

# **Effects of Data Standardization on Hyperparameter Optimization with the Grid Search Algorithm Based on Deep Learning: A Case Study of Electric Load Forecasting**

Tran Thanh Ngoc, Le Van Dai\*, Lam Binh Minh

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

Received 04 January 2022; received in revised form 24 March 2022; accepted 25 March 2022

DOI: <https://doi.org/10.46604/aiti.2022.9227>

## **Abstract**

This study investigates data standardization methods based on the grid search (GS) algorithm for energy load forecasting, including zero-mean, min-max, max, decimal, sigmoid, softmax, median, and robust, to determine the hyperparameters of deep learning (DL) models. The considered DL models are the convolutional neural network (CNN) and long short-term memory network (LSTMN). The procedure is made over (i) setting the configuration for CNN and LSTMN, (ii) establishing the hyperparameter values of CNN and LSTMN models based on epoch, batch, optimizer, dropout, filters, and kernel, (iii) using eight data standardization methods to standardize the input data, and (iv) using the GS algorithm to search the optimal hyperparameters based on the mean absolute error (MAE) and mean absolute percent error (MAPE) indexes. The effectiveness of the proposed method is verified on the power load data of the Australian state of Queensland and Vietnamese Ho Chi Minh city. The simulation results show that the proposed data standardization methods are appropriate, except for the zero-mean and min-max methods.

**Keywords:** deep learning, grid search, data standardization method, hyperparameter, electric load forecasting

## **1. Introduction**

According to the United States Energy Information Administration (US-EIA), worldwide energy demand is expected to rise by 50%, with rising countries in Asia leading the way. This rising demand would place considerable strain on the current energy infrastructure and jeopardize global environmental health by increasing greenhouse gas emissions from conventional power sources [1]. In the United States and Europe, an estimated 40% of electricity consumption and 38% of CO<sub>2</sub> emissions come from the construction industry [2]. Currently, the construction industry tends to use sustainable energy sources to replace limited energy sources. As a result, the use of renewable energy sources has been increasing, the design of buildings must be improved, and the building energy demand needs to be forecasted. Therefore, it is necessary to apply the energy load forecasting method because it has both economic and infrastructure advantages. This method can predict future electricity consumption and help power companies make economically viable plans and decisions [3].

Short-term load forecasting plays an important role in the power industry, including power system planning, power generation planning, and the power supply-demand balance [4-5]. If load forecasting is accurate, significant cost reductions in control operations and decision-making, such as dispatch, unit commitment, fuel allocation, power system security assessment, and off-line analysis, would be realized. On the contrary, if there is an error in the forecast of electricity demand, there will be an increase in operating costs.

---

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

In the past decade, many methodologies and techniques have been proposed to solve the problem of short-term power load forecasting. They can be classified into two groups of methods. The first group relates to using statistical methods, such as multiple regression, exponential smoothing, and autoregressive integrated moving average (ARIMA) [6-7]. The second group employs artificial intelligence techniques, such as support vector machines (SVM) and artificial neural networks (ANNs) [8-9]. Recent development based on ANNs and deep learning (DL) networks is one of the methods applied to solve the problem of load forecasting. The DL architecture includes different models, such as long short-term memory networks (LSTMN), convolutional neural networks (CNN), deep belief networks, and deep Boltzmann machine networks. Among them, LSTMN and CNN are popular in the problem of power load forecasting [10-11]. The main feature of the DL model is that the accuracy of the load-forecasted results highly depends on its hyperparameters. Therefore, determining these hyperparameters for DL models is important [12-13].

Recently, some algorithms, such as grid search (GS), random search (RS), and genetic algorithm (GA), have been applied to determine the hyperparameters of the DL model, among which the GS algorithm was widely applied [14-15]. In addition, the characteristics of the input data are also important factors affecting the accuracy of the DL model. Moolayil [16] introduced some data standardization methods to solve this problem. However, Raschka et al. [17] and Yang et al. [18] did not show any interest in employing data normalization on the GS algorithm. As a result, this may be a leading disadvantage for these studies. To overcome this disadvantage, this study proposes an input data standardization method on the GS algorithm to determine the hyperparameters of the DL model, including CNN and LSTMN, for energy load forecasting.

For this proposed method, the model data are split into training and testing sets. For the training step, the GS algorithm is performed to determine the hyperparameters of the DL model corresponding to each data normalization method. For the testing step, the predicted errors of these optimal models are compared, and thereby the proposed methodology can evaluate the impact of the data normalization methods on the GS algorithm to the DL model. The error value in the DL model is usually determined based on the error evaluation indexes of the actual value and predicted value of the model, such as the mean absolute error (MAE) and the mean absolute percent error (MAPE). The forecasting results of the DL model are significantly affected by the scale and size of the data. Therefore, it is necessary to standardize the data during training and forecasting for the DL model. In this study, the methods, such as zero-mean, min-max, max, decimal, sigmoid, softmax, median, and robust, are proposed to standardize the input data of the DL model, and the hyperparameter values of CNN and LSTMN models are established based on epoch, batch, optimizer, dropout, filters, and kernel.

The novelty and contributions of this study include the following aspects: (i) introduce a data standardization method on the GS algorithm to determine the optimal hyperparameters of the DL model, including CNN and LSTMN for the energy load forecasting; (ii) consider the error evaluation indexes of MAE and MAPE of actual and predicted values for determining the optimal hyperparameters of the DL model through the epoch, batch, optimizer, dropout, filters, and kernel; (iii) conclude that the zero-mean and min-max are two of the data standardization methods and not the best methods on the GS algorithm for determining the optimal hyperparameters of the DL model.

This study consists of five sections. Section 1 presents the urgency, settlement, and unresolved issues of the load forecasting problem. Section 2 describes the principle, hyperparameters, GS algorithm, and data normalization method for the DL model. The experimental procedures and settings are presented in section 3. Section 4 presents the results and discussion. Finally, the conclusions and future research aspects are presented in section 5.

## 2. Methodology

### 2.1. Deep learning structures

Artificial intelligence is the ability of a machine to imitate intelligent human behavior. Machine learning (ML) is part of artificial intelligence that allows a system to learn and automatically improve from experience. DL is an application of ML that

uses complex algorithms and deep neural nets to train a model. DL models are built using several algorithms, such as CNNs, LSTMNs, recurrent neural networks (RNNs), generative adversarial networks (GANs), radial basis function networks (RBFNs), and so on [19]. In this study, two widespread DL networks, LSTMN and CNN, are used to resolve the problem. The procedure is performed as follows.

### 2.1.1. LSTMN network

The difference between the RNN and feed-forward neural network (FFNN) is that the RNN is a model that can create a correlation between the previous information and the current state. A simple RNN structure is shown in Fig. 1, in which the output signal is determined based on a linear transformation and nonlinear activation. The output signal can be calculated under the tangent function as follows [20]:

$$h_t = \tanh(w(h_{t-1}, x_t) + b) \quad (1)$$

where  $h_{t-1}$  denotes the  $(t-1)$ th output signal,  $x_t$  denotes the  $t$  th input signal, and  $b$  denotes the bias.

The LSTMN is a modified RNN model developed by Song et al. [21]. The difference between the LSTMN and the RNN is that the LSTMN can process long-term dependencies. Fig. 2 describes the LSTMN structure. Each block has two parallel lines going in and out, representing the cell state and hidden state information. The general structure of LSTMN has four layers of neural networks composed of three inputs (i.e.,  $C_{t-1}$ ,  $h_{t-1}$ , and  $x_t$ ) and two outputs (i.e.,  $C_t$  and  $h_t$ ). Therefore, this LSTMN structure can be described by the following equations [21]:

The authors identify the information from the previous cell state  $C_{t-1}$  that should be removed by the following forget gate  $f_t$ .

$$f_t = \sigma(w_f \times (h_{t-1}, x_t) + b_f) \quad (2)$$

The authors identify the input signal  $x_t$  that should be stored in the cell state  $C_t$  in the input gate, in which the input information and the candidacy cell state  $\tilde{C}_t$  should be updated by:

$$i_t = \sigma(w_i \times (h_{t-1}, x_t) + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(w_c \times (h_{t-1}, x_t) + b_c) \quad (4)$$

The previous cell state  $C_t$  is updated by combining  $C_{t-1}$  and  $\tilde{C}_t$ :

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (5)$$

The outcome  $h_t$  in the output gate is confirmed based on the output information  $O_t$  and  $C_t$ :

$$O_t = \sigma(w_o \times (h_{t-1}, x_t) + b_o) \quad (6)$$

$$h_t = O_t \times \tanh(C_t) \quad (7)$$

in which  $w$  is the input weight and  $f$ ,  $i$ , and  $O$  represent the forget, input, and output gates, respectively.

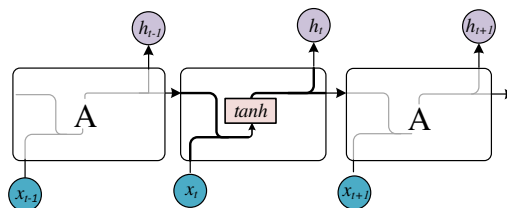


Fig. 1 Simple RNN architecture

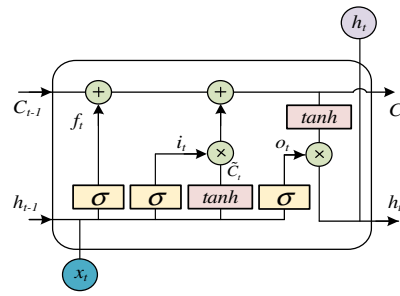


Fig. 2 LSTMN architecture

2.1.2. CNN network

CNN is a feedforward neural network with a structure similar to that of human neurons. Fig. 3 depicts the CNN structure developed based on the convolutional CNN structure introduced in the work of Bon et al. [22], which includes convolution, pooling, and fully linked layers. The input data are convolved using many filters for the convolution layer, and a feature map is formed when a bias term is added. Then, a nonlinear function is applied. The pooling layer’s primary goal is to lower the resolution of the feature maps to aggregate the input data. There are several sorts of pooling procedures, the most prevalent of which is the max-pooling strategy. Finally, fully connected layers process the convolutional layers’ outputs [21].

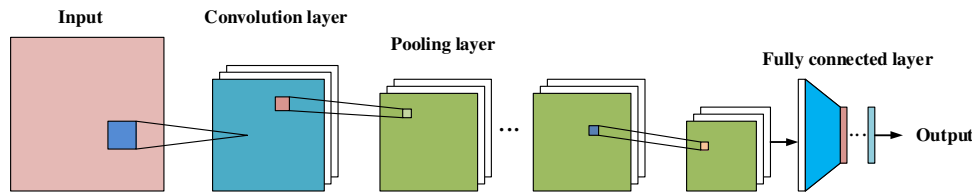


Fig. 3 The architecture of the CNN model

2.2. Hyperparameters

In general, the accuracy of a DL model depends on its hyperparameters. Therefore, determining the hyperparameters for DL plays an extremely important role. A hyperparameter is a configuration that is external to the model, whose value cannot be estimated from data and all values are set before the started network training. In this study, the hyperparameter values of LSTM and CNN are established, as listed in Table 1. *Epoch* refers to the number of times to expose the model to the whole training dataset; *Batch* refers to the number of samples within an epoch after which the weights are updated; *Dropout* refers to the process of randomly omitting a fraction of the hidden neurons.

For each training case, each hidden neuron is randomly omitted from the network with a fixed probability  $p$ , where  $p$  can be chosen in the range  $[0,1]$ . *Optimizer* refers to the optimization algorithm which plays an important role in improving the accuracy of the DL network. The optimizer is a mathematical algorithm that uses derivatives, partial derivatives, and the chain rule in calculus to understand how much change the network will see in the loss function by making a small change in the weight of the neurons. *Filter* is one of the most important CNN hyperparameters, which is the number of filters that will be learned by the convolutional layer. In a CNN, a convolution filter iterates over all of the input components, executing convolution operations to extract input characteristics; 32, 64, 128, and so on are the most frequent number of filters. The convolution window width and height are determined by kernel size. The kernel size might be an odd integer, such as (3, 3), (5, 5), (7, 7), and so on.

There have been many proposed methods to determine the DL hyperparameters in recent years, such as the GS, RS, gradient-based optimization, Bayesian optimization (BO), GA, and particle swarm optimization (PSO). Of these methods, GS is widely employed due to its simplicity and efficiency. Therefore, GS is the choice to determine the DL hyperparameters in this study.

Table 1 Graph representations

LSTMN model	CNN model
Epoch	Epoch
Batch	Batch
Optimizer	Optimizer
Dropout	Filter
-	Kernel

### 2.3. Grid search method

The GS is a comprehensive search process through the predefined subclass of the value's combinatory of the model's hyperparameters. The operation principle of GS is illustrated in Fig. 4. It is composed of two hyperparameters,  $X$  and  $Y$  [23-24]. The  $X$  is established by three values  $\{x_1, x_2, x_3\}$ , and the  $Y$  is established by three values  $\{y_1, y_2, y_3\}$ . As a result, their combination is nine value pairs. The GS will perform a search for the optimal model based on these values, and the optimal hyperparameter corresponds to the DL model with the smallest error.

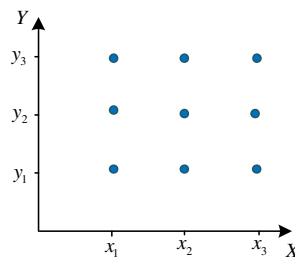


Fig. 4 The operation principle of the GS method

The error value in the DL model is usually determined based on the error evaluation indexes of the actual and predicted values of the model, such as mean square error (MSE), MAE, and MAPE. These evaluation indexes can be described as follows [25]:

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

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \times 100 \quad (10)$$

where  $y_i$  is the actual value  $i^{\text{th}}$ , and  $\hat{y}_i$  is the predicted value  $i^{\text{th}}$ .

### 2.4. Data normalization

Many studies have shown that the forecasting results of the DL model are significantly affected by the scale and size of the data [26]. Thus, it is necessary to standardize the data during training and forecasting for the DL model. In this study, the methods zero-mean, min-max, max, decimal, sigmoid, softmax, median, and robust are proposed to standardize the input data of the DL model. The mathematical models of these methods can be described as follows [16]:

$$\text{Zero-mean normalization: } x' = \frac{x - x_{\text{mean}}}{x_{\text{std}}} \quad (11)$$

$$\text{Min-max normalization: } x' = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \quad (12)$$

$$\text{Max normalization: } x' = \frac{x}{x_{\max}} \quad (13)$$

$$\text{Decimal normalization: } x' = \frac{x}{10^j} \quad (14)$$

$$\text{Sigmoid normalization: } x' = \frac{1}{1 + e^{-a}}, \forall a = \frac{x - x_{\min}}{x_{\text{std}}} \quad (15)$$

$$\text{Softmax normalization: } x' = \frac{1 - e^{-a}}{1 + e^{-a}}, \forall a = \frac{x - x_{\min}}{x_{\text{std}}} \quad (16)$$

$$\text{Median normalization: } x' = \frac{x}{x_{\text{med}}} \quad (17)$$

$$\text{Robust normalization: } x' = \frac{x - x_{\text{med}}}{IQR}, \forall IQR = x_{75} - x_{25} \quad (18)$$

where  $x$  and  $x'$  are the original and standardized data value, respectively;  $x_{\text{mean}}$ ,  $x_{\text{std}}$ ,  $x_{\text{min}}$ ,  $x_{\text{max}}$ , and  $x_{\text{med}}$  are the mean, standard deviation, min, max, and median values of  $x$ , respectively;  $x_{25}$  and  $x_{75}$  are the 25<sup>th</sup> quantile and the 75<sup>th</sup> quantile values of  $x$ , respectively;  $j$  is the smallest integer that satisfies the condition of  $\max|x'| \leq 1$ .

### 2.5. Grid search method based on data normalization

Based on the theoretical base of the DL structure, hyperparameter, and the data standardization method, as presented in sections 2.1, 2.2, and 2.4, the proposed GS algorithm applied to standardize the data for the DL model is shown in Fig. 5. The procedure is done in the six steps below. The procedure is applied to each data standardization method, as introduced in section 2.4, to determine the error value. This error value is then compared to evaluate the effect of these methods on the GS algorithm for the DL network.

- Step 1:** The original data is processed and the input-target pairs of  $(X_{\text{train}}, Y_{\text{train}})$  and  $(X_{\text{test}}, Y_{\text{test}})$  are determined corresponding to the training and test processes, respectively.
- Step 2:** The training and test data are standardized by using the methods described in section 2.4 to determine  $(X'_{\text{train}}, Y'_{\text{train}})$  and  $(X'_{\text{test}}, Y'_{\text{test}})$ . The next step is the training process.
- Step 3:** The GS algorithm is applied to determine the optimal hyperparameters of the DL model from the variable value combination of each hyperparameter  $CFG = \{cfg_i\}$ ,  $i = 1: N$ , in which  $N$  is the total number of combinations. The vector  $cfg_i$  depends on the DL model. For the LSTMN model,  $cfg_i = \{E_i, B_i, O_i, D_i\}$  is used, whereas  $cfg_i = \{E_i, B_i, O_i, F_i, K_i\}$  is used for the CNN model, in which  $E$ ,  $B$ ,  $O$ ,  $D$ ,  $F$ , and  $K$  notate the hyperparameters of epoch, batch, optimizer, dropout, filters, and kernel, respectively. For this step, to overcome overfitting during training, the cross-validation (CV) technique is applied, and the DL model is run repeatedly at the same time. Then, the average value of the model is used to increase its reliability.

The next steps are the test processes:

- Step 4:** The DL is used with the obtained hyperparameters in Step 3 to predict the  $Y'_{\text{predict}}$ .
- Step 5:**  $Y_{\text{predict}}$  is calculated by using the value  $Y'_{\text{predict}}$ .
- Step 6:** The error value of the DL is determined based on the difference between  $Y_{\text{predict}}$  and  $Y_{\text{test}}$  by using Eqs. (8)-(10).

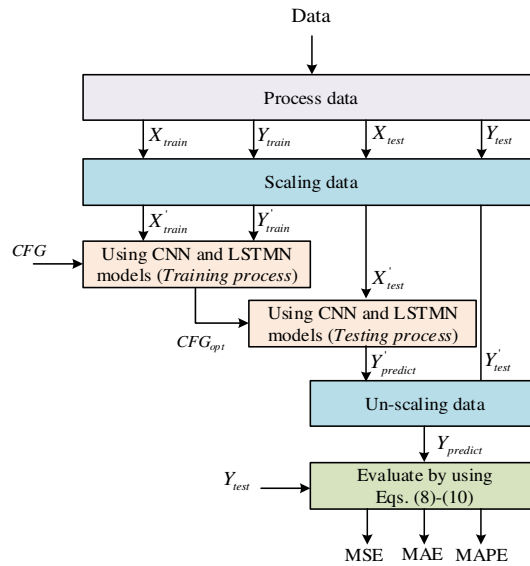


Fig. 5 The GS methodology based on data normalization

### 3. Simulation Data Setup

#### 3.1. Data

The half-hourly load demand data of Queensland state, Australia, and the hourly load demand data of Ho Chi Minh City, Vietnam, are used to verify the effectiveness of the proposed method. The selected data are divided into two different cases, corresponding to the LSTMN and the CNN. These datasets have different periods, and their statistical properties are shown in Table 2. Fig. 6 shows the  $Y_{train}$  value waveform in Case 1 according to the data normalization methods.

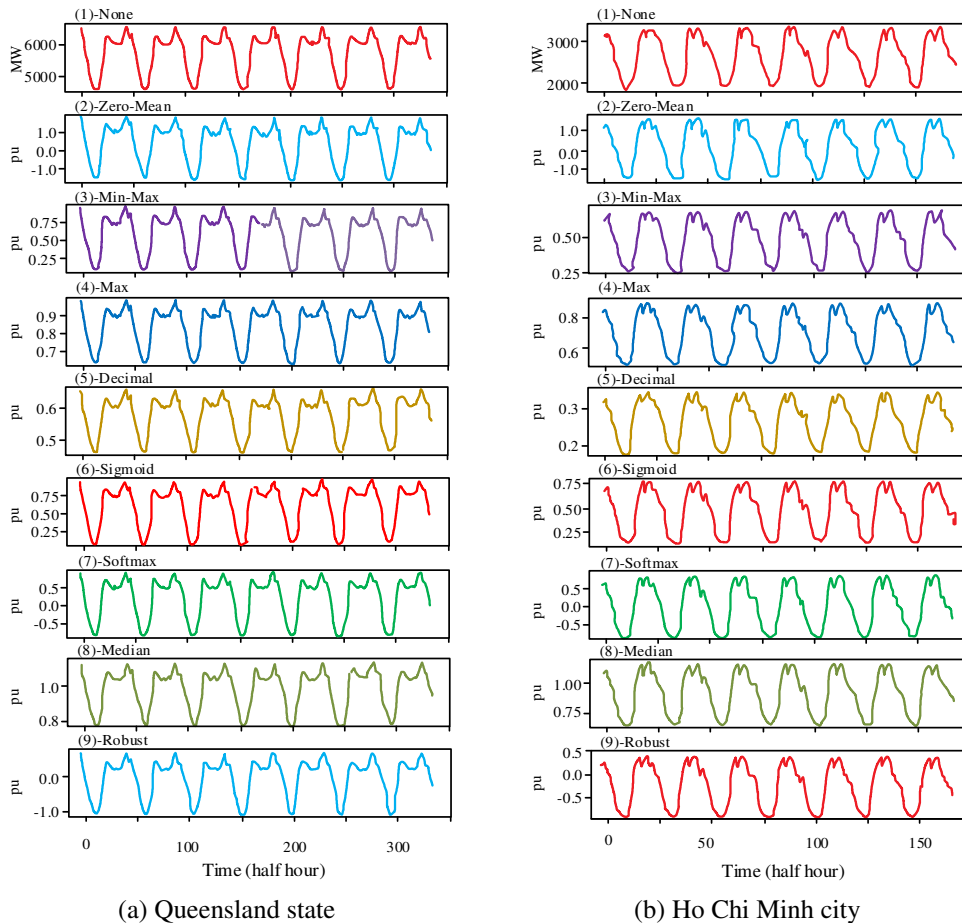


Fig. 6 The  $Y_{train}$  test value waveform in Case 1 according to the data normalization methods

Table 2 Data characteristics

Description	Case 1: LSTM model				Case 2: CNN model			
	Queensland state		Ho Chi Minh city		Queensland state		Ho Chi Minh city	
	$X_{train}$	$X_{test}$	$X_{train}$	$X_{test}$	$X_{train}$	$X_{test}$	$X_{train}$	$X_{test}$
Time (day)	05/10/14 05/23/14	05/24/14 05/30/14	25/11/18 22/12/18	12/23/18 12/29/18	03/29/14 05/23/14	05/24/14 05/30/14	10/28/14 12/22/14	12/23/18 12/29/18
Size	(672,48)	(336, 48)	(672, 24)	(168, 24)	(2688, 48)	(336, 48)	(1344, 24)	(168, 24)
Min (MW)	4,304.46	4,404.48	1,347.70	1,873.90	4,279.21	4,404.48	1,347.70	1,873.90
Mean (MW)	5,535.20	5,591.45	2,917.94	2,844.65	5,589.60	5,591.45	2,951.42	2,844.65
Max (MW)	6,917.66	6,824.76	3,945.90	3,695.20	6,984.78	6,824.76	3,945.9	3,695.20
Std (MW)	6,38.78	6,54.69	6,02.94	553.73	679.70	654.69	589.33	553.73

### 3.2. Simulation value setup

The values of the optimal hyperparameters for the DL model are listed in Table 3. For the LSTM model, the total number of the hyperparameter combinations, represented by  $cfg_i = \{E_i, B_i, O_i, D_i\}$ , is 81. The set value for the CV cycle is 2 (i.e., the training dataset is divided into two subsets corresponding to two times of training and testing). For the CNN model, the total number of the hyperparameter combinations, written as  $cfg_i = \{E_i, B_i, O_i, F_i, K_i\}$ , is 243. The set value for the CV cycle is equal to 3 (i.e., the training dataset is divided into three subsets corresponding to three times of training and testing). The set value for the number of repetitions is two times for both LSTM and CNN (i.e., each model is trained twice). The error measurement of the GS algorithm used in the training process is MAE.

Table 3 The values of the optimal hyperparameters for the DL model

Hyperparameter	LSTM model	CNN model
Epoch ( $E$ )	100, 300, 500	300, 500, 700
Batch ( $B$ )	10, 30, 50	30, 50, 70
Optimizer ( $O$ )	Adadelta, Adam, Adamax	Adagrad, Adam, SGD
Dropout rate ( $D$ )	0.1, 0.3, 0.5	-
Filter ( $F$ )	-	48, 80, 112
Kernel ( $K$ )	-	3, 5, 7
Number of combinations ( $CFG$ )	81	243

## 4. Experimental Results and Analyses

Tables 4 and 5 show the experimental results produced while using LSTM and CNN based on the GS algorithm during training for the data normalization scenarios, respectively. These tables illustrate that the DL model's ideal hyperparameters have distinct values for each data normalization approach and for various Queensland and Ho Chi Minh City datasets. In addition, it shows the same values of the optimal hyperparameter set in some cases. For example, in the instance of Queensland state data where the LSTMN is applied, the max and median approaches yield the same values of the ideal hyperparameters as the normal method (original data).

Table 4 The obtained results of optimal hyperparameters when using LSTMN

Method	Queensland state				Ho Chi Minh city			
	Epoch	Batch	Dropout	Optimizer	Epoch	Batch	Dropout	Optimizer
Normal	500	10	0.1	Adam	500	10	0.3	Adam
Zero-mean	500	10	0.1	Adamax	500	30	0.1	Adam
Min-max	500	10	0.1	Adamax	500	10	0.1	Adam
Max	500	10	0.1	Adam	300	10	0.1	Adam
Decimal	500	10	0.3	Adam	500	10	0.3	Adam
Sigmoid	500	10	0.1	Adamax	500	10	0.1	Adam
Softmax	500	10	0.1	Adamax	500	10	0.1	Adam
Median	500	10	0.1	Adam	300	10	0.1	Adam
Robust	500	50	0.1	Adam	500	10	0.1	Adam



Table 5 The obtained results of optimal hyperparameters when using CNN

Method	Queensland state					Ho Chi Minh city				
	Epoch	Batch	Optimizer	Filter	Kernel	Epoch	Batch	Optimizer	Filter	Kernel
Normal	700	50	Adam	112	7	700	30	Adam	80	5
Zero-mean	700	70	Adam	112	3	500	50	Adam	112	7
Min-max	500	50	Adam	112	3	700	50	Adam	112	7
Max	700	30	Adam	112	3	700	30	Adam	112	5
Decimal	700	50	Adam	112	7	700	70	Adam	112	7
Sigmoid	700	50	Adam	80	7	700	30	Adam	80	7
Softmax	300	70	Adam	80	7	700	50	Adam	80	7
Median	700	50	Adam	80	5	700	30	Adam	80	7
Robust	500	70	Adam	112	7	700	50	Adam	80	5

Table 6 The MAE when using the LSTMN

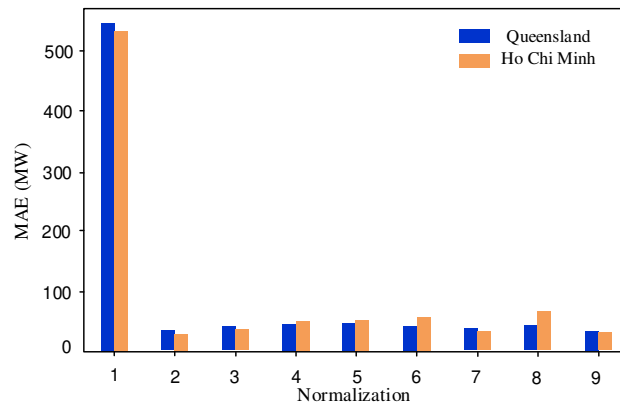
Method	MAE (MW)				MAPE (%)	
	Training		Test		Training	
	Queensland	Ho Chi Minh city	Queensland	Ho Chi Minh city	Queensland	Ho Chi Minh city
Normal	546.17	534.95	567.59	504.53	10.68	20.07
Standard	33.31	26.68	39.94	48.58	0.73	1.73
Min-max	40.21	35.37	39.14	46.81	0.70	1.76
Max	44.04	50.82	44.18	50.09	0.81	1.85
Decimal	45.63	47.32	43.74	53.42	0.80	2.00
Sigmoid	40.19	56.87	40.96	68.12	0.73	2.43
Softmax	34.43	30.88	36.64	43.95	0.66	1.61
Median	41.98	65.02	42.44	60.23	0.77	2.24
Robust	34.16	28.62	37.24	39.31	0.67	1.42

Table 7 The MAE when using CNN

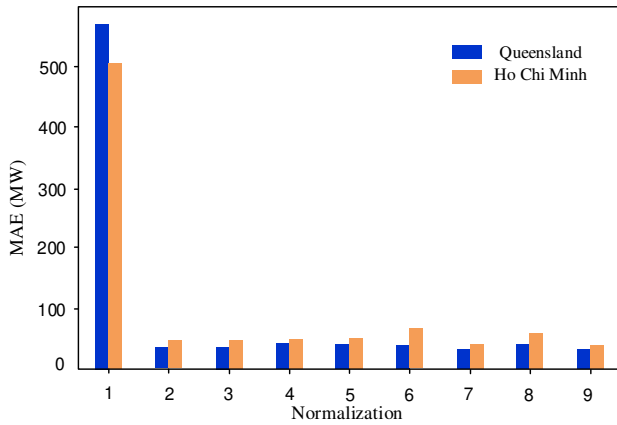
Method	MAE (MW)				MAPE (%)	
	Training		Test		Training	
	Queensland	Ho Chi Minh city	Queensland	Ho Chi Minh city	Queensland	Ho Chi Minh city
Normal	52.66	38.95	52.94	46.97	0.94	1.71
Standard	32.98	25.60	37.03	38.72	0.67	1.41
Min-max	35.98	26.02	37.18	38.65	0.66	1.42
Max	44.57	34.58	41.64	38.62	0.74	1.42
Decimal	40.76	37.25	39.38	43.80	0.71	1.61
Sigmoid	35.79	30.40	38.01	42.89	0.68	1.56
Softmax	33.17	24.48	35.51	36.19	0.64	1.37
Median	40.78	34.01	39.48	39.25	0.70	1.45
Robust	26.21	23.63	33.56	36.03	0.60	1.32

Table 6 presents the MAE and the MAPE error of the training and test stages of the LSTMN model. Fig. 7 shows the boxplot chart of these MAEs and MAPEs corresponding to Table 6. The obtained results show the effectiveness of the data normalization method for the GS algorithm in the LSTM model. Specifically, the MAE is significantly reduced when a data normalization method is applied. For Queensland data, the MAE of the test process is 567.59 MW and the MAPE is 10.68% without using any data normalization technique, while they decrease to 44.18 MW and 0.81% at max, respectively, when the proposed data normalization methods are applied.

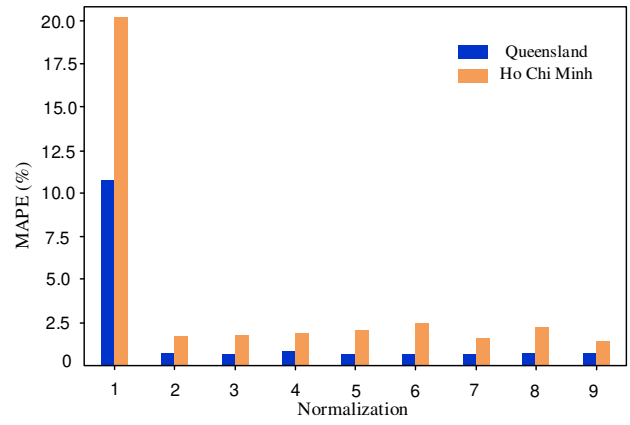
Similarly, Table 7 and Fig. 8 show the MAE, the MAPE, and their boxplots of the training and test stages for the CNN model. Again, the observed results show that applying data normalization methods significantly reduces both the MAE and MPPE. In other words, the performance of the GS algorithm is greatly improved with data normalization. Moreover, the effectiveness of applying data normalization techniques can be divided into three groups. The first group of applying the softmax and robust methods yields small MAEs. The second group, which presents medium MAEs, includes the zero-mean and the min-max. The third group that provides medium MAEs consists of the max, decimal, sigmoid, and median methods.



(a) MAE of the training process

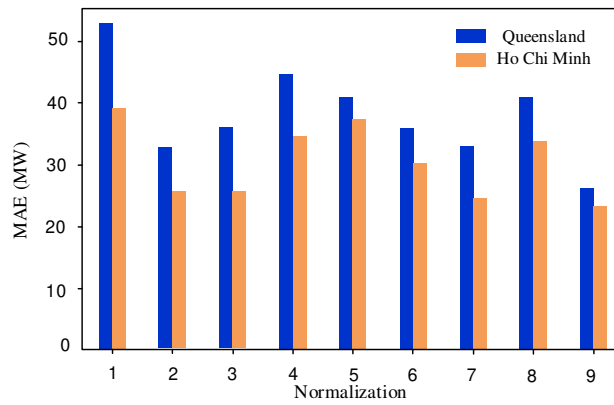


(b) MAE of the test process

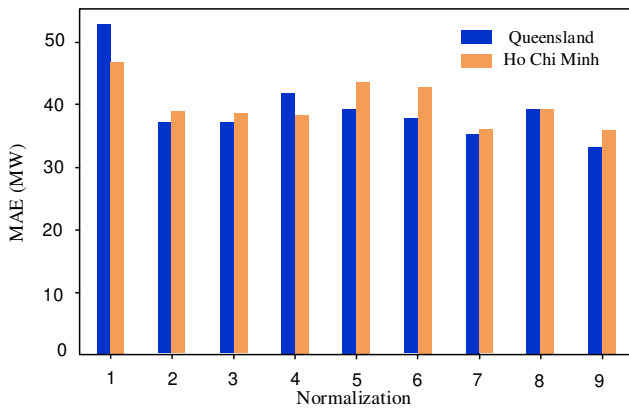


(c) MAPE of the test process

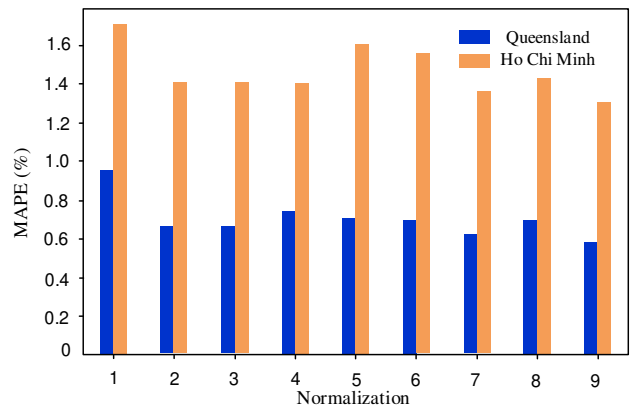
Fig. 7 The boxplot of the MAEs and MAPEs when using LSTMN



(a) MAE of the training process



(b) MAE of the testing process



(c) MAPE of the test process

Fig. 8 The boxplot of the MAEs and MAPEs when using CNN

## 5. Conclusions

This study presents an approach to examine the effect of data normalization methods on the GS algorithm for determining the optimal hyperparameters of the DL model, including the LSTMN and CNN, for energy load forecasting. The power load data of the Australian state of Queensland and the Vietnamese city of Ho Chi Minh were used to verify the reliability of the proposed method. The error evaluation indexes of MAE and MAPE of the actual and predicted values are established based on the epoch, batch, optimizer, dropout, filters, and kernel to determine the optimal hyperparameters of the DL model. The effectiveness of applying data normalization techniques can be divided into three groups. The first group of the applications of the softmax and robust methods yielded small MAEs. The second group, which presented medium MAEs, included the zero-mean and the min-max. The third group that provided medium MAEs consisted of the max, decimal, sigmoid, and median methods. The results showed that both MAE and MAPE were much smaller when applying data normalization. In addition, out of the eight proposed data normalization methods, zero-mean or min-max was not the best method for the GS algorithm for determining the optimal hyperparameters of the DL model.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] A. M. Omer, "Energy Use and Environmental Impacts: A General Review," *Journal of Renewable and Sustainable Energy*, vol. 1, no. 5, Article no. 053100, 2009.
- [2] K. Amasyali, et al., "A Review of Data-Driven Building Energy Consumption Prediction Studies," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1192-1205, January 2018.
- [3] E. Naml, et al., "Artificial Intelligence-Based Prediction Models for Energy Performance of Residential Buildings," *Recycling and Reuse Approach for Better Sustainability*, pp. 141-149, 2019.
- [4] Y. Jiang, et al., "Stochastic Receding Horizon Control of Active Distribution Networks with Distributed Renewables," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1325-1341, March 2019.
- [5] J. C. López, et al., "Parsimonious Short-Term Load Forecasting for Optimal Operation Planning of Electrical Distribution Systems," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1427-1437, March 2019.
- [6] S. Khan, et al., "Forecasting Day, Week and Month ahead Electricity Load Consumption of a Building Using Empirical Mode Decomposition and Extreme Learning Machine," *15th International Wireless Communications and Mobile Computing Conference*, pp. 1600-1605, June 2019.
- [7] T. T. Ngoc, et al., "Grid Search of Exponential Smoothing Method: A Case Study of Ho Chi Minh City Load Demand," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 19, no. 3, pp. 1121-1130, September 2020.
- [8] T. T. Ngoc, et al., "Grid Search of Multilayer Perceptron Based on the Walk-Forward Validation Methodology," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 2, pp. 1742-1751, April 2021.
- [9] K. Krishnakumari, et al., "Hyperparameter Tuning in Convolutional Neural Networks for Domain Adaptation in Sentiment Classification (HTCNN-DASC)," *Soft Computing*, vol. 24, no. 5, pp. 3511-3527, March 2020.
- [10] Y. Yu, et al., "Short-Term Load Forecasting Using Deep Belief Network with Empirical Mode Decomposition and Local Predictor," *IEEE Power and Energy Society General Meeting*, pp. 1-5, August 2018.
- [11] X. Wang, et al., "LSTM-Based Short-Term Load Forecasting for Building Electricity Consumption," *IEEE 28th International Symposium on Industrial Electronics*, pp. 1418-1423, June 2019.
- [12] M. Zahid, et al., "Electricity Price and Load Forecasting Using Enhanced Convolutional Neural Network and Enhanced Support Vector Regression in Smart Grids," *Electronics*, vol. 8, no. 2, Article no. 122, 2019.
- [13] N. M. Aszemi, et al., "Hyperparameter Optimization in Convolutional Neural Network Using Genetic Algorithms," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 6, pp. 269-278, March 2019.
- [14] J. Brownlee, *Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs, and LSTMs in Python*, New York: Machine Learning Mastery, 2018.
- [15] S. Mukhopadhyay, *Deep Learning and Neural Networks, Advanced Data Analytics Using Python*, Berkeley: Apress, 2018.

- [16] J. Moolayil, *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*, New York: Springer, 2019.
- [17] S. Raschka, et al., *Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow 2*, Hoboken: Wiley, 2019.
- [18] L. Yang, et al., "On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice," *Neurocomputing*, vol. 415, pp. 295-316, July 2020.
- [19] S. Motepe, et al., "Power Distribution Networks Load Forecasting Using Deep Belief Networks: The South African Case," *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology*, pp. 507-512, April 2019.
- [20] X. Zhang, et al., "Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning," *International Conference on Neural Information Processing*, pp. 287-295, December 2019.
- [21] X. Song, et al., "Time-Series Well Performance Prediction Based on Long Short-Term Memory (LSTM) Neural Network Model," *Journal of Petroleum Science and Engineering*, vol. 186, Article no. 106682, March 2020.
- [22] N. N. Bon, et al., "Fault Identification, Classification, and Location on Transmission Lines Using Combined Machine Learning Methods," *International Journal of Engineering and Technology Innovation*, vol. 12, no. 2, pp. 91-109, February 2022.
- [23] U. Michelucci, *Applied Deep Learning: A Case-Based Approach to Understanding Deep Neural Networks*, New York: Apress, 2018.
- [24] S. V. Subramanian, et al., "Deep-Learning Based Time Series Forecasting of Go-Around Incidents in the National Airspace System," *AIAA Modeling and Simulation Technologies Conference*, Article no. 0424, January 2018.
- [25] T. T. Ngoc, et al., "Support Vector Regression Based on Grid Search Method of Hyperparameters for Load Forecasting," *Acta Polytechnica Hungarica*, vol. 18, no. 2, pp. 143-158, January 2021.
- [26] A. S. Girsang, et al., "Stock Price Prediction Using LSTM and Search Economics Optimization," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 758-764, November 2020.



Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (<https://creativecommons.org/licenses/by-nc/4.0/>).