SHORT-TERM FORECASTING OF ELECTRICAL LOAD DEMAND IN HANOI BASED ON EXTREME LEARNING MACHINE MODEL

Nguyen Thi Hoai Thu*, Pham Nang Van, Ngo Van Khanh
School of Electrical and Electronic Engineering - Hanoi University of Science and Technology

ARTICLE INFO

Received: **20/3/2024**
Revised: **31/5/2024**
Published: **31/5/2024**

KEYWORDS

Short-term forecasting
Load forecasting
Extreme Learning Machine
Machine Learning
Single-hidden-layer feed-forward neural networks

ABSTRACT

Accurate forecasting of the electrical load is a critical element for grid operators to make well-informed decisions concerning electricity generation, transmission, and distribution. In this study, an Extreme Learning Machine (ELM) model was proposed and compared with four other machine learning models including Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The dataset utilized for evaluating machine learning models were procured from the statistical analysis of the electrical load in the city of Hanoi, Vietnam. Prior to its utilization, the dataset underwent preprocessing procedures involving the removal of outliers and handling of missing values, thereby enhancing the computational efficiency of the models. According to the study results, the proposed model has superior performance when compared with the other four models, achieving the lowest error value. These outcomes substantiate the efficacy of the model, making it a good option for short-term load forecasting.

DỰ BÁO NGẮN HẠN PHỤ TẢI ĐIỆN HÀ NỘI DỰA TRÊN MÔ HÌNH MÁY HỌC CỰC TRỊ

Nguyễn Thị Hoài Thu*, Phạm Năng Văn, Ngô Văn Khánh
Trường Điện - Điện tử - Đại học Bách khoa Hà Nội

THÔNG TIN BÀI BÁO

Ngày nhận bài: **20/3/2024**
Ngày hoàn thiện: **31/5/2024**
Ngày đăng: **31/5/2024**

TỪ KHÓA

Dự báo ngắn hạn
Dự báo phụ tải
Máy học cực trị
Máy học
Mạng nơ-ron truyền thẳng một lớp ẩn

TÓM TẮT

Việc dự báo chính xác phụ tải điện là một yếu tố quan trọng để các kĩ sư vận hành lưới điện đưa ra các quyết định chính xác về sản xuất, truyền tải và phân phối điện. Trong nghiên cứu này, một mô hình máy học cực (Extreme Learning Machine) đã được đề xuất và so sánh với bốn mô hình học máy khác bao gồm mạng thần kinh nhân tạo (Artificial Neural Networks), mạng nơron tích chập (Convolutional Neural Networks), bộ nhớ dài-ngắn hạn (Long Short-Term Memory) và nút hồi tiếp có cổng (Gated Recurrent Unit). Bộ dữ liệu được sử dụng để đánh giá các mô hình được lấy từ dữ liệu phụ tải điện tại thành phố Hà Nội, Việt Nam. Trước khi được sử dụng, dữ liệu được tiền xử lý qua các bước bao gồm việc loại bỏ các giá trị nhiễu và bổ sung các giá trị còn thiếu, nhằm tối ưu hóa khả năng tính toán của các mô hình. Theo kết quả nghiên cứu, mô hình đề xuất có hiệu suất vượt trội khi so sánh với 4 mô hình còn lại với giá trị sai số nhỏ nhất. Những kết quả này đã chứng minh tính hiệu quả của mô hình, khiến nó trở thành một lựa chọn tốt để dự báo phụ tải ngắn hạn.

DOI: https://doi.org/10.34238/tnu-jst.9923

* Corresponding author. *Email: thu.nguyenthihoai@hust.edu.vn*

1. Introduction

Many elements, such as weather conditions and economic factors, are considered when a load forecast for an electric system is performed. Forecasting is an important element in the planning and operation of power systems. It allows electricity companies to efficiently manage their resources, ensuring that the electricity demand can be met. Accurate forecasting of the electrical load can lead to significant cost savings by reducing the need for expensive power plants and minimizing the risk of blackouts. Furthermore, with the increasing integration of renewable energy sources, which are inherently volatile, load forecasting has become more important. With load forecasting models, grid operators can manage the power grid more efficiently and easily, ensuring a reliable source of energy.

Electrical load forecasting can be divided into four categories: very short-term load forecasting, short-term load forecasting, medium-term load forecasting, and long-term load forecasting [1]. Among these types of forecasting, short-term load forecasting (STLF), which forecasts for a period ranging from a few minutes or hours up to one day, aims at economic dispatch and optimal generator unit commitment while addressing real-time control and security assessment [2]. This paper will simulate the impact of machine learning models based on short-term load forecasting.

Over the past decade, data scientists have discovered and developed a variety of forecasting models. These models, which are most commonly used, can be classified into four distinct categories: Statistical methods (time series-based methods), Artificial Intelligence methods, Physical methods, and Mixed methods (hybrid or ensemble methods) [3]. Statistical methods encompass models such as Autoregressive Moving Average (ARMA) [4], [5], and Autoregressive Integrated Moving Average (ARIMA) [6]. While statistical methods were employed in the past, they have been largely surpassed by other methods due to their relative inferiority, even when used in conjunction with machine learning models.

In recent years, the adoption and utilization of various machine learning models have seen a significant surge owing to their demonstrably superior performance compared to traditional methodologies. Notably, models such as the Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) have emerged as prominent choices in diverse applications. However, among these models, the Extreme Learning Machine (ELM) has garnered particular attention owing to its distinctive characteristics and computational efficiency. Unlike conventional neural network architectures, the ELM stands out for its randomized generation of hidden layers, a feature that endows it with rapid convergence speed and commendable performance metrics [7]. Consequently, the model has applications in many areas which also include forecasting [8], [9].

In this paper, we present an Extreme Learning Machine (ELM) model designed for short-term power forecasting. The dataset underwent preprocessing before being utilized by the ELM model. To assess the effectiveness of the ELM model, its results were compared with those of four other machine learning models: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). Section 2 outlines the fundamental methodologies employed, including data preprocessing techniques and the ELM forecasting model. Section 3 presents the prediction outcomes derived from the dataset along with a comparative analysis between the proposed model and the aforementioned four models. Finally, in Section 4, we present the conclusions drawn from this study.

2. Methodology

2.1. Data collection and data preprocessing

The data set utilized by the proposed and comparison models in this study is the recorded load data of Hanoi in 2018. The data was collected from 1/1/2018 0:00 to 12/31/2018 23:00 with a sample every 1 hour. Due to this sample interval, the number of data points were 8760 data point. A visualization of the raw load data is presented in Figure 1. The load data was recorded in MW.

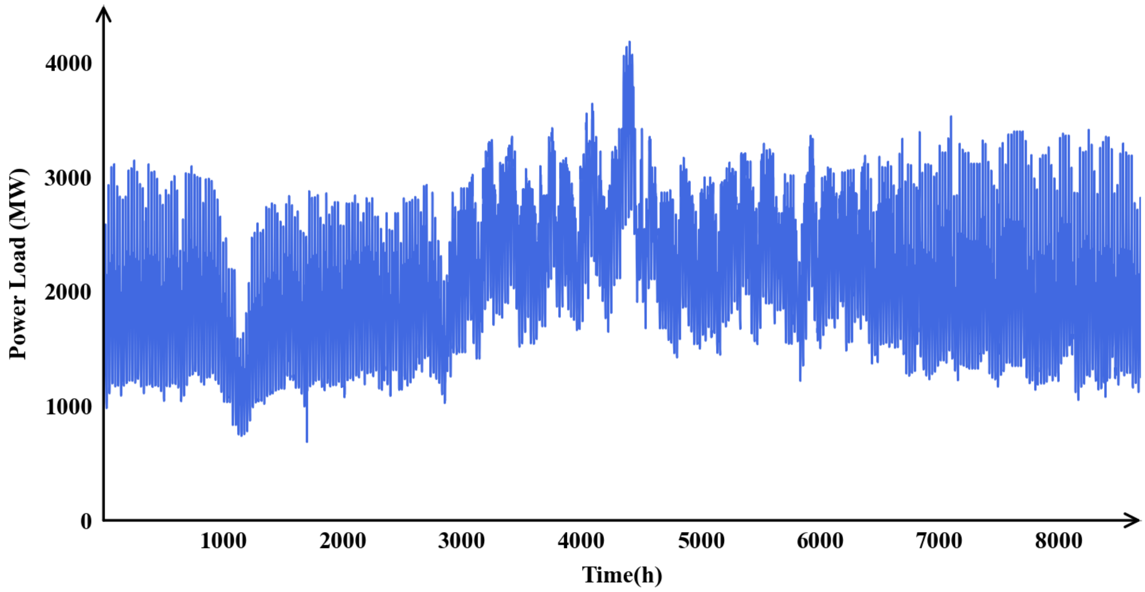

Figure 1. *Hanoi load data in 2018*

Outliers, which are data points significantly different from the rest of the dataset and can have a substantial impact on model performance [10]. Similarly, the presence of missing values represents a common issue that can disrupt analytical processes and model training if left unresolved. Consequently, data preprocessing plays a pivotal role in ensuring the quality and reliability of the data used for the models. In this paper, both of these issues were addressed by using graph visualization techniques. Initially, a graphical representation was generated from the original dataset to facilitate the evaluation process. Subsequently, this graphical representation was scrutinized to identify outliers and missing values, thus facilitating further processing. Any identified anomalies or missing data were subsequently corrected or eliminated from the dataset, ensuring its suitability for subsequent model training endeavors.

Apart from outliers and missing values, it is noteworthy that many machine learning models are sensitive to data scales, leading to the need for data scaling. In this paper, max-min normalization and z-score standardization were used to carry out the scaling process. Max-min normalization rescales the data values to a range between 0 and 1 while z-score standardization is used to transform the values to be normally distributed with a mean of zero and a standard deviation of one [11].

2.2. Extreme Learning Machine

Feed-forward neural networks, such as Convolutional Neural Networks (CNN) or Multilayer Perceptron (MLP), have been utilized in various fields including image classification, natural language processing, and speech recognition [12] - [14]. Extreme Learning machine is a learning algorithm invented for the purpose of training Single-hidden-layer feedforward neural networks (SFLNs). Different from traditional feed-forward neural networks where all parameters need to be tuned, the input weight and biases of the hidden layer in ELM are randomly initialized [7]. Because of this, ELM has been known to perform exceptionally fast with good performance when compared to traditional feed-forward neural networks models.

For N number of random training sets (X_i, Y_i) where $\mathbf{X_i} = \left[x_{i_1}, x_{i_2}, \dots, x_{i_k}\right]^{\mathrm{T}} \in R^k$ is the model input and $\mathbf{Y_i} = \left[y_{i_1}, y_{i_2}, \dots, y_{i_m}\right]^{\mathrm{T}} \in R^m$ is the model output, L is the number of hidden nodes and g(x) is the activation function, because the model output and expected output are equal in ELM model calculation, the output function expression of the ELM can be expressed as :

$$the\ \sum_{l=1}^{L} \beta_l g(b_l + W_l \cdot X_i) = O_i = Y_i,\ i = 1, \dots, N \tag{1}$$

Where, $\mathbf{W_l} = \left[\omega_{l_1}, \omega_{l_2}, \dots, \omega_{l_k}\right]^{\mathrm{T}}$ is the input weight matrix in between input to l[th] hidden nodes, $\boldsymbol{\beta}_\mathbf{l} = \left[\beta_l, \beta_{l_2}, \dots, \beta_{l_m}\right]^{\mathrm{T}}$ is the output weight matrix in between l[th] hidden nodes to the output node. The output of ELM is considered as O_i. The specific structure of the model can be described in Figure 2.

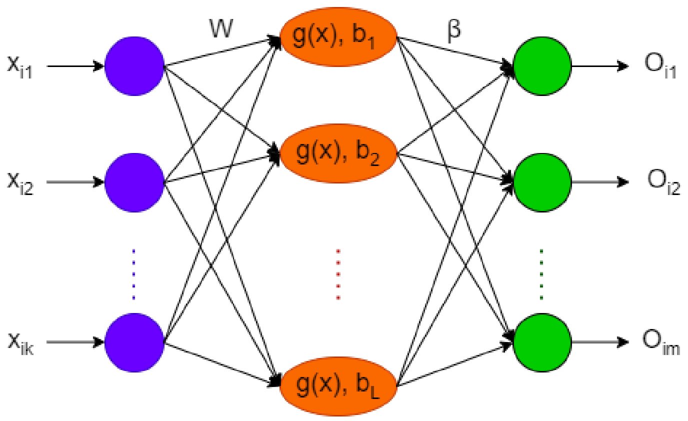

Figure 2. *The structure of ELM*

Eq. (1) can be described using Eq. (2)

$$\mathbf{H\beta} = \mathbf{Y} \tag{2}$$

In Eq. (2), **H** is the hidden layer output matrix. The formulas for H, β, and Y are shown in Eqs. (3) and (4), respectively.

$$\mathbf{H} = \begin{bmatrix} g(X_1 \cdot W_1 + b_1) & \cdots & g(X_1 \cdot W_L + b_L) \\ \vdots & \ddots & \vdots \\ g(X_N \cdot W_1 + b_1) & \cdots & g(X_N \cdot W_L + b_L) \end{bmatrix}_{N \times L} \tag{3}$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \mathbf{Y} = \begin{bmatrix} Y_1^T \\ \vdots \\ Y_N^T \end{bmatrix}_{N \times m} \tag{4}$$

In the ELM model parameter training process, if $\bar{W}_l, \bar{b}_l$, and $\bar{\beta}_l$ can make Eq. (5) hold:

$$E = \min_{W,b,\beta} \sum_{i=1}^{N} \left(\sum_{l=1}^{L} \bar{\beta}_l g\left(\bar{W}_l \cdot X_i + \bar{b}_l\right) - Y_i \right)^2 = \min_{\mathrm{W,b,\beta}} \parallel \mathbf{H\beta} - \mathbf{Y} \parallel \tag{5}$$

Then $\bar{W}_l, \bar{b}_l$ and $\bar{\beta}_l$ are the optimal ELM model parameters. In the ELM model, once the input weight W of the model and the hidden layer threshold b are determined, the output matrix **H** of the hidden layer is uniquely determined. Under this condition, the ELM learning process can be transformed into a linear system (6).

$$\bar{\boldsymbol{\beta}} = \mathbf{H^{-1}Y} \tag{6}$$

In Eq. (6), $\mathbf{H^{-1}}$ is a generalized inverse matrix. It can be seen from the above discussion that the ELM training process involves continuously seeking the optimal solution of the nonlinear system.

2.3. ELM modeling and forecasting process

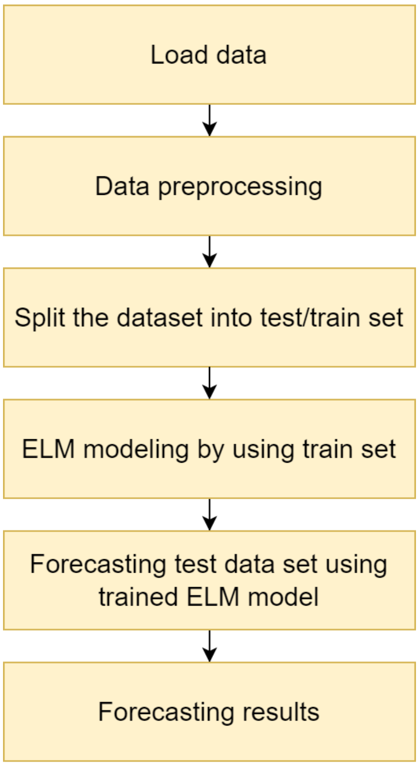

Figure 3. *Flowchart of the forecasting process*

Figure 3 illustrates the forecasting process through a flowchart. Initially, the load data is acquired and subjected to preprocessing, employing techniques such as data cleaning or normalization. Subsequently, the dataset was divided into a training set and test set with a distribution ratio of 90/10. The training set was employed to train the proposed model, whereas the test set was allocated to conduct the forecasting process. The forecasting results are obtained upon the completion of the forecasting process.

The input data for the models comprised 48 historical data points extracted from the dataset sampled at hourly intervals, including the data from time t to t-47. The output of the forecasting models varies depending on the number of forecasting steps, including data at time t+1, t+2,..., t+ k where k is the number of forecasting steps. For instance, in single-step forecasting, the output corresponds to t+1, while in 3-step forecasting, the output comprises the data at t+1, t+2, and t+3.

2.4. Error metrics

In this study, three error metrics were employed for the evaluation of the proposed model. These metrics include the Root Mean Square Error (RMSE), normalized RMSE (n-RMSE), and Mean Absolute Percentage Error (MAPE). The lower the value of these error metrics, the better the forecasting result. RMSE, n-RMSE and MAPE are defined as follows:

$$RMSE = \sqrt{\frac{\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{O_i} - \mathrm{E_i})^2}{\mathrm{N}}} \tag{7}$$

$$N\_RMSE = \frac{\mathrm{RMSE}}{\bar{\mathrm{O}}} \times 100 \tag{8}$$

$$MAPE = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} \left| \frac{O_i - \mathrm{E}_i}{O_i} \right| \tag{9}$$

Where:
N : Number of sample data
O_i: Actual load value
$\overline{O}$: Mean value of Actual load value
E_i: Estimated load value

Typically, RMSE and MAPE serve as standard metrics for assessing the forecasting performance of a model [15] - [17]. Additionally, N-RMSE is also utilized to gauge the accuracy of forecasting models. In literatures, the accuracy of the model was considered excellent when N-RMSE < 10%; good if 10% < N-RMSE < 20%; fair if 20% < N-RMSE < 30% and poor if N-RMSE ≥ 30% [18], [19].

3. Result and Discussion

3.1. Hyperparameter of evaluated models

For each of comparative model, we tested several set of hyperparameters for them and then choose the hyperparameters which obtained the best accuracy among them. The hyperparameters of the models are chosen and outlined in Table 1 and 2. In Table 1, the Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models featured a single dense layer, while the Artificial Neural Network (ANN) incorporated two dense layers, and the Gated Recurrent Unit (GRU) model lacked any dense layers. Moreover, the CNN employed one convolutional layer, the LSTM comprised two LSTM layers, and the GRU similarly utilized two GRU layers. Notably, the ReLU activation function was employed across all four models.

In Table 2, the hyperparameters of the Extreme Learning Machine (ELM) are delineated, comprising four components. These include the number of hidden neurons, the mixing coefficient for distance and dot product input activations denoted as alpha, and the multiplier for the radial basis activation function referred to as rbf_width. The activation function employed by the ELM model is the Multiquadric activation function.

All models were executed on a system equipped with AMD Ryzen 5 5500U with Radeon Graphics and 16GB of LPDDR4x RAM. Python and libraries such as TensorFlow and Scikit-learn were utilized to construct hyperparameters for the machine learning models. All models utilized the 2018 Hanoi load dataset with a consistent 90/10 train/test set ratio.

Table 1. *Layer configuration for ANN, CNN, LSTM and GRU*

Hyperparameter	**Models**			
	ANN	**CNN**	**LSTM**	**GRU**
Conv1D		64		
MaxPooling		2		
LSTM1			100	
LSTM2			50	
GRU1				100
GRU2				32
Dropout	0.2	0.1	0.2	0.2
Dense1	100	100	32	
Dense2	32			
Activation Function	ReLU	ReLU	ReLU	ReLU

Table 2. *Layer configuration for ELM*

Hyperparameter	**Model**
	ELM
Hidden neurons	50
Alpha	0.7
rbf_width	5
Activation Function	Multiquadric

3.2. Comparision of forecasting models by error metrics

To prove the accuracy of ELM models in load forecasting, this paper employed 4 deep-learning models, including: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). Table 3 presents the errors acquired from the 5 models trained by Hanoi load data. The result will be evaluated under 3 error metrics: RMSE, n-RMSE, and MAPE, with 1, 3, 6, 12 and 24-step. While the n-

RMSE serves as the comprehensive evaluation metric for the models, RMSE and MAPE are utilized for comparative analysis of the proposed model against each of the other models.

Overall, it is evident that the proposed model has outperformed the other machine-learning models. In terms of 1-step, the proposed model outperformed other models, reaching a RMSE of 91.0 MW. Conversely, GRU demonstrated the poorest performance in terms of RMSE among the five models, with a RMSE of 146.8 MW. The RMSE values for ANN, CNN, and LSTM were all relatively similar, hovering around 136 MW. Regarding MAPE, the LSTM model performed the best at 4.5%, while CNN performed the worst at 5.7%. Figure 4(a) presents the actual load and the predicted load generated by the evaluated models for the 1-step forecast.

For 3, 6, and 12-step forecasting, the ELM model consistently outperformed all five models, while GRU consistently performed the worst compared to the other four models. The RMSE values for ELM for 3, 6, and 12-step forecasting were 118.5 MW, 132.7 MW, and 146.5 MW, respectively. Consequently, the MAPE values for the proposed model were 4.6%, 5.5%, and 5.7%, respectively. Figure 4(b), Figure 4(c), and Figure 4(d) illustrate the actual load and the forecasted load by all five models for 3, 6, and 12-step forecasts.

For 24-step forecasting, the ELM model was still the best-performing model while GRU was the worst-performing model. The RMSE and MAPE values for ELM were 153.3 MW and 5.9%, respectively, while the corresponding error metrics for GRU were 203.7 MW and 8.3%, respectively. Figure 4(e) illustrates the results produced by the models alongside the actual load data for the 24-step forecast.

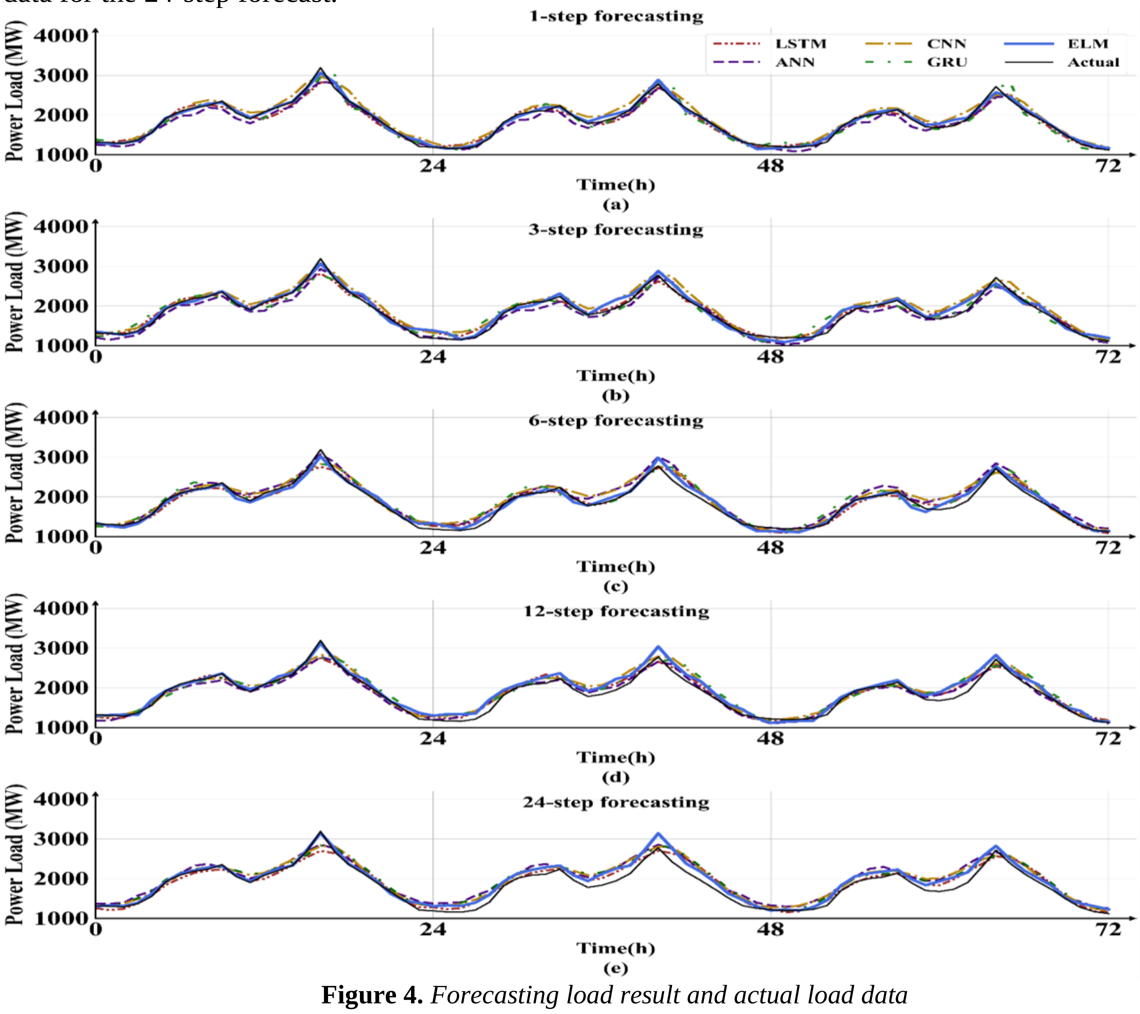

Figure 4. *Forecasting load result and actual load data*

Table 3. *Forecasting results in error metrics of the proposed model and the models used for comparison*

Model used	Error metrics	Forecasting step				
		1-step	3-step	6-step	12-step	24-step
ANN	RMSE	136.7	144.9	157.1	167.0	196.5
	N-RMSE	5.8	6.1	6.7	7.1	8.3
	MAPE	5.5	6.1	6.6	6.4	8.2
CNN	RMSE	139.0	143.4	167.6	172.5	200.6
	N-RMSE	5.9	6.1	7.1	7.3	8.5
	MAPE	5.7	6.2	6.4	6.2	7.8
LSTM	RMSE	134.1	147.4	156.3	168.9	202.5
	N-RMSE	5.8	6.3	6.6	7.2	8.6
	MAPE	4.5	5.5	6.6	6.9	7.6
GRU	RMSE	146.8	160.1	171.6	180.3	203.7
	N-RMSE	6.2	6.8	7.3	7.6	8.6
	MAPE	5.1	6.3	6.7	6.8	8.3
ELM	RMSE	91.0	118.5	132.7	146.5	153.3
	N-RMSE	3.9	5.0	5.6	6.2	6.5
	MAPE	3.0	4.6	5.5	5.7	5.9

3.3. Comparision of forecasting models by training and testing time

Model evaluation may also encompass an analysis of the training and testing durations for each model. The training time denotes the duration required for model training, while the testing time signifies the duration for the trained model to produce the output.

From the data presented in Table 4, it becomes apparent that LSTM and GRU exhibited the least favorable performances in terms of both training and testing time among all models. Conversely, the ELM demonstrated superior efficiency among the five models, with a training time of approximately 1 second and a testing time of only around 0.02 seconds. This outcome underscores the efficacy of ELM's ability to randomly generate weights, thereby facilitating expedited learning processes and mitigating computational complexities.

Although inferior to the ELM, both ANN and CNN showcased significantly lower training and testing times compared to LSTM and GRU, with the combined training and testing durations for both models falling below 1 minute.

It is noteworthy, however, that the sampling interval remained consistent at 1 hour, and the combined testing and training durations for each model remained below 10 minutes. Consequently, the impact of training and testing times on the overall performance of the models is anticipated to be minimal.

Table 4. *Forecasting results in error metrics of the proposed model and the models used for comparison*

Model used	Metrics	Forecasting step				
		1-step	3-step	6-step	12-step	24-step
ANN	Training time (min:sec)	00:13.10	00:23.80	00:21.20	00:23.23	00:23.61
	Testing time (min:sec)	00:00.40	00:00.82	00:00.40	00:00.67	00:00.47
CNN	Training time (min:sec)	00:50.88	00:51.16	00:46.54	00:56.75	00:51.70
	Testing time (min:sec)	00:00.35	00:00.53	00:00.57	00:00.37	00:00.39
LSTM	Training time (min:sec)	04:43.69	05:08.15	05:42.79	07:32.95	09:48.62
	Testing time (min:sec)	00:03.66	00:03.36	00:03.12	00:02.33	00:01.96
GRU	Training time (min:sec)	07:27.48	06:21.53	06:56.40	07:16.82	06:48.72
	Testing time (min:sec)	00:03.08	00:03.72	00:02.13	00:02.48	00:03.06
ELM	Training time (min:sec)	00:00.48	00:00.64	00:00.77	00:00.73	00:00.76
	Testing time (min:sec)	00:00.02	00:00.01	00:00.02	00:00.02	00:00.02

4. Conclusion

In this study, an Extreme Learning Machine (ELM) model is introduced and evaluated utilizing three distinct error metrics: Root Mean Square Error (RMSE), normalized RMSE (n-RMSE), and Mean Absolute Percentage Error (MAPE). The performance of the proposed model is assessed in comparison to four other machine learning models, specifically Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU). The evaluation is conducted using electrical load data from 2018 in Hanoi. Prior to forecasting, rigorous data preprocessing procedures are employed. The findings indicate that the proposed ELM model demonstrates superior efficacy compared to the aforementioned forecasting models, exhibiting the highest accuracy across all error criteria in contrast to its counterparts. It is noted that achieving better forecasting performance in future research can be done by combining the proposed model with other techniques such as decomposition or creating hybrid models.

REFERENCES

[1] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, "Electricity load forecasting: a systematic review," *Journal of Electrical Systems and Inf. Technol.*, vol. 7, no. 1, p. 13, Dec. 2020, doi: 10.1186/s43067-020-00021-8.

[2] A. Irankhah, M. H. Yaghmaee, and S. Ershadi-Nasab, "Optimized short-term load forecasting in residential buildings based on deep learning methods for different time horizons," *Journal of Building Engineering*, vol. 84, p. 108505, May 2024, doi: 10.1016/j.jobe.2024.108505.

[3] N. Q. Nguyen, L. D. Bui, B. V. Doan, E. R. Sanseverino, D. D. Cara, and Q. D. Nguyen, "A new method for forecasting energy output of a large-scale solar power plant based on long short-term memory networks a case study in Vietnam," *Electric Power Systems Research*, vol. 199, p. 107427, Oct. 2021, doi: 10.1016/j.epsr.2021.107427.

[4] K. Benmouiza and A. Cheknane, "Small-scale solar radiation forecasting using ARMA and nonlinear autoregressive neural network models," *Theor. Appl. Climatol.*, vol. 124, no. 3–4, pp. 945–958, May 2016, doi: 10.1007/s00704-015-1469-z.

[5] W. Wang, "Improved short term load forecasting of power system based on ARMA model," in *Proceedings of the 2016 International Conference on Engineering Management (Iconf-EM 2016)*, Guangzhou City, China: Atlantis Press, 2016, doi: 10.2991/iconfem-16.2016.2.

[6] C. Tarmanini, N. Sarma, C. Gezegin, and O. Ozgonenel, "Short term load forecasting based on ARIMA and ANN approaches," *Energy Reports*, vol. 9, pp. 550–557, May 2023, doi: 10.1016/j.egyr.2023.01.060.

[7] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[8] Z. Zhu *et al.*, "A day-ahead industrial load forecasting model using load change rate features and combining FA-ELM and the AdaBoost algorithm," *Energy Reports*, vol. 9, pp. 971–981, Dec. 2023, doi: 10.1016/j.egyr.2022.12.044.

[9] B. Gu, H. Hu, J. Zhao, H. Zhang, and X. Liu, "Short-term wind power forecasting and uncertainty analysis based on FCM–WOA–ELM–GMM," *Energy Reports*, vol. 9, pp. 807–819, Dec. 2023, doi: 10.1016/j.egyr.2022.11.202.

[10] F. E. Grubbs, "Procedures for Detecting Outlying Observations in Samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, Feb. 1969, doi: 10.1080/00401706.1969.10490657.

[11] C. Fan, M. Chen, X. Wang, J. Wang, and B. Huang, "A Review on Data Preprocessing Techniques Toward Efficient and Reliable Knowledge Discovery From Building Operational Data," *Front. Energy Res.*, vol. 9, p. 652801, Mar. 2021, doi: 10.3389/fenrg.2021.652801.

[12] P. Yan and Z. Xiang, "Acceleration and optimization of artificial intelligence CNN image recognition based on FPGA," in *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China: IEEE, Mar. 2022, pp. 1946–1950, doi: 10.1109/ITOEC53115.2022.9734423.

[13] J. A. Botha *et al.*, "Natural Language Processing with Small Feed-Forward Networks," *arXiv*, 2017, doi: 10.48550/ARXIV.1708.00214.

[14] J. Ahmad and H. A. Fatmi, “A novel method of speech recognition using feedforward neural network technology,” in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, TX, USA: IEEE, 1994, pp. 2132–2135, doi: 10.1109/ICSMC.1994.400179.
[15] W. Waheed and Q. Xu, “Data-driven short term load forecasting with deep neural networks: Unlocking insights for sustainable energy management,” *Electric Power Systems Research*, vol. 232, p. 110376, Jul. 2024, doi: 10.1016/j.epsr.2024.110376.
[16] T. Yang, Z. Yang, F. Li, and H. Wang, “A short-term wind power forecasting method based on multivariate signal decomposition and variable selection,” *Applied Energy*, vol. 360, p. 122759, Apr. 2024, doi: 10.1016/j.apenergy.2024.122759.
[17] N. Wei *et al.*, “Short-term load forecasting based on WM algorithm and transfer learning model,” *Applied Energy*, vol. 353, p. 122087, Jan. 2024, doi: 10.1016/j.apenergy.2023.122087.
[18] M.-F. Li, X.-P. Tang, W. Wu, and H.-B. Liu, “General models for estimating daily global solar radiation for different solar radiation zones in mainland China,” *Energy Conversion and Management*, vol. 70, pp. 139–148, Jun. 2013, doi: 10.1016/j.enconman.2013.03.004.
[19] P. D. Jamieson, J. R. Porter, and D. R. Wilson, “A test of the computer simulation model ARCWHEAT1 on wheat crops grown in New Zealand,” *Field Crops Research*, vol. 27, no. 4, pp. 337–350, Nov. 1991, doi: 10.1016/0378-4290(91)90040-3.