Proceedings of Engineering and Technology Innovation, vol. x, no. x, 20xx, pp. xx-xx

Enhancing Load Forecasting Accuracy of Neural Hierarchical Interpolation for Time Series through Hyperparameter Optimization

Thanh Ngoc Tran, Tuan Anh Nguyen*

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

Received 26 March 2026; revised 14 May 2026; accepted 15 May 2026

DOI: https://doi.org/10.46604/peti.2026.16317

Abstract

This study aims to improve the forecasting accuracy of the neural hierarchical interpolation for time series (NHITS) model through a hyperparameter optimization framework. Three optimization strategies, namely random search (RS), tree-structured Parzen estimator (TPE), and genetic algorithm (GA), are compared under the same search space and evaluation budget. Three key hyperparameters, including mlp_units, learning_rate, and max_steps, are optimized because they directly affect model capacity, convergence behavior, and training effort. Experiments are conducted on two Australian half-hourly electricity demand datasets, New South Wales (NSW) and Queensland (QLD), each containing 122,735 records. Forecasting performance is evaluated using MSE, RMSE, MAE, and MAPE. The results show that all optimization methods improve the default NHITS configuration, while TPE achieves the best performance. MAPE decreases from 2.21% to 1.23% for NSW and from 3.34% to 1.79% for QLD.

Keywords: NHITS, electricity load forecasting, random search, tree-structured Parzen estimator, genetic algorithm

1. Introduction

Accurate and reliable load forecasting is essential for secure and economical power system operation, as it supports key decisions such as unit commitment, day-ahead scheduling, market dispatch, and long-term capacity planning. Forecasting performance across different temporal horizons, including very short-, short-, medium-, and long-term horizons, plays an important role in maintaining system stability and facilitating data-driven grid management [1]. In practical applications, electricity load series typically exhibit strong daily and weekly seasonality, holiday effects, abrupt peak variations, and regional differences in consumption behavior. These characteristics require forecasting models that can effectively capture multi-scale temporal patterns while maintaining stable performance across different datasets.

Load forecasting methods have evolved through three major stages: statistical, machine learning, and deep learning approaches. In the early stage, statistical methods such as ARIMA/SARIMA [2] and seasonal regression [3] were widely used due to their interpretability, solid theoretical foundations, and relatively low computational cost. However, their reliance on linear assumptions and rigid model structures limits their ability to capture the nonlinear behaviors and regime shifts often observed in real-world electricity demand data.

To address these limitations, machine learning methods such as support vector machines (SVM) [4-5], random forest [6-7], and XGBoost [8] have been introduced. These methods provide stronger nonlinear modeling capabilities and can capture more complex relationships between load demand and explanatory variables. Nevertheless, their performance often depends heavily on manual feature engineering, including the construction of calendar and weather-related features, as well as the careful selection of hyperparameters for each forecasting task.

* Corresponding author. E-mail address: nguyenanhtuan@iuh.edu.vn

More recently, deep learning models have achieved higher forecasting accuracy by automatically extracting representative features from raw or minimally processed data. Recurrent architectures such as RNN [9], LSTM [10-11], and GRU [12] are effective in modeling long-term temporal dependencies, whereas convolution-based models such as CNN [13-14] and TCN [15] are suitable for extracting local temporal patterns. Hybrid models, including CNN–LSTM [16-17] and task-specific Transformer variants [18-19], further enhance forecasting performance by combining local and global temporal information within the load time series.

Among recent deep forecasting models, neural hierarchical interpolation for time series (NHITS) [20] has emerged as a promising approach for multi-horizon forecasting. Built upon the N-BEATS family [21-22], NHITS employs hierarchical interpolation and multi-frequency aggregation to improve forecasting stability and computational efficiency. These mechanisms allow the model to capture temporal information at different resolutions, making it suitable for electricity load series with both short-term fluctuations and longer-term periodic patterns. Despite these advantages, NHITS remains sensitive to hyperparameter settings across different datasets [23]. In particular, the number of MLP units, learning rate, and maximum training steps strongly influence model capacity, optimization stability, and the training budget, respectively. Inappropriate settings may reduce forecasting accuracy, weaken peak tracking ability, and increase prediction errors.

Although NHITS has shown considerable potential, the effect of hyperparameter optimization on its performance in short-term electricity load forecasting remains insufficiently explored. In many practical applications, NHITS is configured using default settings or manually selected hyperparameters. This practice may lead to unstable forecasting performance when the model is applied to datasets with different regional load characteristics. Moreover, while hyperparameter optimization is widely used in machine learning, few studies provide a controlled comparison of different optimization paradigms for NHITS-based load forecasting under identical search spaces, random seeds, and evaluation budgets. This issue is important because different optimizers may produce different forecasting results not only because of their search mechanisms, but also because of differences in search ranges, computational budgets, and experimental conditions. Therefore, a fair and reproducible comparison is needed to identify the optimization strategy best suited for NHITS in short-term electricity load forecasting.

To address this gap, this study develops a fixed-budget hyperparameter optimization framework to improve the short-term load forecasting accuracy of NHITS. Three representative search strategies are considered: random search (RS) [24], tree-structured Parzen estimator (TPE) [25], and genetic algorithm (GA) [26]. These methods represent random, Bayesian, and evolutionary optimization paradigms, respectively, and provide complementary search characteristics for identifying suitable NHITS hyperparameters. The framework focuses on three influential hyperparameters, namely mlp_units, learning_rate, and max_steps, because they directly affect model capacity, training convergence, and the computational training budget. Accordingly, this study aims to evaluate the effectiveness of different optimization strategies for improving NHITS performance under a consistent and reproducible experimental protocol.

The main contributions of this study are summarized as follows. First, a controlled and reproducible optimization framework is developed for NHITS-based short-term electricity load forecasting. Second, three different optimization paradigms, namely RS, TPE, and GA, are compared under the same search space, random seed, and evaluation budget. Third, the study analyzes the effects of optimizing three key NHITS hyperparameters and investigates their impact on forecasting performance. Fourth, the optimized NHITS configurations are evaluated on two Australian regional electricity demand datasets, NSW and QLD, using MSE, RMSE, MAE, and MAPE. Finally, the study provides practical insights into selecting effective optimization strategies for NHITS-based load forecasting under limited computational resources.

The remainder of this manuscript is organized as follows. The next section describes the proposed methodology, including the NHITS model and the hyperparameter optimization strategies. The subsequent section presents the experimental setup, datasets, search space, and evaluation metrics. Subsequently, the forecasting results and comparative analysis are discussed. Finally, the main findings, limitations, and conclusions are summarized in the last section.

2. Methodology

This section presents the methodology of the proposed NHITS-based short-term load forecasting framework. It first describes the structure and main operating principles of the NHITS model. Then, the hyperparameter optimization problem and the three optimization algorithms, including RS, TPE, and GA, are introduced. Finally, the overall optimization procedure and the corresponding framework are provided to clarify the complete training, optimization, and forecasting process.

2.1 NHITS Model

Neural hierarchical interpolation for time series (NHITS) is a neural network architecture designed for long-horizon time-series forecasting. The model consists of a sequence of forecasting blocks, each implemented as a multi-layer perceptron (MLP). Each block produces two outputs: a backcast, which estimates and removes the explained component from the input signal, and a forecast, which is aggregated into the final prediction.

The main advantage of NHITS lies in its hierarchical interpolation and multi-frequency sampling mechanisms. The input signal is downsampled to extract features at different temporal resolutions, covering both low- and high-frequency components. The resulting outputs are then upsampled to a common resolution and aggregated to generate the final forecast. This hierarchical representation enables the model to capture both short-term fluctuations and long-term temporal dependencies, thereby improving forecasting stability and reducing computational complexity for long-horizon prediction tasks.

By progressively refining the residual signal across successive blocks, NHITS forms a residual learning framework in which each block focuses on correcting the remaining error after backcast subtraction from the previous block. As a result, NHITS provides an effective and scalable framework for electricity load forecasting applications, particularly when the load series contains multi-scale temporal patterns such as daily cycles, weekly seasonality, and short-term peak variations.

The fundamental equations describing the operation of the NHITS model are summarized as follows:

$$(\hat{y}_t^{(b,(i))}, \hat{y}_t^{(f,(i))}) = f_{\theta_i}(x_t^{(i)}) \tag{1}$$

Residual update for the next block:

$$x_t^{(i+1)} = x_t^{(i)} - \hat{y}_t^{(b,(i))} \tag{2}$$

Final aggregated forecast from all blocks:

$$\hat{y}_t^{f} = \sum_{i=1}^{K} \hat{y}_t^{(f,(i))} \tag{3}$$

Hierarchical interpolation (downsampling and upsampling):

$$\hat{y}_i^{(f,(i))} = U_l\left(f_{\theta_l}\left(D_i(x_i) \right) \right) \tag{4}$$

Training loss function (mean squared error):

$$L(\theta) = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2 \tag{5}$$

These equations describe the core mechanism of NHITS, in which each forecasting block refines the residual error through backcast-forecast operations and hierarchical interpolation. Through this mechanism, NHITS can learn temporal patterns at different resolutions and generate stable forecasts for long-horizon time-series prediction tasks.

Similar to other deep learning models, NHITS contains several important hyperparameters that must be properly configured to ensure stable convergence and effective forecasting performance. In this study, particular attention is given to three hyperparameters: the number of MLP units, the learning rate, and max_steps. These parameters were selected because they directly influence three important aspects of the training process. First, MLP units determine the number of hidden-layer neurons in each forecasting block, thereby controlling the model's representational capacity. A very small number of units may lead to underfitting, whereas an excessively large number may increase model complexity and the risk of overfitting. Second, the learning rate controls the step size during parameter updates and strongly affects convergence stability. An inappropriate learning rate may cause slow convergence or unstable training. Third, max_steps define the maximum number of training iterations and therefore determines the training budget and the degree to which the model can learn from the available data.

Other NHITS parameters, including batch size, dropout rate, number of blocks, and stack expansion coefficients, were kept fixed according to the standard NeuralForecast configuration. This setting was adopted to maintain a controlled experimental design and to isolate the effects of the three selected hyperparameters. Optimizing all NHITS parameters simultaneously would substantially enlarge the search space and increase computational cost, especially under the fixed-budget comparison adopted in this study. Therefore, the present work focuses on three hyperparameters closely related to model capacity, optimization behavior, and training effort. According to the standard configuration provided in the NeuralForecast library, the default NHITS model uses an MLP architecture with 512 neurons per hidden layer, a learning rate of 0.001, and a maximum of 1,000 training steps.

However, these default values may not be optimal for every dataset. In electricity load forecasting applications, careful hyperparameter tuning is essential because different regional load patterns may require different levels of model capacity, learning stability, and training effort. Based on the NHITS configuration considered in this study, the default and fixed parameters are summarized in Table 1.

Table 1 Default and fixed NHITS configuration

Hyperparameter	Description	Default Value	Optimized
MLP units	Number of neurons in the hidden layers of each MLP block	512	Tune
Learning rate	Step size for updating model weights	0.001	Tune
Max steps	Maximum number of training iterations	1,000	Tune
Batch size	Number of samples per training batch	32	Fixed
Dropout rate	A fraction of neurons is randomly dropped to reduce overfitting	0.1	Fixed
Number of blocks	Number of hierarchical blocks (stacks) in NHITS architecture	3	Fixed
Stack expansion	Expansion factor controlling the width of MLP layers per block	2	Fixed

The hyperparameters listed in Table 1 include both optimized and fixed parameters used in this study. Among them, MLP units, learning rate, and max_steps were selected for optimization. In contrast, batch size, dropout rate, number of blocks, and stack expansion were kept at their default values to maintain a consistent NHITS architecture across all experiments. This design allows comparisons among RS, TPE, and GA to focus on the effects of the selected hyperparameters within the same model structure and computational budget. Therefore, the selected configuration provides a controlled and practical basis for assessing the effectiveness of different hyperparameter optimization strategies for NHITS-based load forecasting.

2.2 Hyperparameter Optimization Algorithms

In this study, hyperparameter optimization is formulated as a minimization problem that identifies the NHITS hyperparameter configuration that minimizes forecasting error. The hyperparameter vector is defined as:

$$\theta = \{mlp\_units,\ learning\_rate,\ max\_steps\} \tag{6}$$

The optimal configuration can be expressed as:

$$\theta^* = arg\ min\, f(\theta),\ \theta \in \Omega \tag{7}$$

where θ denotes a candidate hyperparameter configuration, Ω is the predefined search space, and $f(\theta)$ represents the forecasting error obtained by training and evaluating the NHITS model with θ. In this study, MAPE is used as the main objective function during optimization because it provides a scale-independent measure of percentage error and allows intuitive comparison between the NSW and QLD datasets.

Three representative hyperparameter optimization algorithms are considered in this study: RS, TPE, and GA. These methods are selected because they represent three distinct optimization paradigms: random sampling, Bayesian optimization, and evolutionary optimization. Applying these methods under the same search space, random seed, and evaluation budget enables a fair comparison of their effectiveness in NHITS-based short-term load forecasting. This controlled setting allows performance differences to be mainly attributed to the search strategies.

Random Search (RS): Random search is a simple and widely used hyperparameter optimization method in which candidate parameter combinations are independently sampled from a predefined search space [24]. For each trial, a hyperparameter configuration θ_i is randomly generated from Ω, and the NHITS model is trained and evaluated using this configuration. The best configuration is selected according to the lowest MAPE value:

$$\theta_{RS}^{*} = \arg\min_{\theta_i \in \Omega} f(\theta_i), \quad i = 1, 2, \ldots, N \tag{8}$$

where N is the total number of trials. Despite its simplicity, RS can be effective when only a few hyperparameters strongly influence model performance. Its main advantages are ease of implementation and low algorithmic complexity. However, RS does not use information from previous trials to guide the search process. Therefore, promising regions of the search space may not be adequately explored when the number of trials is limited.

Tree-structured Parzen Estimator (TPE): TPE is a Bayesian optimization algorithm widely used for hyperparameter tuning [25]. Unlike RS, TPE uses information from previous trials to guide the selection of subsequent configurations. Instead of directly modeling the objective function, TPE divides previous observations into two groups based on a threshold value y^*. The probability model can be expressed as:

$$p(\theta \mid y) = l(\theta), \ if \ y < y^* \tag{9}$$

$$p(\theta \mid y) = g(\theta), \ if \ y \geq y^* \tag{10}$$

where $l(\theta)$ represents the distribution of promising configurations with lower error values, and $g(\theta)$ represents the distribution of less promising configurations with higher error values. New candidate configurations are selected by favoring regions where the ratio $l(\theta)/g(\theta)$ is high. This mechanism allows TPE to concentrate the search on promising regions of the hyperparameter space while still maintaining exploration. Therefore, TPE is expected to be more sample-efficient than RS, especially under a limited evaluation budget.

Genetic Algorithm (GA): GA is an evolutionary optimization method inspired by natural selection [26]. In GA, each individual in the population represents a candidate configuration of NHITS hyperparameters. The quality of each individual is evaluated using a fitness function. Since this study aims to minimize MAPE, the fitness function can be defined as:

$$\text{Fitness}(\theta_i) = \frac{1}{f(\theta_i)} \tag{11}$$

Since lower MAPE values correspond to higher fitness values, the best-performing individuals are selected as parents, and new offspring are generated via crossover and mutation. Crossover combines information from two parent configurations, while mutation randomly modifies one or more hyperparameters to maintain population diversity. Over successive generations, the population gradually evolves toward better hyperparameter configurations. GA is useful for exploring complex search spaces and reducing the risk of being trapped in local optima. However, it usually requires more evaluations than RS and TPE because multiple individuals must be trained and evaluated within each generation.

Table 2 summarizes the experimental settings for the three hyperparameter optimization methods applied to NHITS. The same search space, optimization objective, random seed, and evaluation budget are used to ensure a fair and reproducible comparison.

Table 2 Configuration summary of hyperparameter optimization methods for NHITS

Configuration	RS	TPE	GA
Trials/evaluations	20 trials	20 trials	20 evaluations
Objective	Minimize MAPE		
Seed	42		
Search space	mlp_units: 32 – 960, step 32; learning_rate: log-uniform 10^{-4} – 10^{-2}; max_steps: integer 10 – 2000		
Algorithm settings	Randomly sample one configuration per trial and select the configuration yielding the minimum MAPE.	Use TPESampler (seed = 42, direction = “minimize”); previous trials guide the search after the initial random sampling stage.	Population size = 10, number of generations = 2, elitism = 2, top 50% selection, uniform crossover with a probability of 0.5 per gene, and mutation rate = 0.15.

As shown in Table 2, RS and TPE are each limited to 20 trials, whereas GA performs 20 evaluations through its population–generation structure, with a population size of 10 and 2 generations. The same random seed (42) is used for all methods to improve reproducibility and reduce randomness in the comparison. In addition, all optimization methods explore the same search space defined by the three selected NHITS hyperparameters.

The use of 20 evaluations was adopted as a fixed-budget setting to ensure that all three methods were compared under the same computational constraint. This setting is relevant for practical load forecasting applications where computational resources and training time are limited. Although increasing the number of trials may further improve the performance of each optimization method, using the same evaluation budget allows the observed differences in forecasting accuracy to be mainly attributed to the search strategy rather than to differences in computational cost. Therefore, Table 2 provides a controlled and reproducible basis for evaluating RS, TPE, and GA in NHITS-based short-term electricity load forecasting.

2.3 General Hyperparameter Optimization Procedure

Algorithm 1 summarizes the general optimization procedure used to determine the optimal hyperparameters of the NHITS model using RS, TPE, and GA. Although these optimization methods adopt different search mechanisms, they are evaluated under the same objective function, search space, and evaluation budget. After the best hyperparameter configuration is identified, the final NHITS model is retrained and evaluated using MAPE, MAE, MSE, and RMSE.

Algorithm 1 General pseudo-code for NHITS hyperparameter optimization using RS, TPE, and GA

```
Input:
1. D_train, D_test, NHITS model
2. Forecast horizon h, input size L
3. Search space Ω
4. Method M ∈ {RS, TPE, GA}
5. Max evaluations N
Output:
6. Best configuration θ* and performance (MAPE, MAE, MSE, RMSE)
7. Define Ω:
8. mlp_units ∈ {32, 64, …, 960}
9. learning_rate ∈ [10⁻⁴, 10⁻²]
10. max_steps ∈ [10, 2000]
11. Initialize MAPE_best ← +∞, θ* ← None
12. Set random seed
13. If M = GA initialize population; if TPE initialize sampler
14. For i = 1 to N:
15.   If M = RS:
16.      Sample θi randomly from Ω
17.   If M = TPE:
18.      Sample θi using TPE model
19.   If M = GA:
20.      Generate θi via selection, crossover, mutation
21.   Train NHITS with θi on D_train
22.   Evaluate MAPEi
23.   Store (θi, MAPEi)
24.   If MAPEi < MAPE_best:
25.      MAPE_best ← MAPEi
26.      θ* ← θi
27. End For
28. Train final NHITS with θ*
29. Forecast on D_test
30. Compute MAPE, MAE, MSE, RMSE
31. Return θ* and results
```

2.4 Flowchart of the Proposed Method Evaluation

Fig. 1 presents the overall workflow of the proposed NHITS optimization and evaluation framework for short-term electricity load forecasting. The procedure includes input data preparation, data preprocessing, training–testing data separation, hyperparameter search space definition, optimization using RS, TPE, and GA, NHITS model training, optimal hyperparameter selection, forecasting, and final performance evaluation. This workflow is designed to ensure that all optimization methods are compared under the same data structure, search space, objective function, and evaluation procedure. As shown in Fig. 1, the process begins with the original time-series electricity load data. These data are preprocessed and formatted into the input structure required by NHITS before sample generation for forecasting.

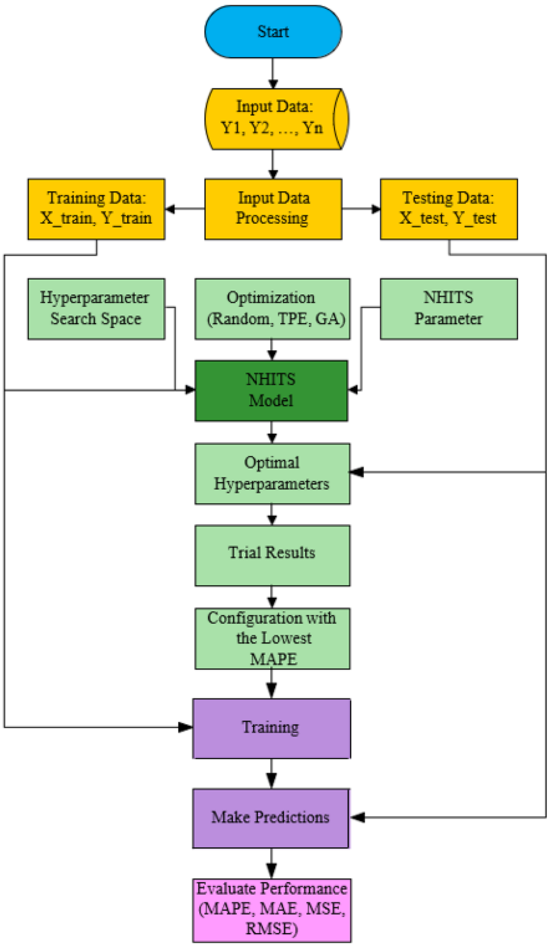

Fig. 1 NHITS model optimization and evaluation flow

After preprocessing, the dataset is divided into training and testing subsets. The training data, represented by X_train and Y_train, are used to train the NHITS model and evaluate candidate hyperparameter configurations. The testing data, represented by X_test and Y_test, is reserved for final forecasting evaluation. This separation is important because it ensures that the model performance is assessed using unseen data, thereby providing a more reliable evaluation of the generalization capability of the NHITS model. Next, the search space of the three selected hyperparameters, namely mlp_units, learning_rate, and max_steps, is defined. The search space defines the possible values that each optimization method can explore. By using the same search space for RS, TPE, and GA, the comparison among the three methods becomes fairer and more controlled.

RS, TPE, and GA then generate candidate hyperparameter configurations according to the search mechanism of the corresponding optimization method. This configuration is then passed to the NHITS model. The model is trained using the training data, and its forecasting performance is calculated. The MAPE value of each trial is recorded together with the corresponding hyperparameter configuration. In this way, all trial results are stored and compared to identify which configuration provides the best forecasting accuracy.

The optimization block in Fig. 1 searches for the most suitable NHITS hyperparameter configuration within the predefined search space. Although RS, TPE, and GA use different search mechanisms, they are evaluated under the same number of trials and objective function to ensure a fair comparison. After all trials are completed, the configuration with the lowest MAPE is selected as the optimal setting and used in the final training stage to generate the final forecasts.

In the prediction stage, the trained NHITS model produces load forecasts for the testing horizon. These predicted values are then compared with the actual load values in the testing dataset. The final evaluation is conducted using four standard forecasting metrics: MAPE, MAE, MSE, and RMSE. MAPE serves as the primary selection criterion because it expresses forecasting error as a percentage and is independent of the scale of the load data, making it suitable for comparing results across different regional datasets. Meanwhile, MAE, MSE, and RMSE are used as complementary metrics to provide a more comprehensive assessment of forecasting accuracy, including absolute error magnitude and squared error behavior. Overall, Fig. 1 summarizes the optimization, training, and evaluation procedure adopted in this study.

3. Experimental Setup

This section describes the experimental setup used to evaluate the proposed NHITS optimization framework. It includes the description of the NSW and QLD electricity demand datasets, the data preprocessing procedure, and the input–output structure used for one-day-ahead forecasting. The NHITS model configuration and selected hyperparameter search ranges are also presented. In addition, the experimental environment is described to ensure reproducibility and provide a clear basis for comparing the optimization methods under the same conditions.

3.1 Dataset Description

The experimental data used in this study consist of two regional electricity demand datasets, namely New South Wales (NSW) and Queensland (QLD), both obtained from the Australian National Electricity Market (NEM). Each dataset contains 122,735 records and two variables: settlementdate and totaldemand. The settlement date variable represents the timestamp of half-hourly electricity demand observations starting from January 1, 2015, while total demand denotes the total electrical load recorded at each settlement interval in megawatts (MW). Since the observations are recorded every 30 minutes, each day contains 48 load values.

The NSW and QLD datasets have the same temporal resolution, data format, number of records, and target variable. This consistency allows the same preprocessing procedure, input window, forecasting horizon, and evaluation protocol to be applied to both regions. Therefore, the forecasting performance of the optimized NHITS models can be compared under consistent experimental conditions. The main characteristics of the datasets are summarized in Table 3.

Table 3 Dataset summary for NSW and QLD

Region	Number of Records	Variables	Time Interval	Target Variable
NSW	122,735	settlementdate, totaldemand	30 minutes	totaldemand
QLD	122,735	settlementdate, totaldemand	30 minutes	totaldemand

As shown in Table 3, although both datasets share identical data structures, they represent different regional electricity demand patterns. This consistency reduces the influence of data-format differences on model performance and allows the comparison to focus on the effect of hyperparameter optimization. Moreover, using two regional datasets provides a more meaningful evaluation than using a single load series, because NSW and QLD may exhibit different demand patterns, peak-load behaviors, and regional consumption characteristics.

During data preprocessing, the settlement date column was converted into a standard datetime format, and the observations were sorted in ascending chronological order to preserve the temporal continuity of the load series. The total demand column was selected as the target for forecasting. After preprocessing, the data were split into training and test sets using the same procedure for both regions. Specifically, the last 28 days before the forecasting period were used for model training, with the final day reserved for testing and evaluation. Because the data were recorded at 30-minute intervals, the 28-day training period contained 1,344 observations, and the one-day testing period contained 48 observations.

The input structure was defined using a sliding window with the previous seven days of load observations as model input. Since each day contains 48 observations, the input size was set to 336 time steps (48 × 7). The forecasting horizon was set to 48 time steps, corresponding to a one-day-ahead load forecast. This setting allows the NHITS model to learn recent daily and weekly temporal patterns and generate short-term forecasts for the next 24 hours.

After the data split, the datasets were reformatted according to the requirements of the NeuralForecast library. Three columns were used: unique_id, which represents the regional time-series identifier such as "NSW" or "QLD"; ds, which represents the timestamp; and y, which represents the corresponding load value. Finally, the data frequency was specified as 30 minutes to ensure compatibility with the NHITS model and maintain temporal consistency across training, optimization, and testing. The same preprocessing procedure was applied to both datasets to ensure consistency in subsequent experiments.

3.2 NHITS Model Configuration

In the experimental setup, the NHITS model was implemented using the NeuralForecast library. The data frequency was set to freq = 30 minutes to match the half-hourly sampling interval of the NSW and QLD datasets. The forecasting horizon was defined as h = 48 steps, corresponding to a one-day-ahead forecast. The input window was set to input_size = 336 (48 × 7), meaning the model uses the previous 7 days of historical load observations to generate a forecast for the next 24 hours. This setting was selected because electricity demand usually exhibits strong daily and weekly patterns, and a seven-day input window allows the model to capture recent weekly temporal dependencies.

The random seed was fixed at 42 to improve reproducibility and reduce the influence of random initialization on the experimental results. The default NHITS configuration provided by the NeuralForecast library was adopted as the baseline model, and the corresponding parameter settings are summarized in Table 1. This baseline configuration serves as a standard reference for evaluating the effects of hyperparameter optimization.

To ensure a controlled comparison, only three hyperparameters, namely mlp_units, learning_rate, and max_steps, were optimized in this study. The remaining parameters were retained at their default values. This design prevents the search space from becoming excessively large. It allows comparisons among RS, TPE, and GA, focusing on the influence of the selected key hyperparameters within the same model structure.

3.3 Hyperparameter Range

The hyperparameter search space was defined to include three NHITS parameters: mlp_units, learning_rate, and max_steps. These parameters were selected to represent model capacity, optimization behavior, and training budget, respectively. The search ranges were selected to include values both below and above the default NHITS configuration. For mlp_units, the default value is 512; therefore, the search range was set from 32 to 960 with a step size of 32 to evaluate both compact and larger network structures. For learning_rate, the default value is 10−3; therefore, a log-uniform range from 10−4 to 10−2 was used to explore slower and faster learning rates around the default setting. For max_steps, the default value is 1000; therefore, the search range was set from 10 to 2000 to examine both shorter and longer training budgets. The corresponding search ranges are summarized in Table 4.

Table 4 Hyperparameter search space for NHITS optimization

Hyperparameter	Search Range	Type
mlp_units	32 – 960, step 32	Discrete
learning_rate	10^{-4} – 10^{-2}	Continuous, log-uniform
max_steps	10 – 2000	Integer

As shown in Table 4, the search space was defined to provide sufficiently broad but computationally feasible ranges for the selected NHITS hyperparameters. Other NHITS parameters were kept fixed to ensure a consistent model structure and a fair comparison across all experiments. The number of evaluations for each optimization method was set to 20 to ensure that RS, TPE, and GA were compared under the same computational constraint.

3.4 Experimental Environment

All experiments were conducted using Google Colab, a cloud-based computing platform that provides a flexible environment for developing and evaluating deep learning models. The runtime environment was configured with Python 3. The main software libraries included NeuralForecast for implementing the NHITS model, NumPy and Pandas for data preprocessing and manipulation, Matplotlib for visualization and result plotting, and Scikit-learn for calculating the evaluation metrics, including MAPE, MAE, MSE, and RMSE.

For hardware configuration, the experiments were executed on a CPU with high-RAM mode enabled to provide sufficient memory for model training and data processing. Although Google Colab also supports GPU and TPU acceleration, the CPU setting was adopted to maintain consistency across experimental runs and to evaluate the proposed optimization framework under a common computing environment. This setting also reflects a practical scenario in which high-performance hardware may not always be available.

The computational cost of the proposed framework mainly depends on the number of optimization evaluations and the training budget defined by max_steps. Since each candidate hyperparameter configuration requires training and evaluating one NHITS model, increasing the number of trials or expanding the search space would increase the total computational time. Overall, this experimental environment provided a stable and reproducible basis for evaluating NHITS-based short-term electricity load forecasting. The use of a fixed computational setting also supports a fair comparison of the optimization methods. It provides practical insight into the applicability of the proposed framework under limited computational resources.

4. Results and Analysis

This section presents and discusses the forecasting results obtained from the default and optimized NHITS models. The effects of random search, tree-structured Parzen estimator, and genetic algorithm on forecasting accuracy are compared using the NSW and QLD datasets. The results are evaluated using MSE, RMSE, MAE, and MAPE, together with graphical comparisons of actual and predicted load profiles. In addition, a point-wise statistical analysis is conducted to further examine the significance and stability of the forecasting improvements.

4.1 Results

Table 5 presents the default configuration together with the best hyperparameter sets obtained by RS, TPE, and GA for the NSW and QLD datasets. In addition to the optimized values of the selected hyperparameters, the table also reports the MAPE values achieved by each configuration. This comparison provides a clear overview of how the optimization methods modify the default NHITS settings and how these changes affect forecasting accuracy across the two datasets.

Table 5 NHITS hyperparameters and MAPE on NSW and QLD

Region	Model	mlp_units	Learning_rate	max_steps	MAPE (%)
NSW	Default	512	0.0010	1000	2.21
	RS-best	160	0.0005	1519	1.77
	TPE-best	960	0.0001	1700	1.23
	GA-best	672	0.0008	476	1.49
QLD	Default	512	0.0010	1000	3.34
	RS-best	416	0.0025	1719	2.73
	TPE-best	320	0.0009	63	1.79
	GA-best	448	0.0001	201	2.87

The results in Table 5 show that hyperparameter optimization substantially improves the forecasting performance of NHITS for both NSW and QLD. For NSW, MAPE decreases from 2.21% under the default setting to 1.77% with RS, 1.49% with GA, and 1.23% with TPE. This result indicates that all three optimization methods are effective in enhancing forecasting accuracy, with TPE providing the largest improvement. Similarly, for QLD, MAPE is reduced from 3.34% to 2.73% with RS, 2.87% with GA, and 1.79% with TPE. Among the three optimization methods, TPE consistently achieves the best performance, showing its superior ability to identify effective hyperparameter configurations under the fixed evaluation budget. Although RS and GA also improve the default model, their performance is less consistent across the two datasets.

Table 6 presents the forecasting performance of the four NHITS model configurations, namely Default, RS, TPE, and GA, for the NSW and QLD datasets using MSE, RMSE, MAE, and MAPE. These results provide a quantitative comparison of the effects of hyperparameter optimization on forecasting performance relative to the default NHITS configuration.

Table 6 Forecasting performance for NSW and QLD

Region	Model	MSE	RMSE	MAE	MAPE (%)
NSW	Default	35,616	189	150	2.21
	RS	24,926	158	122	1.77
	TPE	15,130	123	88	1.23
	GA	17,393	132	104	1.49
QLD	Default	71,839	268	198	3.34
	RS	42,384	206	162	2.73
	TPE	18,900	137	105	1.79
	GA	42,322	206	175	2.87

For the NSW dataset, TPE achieves the best overall performance among the four model configurations, with MSE = 15,130, RMSE = 123 MW, MAE = 88 MW, and MAPE = 1.23%. Compared with the Default model, TPE substantially reduces forecasting errors, particularly in MAE and MAPE, confirming the effectiveness of hyperparameter optimization in improving NHITS forecasting accuracy. GA ranks second with a MAPE of 1.49%, while RS also improves the default configuration but remains less effective than TPE and GA.

A similar pattern is observed for the QLD dataset. TPE again delivers the best performance, achieving the lowest MAPE value of 1.79%, compared with 3.34% for the Default model. This result demonstrates a clear improvement in forecasting accuracy and confirms TPE’s ability to identify suitable hyperparameter settings across different regional load patterns. Although RS and GA also outperform the Default model, their improvements are smaller, with MAPE values of 2.73% and 2.87%, respectively.

Fig. 2 compares the MAPE values of the Default, RS, TPE, and GA configurations for the NSW and QLD datasets. The figure provides a visual comparison of the forecasting accuracy across the different hyperparameter optimization methods and the default NHITS configuration.

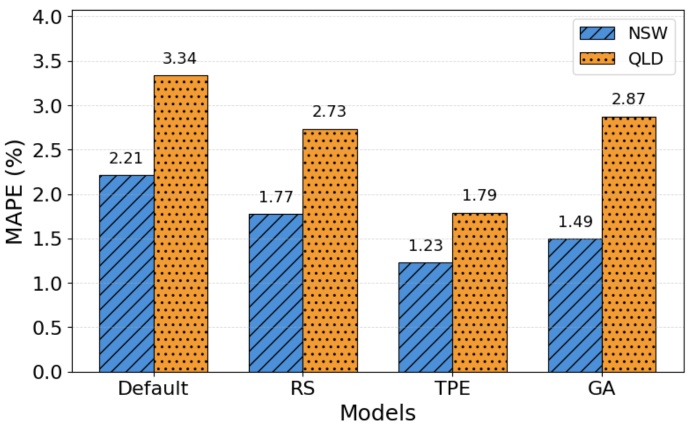

Fig. 2 MAPE comparison (NSW vs QLD)

As shown in Fig. 2, TPE achieves the lowest MAPE values for both NSW and QLD, while the Default NHITS model shows the highest errors. RS and GA also improve forecasting accuracy compared with the Default configuration, but their improvements are less pronounced than those obtained by TPE.

The consistent superiority of TPE across both datasets suggests that Bayesian optimization provides a more efficient search mechanism for the NHITS hyperparameter space. By leveraging information from previous evaluations, TPE is better able to focus the search on promising regions, thereby increasing the likelihood of finding configurations with lower forecasting error. In comparison, RS relies entirely on random exploration, while GA depends on evolutionary operations that may require a larger search budget to achieve more competitive results. This difference explains why TPE provides the strongest and most stable performance in both regional forecasting tasks.

Because all optimization methods were evaluated under the same budget of 20 model evaluations, the differences in forecasting performance mainly reflect the effectiveness of their search strategies. Under this fixed-budget setting, TPE consistently provides the largest reduction in forecasting error for both NSW and QLD, indicating the best trade-off between forecasting accuracy and computational effort.

Figs. 3 and 4 compare the actual and predicted load profiles over the last seven training days and the test day for the NSW and QLD datasets, respectively. These figures provide a broader temporal view of how the four NHITS model configurations, namely Default, RS, TPE, and GA, follow the overall load pattern before and during the forecasting period.

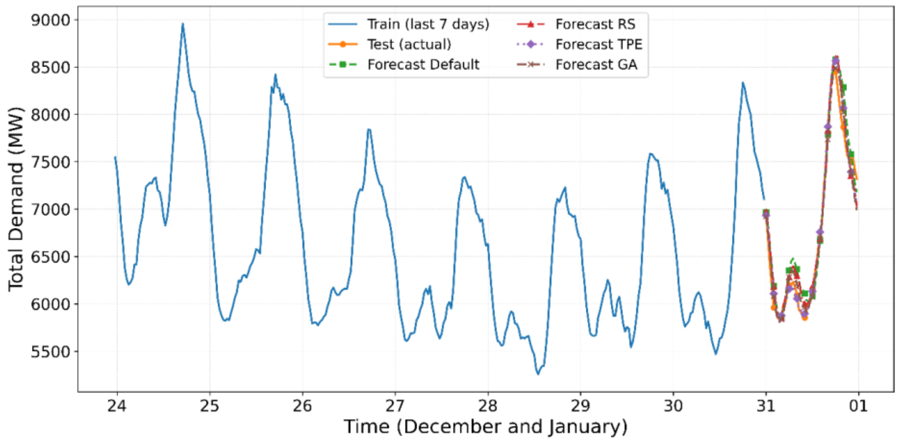

Fig. 3 Actual and predicted load profiles for NSW

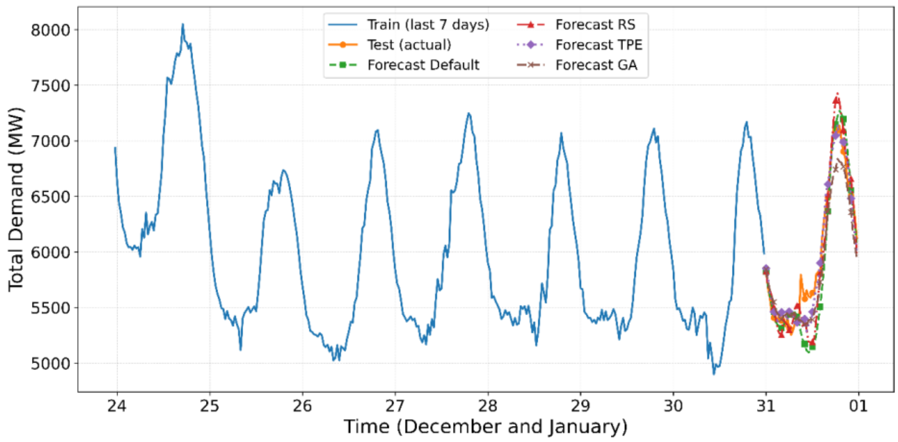

Fig. 4 Actual and predicted load profiles for QLD

As shown in Figs. 3 and 4, all model configurations capture the dominant daily load cycle for both datasets. However, differences in forecasting quality are evident. In both NSW and QLD, the TPE-based model follows the actual load trajectory more closely, particularly during periods of increasing demand and around peak-load regions. The Default configuration exhibits larger deviations, whereas RS and GA provide noticeable improvements but remain less consistent than TPE. Overall, the visual comparisons indicate that hyperparameter optimization enhances the ability of NHITS to track regional load dynamics, with TPE providing the closest agreement with the observed load profiles.

Following the broader comparison presented in Figs. 3 and 4, Figs. 5 and 6 provide a more detailed view of the forecasting results during the 24-hour test period for the NSW and QLD datasets, respectively. These figures highlight how the four NHITS model configurations, namely Default, RS, TPE, and GA, to capture short-term load variations within a single day.

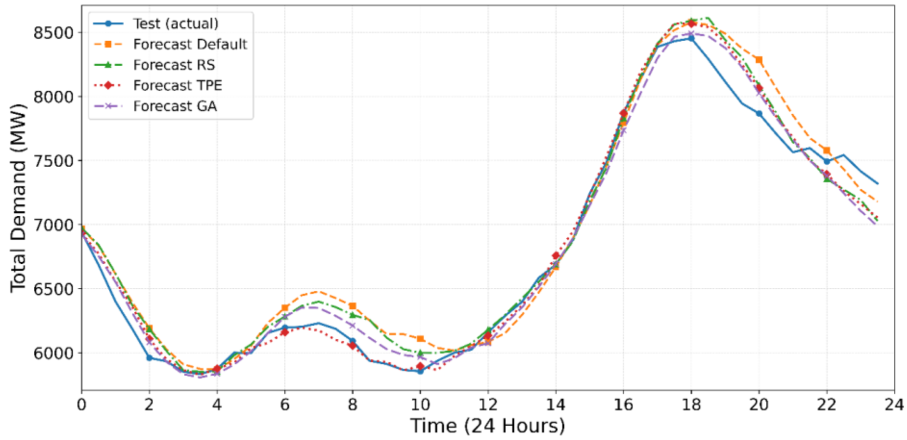

Fig. 5 24-hour load forecasting results for NSW

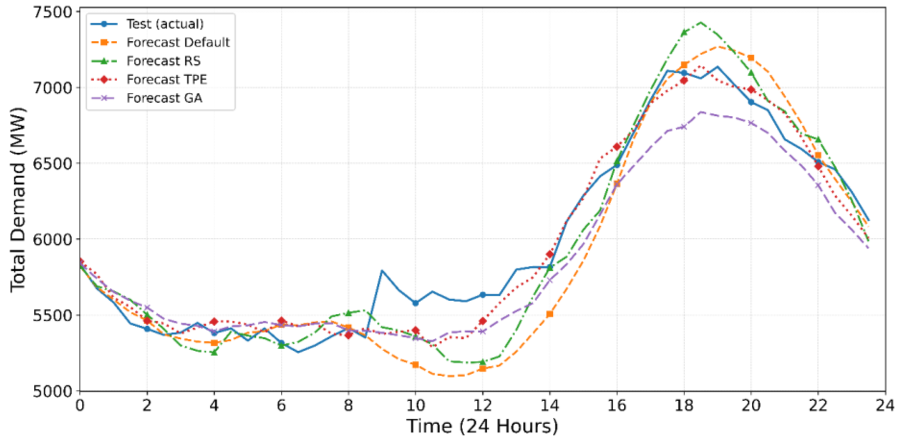

Fig. 6 24-hour load forecasting results for QLD

As illustrated in Figs. 5 and 6, the TPE-based model provides the closest agreement with the actual load profiles during the 24-hour test period for both datasets. In particular, TPE captures the timing and magnitude of demand variations more accurately, especially during periods of rapidly increasing or decreasing load and around peak-demand regions. The Default configuration exhibits larger deviations, whereas RS and GA provide intermediate improvements. These observations are consistent with the quantitative results presented in Table 6 and further demonstrate the effectiveness of TPE for NHITS-based load forecasting.

4.2 Statistical Analysis of Point-Wise Forecasting Errors

To further examine the forecasting behavior of the four NHITS configurations during the 24-hour test period, a point-wise error analysis was conducted for both the NSW and QLD datasets. The absolute percentage error (APE) was calculated at each of the 48 forecasting steps. In addition to the conventional evaluation metrics, the standard deviation, median, and maximum APE values were used to provide a more detailed assessment of forecasting stability. The descriptive statistics of the point-wise forecasting errors are presented in Table 7.

Table 7 Descriptive statistics of point-wise forecasting errors for the 24-hour test period

Region	Model	MSE	RMSE	MAE	MAPE (%)	APE Std	Median APE	Max APE
NSW	Default	35,616	188.7	149.6	2.21	1.67	1.66	5.39
NSW	RS	24,925	157.9	121.6	1.77	1.44	1.39	5.35
NSW	TPE	15,130	123.0	87.8	1.23	1.13	0.92	3.81
NSW	GA	17,392	131.9	103.8	1.49	1.11	1.37	4.54
QLD	Default	71,838	268.0	197.5	3.34	3.18	2.09	9.57
QLD	RS	42,383	205.8	162.5	2.73	2.19	2.33	7.88
QLD	TPE	18,899	137.5	104.8	1.79	1.58	1.34	7.24
QLD	GA	42,322	205.7	175.4	2.87	1.71	2.87	7.08

As shown in Table 7, the TPE-based configuration shows more stable forecasting behavior for both datasets. For NSW, TPE achieves the lowest median APE (0.92%) and maximum APE (3.81%), indicating smaller deviations across the 48 prediction steps. For QLD, TPE provides the lowest APE standard deviation (1.58%) and median APE (1.34%), suggesting a more consistent error distribution. Although GA has a slightly lower maximum APE than TPE in QLD, its higher median APE indicates less stable overall performance. These results show that TPE improves not only average accuracy but also prediction stability during the 24-hour forecasting period.

To quantify the relative improvement of the TPE-based NHITS configuration, the percentage reductions in MAPE, MAE, and RMSE were calculated with respect to the Default, RS, and GA configurations. Table 8 summarizes these reduction rates for both the NSW and QLD datasets under the same testing conditions. In addition, the table reports the Wilcoxon test conclusion for each comparison. This presentation provides a concise basis for assessing the level of improvement in TPE before discussing the detailed numerical results.

Table 8 Error reduction achieved by TPE compared with the other NHITS configurations.

Region	Reference model	MAPE reduction (%)	MAE reduction (%)	RMSE reduction (%)	Wilcoxon result
NSW	Default	44.55	41.33	34.82	Significant at 5%
NSW	RS	30.81	27.82	22.09	Significant at 5%
NSW	GA	17.83	15.41	6.73	Not significant at 5%
QLD	Default	46.36	46.92	48.71	Significant at 5%
QLD	RS	34.45	35.46	33.22	Significant at 5%
QLD	GA	37.71	40.24	33.17	Significant at 5%

Table 8 shows that TPE substantially reduced forecasting errors relative to the other configurations. Compared with the Default configuration, TPE reduced MAPE by 44.55% for NSW and 46.36% for QLD. The corresponding MAPE reductions relative to RS were 30.81% and 34.45%, respectively. TPE also achieved lower MAPE values than GA, with reductions of 17.83% for NSW and 37.71% for QLD. The Wilcoxon result further indicates that these improvements are statistically significant at the 5% level for all comparisons except the comparison between TPE and GA in NSW. Therefore, TPE provides the most consistent overall error reduction, although the comparison with GA in NSW should be interpreted cautiously.

Finally, to examine whether the observed improvements were statistically significant, the Wilcoxon signed-rank test was applied to the paired APE values obtained at the same forecasting time points. For each dataset, the TPE-based configuration was compared with the Default, RS, and GA configurations. The results are reported in Table 9.

Table 9 Wilcoxon signed-rank test results using point-wise APE values

Region	Comparison	Mean diff	Median diff	TPE lower points	W	p-value	Conclusion
NSW	TPE vs Default	-0.986	-0.792	33	234.000	0.000169	Significant at 5%
NSW	TPE vs RS	-0.547	-0.262	31	311.000	0.003915	Significant at 5%
NSW	TPE vs GA	-0.266	-0.199	28	436.000	0.120914	Not significant at 5%
QLD	TPE vs Default	-1.547	-0.605	32	277.000	0.001095	Significant at 5%
QLD	TPE vs RS	-0.941	-0.730	31	311.000	0.003915	Significant at 5%
QLD	TPE vs GA	-1.084	-0.785	32	219.000	8.15e-05	Significant at 5%

The Wilcoxon signed-rank test results in Table 9 show that TPE significantly outperformed the Default and RS configurations for both NSW and QLD at the 5% significance level. For QLD, the improvement of TPE over GA was also statistically significant. For NSW, although TPE achieved lower average error values than GA, the difference was not statistically significant. Therefore, the statistical results indicate that TPE provides a consistent and clear improvement over the Default and RS configurations in both regions, whereas its advantage over GA is less pronounced for NSW.

5. Conclusion

This study developed and evaluated a controlled hyperparameter optimization framework to improve the short-term accuracy of the NHITS model for electricity load forecasting. Three representative optimization strategies, namely RS, TPE, and GA, were compared under the same search space, random seed, and fixed evaluation budget. The optimization focused on three influential NHITS hyperparameters, including mlp_units, learning_rate, and max_steps, which directly affect model capacity, convergence behavior, and training effort. The proposed framework was tested on two Australian regional electricity demand datasets, NSW and QLD, using MSE, RMSE, MAE, and MAPE as evaluation metrics. The main conclusions are summarized as follows:

(1) Hyperparameter optimization improved the forecasting performance of the default NHITS configuration on both datasets. This confirms that the default NHITS settings are not necessarily optimal for regional short-term electricity load forecasting.

(2) All three optimization methods reduced forecasting errors compared with the default model. The optimized NHITS models also provided better agreement between predicted and actual load profiles during both the multi-day visualization and the 24-hour test horizon.

(3) Among the evaluated methods, TPE achieved the best and most consistent forecasting performance. It produced the lowest errors in both NSW and QLD, with MAPE reduced from 2.21% to 1.23% in NSW and from 3.34% to 1.79% in QLD.

(4) The relative performance of RS and GA varied across datasets. GA performed better than RS on NSW, whereas RS slightly outperformed GA on QLD, indicating that optimization performance can depend on regional load characteristics and the available search budget.

(5) The point-wise statistical analysis further confirmed that TPE significantly outperformed the Default and RS configurations in both datasets. However, although TPE achieved lower average errors than GA, its improvement over GA was statistically significant only for QLD and not for NSW.

Overall, the results demonstrate that appropriate hyperparameter selection is essential for improving NHITS-based short-term load forecasting. Future work should extend the framework by optimizing additional NHITS parameters, using more datasets, increasing the evaluation budget, and conducting multi-seed robustness analysis.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Y. Eren and İ. Küçükdemiral, “A Comprehensive Review on Deep Learning Approaches for Short-Term Load Forecasting,” Renewable and Sustainable Energy Reviews, vol. 189, part B, article no. 114031, 2024.

[2] N. T. N. Anh, N. N. Anh, T. N. Thang, V. K. Solanki, R. G. Crespo, and N. Q. Dat, “Online SARIMA Applied for Short-Term Electricity Load Forecasting,” Applied Intelligence, vol. 54, no. 1, pp. 1003-1019, 2024.

[3] H. M. Al-Hamadi, “Long-Term Electric Power Load Forecasting Using Fuzzy Linear Regression Technique,” Proceedings of the 2011 IEEE Power Engineering and Automation Conference, pp. 96-99, 2011.

[4] T. T. Ngoc, L. Van Dai, and C. M. Thuyen, “Support Vector Regression Based on Grid Search Method of Hyperparameters for Load Forecasting,” Acta Polytechnica Hungarica, vol. 18, no. 2, pp. 143-158, 2021.

[5] P. F. Pai and W. C. Hong, “Forecasting Regional Electricity Load Based on Recurrent Support Vector Machines with Genetic Algorithms,” Electric Power Systems Research, vol. 74, no. 3, pp. 417-425, 2005.

[6] S. Deng, X. Dong, L. Tao, J. Wang, Y. He, and D. Yue, “Multi-Type Load Forecasting Model Based on Random Forest and Density Clustering with the Influence of Noise and Load Patterns,” Energy, vol. 307, article no. 132635, 2024.

[7] S. Wang and J. Ma, “A Novel Ensemble Model for Load Forecasting: Integrating Random Forest, XGBoost, and Seasonal Naive Methods,” Proceedings of the 2023 2nd Asian Conference on Frontiers of Power and Energy (ACFPE), pp. 114-118, 2023.

[8] L. Zhang and D. Jánošík, “Enhanced Short-Term Load Forecasting with Hybrid Machine Learning Models: CatBoost and XGBoost Approaches,” Expert Systems with Applications, vol. 241, article no. 122686, 2024.

[9] A. O. Aseeri, “Effective RNN-Based Forecasting Methodology Design for Improving Short-Term Power Load Forecasts: Application to Large-Scale Power-Grid Time Series,” Journal of Computational Science, vol. 68, article no. 101984, 2023.

[10] J. Wen, Y. Peng, W. Zhang, X. Huang, and Z. Wang, “Short-Term Power Load Forecasting Based on TCN-LSTM Model,” Proceedings of the 2024 IEEE 6th International Conference on Civil Aviation Safety and Information Technology (ICCASIT), pp. 734-738, 2024.

[11] A. Kathirgamanathan, A. Patel, A. S. Khwaja, B. Venkatesh, and A. Anpalagan, “Performance Comparison of Single and Ensemble CNN, LSTM and Traditional ANN Models for Short-Term Electricity Load Forecasting,” The Journal of Engineering, vol. 2022, no. 5, pp. 550-565, 2022.

[12] H. Eskandari, M. Imani, and M. Parsa Moghaddam, “Best-Tree Wavelet Packet Transform Bidirectional GRU for Short-Term Load Forecasting,” The Journal of Supercomputing, vol. 79, no. 12, pp. 13545-13577, 2023.

[13] J. Ran, Y. Gong, Y. Hu, and J. L. Cai, “EV Load Forecasting Using a Refined CNN-LSTM-AM,” Electric Power Systems Research, vol. 238, article no. 111091, 2025.

[14] Q. Hua et al., “A Short-Term Power Load Forecasting Method Using CNN-GRU with an Attention Mechanism,” Energies, vol. 18, no. 1, article no. 106, 2025.

[15] D. Tan, Z. Tang, F. Zhou, and Y. Xie, “A Novel Hybrid Model Based on EMD-Improved TCN-Improved TST for Short-Term Railway Traction Load Forecasting,” IEEE Transactions on Transportation Electrification, vol. 11, no. 2, pp. 6418-6427, 2024.

[16] X. Guo, Q. Zhao, D. Zheng, Y. Ning, and Y. Gao, “A Short-Term Load Forecasting Model of Multi-Scale CNN-LSTM Hybrid Neural Network Considering the Real-Time Electricity Price,” Energy Reports, vol. 6, pp. 1046-1053, 2020.

[17] J. Zhang, Z. Zhu, and Y. Yang, “Electricity Load Forecasting Based on CNN-LSTM,” Proceedings of the 2023 IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE), pp. 1385-1390, 2023.

[18] Z. Tian, W. Liu, W. Jiang, and C. Wu, “CNNs-Transformer Based Day-Ahead Probabilistic Load Forecasting for Weekends with Limited Data Availability,” Energy, vol. 293, article no. 130666, 2024.

[19] J. W. Chan and C. K. Yeo, “A Transformer Based Approach to Electricity Load Forecasting,” The Electricity Journal, vol. 37, no. 2, article no. 107370, 2024.

[20] S. F. Stefenon, L. O. Seman, C. K. Yamaguchi, L. D. S. Coelho, V. C. Mariani, J. P. Matos-Carvalho, et al., “Neural Hierarchical Interpolation Time Series (NHITS) for Reservoir Level Multi-Horizon Forecasting in Hydroelectric Power Plants,” IEEE Access, vol. 13, pp. 54853-54865, 2025.

[21] W. Zhou, H. Qi, D. Boland, and P. H. W. Leong, “FPGA Implementation of N-BEATS for Time Series Forecasting Using Block Minifloat Arithmetic,” Proceedings of the 2022 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), pp. 546-550, 2022.

[22] M. Dai, M. Y. Seow, R. Shirota Filho, A. Sclip, K. Chawla, R. S. M. Goh, et al., “N-BEATS-GAN: A Risk-Aware Financial Time Series Forecasting with Generative Adversarial Networks,” Applied Soft Computing, vol. 186, part D, article no. 114235, 2026.

[23] H. Hou, C. Liu, Q. Wang, X. Wu, J. Tang, Y. Shi, et al., “Review of Load Forecasting Based on Artificial Intelligence Methodologies, Models, and Challenges,” Electric Power Systems Research, vol. 210, article no. 108067, 2022.

[24] O. Rubasinghe, X. Zhang, T. K. Chau, T. Fernando, and H. H. C. Iu, “A Novel Sequence to Sequence based CNN-LSTM Model for Long Term Load Forecasting,” Proceedings of the 2022 IEEE Sustainable Power and Energy Conference (iSPEC), pp. 1-5, 2022.

[25] H. Hu and B. Zheng, “Short-Term Electricity Load Forecasting Based on CEEMDAN-FE-BiGRU-Attention Model,” International Journal of Low-Carbon Technologies, vol. 19, pp. 988-995, 2024.

[26] Ş. Özdemir, Y. Demir, and Ö. Yildirim, “The Effect of Input Length on Prediction Accuracy in Short-Term Multi-Step Electricity Load Forecasting: A CNN-LSTM Approach,” IEEE Access, vol. 13, pp. 28419-28432, 2025.