsein Dzakurnain. "Energy Security Policy Shift of North America and Ontario, Canada Following 2003 Power Blackout: A Review." *Progress in Energy and Environment* (2018): 14-24.