Research on the Influence of Hyperparameters on the LightGBM Model in Load Forecasting

Khanh-Toan Nguyen
Industrial University of Ho Chi Minh City, Vietnam
21141151.toan@student.iuh.edu.vn

Thanh-Ngoc Tran
Industrial University of Ho Chi Minh City, Vietnam
tranthanhngoc@iuh.edu.vn (corresponding author)

Huy-Tuan Nguyen
Go Vap Power Company, Ho Chi Minh Power Corporation (EVNHCMC), Vietnam
vuhuytuan77@gmail.com

Received: 30 June 2024 | Revised: 31 July 2024 | Accepted: 13 August 2024

 DOI: https://doi.org/10.48084/etasr.8266

ABSTRACT

Electric load forecasting plays a vital role in all aspects of the electrical system, including generation, transmission, distribution, and electricity retail. The LightGBM ensemble learning method has been widely applied in load forecasting and has yielded many positive results. This study presents an algorithm combining the grid space of hyperparameters with cross-validation to evaluate the accuracy of LightGBM models across different hyperparameter values. Peak load data from Ho Chi Minh City were used to enhance the reliability of the results. Analysis of the results based on boxplot statistical charts indicated that the accuracy of the LightGBM model significantly depends on the hyperparameter values. Moreover, using default hyperparameter values may result in large errors in load forecasting.

Keywords-load forecasting; LightGBM; cross-validation; hyperparameters

I. INTRODUCTION

Electric load forecasting estimates the future electricity consumption for a specific area, system, or grid. This forecasting is critical in assisting power plants, utility companies, and grid operators in effectively planning and managing electricity production, transmission, and distribution. Accurate load forecasting ensures an adequate supply of electricity to support socio-economic activities, such as allowing businesses to schedule production efficiently while optimizing the performance of an electrical system, minimizing losses, and reducing costs [1]. Numerous methods have been proposed for electric load forecasting, including regression methods [2, 3], exponential smoothing [4], cluster analysis methods [3], Artificial Neural Networks (ANNs) [5], machine learning models [6], deep learning [7], and ensemble learning [8]. In recent years, among state-of-the-art load forecasting methods, the LightGBM model has been proven to be effective in time series forecasting problems [9-10].

The performance of ensemble learning models, including LightGBM, generally depends on their hyperparameters. In this regard, evaluating the impact of hyperparameter values is crucial for applying the LightGBM model. To the best of our knowledge, very few studies have focused on this topic. For example, in [11], the characteristics of the learning rate (γ) and num_leaves were examined, identifying the optimal values for these two LightGBM hyperparameters. In [12], the influence of the n estimators and the learning_rate on the performance of the LightGBM model was established, obtaining the optimal values. In addition, some studies simply applied the LightGBM model with default hyperparameter values to predict loads [13-15]. Therefore, conducting a comprehensive evaluation of the influence of key hyperparameters on the performance of the LightGBM model is important for its application, especially compared to the default hyperparameter values. This study uses the grid space for hyperparameters and cross-validation procedures to evaluate their overall influence on the performance of LightGBM, utilizing peak load data from Ho Chi Minh City for experimentation. In addition, a boxplot chart was used to analyze the results under various experimental scenarios.

II. RESEARCH METHODS

A. LightGBM Model

LightGBM is a powerful machine learning algorithm that is widely applied in solving classification and regression problems. Developed by Microsoft, LightGBM is a high-performance decision tree-based model that integrates several advanced techniques such as the histogram algorithm, leaf-wise strategy, gradient-based one-side sampling, and exclusive feature bundling [16]. Combining these techniques enhances its efficiency, providing many advantages over other gradient-boosting models. The basic steps of the LightGBM model are [17]:

- Define a specific loss function: LightGBM requires a suitable loss function for the specific problem. This loss function will be optimized during the training process.
- Gradient-based one-side sampling: LightGBM uses this procedure to create subtrees. Instead of random sampling, the algorithm focuses on samples with large gradients to optimize performance.
- Histogram algorithm to identify the optimal segmentation point: This algorithm identifies the optimal segmentation point. Instead of processing each data point, the algorithm uses histograms to optimize tree splitting.
- Feature dimension by exclusive feature bundling: LightGBM can automatically combine similar features into a single feature, reducing data dimensionality and accelerating the training process.
- Leaf-wise algorithm with depth limitation: LightGBM grows trees vertically (leaf-wise) instead of horizontally (level-wise). This approach selects leaves with the most significant loss to grow, optimizing performance.
- The leaf nodes to which the samples belong are combined to fit the residuals: LightGBM combines the leaf nodes to fit the residuals of the samples, improving the model accuracy.
- Split the nodes of a tree by scoring the tree structure: LightGBM uses tree structure scores to decide how to split the nodes, optimizing the tree structure.
- Stop the growth and generate the decision tree: LightGBM halts tree growth when certain conditions are met (e.g., maximum depth, maximum number of trees). The resulting decision tree is then used for the prediction.

TABLE I. SUMMARY OF LIGHTBM HYPERPARAMETERS

Hyperparameters	Description
learning_rate	Adjusts how much the model's weights are updated at each iteration.
min_child_samples	The minimum number of samples required in a node to be split into two child nodes.
colsample_bytree	The percentage of columns to be randomly sampled and used for constructing each decision tree.
n_estimators	The number of decision trees to be built during the training process.
num_iterations	The number of iterations to train the model.
max_depth	The maximum depth of the decision trees.
num_leaves	The maximum number of leaves that a decision tree can have.
max_bin	The maximum number of bins to be used in constructing histograms.
bagging fraction	The fraction of samples that are randomly sampled for each iteration.
feature_fraction	The fraction of features (or columns) that are randomly selected to build each decision tree split.

Similarly to other machine learning models, the performance of LightGBM also depends on its hyperparameter values [18]. Table I presents the important hyperparameters of the LightGBM model and their descriptions.

B. Proposed Method

To evaluate the impact of hyperparameters on the performance of the LightGBM model, it is essential to assess model performance while varying their values around their default settings. Figure 1 illustrates an example of setting up a grid space with two hyperparameters, denoted as a and b. Hyperparameter a is configured with three values $\{a_1, a_2, a_3\}$, where a_2 is the default value. Similarly, $\{b_1, b_2, b_3\}$ are configurations for hyperparameter b, with b_2 being the default value. The combination of these two hyperparameters creates 9 parameter sets. Comparing the performance of the default combination (a_2, b_2) with the others provides a basis for assessing the role of different hyperparameter values relative to their defaults. This study set up a grid of values for fundamental hyperparameters of the LightGBM model, including colsample_bytree, n_estimators, min_child_samples, and learning_rate. This approach aims to evaluate the roles of these hyperparameters in model performance.

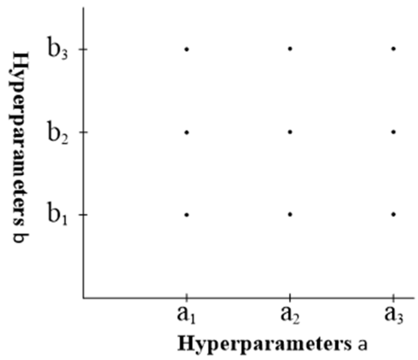

Fig. 1. A grid space model with two hyperparameters of a and b.

Ensemble learning models, specifically LightGBM, often encounter overfitting issues in which they perform well on training data but are less effective on new data. In this scenario, a technique known as k-fold cross-validation can be applied to mitigate overfitting during hyperparameter tuning processes, as shown in Figure 2 [19].

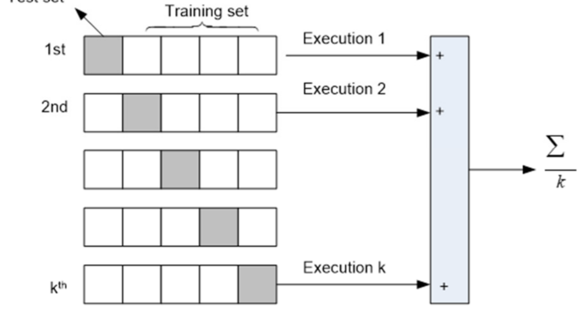

Fig. 2. The k-fold cross-validation procedure.

This technique divides the dataset into k equal parts (folds). The model is trained k times each time, using (k-1) folds for training and the remaining fold for validation. The performance of the model is measured in the validation fold in each iteration. The results from k cross-validation runs are then averaged to estimate the model performance. The k-fold cross-validation technique helps assess the generalization of a model on new datasets, thereby optimizing performance and mitigating overfitting issues for models.

An algorithm was proposed by combining the hyperparameter grid space and cross-validation, as shown in Figure 3, including the following main steps:

- Step 1: Data preprocessing. The data of the maximum electric load are processed and split into the input (X) and target (Y) datasets.
- Step 2: Grid setup based on predefined ranges of hyperparameter values. At the same time, the cross-validation procedure is established using the selected k-fold values.
- Step 3: Training the LightGBM model. The model error is measured corresponding to each combination within the grid space. The Mean Square Error (MSE) is used to estimate the discrepancy between the actual and predicted values [20]. These MSE values are used to evaluate and analyze the roles of the hyperparameters in the performance of the LightGBM model.

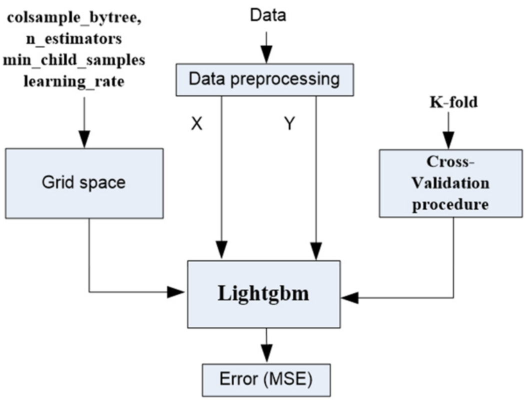

Fig. 3. Impact assessment of hyperparameters.

Furthermore, to evaluate the influence of hyperparameters on the effectiveness of the LightGBM model in more detail, different combinations of hyperparameters were considered, as shown in Table II. These models consist of the following:

- Model M1 uses default values for four hyperparameters: colsample_bytree, n_estimators, min_child_samples, and learning_rate.
- In Model M2, three hyperparameters retain their default values, with only colsample_bytree varying within the predefined range.
- Model M3 keeps two hyperparameters unchanged while adjusting the colsample_bytree and n_estimators.
- Model M4 maintains one hyperparameter at its default value while adjusting the remaining three: colsample_bytree, n_estimators, and min_child_samples.
- Finally, model M5 does not retain any hyperparameters at their default values.

TABLE II. PROPOSED HYPERPARAMETER MODELS

Model	Combinations of hyperparameters
M1	colsample_bytree = 1 n_estimators =100 min_child_samples = 20 learning_rate = 0.1
M2	colsample_bytree = [min - max] n_estimators =100, min_child_samples = 20 learning_rate = 0.1
M3	colsample_bytree = [min - max] n_estimators =[min - max] min_child_samples = 20 learning_rate = 0.1
M4	colsample_bytree = [min - max] n_estimators = [min - max] min_child_samples = [min - max] learning_rate = 0.1
M5	colsample_bytree = [min - max] n_estimators = [min - max] min_child_samples = [min - max] learning_rate = [min - max]

Analyzing the errors across these models allows for assessing the impact of adjusting hyperparameters on the predictive performance of the model. Consequently, the study can demonstrate how increasing the number of hyperparameter combinations can improve or affect the accuracy of the model. The findings of this analysis will support optimizing the LightGBM model, particularly in complex forecasting applications, to achieve the best possible performance.

The statistical results are presented in the form of boxplots. A boxplot is a standard method for displaying data distributions by dividing the dataset into four equal parts, as shown in Figure 4. The first (Q1), second (Q2), and third (Q3) quartiles correspond to the 25th, 50th (median), and 75th percentiles of the dataset. The second quartile (Q2) lies in the middle and divides the data into two halves, therefore, Q2 is also called the median [20].

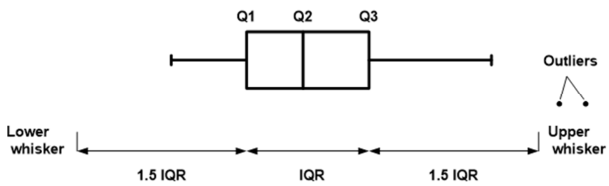

Fig. 4. Boxplot and its components.

III. RESULTS AND DISCUSSION

A. *Experimental Setup*

This study used a peak electric load dataset from Ho Chi Minh City. These datasets were extracted and preprocessed to create the input (X) and target (Y) datasets corresponding to the

LightGBM model's inputs and outputs. Figure 5 shows the graph of the Y data corresponding to the dataset.

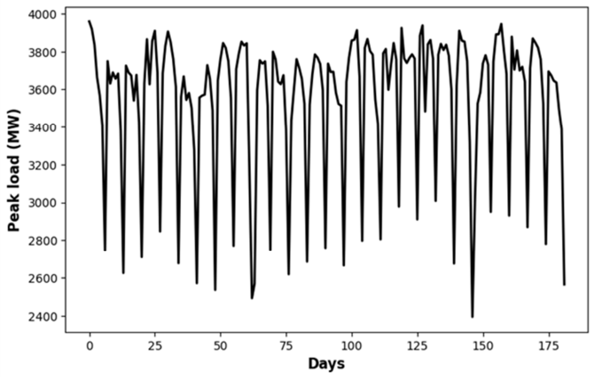

Fig. 5. Graph of target values (Y).

Table III presents the range of surveyed values for the hyperparameters of LightGBM and the range of surveyed values for the k-fold cross-validation process. The range of surveyed hyperparameter values is set within their respective limits, ensuring that the default values of the hyperparameters are positioned in the middle of the surveyed range. By exploring the values within this range, the study aims to evaluate the impact of hyperparameters on the performance of the LightGBM model, particularly when using default values. The default values of the hyperparameters are indicated in bold.

TABLE III. SURVEYED VALUE RANGE

Hyperparameters	Value range
colsample_bytree	[0.15, 0.3, 0.45, 0.6, 0.75, 0.9, **1**]
n_estimators	[25, 50, 75, **100**, 125, 150, 175]
min_child_samples	[1, 5, 10, **20**, 25, 30, 35]
learning_rate	[0.025, 0.05, 0.075, **0.1**, 0.125, 0.15, 0.175]
k-fold	[2, 3, 4, **5**, 6, 7, 8]

B. Results and Evaluation

1) Assessment of the Impact of Cross-Validation

Figure 6 presents the analysis of the impact of the k-fold parameter in the cross-validation on the prediction errors of the LightGBM model in load forecasting. The results show that the prediction error at the default value $k = 5$ is not optimal. Specifically, at $k = 5$, the median error is 186,281 MW, and the minimum error is 142,186 MW. At $k = 7$, these values decrease to 180,737 MW and 140,711 MW, respectively. More decreasing errors are observed at $k = 8$, with values of 176,046 MW for the median and 140,250 MW for the minimum error. Therefore, selecting an appropriate k-fold value instead of the default $k = 5$ in the cross-validation algorithm can help reduce errors in the LightGBM model forecasting process.

2) Assessment of the Impact of Each Hyperparameter

Figure 7 illustrates the impact of the colsample_bytree hyperparameter on prediction errors. The chart indicates that the default value colsample_bytree = 1 is not optimal regarding model error. The results show that both the median and minimum errors start to increase from colsample_bytree = 0.15 (with a median of 167,972 MW and a minimum of 140,250 MW) up to colsample_bytree = 1.0 (with a median of 189,818 MW and a minimum of 143,995 MW). These results suggest that reducing the colsample_bytree value can improve the model's prediction performance.

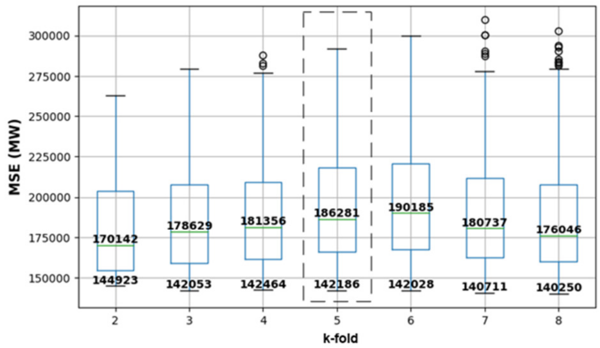

Fig. 6. Boxplots of prediction errors when altering the k-fold parameter.

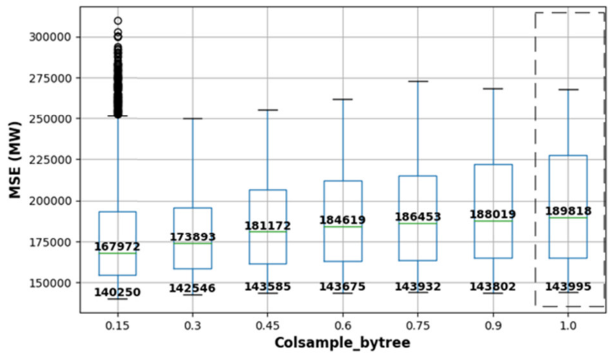

Fig. 7. Boxplots of prediction errors when altering colsample_bytree.

Figure 8 illustrates the impact of the n_estimators hyperparameter on the prediction errors of the model. The chart shows that the default value n_estimators = 100 is not optimal. Specifically, the errors tend to increase from n_estimators = 25 (with a median of 156,690 MW and a minimum of 140,250 MW) to n_estimators = 100 (with a median increasing to 183,291 MW and a minimum of 143,724 MW). These data indicate that adjusting the value of n_estimators value instead of using the default can improve the accuracy of predictions, thus enhancing the model's performance.

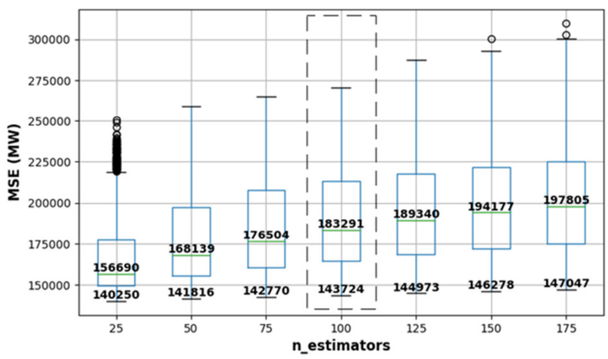

Fig. 8. Boxplots of prediction errors when altering n_estimators.

Figure 9 illustrates the impact of the min_child_samples hyperparameter on prediction errors. The chart shows that the default value min_child_samples = 20 is not optimal. Prediction errors decrease significantly when min_child_samples is increased from 20 to 35. Specifically, the analysis shows a decrease in errors from a median of 178,233 MW and a minimum of 141,551 MW at min_child_samples = 20 to 156,945 MW and 140,711 MW at min_child_samples = 35, respectively. These results show that adjusting the min_child_samples value appropriately can enhance the model's prediction accuracy.

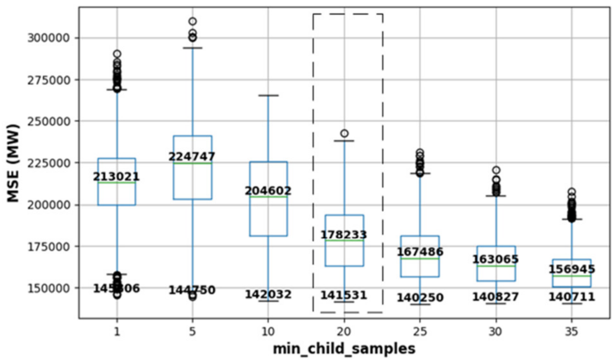

Fig. 9. Boxplots of prediction errors when altering min_child_samples.

Figure 10 illustrates the impact of learning_rate on prediction errors. As shown in the graph, the default learning_rate = 0.1 is not optimal. When reducing the learning_rate from 0.1 to 0.025, prediction errors decrease significantly. Specifically, errors decrease from a median of 183,840 MW and a minimum of 142,514 MW at learning_rate = 0.1 to a median of 156,203 MW and a minimum of 140,711 MW at learning_rate = 0.025. These data show that adjusting the learning_rate can appropriately enhance the prediction accuracy of the model compared to using the default value.

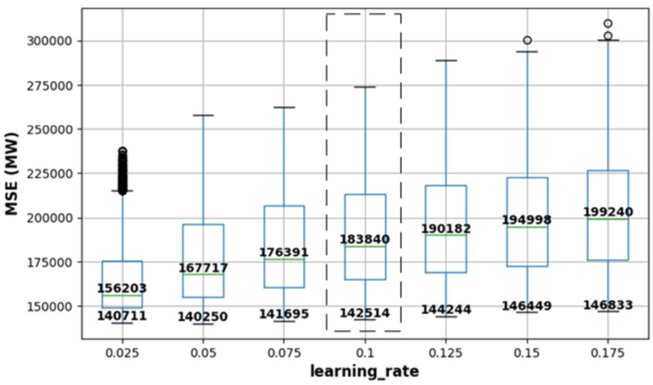

Fig. 10. Boxplots of prediction errors when altering learning_rate.

In summary, analyzing boxplot charts of prediction errors for each case study of the LightGBM model's hyperparameters (such as colsample_bytree, n_estimators, min_child_samples, and learning_rate) reveals that using the default values may not yield optimal results. Therefore, selecting appropriate values for these hyperparameters tailored to each forecasting problem is crucial to enhancing model performance.

3) *Assessment of the Influence of the Hyperparameters Combination*

Figure 11 presents the results on the impact of hyperparameter combinations. The graphical analysis indicates an increase in the number of hyperparameter combinations, helping to reduce the prediction error compared to those of the default values. Specifically, using the default values (Model M1), the corresponding error is 194,398 MW. For Model M2, statistical values with a median of 187,683 MW and a minimum of 173,880 MW are observed. For Model M3, the corresponding values are 187,420 MW and 151,508 MW. For Model M4, it is 187,683 MW and 146,963 MW. Finally, for Model M5, it is 186,280 MW and 142,186 MW.

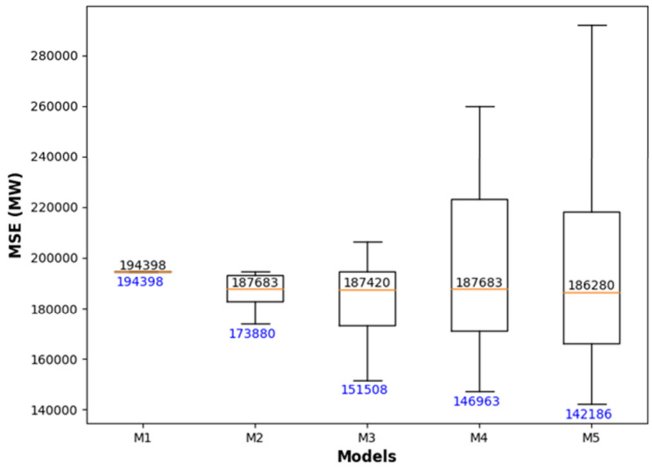

Fig. 11. Boxplots of prediction errors based on hyperparameter combinations.

IV. CONCLUSION

To assess the impact of the LightGBM model's hyperparameters on load forecasting, an algorithm was proposed based on a grid space of hyperparameter values combined with cross-validation cycles. Various cases were suggested for investigation, and the results were evaluated using boxplot charts. The findings showed that the accuracy of the LightGBM model significantly depends on its hyperparameter values. The LightGBM model with default hyperparameters results in relatively large errors in the survey range, whereas there are some other hyperparameter values with better error results. Moreover, increasing the number of hyperparameter combinations also tends to achieve better forecasting results. These results underscore the importance of optimizing hyperparameter values for the LightGBM model and other machine learning models for load forecasting and general time series prediction. The obtained results allow for more in-depth research on hyperparameter optimization for the LightGBM model.

REFERENCES

[1] V. Gupta and S. Pal, "An overview of different types of load forecasting methods and the factors affecting the load forecasting," *International*

Journal for Research in Applied Science & Engineering Technology, vol. 5, no. IV, pp. 729–733, 2017.

[2] T. Hong, P. Wang, and H. L. Willis, "A Naïve multiple linear regression benchmark for short term load forecasting," in *2011 IEEE Power and Energy Society General Meeting*, Detroit, MI, USA, Jul. 2011, pp. 1–6, https://doi.org/10.1109/PES.2011.6038881.

[3] S. K. Filipova-Petrakieva and V. Dochev, "Short-Term Forecasting of Hourly Electricity Power Demand: Reggresion and Cluster Methods for Short-Term Prognosis," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8374–8381, Apr. 2022, https://doi.org/10.48084/etasr.4787.

[4] J. W. Taylor, "Short-term electricity demand forecasting using double seasonal exponential smoothing," *Journal of the Operational Research Society*, vol. 54, no. 8, pp. 799–805, Aug. 2003, https://doi.org/10.1057/palgrave.jors.2601589.

[5] J. Chakravorty, S. Shah, and H. N. Nagraja, "ANN and ANFIS for Short Term Load Forecasting," *Engineering, Technology & Applied Science Research*, vol. 8, no. 2, pp. 2818–2820, Apr. 2018, https://doi.org/10.48084/etasr.1968.

[6] N. T. Dung and N. T. Phuong, "Short-Term Electric Load Forecasting Using Standardized Load Profile (SLP) And Support Vector Regression (SVR)," *Engineering, Technology & Applied Science Research*, vol. 9, no. 4, pp. 4548–4553, Aug. 2019, https://doi.org/10.48084/etasr.2929.

[7] C. Shang, J. Gao, H. Liu, and F. Liu, "Short-Term Load Forecasting Based on PSO-KFCM Daily Load Curve Clustering and CNN-LSTM Model," *IEEE Access*, vol. 9, pp. 50344–50357, 2021, https://doi.org/10.1109/ACCESS.2021.3067043.

[8] Y. Liu, H. Luo, B. Zhao, X. Zhao, and Z. Han, "Short-Term Power Load Forecasting Based on Clustering and XGBoost Method," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, Nov. 2018, pp. 536–539, https://doi.org/10.1109/ICSESS.2018.8663907.

[9] Y. Wang *et al.*, "Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1984–1997, Feb. 2021, https://doi.org/10.1109/TPWRS.2020.3028133.

[10] Z. Fang, J. Zhan, J. Cao, L. Gan, and H. Wang, "Research on Short-Term and Medium-Term Power Load Forecasting Based on STL-LightGBM," in *2022 2nd International Conference on Electrical Engineering and Control Science (IC2ECS)*, Nanjing, China, Dec. 2022, pp. 1047–1051, https://doi.org/10.1109/IC2ECS57645.2022.10088145.

[11] Y. Tan, Z. Teng, C. Zhang, G. Zuo, Z. Wang, and Z. Zhao, "Long-Term Load Forecasting Based on Feature fusion and LightGBM," in *2021 IEEE 4th International Conference on Power and Energy Applications (ICPEA)*, Busan, Korea, Republic of, Oct. 2021, pp. 104–109, https://doi.org/10.1109/ICPEA52760.2021.9639313.

[12] Y. Liang *et al.*, "Product marketing prediction based on XGboost and LightGBM algorithm," in *Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition*, Beijing, China, Aug. 2019, pp. 150–153, https://doi.org/10.1145/3357254.3357290.

[13] X. Liang, Y. Feng, J. Jiang, W. Wang, X. Liu, and Z. Gong, "Short-term Load Forecasting of a Technology Park Based on a LightGBM-LSTM Fusion Algorithm," in *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, Shenyang, China, Nov. 2022, pp. 151–155, https://doi.org/10.1109/AUTEEE56487.2022.9994355.

[14] Y. Miao, J. Zhu, H. Dong, Z. Chen, S. Li, and X. Wen, "Short-term Load Forecasting Based on Echo State Network and LightGBM," in *2023 IEEE International Conference on Predictive Control of Electrical Drives and Power Electronics (PRECEDE)*, Wuhan, China, Jun. 2023, pp. 1–6, https://doi.org/10.1109/PRECEDE57319.2023.10174609.

[15] Y. Zhou, Q. Lin, and D. Xiao, "Application of LSTM-LightGBM Nonlinear Combined Model to Power Load Forecasting," *Journal of Physics: Conference Series*, vol. 2294, no. 1, Mar. 2022, Art. no. 012035, https://doi.org/10.1088/1742-6596/2294/1/012035.

[16] G. Ke *et al.*, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.

[17] D. Zhang and Y. Gong, "The Comparison of LightGBM and XGBoost Coupling Factor Analysis and Prediagnosis of Acute Liver Failure," *IEEE Access*, vol. 8, pp. 220990–221003, 2020, https://doi.org/10.1109/ACCESS.2020.3042848.

[18] K. Huang, "An Optimized LightGBM Model for Fraud Detection," *Journal of Physics: Conference Series*, vol. 1651, no. 1, Aug. 2020, Art. no. 012111, https://doi.org/10.1088/1742-6596/1651/1/012111.

[19] P. Pokhrel, "A LightGBM based Forecasting of Dominant Wave Periods in Oceanic Waters." arXiv, Jul. 14, 2021, https://doi.org/10.48550/arXiv.2105.08721.

[20] N. T. Tran, T. T. G. Tran, T. A. Nguyen, and M. B. Lam, "A new grid search algorithm based on XGBoost model for load forecasting," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 1857–1866, Aug. 2023, https://doi.org/10.11591/eei.v12i4.5016.