Bulletin of Electrical Engineering and Informatics
Vol. 12, No. 4, August 2023, pp. 1857~1866
ISSN: 2302-9285, DOI: 10.11591/eei.v12i4.5016

A new grid search algorithm based on XGBoost model for load forecasting

Ngoc Thanh Tran[1], Thanh Thi Giang Tran[2], Tuan Anh Nguyen[2], Minh Binh Lam[2]
[1]Department of Electrical Power Supply, Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh, Vietnam
[2]Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh, Vietnam

Article Info

Article history:

Received Oct 16, 2022
Revised Dec 21, 2022
Accepted Jan 5, 2023

Keywords:

Load forecasting
Machine learning
Mean square error
XGBoost

ABSTRACT

XGBoost is a highly effective and widely used machine learning model and its hyperparameters take an important role on the performance of the model. This paper presents a new grid search (GS) algorithm for obtaining optimal hyperparameters of the XGBoost model based on the median values of their error loss. A benchmark method used to evaluate the proposed and original GS algorithms is introduced. Datasets with measured daily electricity demand load values of Ho Chi Minh City, Vietnam and Tasmania state, Australia are analyzed for the performance of both algorithms. The error metrics, mean squared errors (MSEs), of the proposed algorithm are found to be 2,282 MW and 501 MW that are smaller than those of original algorithms, which are 2,424 MW and 537 MW in case of Ho Chi Minh City and Tasmania state, respectively. These results then verify the accuracy of the proposed algorithm.

Corresponding Author:

Ngoc Thanh Tran
Department of Electrical Power Supply, Faculty of Electrical Engineering Technology
Industrial University of Ho Chi Minh City
12 Nguyen Van Bao, Ho Chi Minh City, Vietnam
Email: tranthanhngoc@iuh.edu.vn

1. INTRODUCTION

The XGBoost algorithm was introduced by Chen and Guestrin [1] and made up an effective direction of applying in machine learning. There have been many studies which successfully employed the XGBoost model to solve time series forecasting problems, such as forecasting stock price [2], energy [3], hemorrhagic fever [4], oil price [5], and traffic flow [6]. Not out of this trend, the power load forecasting problem has also been investigated by many scholars using the XGBoost model and obtained impressive results [7]–[12]. One of the features of the XGBoost model is that its accuracy depends on hyperparameters including the number of gradient boosted trees, maximum tree depth, boosting learning rate, minimum sum of instance weight, subsample ratio of columns, and so on. Therefore, determining the optimal hyperparameters is essential for the application of XGBoost model [13]–[15]. Several algorithms have been used to determine these optimal hyperparameters, for which the grid search (GS) algorithm combined with cross-validation technique is preferred to use due to high efficiency and simplicity. The GS algorithm runs a search over all hyperparameter sets in a grid space while recording error metric – the criterion for evaluation of model performance. GS algorithm returns the optimal model with optimal hyperparameters based on a selection criterion for getting the smallest of error metric in the training process [16]–[19]. Objectively, the minimum value of a dataset will normally fluctuate, so the obtained optimal model of the GS algorithm determined during the training process may not be the best value in the testing process.

In this regard, the present work proposes a GS algorithm based on the median values instead of the minimum values. The boxplot chart allows to analyze statistical characteristics of the data according to 5 distribution positions embedded in the proposed GS algorithm, namely, minimum value (min), first quartile (Q1), median (Q2), third quartile (Q3), and maximum value (max) [17], [20], [21]. Method used to compare the proposed algorithm with the traditional one is set up to evaluate the accuracy of the two models. daily load data of the Ho Chi Minh City (HCM) (Vietnam) and Tasmania state (TA) (Australia) were employed in the experiments to verify the accuracy of this study.

The rest of the paper is organized as follows. Sections 2 and 3 describe an overview of the XGBoost model, GS algorithm, as well as the proposed GS algorithm and the method for evaluating both algorithms. Section 4 presents the analysis and discussion of the experimental results. Conclusions of the paper are given in section 5.

2. METHOD

2.1. The XGBoost model

XGBoost, a kind of boosting algorithm, is a powerful method for regression as well as classification [22], [23]. Support the dataset D={(x_i, y_i)} (x_i ∈ Rm, y_i ∈ R), m is the dimension of sample and n is the number of samples. A tree ensemble model including K decision trees with predicted value is obtained by:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i),\ f_k \in F \tag{1}$$

$$F = \{f(x) = \omega_{q(x)}\}\ \ (q\colon R^m \to T, \omega \in R^T) \tag{2}$$

where F represents the space of regression trees, q indicates the structure of each tree, T is the number of leaves in the tree, ω is the leaf weight, and f_k corresponds to an independent tree.
The goal of the model is to learn the (1), and the objective function defined as (3):

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{3}$$

where $\hat{y}_i$ and y_i are the predicted and real values; l is the training loss measuring the difference between $\hat{y}$ and y; Ω is the complexity of the model, which is used to prevent over fitting of the model, Ω is indicated by:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda\|\omega\|^2 \tag{4}$$

In (4), γ is the penalty coefficient which controls the complexity of the model; λ is the penalty coefficient of leaf weight. Let $\hat{y}_i^{(t)}$ is the prediction of the i-th instance at the t-th iteration, and $\hat{y}_i^{(t)}$ can be obtained in (5):

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{5}$$

Substituting (5) into the objective (3), the objective function can be rewritten as (6):

$$L^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{6}$$

To improve the convergence speed and accuracy, the second-order Taylor approximation is used, and the (6) is transformed into (7):

$$\begin{gathered} L^{(t)} \simeq \sum_{i=1}^{n}[l(y_i, \hat{y}_i^{(t-1)} + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \\ g_i = \partial_{\hat{y}^{(t-1)}} l\big(y_i, \hat{y}^{(t-1)}\big), h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)}) \end{gathered} \tag{7}$$

where g_i, h_i are first and second order gradient statistics of loss function. Remove the constant term, the specific objective function at step t is being the new optimization goal for the new tree as (8):

$$\tilde{L}^{(t)} = \sum_{i=1}^{n}[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \tag{8}$$

Considering that I_j={i|q(xi)=j} is the instance set of leaf j, replacing $f_t(x_i)$ by the tree definition $\omega q_{(x)}$, expanding Ω, then (8) can be changed into (9):

$$\begin{aligned} \tilde{L}^{(t)} &= \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T}\omega_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)\omega_j^2] + \gamma T \end{aligned} \tag{9}$$

The optimal weight ϖ_j^* of leaf j can be obtained in (10):

$$\omega_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda} \quad (10)$$

And then the tree with the corresponding optimal value is (11):

$$\tilde{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T} \frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T \quad (11)$$

In (11) is used as a scoring function to measure the quality of a q-tree structure.

2.2. The hyperarameters of XGBoost

XGBoost is a powerful algorithm. So, it will take more design decisions and hence large hyperparameters [24]–[26]. Hyperparameters are certain values or weights that define the learning process of an algorithm. The hyperparameters of the XGBoost model can be classified into three categories:
- General parameters: define the overall functionality of XGBoost model, such as booster, verbosity, and nthread
- Booster parameters: control the performance of the XGBoost model, such as learning_rate (LR), min_child_weight (MC), max_depth (MD), subsample, lambda, and alpha.
- Learning task parameters: define the optimization objective to be calculated at each step, such as objective, eval_metric, and seed.

So, there are many tuning hyperparameters for tree-based learners in the XGBoost model, and the most common ones are described in Table 1.

Table 1. The common hyperparameters of XGBoost

Hyperparameters	Definition
booster	Select the type of model to run at each iteration
nthread	The number of parallel threads used to run XGBoost, is used for parallel processing and number of cores in the system should be entered
learning_rate	Shrinking the step size is used to prevent overfitting. Range is [0,1]
min_child_weight	Determines the minimum weighted sum of all required observations in a child
max_depth	Determines how deep each tree is allowed to grow in any boost loop
subsample	Percentage of samples used per tree. Low values can lead to underfitting
objective	Defines the loss function to be minimized
eval_metric	The metric to be used for validation data

3. GRID SEARCH AND PROPOSED GRID SEARCH ALGORITHM

3.1. The original grid search algorithm

The accuracy of machine learning models in general and XGBoost networks in particular depends on their hyperparameters. There are many algorithms used to determine these optimal hyperparameters, typically, GS, random search (RS), genetic algorithm (GA), particle swarm optimization (PSO), bayesian optimization (BO) [13], [14], [25]. In which, the GS algorithm will be explored in this study. The principle of the original GS algorithm is to generate a grid of possible values for the hyperparameters. During iteration, the hyperparameters will be combined in a specific order, fits the model, and recorded the performance (error metric) of the model. Finally, the algorithm determines the optimal hyperparameters with the best performance. In the paper, the procedure of the original GS algorithm is reviewed based on the boxplot definition, which the form of boxplot is illustrated in Figure 1. Boxplot is a method for describing the distribution of data in statistics [20], [21]. It determines the lower quartile (Q1), median (Q2), and upper quartile (Q3) values. The interquartile quartile range (IQR)=Q3-Q1, the maximum whisker length is IQR*1.5, and outliers are points which lies outside that range.

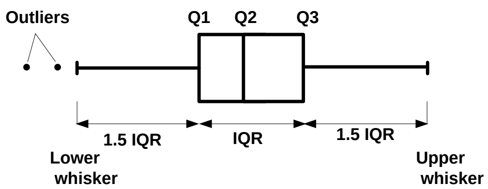


Figure 1. The boxplot definition

The XGBoost have many hyperparameters as shown in Table 1, and in this paper the author focuses on three hyperparameters namely LR, MD, and MC. The procedure of the GS algorithm with three hyperparameters based on the boxplot is shown in Figure 2. Firstly, it is needed to determine the range of hyperparameters to tune and their search space. The next step is a process of finding an optimal value of LR hyperparameter (LR_{opt}) to achieve minimum error loss based on the boxplot drawn by LR. Fixed $LR=LR_{opt}$, the new search space is created by only the combination of MD and MC hyperparameters, and the optimal value of the MD hyperparameter is obtained by the same process of the LR hyperparameter. Continuously fixed $MD=MD_{opt}$, the search space is made up of just the range of the last hyperparameter MC, and then we also obtain the optimal value MC_{opt}. The output of this procedure is the optimal hyperparameter {LR_{opt}, MD_{opt}, MC_{opt}}.

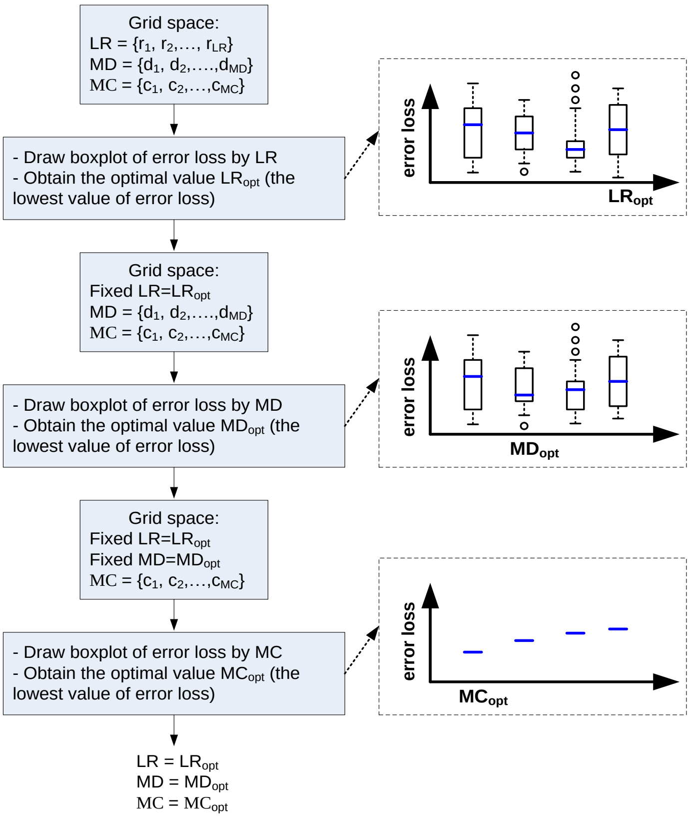
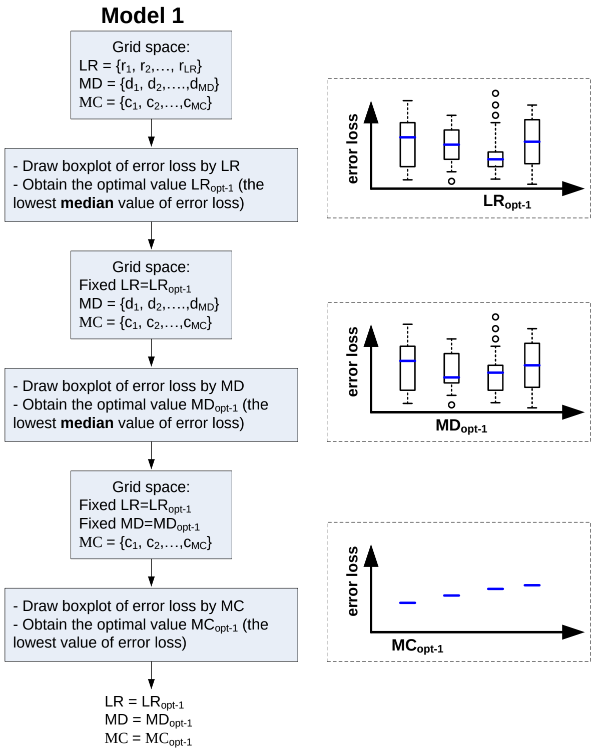


Figure 2. The procedure of the original grid search for XGBoost

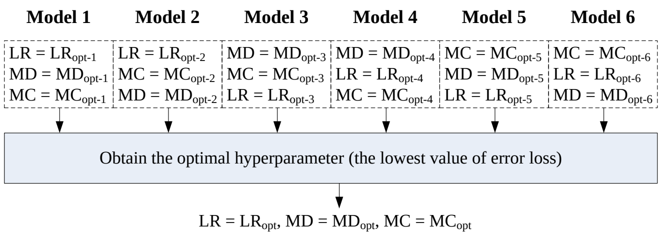
To enhance the effeteness, the cross-validation technique is commonly combined with GS algorithm for searching optimal parameters. The technique is performed using k-fold cross validation. The data is divided into k equal subsets where one subset is used for testing while the remaining ones are used for training purpose. Once the model has been trained k times, the overall training performance is evaluated by the average of the training results obtained in each iteration.

3.2. The proposed grid search algorithm

In this paper, the proposed algorithm is based on the procedure of the original GS algorithm described above, where the difference is that the processes for determining the optimal hyperparameters are based on the minimum value of the median, instead of the minimum values of the original GS algorithm. The procedure of the proposed GS algorithm using the boxplot is shown in Figure 3. In Figure 3(a), model 1 presents the procedure based on the median values, corresponding to the order of sequentially determined optimal parameters of LR, MD and MC. As a result of this procedure, the optimal hyperparameters {LR_{opt-1}, MD_{opt-1}, MC_{opt-1}} are obtained. Since the selection criterion is based on median values, changing the sequence of the obtained hyperparameters for the XGBoost model may lead to different optimal values. In this study, we just consider 03 hyperparameters of LR, MD, MC for the XGBoost model, thereby it allows to establish 5 more sequence combinations of hyperparameters, corresponding to models from 2 to 6. These models are also used in turn with the same procedure of model 1, thereby determining a total of 6 optimal sets of hyperparameters. In the last step as shown in Figure 3(b), the error metric values of 6 optimal sets of hyperparameters are evaluated to obtain the set of hyperparameter with the smallest error metric, which is the output of the proposed GS algorithm.



(a)



(b)

Figure 3. The procedure of the proposed grid search for XGBoost (a) the procedure of the model 1 based on the median values and (b) the combination of 6 models

3.3. Method for evaluating

Figure 4 shows a method used as a benchmark for the proposed GS algorithm. This method consists of three processes: data extracting, training and testing. The data extracting process: the data is divided into training and testing dataset. The training dataset (X_{train}, Y_{train}) may be used to train the model while the testing dataset (X_{test}, Y_{test}) - evaluate the proposed and original algorithms.

The training processing: Define the search space based on the combination of the tuning values of the hyperparameters LR, MD, MC. The original and proposed GS algorithm will be performed according to the procedure illustrated in Figures 2 and 3, respectively. After this step, the optimal hyperparameters of original and proposed GS algorithm will be obtained. Performance of the original and proposed GS algorithms is evaluated using mean squared error (MSE) error loss defined as [27], [28].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|^2 \quad (12)$$

where [y_1, y_2, …, y_n] and [$\hat{y}_1$, $\hat{y}_2$, …, $\hat{y}_n$] are the test and prediction values, respectively. The testing process: Both these optimal XGBoost models will generate the predicted value $Y_{predict}$, then calculate the error metric MSE. And these error metrics of the original and proposed GS algorithms will be evaluated by comparing to each other.

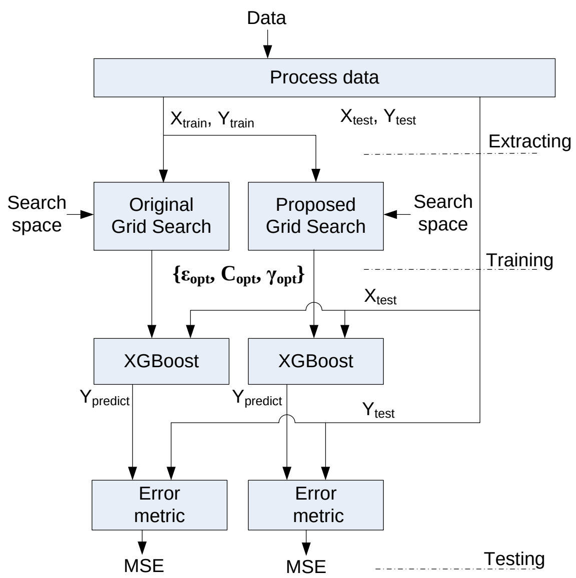


Figure 4. The method of evaluating the proposed algorithm

4. EXPERIMENTS RESULTS

4.1. Parameter settings and dataset description

This study employed the GS CV tool provided by scikit-learn [29] to implement the GS algorithm using the cross-validation technique. The experiments were conducted on Google Colab's TPU [30]. Table 2 below highlights the hyperparameters tuning considered in this study. The length of LR hyperparameter of 30, MD and MC of 20, so the total number of combinations is equal to 30×20×20=12000. In addition, we will perform the original and original GS algorithms using k-fold cross-validation with 2 folds.

The electric load data of both TA (Australia) and HCM (Vietnam) are used in the paper to verify the effectiveness of the proposed algorithm. The Tasmania dataset records the electric load every half-hour, produces 48 daily points. Meanwhile, the Ho Chi Minh dataset records the electric load every hour, produces 24 daily points. Here we considered 63 days in the provided dataset with the training dataset for two typical months of 56 days and the testing dataset for one week of 7 days. The dataset description of TA and HCM is shown in the Table 3.

Table 2. The ranges and options for hyperparameter

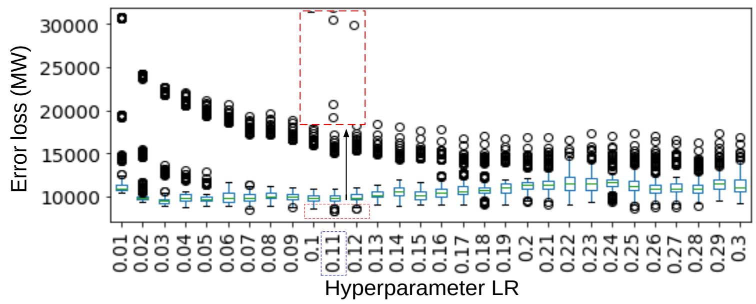
Abbreviation	Hyperparameter	Range		
		min	max	step
LR	learning_rate	0.01	.30	0.01
MD	max_depth	1	20	1
MC	min_child_weight	1	20	1

Table 3. The data description

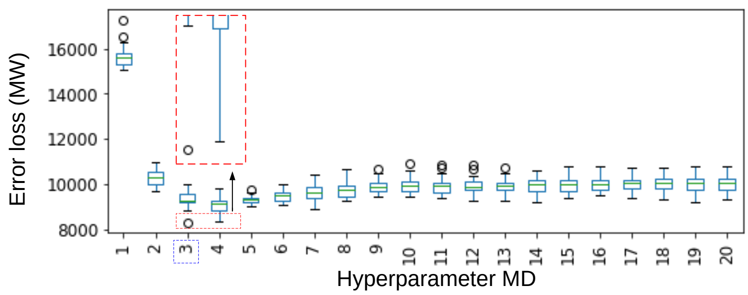
Data	Training	Testing
TA	- Dimension: X_{train}: (2688, 48), Y_{train}: (2688,) - Time: from 3/23/14 to 5/17/14	- Dimension: X_{test}: (336, 48), Y_{test}: (336,) - Time: from 5/18/14 to 5/24/14
HCM	- Dimension: X_{train}: (1344, 24), Y_{train}: (1344,) - Time: from 10/22/18 to 12/16/18	- Dimension: X_{test}: (168, 24), Y_{test}: (168,) - Time: from 12/17/18 to 12/23/18

4.2. The training process

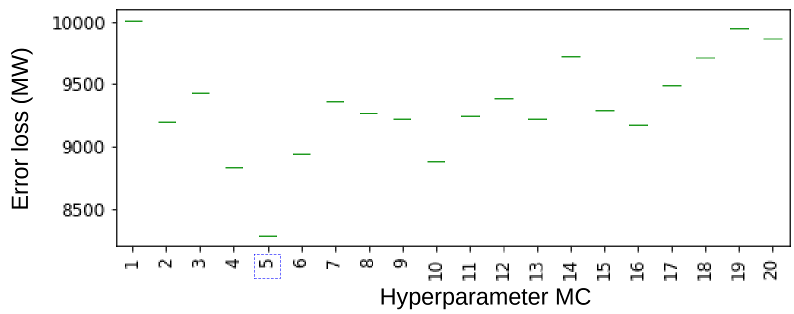
Figure 5 illustrates step by step the procedure of the original GS algorithm for the case of HCM data described in Figure 2. Figure 5(a) shows the boxplot of error loss by LR and the obtained optimal value of LR=0.11. At the same time, the boxplot of error loss by MD is presented in Figure 5(b) with the optimal value of MD=3. The last one is reported in Figure 5(c) with the error loss by MC and the optimal value of MC=5. So, the optimal hyperparameter of the original GS algorithm is {LR=0.11, MD=3, MC=5} for the case of HCM. That is also the result of the training process of the original GS algorithm which is shown in Figure 4. The same result was also obtained for the case of TA data as shown in Table 4.



(a)



(b)



(c)

Figure 5. The training process of original GS algorithm, HCM (a) hyperparameter LR, (b) hyperparameter MD, and (c) hyperparameter MC

Table 4. The training process of the original grid search

Data	Optimal hyperparameters	Error loss (MSE)
HCM	LR=0.11; MD=3; MC=5	8284.3
TA	LR=0.27; MC=1; MD=4	805.2

Figure 6 reports step by step the procedure of the proposed GS algorithm (model 1) in the case of HCM as described in Figure 3(a). The boxplots of error loss by LR and MD with the optimal values of LR=0.03 and MD with the optimal value as MD=5 is illustrated in Figures 6(a) and 6(b), respectively. Figure 6(c) shows the boxplot of error loss by MC with the optimal value of MC=12. As a result, the optimal hyperparameter of the model 1 based on the proposed GS algorithm is {LR=0.03, MD=5, MC=12}.

In the same way, the optimal hyperparameters of Models 2 to 6 were obtained as shown in Table 5. Comparing these models and choosing the model with the smallest error loss values allowed to obtain the optimal hyperparameter of {MD=4; MC=5; LR=0.11} for the proposed GS algorithm toward the case of HCM data. In the case of TA data, the optimal hyperparameter was obtained with {MD=1; MC=6; LR=0.12} (Table 6). Note that during training, the error loss value of the original GS algorithm (8284.3 MW for HCM data, and 805.2 MW for TA data) is obviously smaller than that of the proposed GS algorithm with 8,322.14 and 806.26 MW for TA and HCM data, respectively.

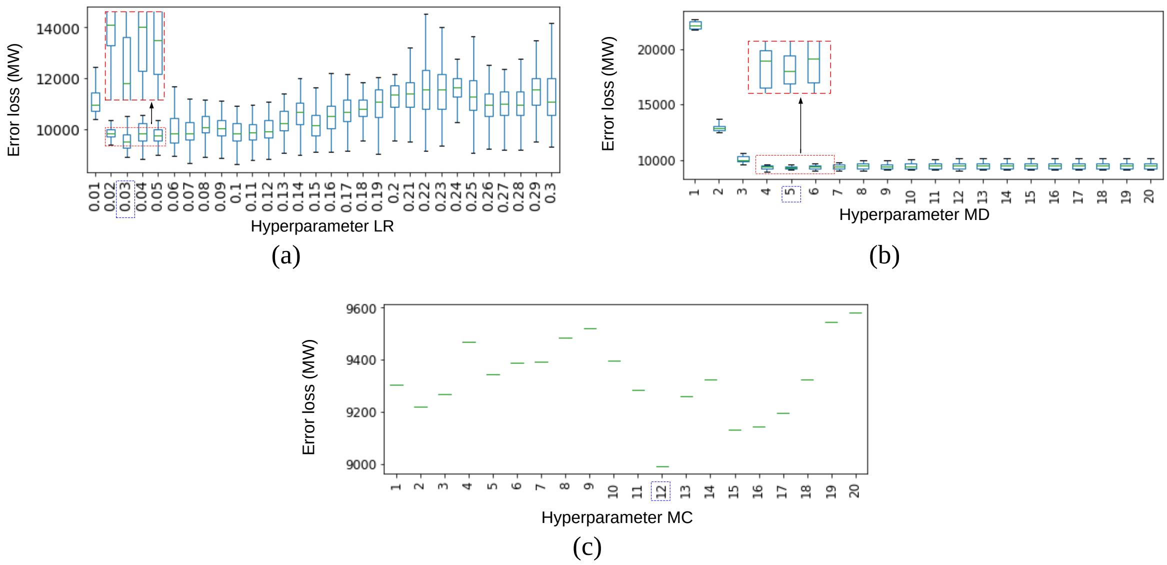


Figure 6. The training process of proposed GS algorithm, model 1, HCM (a) hyperparameter LR, (b) hyperparameter MD, and (c) hyperparameter MC

Table 5. The proposed grid search, training process, HCM

Model	Optimal hyperparameters	Error loss (MSE)
1	LR=0.03; MD=5; MC=12	8,989.92
2	LR=0.03; MC=17; MD=12	9,055.84
3	MD=4; MC=5; LR=0.11	8,322.14
4	MD=4; LR=0.11; MC=5	8,322.14
5	MC=17; MD=3; LR=0.16;	9,168.17
6	MC=17; LR=0.03; MD=12	9,055.84
The smallest error loss: MD=4; MC=5; LR=0.11		

Table 6. The proposed grid search, training process, TA

Model	Optimal Hyperparameters	Error loss (MSE)
1	LR=0.02; MD=20; MC=6	814.53
2	LR=0.02; MC=6; MD=16	810.35
3	MD=1; MC=6; LR=0.12	806.26
4	MD=1; LR=0.12; MC=6	806.26
5	MC=7; MD=1; LR=0.13	819.34
6	MC=7; LR=0.02; MD=19	830.42
The smallest error loss: MD=1; MC=6; LR=0.12		

4.3. The testing process

Based on the training process given in subsection 4.2, the optimal hyperparameter of the original and proposed GS algorithms can be obtained as shown in the column 'optimal hyperparameters' of the Table 7. The last column of the Table 7 depicts the error metrics MSE in the testing process of these optimal hyperparameters for the original and proposed GS algorithms using the data of HCM and TA, respectively. It is indicated that the error metric MSE for the proposed algorithm has values of 2,282 MW and 501 MW i.e. smaller than those for original one with 2,424 MW and 537 MW, respectively. Apparently, the results demonstrate a huge advantage of the proposed over original algorithm by means of the error metrics.

Table 7. The results of testing process

Data	Method	Optimal hyperparameters (in the training process) LR	MD	MC	Error metric MSE (in the testing process)
HCM	Original grid search	0.11	3	5	2,424
	Proposed grid search	0.11	4	5	2,282
TA	Original grid search	0.27	1	4	537
	Proposed grid search	0.12	1	6	501

5. CONCLUSION

This paper proposes a new GS algorithm for obtaining the optimal hyperparameters of the XGBoost model. The proposed algorithm is established based on the minimum median values of the error loss instead of the minimum values for the original algorithm. The boxplot distribution is embedded to conduct the proposed and original GS algorithms. The benchmark method is capable of evaluating the performance of the proposed and original GS algorithms using the daily electric load demand of the HCM, Vietnam and TA state, Australia. According to the experimental results, the satisfying performance of the proposed algorithm over the original one was demonstrated to verify the effectiveness of the proposed GS algorithm.

REFERENCES

[1] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[2] T. Liwei, F. Li, S. Yu, and G. Yuankai, "Forecast of LSTM-XGBoost in stock price based on bayesian optimization," *Intelligent Automation & Soft Computing*, vol. 29, no. 3, pp. 855–868, 2021, doi: 10.32604/iasc.2021.016805.

[3] W. Yucong and W. Bo, "Research on EA-Xgboost hybrid model for building energy prediction," *Journal of Physics: Conference Series*, vol. 1518, no. 1, pp. 1–10, 2020, doi: 10.1088/1742-6596/1518/1/012082.

[4] C. -X. Lv, S. -Y. An, B. -J. Qiao, and W. Wu, "Time series analysis of hemorrhagic fever with renal syndrome in mainland China by using an XGBoost forecasting model," *BMC Infectious Diseases*, vol. 21, no. 1, p. 839, 2021, doi: 10.1186/s12879-021-06503-y.

[5] M. Gumus and M. S. Kiran, "Crude oil price forecasting using XGBoost," in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 1100–1103, doi: 10.1109/UBMK.2017.8093500.

[6] J. Cao, G. Cen, Y. Cen, and W. Ma, "Short-term highway traffic flow forecasting based on XGBoost," in *2020 15th International Conference on Computer Science & Education (ICCSE)*, 2020, pp. 359–362, doi: 10.1109/ICCSE49874.2020.9201834.

[7] R. Gupta, A. K. Yadav, S. Jha, and P. K. Pathak, "Time series forecasting of solar power generation using Facebook prophet and XG Boost," in *2022 IEEE Delhi Section Conference (DELCON)*, 2022, pp. 1–5, doi: 10.1109/DELCON54057.2022.9752916.

[8] X. Liao, N. Cao, M. Li, and X. Kang, "Research on short-term load forecasting using XGBoost based on similar days," in *2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*, 2019, pp. 675–678, doi: 10.1109/ICITBS.2019.00167.

[9] J. Cai, H. Cai, Y. Cai, L. Wu, and Y. Shen, "Short-term forecasting of user power load in China based on XGBoost," in *2020 12th IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 2020, pp. 1–5, doi: 10.1109/APPEEC48164.2020.9220335.

[10] 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)*, 2018, pp. 536–539, doi: 10.1109/ICSESS.2018.8663907.

[11] R. A. Abbasi, N. Javaid, M. N. J. Ghuman, Z. A. Khan, S. U. Rehman, and Amanullah, "Short term load forecasting using XGBoost," in *Web, Artificial Intelligence and Network Applications*, Cham: Springer, 2019, pp. 1120–1131, doi: 10.1007/978-3-030-15035-8_108.

[12] M. Fan, Y. Hu, X. Zhang, H. Yin, Q. Yang, and L. Fan, "Short-term load forecasting for distribution network using decomposition with ensemble prediction," in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 152–157, doi: 10.1109/CAC48633.2019.8997169.

[13] Z. Zheng, S. Pan, H. Luo, and Z. Guo, "Porosity prediction based on GS+GA-XGBoost," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, 2020, pp. 1014–1020, doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00153.

[14] C. Qin, Y. Zhang, F. Bao, C. Zhang, P. Liu, and P. Liu, "XGBoost optimized by adaptive particle swarm optimization for credit scoring," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–18, 2021, doi: 10.1155/2021/6655510.

[15] E. A. Madrid and N. Antonio, "Short-term electricity load forecasting with machine learning," *Information*, vol. 12, no. 2, pp. 1–21, 2021, doi: 10.3390/info12020050.

[16] S. Malik, R. Harode, and A. S. Kunwar, "XGBoost: a deep dive into boosting," *Simon Fraser University*, pp. 1–21, 2020, doi: 10.13140/RG.2.2.15243.64803.

[17] G. Duan and X. Ma, "A coupon usage prediction algorithm based on XGBoost," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, Jul. 2018, pp. 178–183, doi: 10.1109/FSKD.2018.8686924.

[18] T. Vantuch, A. G. Vidal, A. P. R. -Gonzalez, A. F. Skarmeta, and S. Misak, "Machine learning based electric load forecasting for short and long-term period," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 511–516, doi: 10.1109/WF-IoT.2018.8355123.

[19] J. Brownlee, *XGBoost with python: gradient boosted trees with XGBoost and scikit-learn*. Machine Learning Mastery, 2021.

[20] J. Brownlee, *Statistical methods for machine learning: discover how to transform data into knowledge with Python*. Machine Learning Mastery, 2018.

[21] J. Brownlee, *Machine learning mastery with python: understand your data, create accurate models and work projects end-to-end.* Machine Learning Mastery, 2021.

[22] F. Fahiman, S. M. Erfani, and C. Leckie, "Robust and accurate short-term load forecasting: a cluster oriented ensemble learning approach," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8, doi: 10.1109/IJCNN.2019.8852460.

[23] C. B. Escabias, "Tree boosting data competitions with XGBoost," Universitat Politècnica de Catalunya, 2017.
[24] B. Quinto, *Next-generation machine learning with spark*. Berkeley, CA: Apress, 2020, doi: 10.1007/978-1-4842-5669-5.
[25] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: 10.1016/j.neucom.2020.07.061.
[26] C. Bentéjac, A. Csörgő, and G. M. -Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, no. 3, pp. 1937–1967, 2021, doi: 10.1007/s10462-020-09896-5.
[27] B. Dursun, F. Aydin, M. Zontul, and S. Sener, "Modeling and estimating of load demand of electricity generated from hydroelectric power plants in turkey using machine learning methods," *Advances in Electrical and Computer Engineering*, vol. 14, no. 1, pp. 121–132, 2014, doi: 10.4316/AECE.2014.01019.
[28] A. Utku and M. A. Akcayol, "Deep learning based prediction model for the next purchase," *Advances in Electrical and Computer Engineering*, vol. 20, no. 2, pp. 35–44, 2020, doi: 10.4316/AECE.2020.02005.
[29] D. K. Barupal and O. Fiehn, "Generating the blood exposome database using a comprehensive text mining and database fusion approach," *Environmental Health Perspectives*, vol. 127, no. 9, pp. 1–10, 2019, doi: 10.1289/EHP4713.
[30] E. Bisong, *Building machine learning and deep learning models on Google cloud platform*. Berkeley, CA: Apress, 2019, doi: 10.1007/978-1-4842-4470-8.

BIOGRAPHIES OF AUTHORS

Ngoc Thanh Tran received the B.Sc. and M.Sc. degrees from the Ho Chi Minh University of Technology, Vietnam in 2001 and 2006, respectively. He received his Ph.D. degree in Information and electricity engineering from Hunan University, Changsha, China in 2012. Currently, he is a Lecturer with the Faculty of Electrical Engineering Technology, Industry University of Ho Chi Minh City, Vietnam. His main research interests include HVDC transmission systems, active filters, power quality and load forecasting. He can be contacted at email: tranthanhngoc@iuh.edu.vn.

Thanh Thi Giang Tran received the B.Sc. and M.Sc. degrees from the Ho Chi Minh University of Technology, Vietnam in 2001 and 2006, respectively. Currently, he is a Lecturer with the Faculty of Electrical Engineering power quality and load forecasting. He can be contacted at email: tranthigiangthanh@iuh.edu.vn.

Tuan Anh Nguyen received the B.Sc degrees from the Industrial University of Ho Chi Minh City, Vietnam in 2012 and M.Sc degrees from the HCMC University Of Technology and Education, Vietnam in 2015, respectively. Currently, he is a Lecturer with the Faculty of Electrical Engineering Technology, Industry University of Ho Chi Minh City, Vietnam. His main research interests include HVDC transmission systems, active filters, power quality and load forecasting, Scada, PLC, automatic system. He can be contacted at email: nguyenanhtuan@iuh.edu.vn.

Minh Binh Lam received the B.E. degree in Electrical - Electronics Engineering from BachKhoa University, Vietnam, in 2002, and the M.S. degree in Telecommunication and Information Systems from University of Essex, U.K, in 2004. She is currently a lecturer of the Faculty of Electrical Engineering Technology of the Industrial University of Ho Chi Minh City, Vietnam. She is also pursuing a Ph.D. degree at Department of Electronic Engineering, Pukyong National University, Busan, Korea. Currently she is a Lecturer with the Faculty of Electrical Engineering Technology, Industry University of Ho Chi Minh City, Vietnam. Her current research interests include high-frequency radio frequency identification, far field communication, and low-power sensor networking. She can be contacted at email: lambinhminh@iuh.edu.vn.