Hyperparameter Optimization for Deep Learning Modeling in Short-Term Load Forecasting

Original Scientific Paper

Thanh Ngoc Tran
Faculty of Electrical Engineering Technology
Industrial University of Ho Chi Minh City,
12 Nguyen Van Bao Street, Ward 1, Go Vap District,
Ho Chi Minh City, Viet Nam
tranthanhngoc@iuh.edu.vn

Tuan Anh Nguyen*
Faculty of Electrical Engineering Technology
Industrial University of Ho Chi Minh City,
12 Nguyen Van Bao Street, Ward 1, Go Vap District,
Ho Chi Minh City, Viet Nam
nguyenanhtuan@iuh.edu.vn

*Corresponding author

Abstract *– The evolution of new technologies has made short-term power load forecasting an essential part of the streamlining process in the management of power grid systems. Machine learning algorithms have been applied widely in this area but with little success towards achieving better accuracy rates. These gaps point out the necessity for better forecasting methods . This study is about the power grid system from Ho Chi Minh city in Vietnam. Ho Chi Minh operates as a metropolitan area on the rise with economic activity and seasonal factors greatly influencing electricity consumption. Due to its intricate fluctuations in consumption pattern, the city is known for having a high level of energy. This makes the city suitable for an in-depth investigation regarding a case study on short-term load forecasting approaches. In this study, the goal is to evaluate the effectiveness of three hyperparameter optimization methods: Random Search, Grid Search, and Bayes Search. All these methods optimize the performance of Convolutional Neural Network (CNN) models for short-term electricity load forecasting in Ho Chi Minh City. The results obtained through this work can also be used as a basis for introducing the methods to other locations in Vietnam. The assessment of the techniques is performed using fundamental error measures such as Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). Bayes Search completed with an MAE of 77.93, MAPE of 2.94%, MSE of 10,376.7, and RMSE of 101.9. These results indicate a noticeable enhancement in prediction accuracy when compared with the outcomes from Grid Search and Random Search. Grid Search provided an MAE of 106.23, MAPE of 3.95%, MSE of 17,033.7, and RMSE of 130.5. Random Search produces results of an MAE of 96.8, MAPE of 3.57%, MSE of 14,951.0, RMSE of 122.3. These results are evidence that Bayes Search is better for short-term electricity load forecasting in Ho Chi Minh City. The study also proposes an evaluation framework, which is meant for load forecasting in Vietnam. It is designed for Ho Chi Minh City predicting purposes, thus, integrating innovative concepts with actual forecasting functions. The framework is also applicable to other areas in Vietnam, both rural and urban, having different power consumption patterns. The reduction in forecasting inaccuracies through the use of Bayes Search is found to be promising as observed in the research. This automation supports better decision-making in energy management. It helps reduce costs in dynamic and complex power grid environments. These findings have practical value. They support efforts to build more flexible and efficient energy grids in Vietnam.*

Keywords*:* *CNN network, hyperparameter optimization, Bayes Search, Random Search, Grid Search, Short-term load forecasting*

Received: September 30, 2024; Received in revised form: March 21, 2025; Accepted: April 7, 2025

1. INTRODUCTION

Load forecasting in the near future is essential for adequate power grid control in terms of power generation, distribution, and supply. Precise estimation of electricity consumption is key to preserving a stable provision of energy, and subsequently minimizing expenses.

It guarantees rational energy distribution, which is particularly important in major metropolitan regions. The demand for electricity in cities is volatile due to the expansion of the commercial and industrial activities. This necessitates the need for accurate forecasting. There is considerable improvement in forecasting techniques over the years. Still, there are difficulties.

One of the difficulties has always been the validity of the results. One issue is reliability. Conventional approaches, which bear this burden, have their own difficulties. Linear Regression [1], Time Series Forecasting [2], Kalman Filtering [3] are models that tend to fail for a variety of reasons. They have great difficulties deals with large, complex modern power system datasets. These datasets are often produced by contemporary power systems. To overcome these challenges, Deep Learning models are useful. Convolutional Neural Networks (CNN) is one of it. CNNs are capable of processing data with non-linear multi-dimensional structures. Thus, they are able to make better predictions [4]. They can uncover deeper patterns within the data than other methods. It also increases the overall accuracy of forecasts offered by the models.

Now we see more and more interest in load forecasting models in developing countries. There are so many techniques for improving forecasting. Study [5] proposed a hybrid model, HHO-GCN-LSTM. This model uses a sophisticated combination of deep learning architectures and optimization methods. It aims towards achieving more precise load forecasting. Study [6] recommends the development of ensemble based techniques for short term load forecasting. With this approach, wavelet transform, Extreme Learning Machine and Partical Least Squares Regression (PLSR) are used. The improvement of forecast accuracy is achieved by these methods. They also minimize overfitting. This enhances the reliability of the model's predictions. Study [7] introduces a spatial-temporal forecasting framework using Graph Neural Networks, leveraging individual and aggregated load data to capture hidden dependencies among residential units, significantly enhancing prediction accuracy over conventional methods. These studies demonstrate significant progress in load forecasting in developing countries and contribute to shaping advanced technological trends in energy management, particularly in the context of increasingly scarce resources that must be utilized efficiently. However, recent studies also suggest that integrating signal decomposition techniques with deep learning models, as demonstrated in Article [8], can result in higher accuracy for short-term forecasting.

In recent years, the development of short-term power load forecasting models has achieved notable advancements, driven by the progression of deep learning technologies. Studies [9-12] have explored these innovative methods, particularly the combination of Convolutional Neural Networks (CNNs) and Long Short-Term Memory Networks (LSTMs). Research [9] developed a Pyramid-CNN model for customers with similar energy usage profiles, enhancing forecasting accuracy through cluster analysis. Meanwhile, studies [10] have focused on integrating CNN and LSTM for short-term power load forecasting, demonstrating superior accuracy compared to traditional models. Further studies, such as those in [11], propose convolutional multi-integration models and convolutional wavelet models to handle complex fluctuations in load data effectively. Research [12] further expands model capabilities by utilizing evolutionary methods and encoder-decoder combinations in power load forecasting. Article [13] highlights the utilization of technical indicators such as EMA and chaotic optimization algorithms for time series prediction, showcasing their potential for applications in power load forecasting. Building on this foundation, research [14] delves into a CNN- LSTM combined deep learning model, demonstrating its high efficiency in addressing dynamic load fluctuations within specific power systems. Together, these studies underscore the advancements in predictive modeling techniques for power systems.

The key issue to note in the aforementioned advanced methods is that many studies have not adequately addressed hyperparameter optimization, which is a critical factor that can significantly influence the performance and accuracy of the model. Failure to optimize hyperparameters may lead to suboptimal outcomes, even though deep learning models like CNNs and LSTMs have demonstrated their efficacy in short-term power load forecasting. Integrating hyperparameter optimization techniques is thus an essential step to enhance the performance of these models further. Feature selection and model simplifications are covered in reference [15]. Some optimizational strategies, like FPA, outperform their non-optimized counterparts. These strategies cut down the cost for computation. Such techniques can be implemented in applications that need near real-time processing like power load forecasting. This pertains to hyperparameter tuning of the CNN model [16]. The aim is to refine load prediction in metropolitan areas. It employs sophisticated optimization techniques [17]. These techniques are Grid Search, Random Search and Bayes Search [18]. They assist in pinpointing optimal values for hyperparameters to maximize precision in load forecasting. In this case, the MAE, MAPE, RMSE, and MSE error metrics are used to validate the model's accuracy.

What stands out its novelty is that it analyses the power grid of Ho Chi Minh City in Vietnam. The research focuses on certain anomalies of electricity usage in the area. These anomalies are correlated with socioeconomics and seasonal changes. This research work differs from the general research works on the CNNs and the optimization methods. It focuses on the particular issues of the region. One of them is the problem of excessive volatility of electric power consumption. Another issue is the existing attitude regionally power systems applied modern optimization techniques. The research investigates three approaches to hyperparameter optimization. These were the Grid Search, Random Search, and Bayes Search methods. Each of them is assessed concerning the degree to which they are able to enhance the CNN model. The purpose of the modeling is to satisfy the conditions of the power grid system of Ho Chi Minh City.

This work attempt solve the problem of neglected hyperparameter tuning on the city's power grid systems.

It illustrates the use of Bayes Search in combination with self-training algorithm and how it enhances a customer's computing performance. The results show that Bayes Search can lead to more precise predictions compared to other methods. The analysis was performed on data obtained from a specific part of a city. It measures the performance of the developed models using MAE, MAPE, RMSE, and MSE which are considered error metrics. Evaluations of this nature are intended to complement the findings. This work commences further research in this area. It demonstrates the application of contemporary methods in load forecasting at short time intervals. Other cities which have the same power difficulties may use the methods presented in this stud.

2. THEORETICAL BASIS

2.1. CNN MODEL

Deep Learning is a part of machine learning. It uses artificial neural networks to process data in layers. It helps the model understand complex patterns. Convolutional Neural Network (CNN) is one type of neural network. It is made to process data in multiple arrays. Images are a common example. CNN models are widely used in vision tasks. These tasks include face detection, image recognition, and object identification.

A CNN has many parts. Some of them are key components. The convolutional layer is one of them. It uses filters to extract features from the input data. The activation layer is another part. It adds non-linearity. This helps the model learn complex patterns. The pooling layer works with the two layers. It reduces data size but keeps important information. The fully connected layer comes at the end. It brings together all extracted features. It handles the final classification or prediction [19].

Convolutional Layer: A filter (or filters) is slid through the input image to create a featured map in a convolutional layer. Each filter is small (3x3 or 5x5) and is applied to the entire input image to create a new featured map. The mathematics of this process can be represented as follows: For the input image I and filter F, the characteristic map is calculated by the convolutional product:

$$S(i, j)=(F*I)(i, j)= \sum_m\sum_n F(m,n)I(i-m,j-n) \tag{1}$$

Where:

(i, j): is the location on the characteristic map.

$S(i, j)$: The output value at position (i,j) after applying the convolution operation.

$F*I$: The convolution operation between the filter F and the input image I.

$F(m, n)$: The value at position (m,n) in the filter (also called a kernel).

$I(i-m, j-n)$: The value at position $(i-m, j-n)$ in the input image I.

$\sum_m\sum_n$: Summation over all elements of the filter.

Activation Layer: After the filter is applied, the values on the characteristic map are passed through a nonlinear trigger function, usually ReLU (Rectified Linear Unit). The ReLU function is defined as:

$$ReLU(x)=max(0,x) \tag{2}$$

Where:

x: The input value to the activation function (which can be the output from a previous layer in a neural network).

$ReLU(x)$: The output value of the ReLU activation function.

$max(0,x)$: The function returns the more excellent value between 0 and x.

Pooling Layer: Pooling typically uses max pooling or average pooling to reduce the spatial size of featured maps, highlight essential features, and reduce the number of parameters. Maximum compounding is defined as:

$$P(i, j)=max_{k,\, l \in window}\, I(i+k, j+l) \tag{3}$$

Where:

$P(i, j)$: The output value of the max pooling operation at position (i, j).

$max_{k,\, l \in window}$: The maximum value is selected from the specified window.

$I(i+k, j+l)$: The input characteristic map (or feature map), where $i+k, j+l$ represents the elements inside the pooling window.

Fully Connected Layer: Data from the fully connected layer is flattened and fed into one (or more) fully connected layer (s). Each neuron in this layer is connected to all the neurons in the previous layer, each with its own weight. The output of this class is:

$$y=W_x+b \tag{4}$$

Where

x: is the input from the previous layer.

W: i s the weighted matrix.

B: is the bias vector.

CNN's mathematical model is complex and requires optimization during training to learn effective weights and filters. Typically, this is done with the aid of backpropagation algorithms. If there is optimization involved, one method is Gradient Descent.

2.2. CNN NETWORK HYPERPARAMETERS

CNNs are mostly used for image processing. However, they can also be used for time series tasks like power load forecasting. Many hyperparameters are important for this type of work. These parameters help define the model's performance and relevance.

- Fewer filters make it harder for the model to detect features. More filters help the model learn complex patterns. Too many filters increase computation time. The number of filters must balance complexity and performance.

- Larger filter sizes increase the visible area during convolution. They help the model capture broader patterns. Smaller filters focus on fine details. The filter size affects how well the model detects time-based patterns.
- Stride controls how far the filter moves. A large stride reduces computations but may miss important data. A small stride captures more detail but increases cost.
- Padding defines how the model treats the input edges. 'Same' padding keeps the input and output size equal. 'Valid' padding reduces the output size. Padding affects how the model reads edge data.
- Activation functions add non-linearity. ReLU is the most common. It helps reduce the vanishing gradient and speeds up training.
- Pooling layers reduce the number of feature maps. Max pooling and average pooling are the most used. Pooling helps reduce overfitting and focuses on key features.
- The learning rate controls how fast weights update. A high learning rate speeds up training but may cause errors. A low learning rate trains slowly but more carefully.
- Batch size is the number of training samples processed at once. Small batches train slowly but learn fine patterns. Large batches train fast but may miss details.
- Epochs define how many times the model goes over the data. More epochs give more learning chances. Too many may cause overfitting.
- Regularization helps reduce overfitting. L1 and L2 add penalties to large weights. Dropout turns off some neurons during training. These methods help the model generalize better.

Choosing correct values for all parameters improves CNN performance on time series data. Hyperparameters also affect the CNN structure and real-world use.

3.2. HYPERPARAMETER OPTIMIZATION METHOD

Tuning is a key component when training the machine learning model because it fine tunes the performance of the model. It determines the optimal values for a number of adjustable predefined settings like learning rate, batch size, hidden layers, neurons per layer, etc. All of these must be configured prior to commencing the training thus rendering them immutable during the training process. Hyperparameters are distinct from model weights which are changed during training. Hyperparameters remain unchanged thereby making tuning a time-consuming process that requires enormous computational resources. The accuracy of the model on new data is improved, though. In the absence of tuning, overfitting or underfitting of the model is likely resulting in poor performance in real life scenarios.

Grid Search

Grid search is a method for hyperparameter optimization. It tests all possible combinations of hyperparameter values. It works well in small search spaces and when high precision is needed. Bergstra and Bengio [20] said Grid Search is easy to learn and use for beginners. However, it becomes inefficient in large search spaces because it takes a lot of time and resources. For example, three hyperparameters with four values each require 64 runs. Hutter et al. [21] called it "brute-force" and said it does not learn from past trials. Because of this, deep learning and decision tree models often use other methods. Random Search and Bayesian Search are common alternatives. They focus on promising regions and save resources. Petro and Pavlo Liashchynskyi [22] noted the strength of these methods.

Random Search

Among many methods, Random Search is a practical option for hyperparameter tuning. It works well in large and complex search spaces. Grid Search tests every possible configuration. In contrast, Random Search tests a limited number of random configurations. Navon and Bronstein [23] showed that Random Search can still give good results. It also uses fewer resources, which helps when computational power is limited. First, the user selects the hyperparameters. Then, the user sets the value ranges. The algorithm randomly generates a few configurations to test. Florea and Andonie [24] suggested a new version called Weighed Random Search. This version still uses random generation. However, it adds probabilistic rules to focus on better parts of the search space. This makes the search more efficient. Still, Random Search is not always precise. If too few configurations are tested, the results may be poor. In such cases, Bayes Search is better. Bischl et al. recommended it for better accuracy [25]. This method learns from past tests. It helps the model find the best areas to search.

Bayes Search

Like other forms of hyperparameter tuning, BayesSearchCV attempts to minimize some objective function. The difference is its use of Bayesian optimization, an advanced technique used to optimize functions that are costly or time-consuming to evaluate. It usually employs Gaussian Processes to try and direct the search to better places. The model first looks to find the promising regions within the space to search. After that, it tries to focus on evaluating those regions. Compared to Grid Search or Random Search, this approach considerably lowers the amount of attempts required [26]. The model decides upon a new set of parameters, evaluates the function and adjusts the estimate of its performance accordingly. BayesSearchCV is especially great when resources are limited [27]. However, it sacrifices some accuracy due to the surrogate model. It also needs extra settings which makes it harder to use. Still, it performs better with complex models. Deep neural networks, for example, benefit a lot. They need fewer trials to find the best configurations [25].

3. PROPOSED OPTIMIZATION MODEL STRUCTURE

3.2. ALGORITHMIC FLOWCHART

The authors explain an algorithm flowchart in this article. Its purpose is to evaluate the efficiency of three optimization methods which include: Grid Search, Random Search, and Bayes Search. The analysis is conducted on the CNN model. Figure 1 shows the flowchart.

Where:

X_{train}, Y_{train}: are the training inputs for the model.

X_{test}, Y_{test}: are the testing inputs for the model.

Y_{pred}: symbolizes the output from the model after execution.

$Y1, Y2, \ldots, Yn$: represent the past load values

The main processes of the flowchart are described as follows:

- Input Data: The process begins with the input data, consisting of historical load values ($Y1, Y2, \ldots, Yn$).
- Input Data Processing: The input data is preprocessed and divided into Training Data (X_{train}, Y_{train}) used to train the CNN model and testing Data (X_{test}, Y_{test}) used to evaluate the model's performance.
- Search Space of Hyperparameters: A predefined search space of hyperparameters is established, specifying possible configurations for the CNN model.
- Optimization Algorithms: Three optimization algorithms, Grid Search, Random Search, and Bayes Search, are applied to explore the hyperparameter search space and identify the Optimal Hyperparameters.
- Training the CNN Model: The CNN model is trained on the training dataset (X_{train}, Y_{train}) using the optimal hyperparameters.
- Prediction: The trained CNN model predicts the output (Y_{pred}) based on the test input data (X_{test}).
- Evaluation: The predicted values (Y_{pred}) are compared with the actual test data (Y_{test}), and the errors are calculated using metrics such as MAE, MAPE, MSE, and RMSE.

The following formula describes the mathematical model of the error rates used in the paper. Where y_i is the actual value and $\hat{y}_i$ is the predicted value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (5)$$

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \quad (6)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (8)$$

Where

MAE: Mean Absolute Error.

$MAPE$: Mean Absolute Percentage Error.

MSE: Mean Squared Error.

$RMSE$: Root Mean Squared Error.

n: The total number of data points.

yi: The actual value at index i.

$\hat{y}_i$:The predicted value at index i.

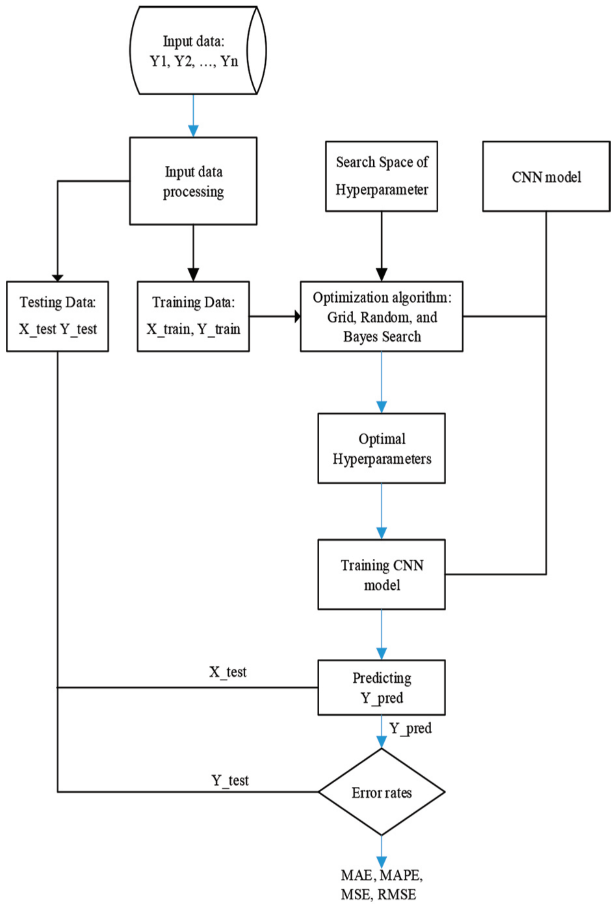

Fig. 1. Hyperparameter optimization algorithm diagram

4. CNN HYPERPARAMETER OPTIMIZATION ANALYSIS RESULTS

4.1. EXPERIMENTAL SETUP

In this study, the authors employed the electricity load dataset of Ho Chi Minh City, Vietnam, as presented in Table 1 below. The data sampling interval is 60 minutes, resulting in 24 data points daily. A sliding window approach with a window size of 24 generated Input-Target pairs (X, Y). The dataset (X, Y) consists of 840 samples, which were divided into a training dataset (X_{train}, Y_{train}) and a testing dataset (X_{test}, Y_{test}) with a ratio of 8:2.

Table 1. Historical Load Data in Ho Chi Minh City from 12/9/2016 to 31/12/2018

Date	00:00	01:00	……….	22:00	23:00
12/09/2016	1842.1	1795.1	……….	2337.2	2110.1
14/09/2016	1975.7	1914.6	……….	2297.5	2106.2
……….	……….	……….	……….	……….	……….
30/12/2018	2083.3	1980.9.	……….	2325.4	2127.8
31/12/2018	1902.7	1776.4	……….	2233.8	2059.5

Fig. 2 presents the electricity load profile for December 31, 2018. The chart shows an apparent fluctuation in the electricity load throughout the day, with the minimum load occurring in the early morning (1072.4 MW) and the maximum load during peak hours (2032.9 MW). This reflects the low electricity demand at night and early morning, while the demand increases significantly around midday when people and facilities use the most electricity. This chart helps identify usage trends throughout the day, allowing for better planning of an efficient power supply. Understanding these fluctuations not only aids in managing electricity distribution more effectively but also helps optimize operational strategies and distribution, ensuring that peak demand is met while saving resources during off-peak hours.

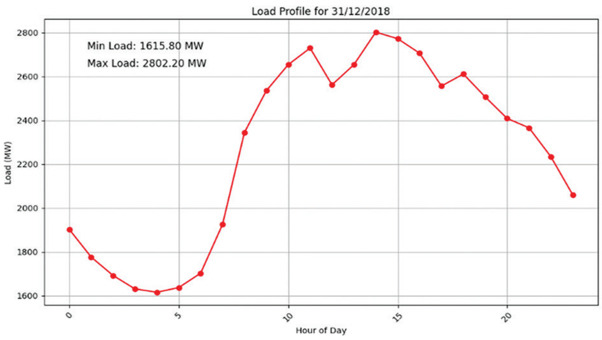

Fig. 2. Electricity Load on December 31, 2018

Table 2 presents the search space for the hyperparameters of the CNN model under investigation, including the number of filters, kernel size, batch size, and epochs. These search spaces are consistently applied across the Grid Search, Random Search, and Bayes Search algorithms.

Table 2. Hyperparameter search space

Hyperparameter	Search Space	Description
Filters	[16, 32, 64, 96, 128]	Number of filters in the Conv1D layer
Kernel Size	[3, 5, 7]	Size of the convolutional window
Batch Size	[16, 32, 64, 128, 256]	Number of samples processed in each training step
Epochs	[50, 100, 150, 200, 250]	Number of complete passes through the training dataset.

4.2. EXPERIMENTAL RESULTS

Table 3 presents the optimal hyperparameter values obtained from the three algorithms: Grid Search, Random Search, and Bayes Search. These results correspond to the hyperparameter search space described in Table 2 for the CNN model.

Table 3. Hyperparameter sets with optimal methods

	Randon Search	Gird Search	Bayes Search
Filters	32	32	64
Kernel Size	7	7	5
Batch Size	16	32	32
Epochs	250	250	200

Figs. 3, 4, and 5 illustrate how the predicted values from Grid Search, Random Search, and Bayes Search align with actual observations. The visual similarity between predicted and real values suggests that each algorithm successfully fine-tunes the CNN model to an acceptable level of accuracy. However, the differences in precision still matter when choosing the optimal method. The results prove the effectiveness and accuracy of the methodologies employed.

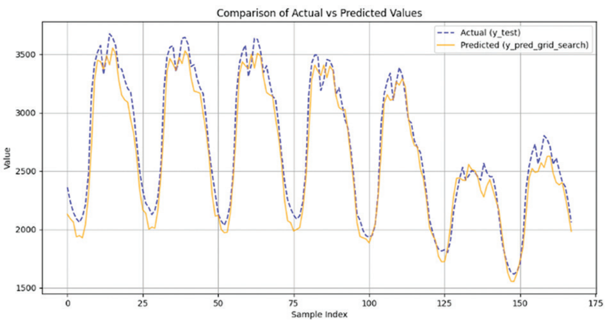

Fig. 3. The graph of predicted values using the Grid Search algorithm

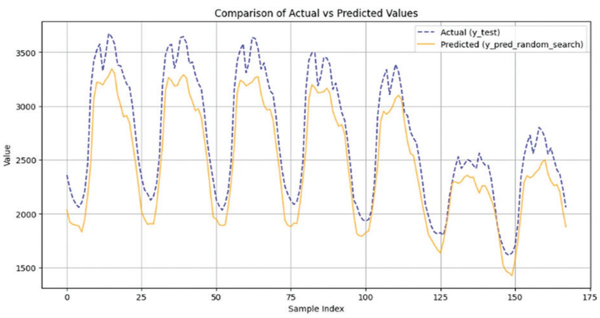

Fig. 4. Predicted values graph using Random Search

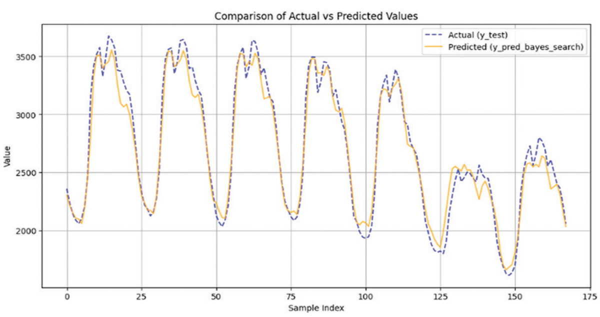

Fig. 5. Predicted values graph using Bayes Search

The error metrics for Grid Search, Random Search, and Bayes Search are shown in Table 4. The metrics include MAE, MAPE, MSE, and RMSE. Bayes Search gives the best results. It has the lowest MAE of 81.94. It also gives a MAPE of 3.09%, an MSE of 11,458.0, and an RMSE of 107.1. These values show high accuracy. They also show that Bayes Search improves model robustness. This makes it the most effective method. Random Search gives the worst results. It has an MAE of 166.7, a MAPE of 5.92%, an MSE of 36,783.19, and an RMSE of 191.7. These values show poor performance in finding good hyperparameters. Random Search is simple but not efficient here. Grid Search performs better than Random Search. It gives an MAE of 124.15, a MAPE of 4.58%, an MSE of 22,116.2, and an RMSE of 148.7. Grid Search can find near-optimal values. However, it is still less precise than Bayes Search.

Table 4. The results of error rates

Search_Method	MAE	MAPE	MSE	RMSE
Grid Search	124.15	4.58	22116.2	148.7
Random Search	166.7	5.92	36783.19	191.7
Bayes Search	81.94	3.09	11456.0	107.1

Fig. 6 presents the execution time for the three algorithms: Grid Search, Random Search, and Bayes Search. The runtime comparison shows that Grid Search is the slowest, taking nearly 9,551.44 seconds due to its exhaustive evaluation. Random Search improves efficiency with a runtime of around 3,796.13 seconds. Bayes Search is the fastest, completing it in just over 4,218.63 seconds, making it the most efficient method, according to the execution time.

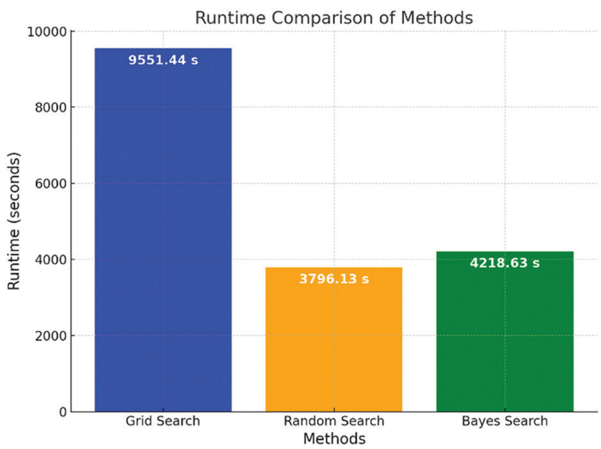

Fig. 6. Runtime of Optimization Methods

5. CONCLUSION

This study underlines how hyperparameter optimization—using techniques like Grid Search, Random Search, and Bayes Search—can significantly enhance CNN model performance for forecasting electricity demand. Among them, Bayes Search showed the highest predictive accuracy and the shortest processing time. In contrast, while Grid Search was moderately accurate, its longer runtime made it less practical. Random Search was quicker but less precise, making it a less reliable option for pinpointing the best parameter combinations Although moderately accurate, the Grid Search algorithm had the longest execution time, reflecting its inefficiency for large-scale problems. The Random Search algorithm showed better runtime efficiency than the Grid Search algorithm but produced higher error values, making it less dependable for optimal configurations. Future studies could further investigate advanced optimization algorithms and their application to larger datasets to enhance forecasting performance.

6. REFERENCES

[1] T. Zhao, J. Wang, Y. Zhang, "Day-Ahead Hierarchical Probabilistic Load Forecasting with Linear Quantile Regression and Empirical Copulas", IEEE Access, Vol. 7, 2019, pp. 80969-80979.

[2] P. Li et al. "Dynamic Similar Sub-Series Selection Method for Time Series Forecasting", IEEE Access, Vol. 6, 2018, pp. 32532-32542.

[3] S. Sharma, A. Majumdar, V. Elvira, E. Chouzenoux, "Blind Kalman Filtering for Short-Term Load Forecasting", IEEE Transactions on Power Systems, Vol. 35, No. 6, 2020, pp. 4916-4919.

[4] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, Z. Zhu, "Multi-scale convolutional neural network with time-cognition for multi-step short-Term load forecasting", IEEE Access, Vol. 7, 2019, pp. 88058-88071.

[5] D. N. Hung, N. M. Tam, N. T. Linh,

N. T. Hoan, N. T. Duy "Application of combining data preprocessing with wavelet filtering for GCN-LSTM network with HHO optimization algorithm in load forecasting model", TNU Journal of Science and Technology, Vol. 229, No. 6, 2024, pp. 160-169.

[6] S. Li, L. Goel, P. Wang, "An ensemble approach for short-term load forecasting by extreme learning machine", Applied Energy, Vol. 170, 2016, pp. 22-29.

[7] W. Lin, D. Wu, B. Boulet, "Spatialoral Residential Short-Term Load Forecasting via Graph Neural Networks", IEEE Transactions on Smart Grid, Vol. 12, No. 6, 2021, pp. 5373-5384.

[8] C. Barış, C. Yanarateş, A. Altan, "A robust chaos-inspired artificial intelligence model for dealing with nonlinear dynamics in wind speed forecasting", PeerJ - Computer Science, Vol. 10, 2024, pp. 1-40.

[9] K. Aurangzeb, M. Alhussein, K. Javaid, S. I. Haider, “A Pyramid-CNN based deep learning model for power load forecasting of similar-profile energy customers based on clustering”, IEEE Access, Vol. 9, 2021, pp. 14992-15003.

[10] S. H. Rafi, N. Al-Masood, S. R. Deeba, E. Hossain, “A short-term load forecasting method using integrated CNN and LSTM network”, IEEE Access, Vol. 9, 2021, pp. 32436-32448.

[11] H. U. I. H. Goh, S. Member, B. He, “Multi-Convolution Feature Extraction and Recurrent Neural Network Dependent Model for Short-Term Load Forecasting”, IEEE Access, Vol. 9, 2021, pp. 118528-118540.

[12] S. M. J. Jalali et al. “A Novel Evolutionary-Based Deep Convolutional Neural Network Model for Intelligent Load Forecasting”, IEEE Transactions on Industrial Informatics, Vol. 17, No. 12, 2021, pp. 8243-8253.

[13] A. Altan, S. Karasu, “Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization”, Energy, Vol. 242, p. 122964.

[14] F. Xu, G. Weng, Q. Ye, Q. Xia, “Research on Load Forecasting Based on CNN-LSTM Hybrid Deep Learning Model”, Proceedings of the IEEE 5th International Conference on Electronics Technology, Chengdu, China, 13-16 May 2022, pp. 1332-1336.

[15] İ. Yağ, A. Altan, “Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments”, Biology, Vol. 11, No. 12, 2022.

[16] H. Aldosari, “A Hybrid Support Vector Machine with Grasshopper Optimization Algorithm based Feature Selection for Load Forecasting”, Proceedings of the International Conference on Distributed Computing and Optimization Techniques, Bengaluru, India, 15-16 March 2024, pp. 1-4.

[17] L. Yang, A. Shami, “On hyperparameter optimization of machine learning algorithms: Theory and practice”, Neurocomputing, Vol. 415, 2020, pp. 295-316.

[18] M. Munem, T. M. Rubaith Bashar, M. H. Roni, M. Shahriar, T. B. Shawkat, H. Rahaman, “Electric power load forecasting based on multivariate LSTM neural network using Bayes Search”, Proceedings of the IEEE Electric Power and Energy Conference, Edmonton, AB, Canada, 9-10 November 2020, pp. 5-10.

[19] R. Patel, S. Patel, “A comprehensive study of applying convolutional neural network for computer vision”, International Journal of Advanced Science and Technology, Vol. 29, No. 6, 2020, pp. 2161-2174.

[20] J. Bergstra, Y. Bengio,“Random Search for Hyper-Parameter Optimization”, Journal of Machine Learning Research, Vol. 13, 2012, pp. 281-305.

[21] C. A. Coello Coello, "Learning and Intelligent Optimization", Vol. 3618, Springer, 2005.

[22] P. Liashchynskyi, P. Liashchynskyi, “Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS”, arXiv:1912.06059, 2019.

[23] D. Navon, A. M. Bronstein, “Random Search Hyper-Parameter Tuning: Expected Improvement Estimation and the Corresponding Lower Bound”, arXiv:2208.08170, 2022.

[24] A. C. Florea, R. Andonie, “Weighted Random Search for hyperparameter optimization”, International Journal of Computers Communications & Control, Vol. 14, No. 2, 2019, pp. 154-169.

[25] B. Bischl et al. “Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges”, WIREs Data Mining and Knowledge Discovery, Vol. 13, No. 2, 2023, pp. 1-43.

[26] J. Snoek, H. Larochelle, R. P. Adams, “Practical Bayes Search of machine learning algorithms”, Advances in Neural Information Processing Systems, Vol. 4, 2012, pp. 2951-2959.

[27] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, N. De Freitas, “Taking the human out of the loop: A review of Bayes Search”, Proceedings of the IEEE, Vol. 104, No. 1, 2016, pp. 148-175.