

## Article

# Metaheuristic Extreme Learning Machine for Improving Performance of Electric Energy Demand Forecasting

Sarunyoo Boriratrith<sup>1,2</sup>, Chitchai Srithapon<sup>3</sup>, Pradit Fuangfoo<sup>2</sup> and Rongrit Chatthaworn<sup>1,\*</sup><sup>1</sup> Department of Electrical Engineering, Faculty of Engineering, Khon Kaen University, Khon Kaen 40002, Thailand; sarunyooob@kkumail.com<sup>2</sup> Provincial Electricity Authority of Thailand (PEA), Bangkok 10900, Thailand; pradit.fua@pea.co.th<sup>3</sup> Department of Electrical Engineering, KTH Royal Institute of Technology, 11428 Stockholm, Sweden; chitchai@kth.se

\* Correspondence: rongch@kku.ac.th; Tel.: +66-84-685-2286

**Abstract:** Electric energy demand forecasting is very important for electric utilities to procure and supply electric energy for consumers sufficiently, safely, reliably, and continuously. Consequently, the processing time and accuracy of the forecast system are essential to consider when applying in real power system operations. Nowadays, the Extreme Learning Machine (ELM) is significant for forecasting as it provides an acceptable value of forecasting and consumes less computation time when compared with the state-of-the-art forecasting models. However, the result of electric energy demand forecasting from the ELM was unstable and its accuracy was increased by reducing overfitting of the ELM model. In this research, metaheuristic optimization combined with the ELM is proposed to increase accuracy and reduce the cause of overfitting of three forecasting models, composed of the Jellyfish Search Extreme Learning Machine (JS-ELM), the Harris Hawk Extreme Learning Machine (HH-ELM), and the Flower Pollination Extreme Learning Machine (FP-ELM). The actual electric energy demand datasets in Thailand were collected from 2018 to 2020 and used to test and compare the performance of the proposed and state-of-the-art forecasting models. The overall results show that the JS-ELM provides the best minimum root mean square error compared with the state-of-the-art forecasting models. Moreover, the JS-ELM consumes the appropriate processing time in this experiment.

**Keywords:** electricity forecasting; Extreme Learning Machine; improvement model; machine learning; metaheuristic; Jellyfish Search Optimization; Harris Hawk Optimization; Flower Pollination Algorithm



**Citation:** Boriratrith, S.; Srithapon, C.; Fuangfoo, P.; Chatthaworn, R.

Metaheuristic Extreme Learning Machine for Improving Performance of Electric Energy Demand Forecasting. *Computers* **2022**, *11*, 66.

<https://doi.org/10.3390/computers11050066>

Academic Editors: Pedro Pereira, Luis Gomes and João Goes

Received: 16 March 2022

Accepted: 24 April 2022

Published: 27 April 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

For several decades, electric energy has been essential for living, and it is a basic utility that governments must provide for their people. Indeed, power utilities or electricity providers must procure and supply electric energy to consumers sufficiently, safely, reliably, and continuously [1]. In order to achieve all those factors previously mentioned, forecasting electric energy demand is necessary for utilities. Forecasting is the famous well-known method that learns the historical data and then predicts the expected data [2]. Many works have used various forecasting models [3], such as the dynamic regression model [4], ARIMA model [5], exponential smoothing model [6], neural network model [7], and so on, to generate the trend of future forecasted data and select the best model that is evaluated from its performance and accuracy.

Many researchers have proposed machine learning models to increase the accuracy of electric energy demand forecast. Gholamreza Memarzadeh and Farshid Keynia [8] proposed the new optimal long short-term memory model to predict electric energy demand and price based on Pennsylvania, New Jersey, and Maryland databases. Zongying Liu et al. [9] proposed the novel adaptive method applied with the kernel ELM model called the error-output recurrent two-layer ELM and used the quantum particle swarm

optimization to increase the forecast accuracy of the time-series predicted demand. Mikel Larrea et al. [10] proposed the particle swarm optimization algorithm to optimize weight parameters in the ensemble ELM model for forecasting the Spanish time-series electric consumption. Yanhua Chen et al. [11] proposed the empirical mode decomposition to decompose the time-series forecasting data and mixed kernel with radial basis function and UKF to implement on the ELM model. The New South Wales, Victoria, and Queensland electric load databases were used for testing the proposed model. Qifang Chen et al. [12] proposed the novel deep learning model by using the stacked auto-encoder framework applied to the ELM model to improve the capabilities of forecasting. Moreover, the raw time-series data were analyzed by using the empirical mode decomposition method. Shafiul Hasan Rafi et al. [13] proposed the hybrid methodology of deep learning by integrating the convolutional neural network and long short-term memory to improve the forecast accuracy for forecasting the Bangladesh power system dataset. Muhammad Sajjad et al. [14] proposed the hybrid convolutional neural network and gated recurrent units to maximize forecast performance for appliance energy demand and individual household electric power demand. Yusha Hu et al. [15] proposed the optimization methods consisting of the genetic algorithm and particle swarm optimization applied to the backpropagation model to forecast the electric demand dataset. Mohammad-Rasool Kazemzadeh et al. [16] proposed the novel hybrid optimization method, which combines the particle swarm optimization with a machine learning model, to forecast the long-term electricity peak demand and electric energy demand. Ghulam Hafeez et al. [17] proposed the novel hybrid model of modified mutual information, factored condition restricted Boltzmann machine, and genetic wind driven optimization to forecast the hourly load data of FE, Dayton, and EKPC USA power grids.

From the aforementioned works, one of the machine learning models that can learn rapidly and provide acceptable results is the Extreme Learning Machine (ELM) [18]. The concept of this model is to learn the data without iterative tuning in the hidden layer phase and then calculate the output weight parameter with a pseudo-inverse matrix called the Moore–Penrose inverse matrix [19] in the output layer phase. However, the disadvantage of the ELM model [20–22] is that it uses the standard randomization of the input weight to compute in the input layer phase that can cause overfitting and has a high probability to fall into local optima.

To improve the performance of the electric energy demand forecasting model, this paper proposes a method to develop the ELM model by using metaheuristic optimization algorithms to adjust the appropriate result of the output weight, which can reduce the cause of overfitting. The Jellyfish Search Optimization (JSO) [23], the Harris Hawk Optimization (HHO) [24], and the Flower Pollination Algorithm (FPA) [25] were selected for adjusting the output weight and reducing the cause of overfitting as mentioned before. The performances of these three proposed forecasting models were evaluated via an experiment that used the seven groups of the electric energy demand datasets in Thailand. The main contributions of this work can be summarized as follows:

- This paper presents the novel hybrid method combining the ELM and metaheuristic optimization consisting of the JSO, the HHO, and the FPA to forecast the electric energy demand. Furthermore, the proposed model is investigated with the real-life dataset of electric energy demand to challenge the forecasting performance in terms of forecasting accuracy and stability.
- To increase the robustness of forecasting in the training and testing processes of the traditional ELM model, the randomization process of the initial weight parameter in the ELM is developed to obtain the optimal weight parameter. The proposed metaheuristic optimization algorithms are not complex to implement, leading to less processing time, low population usage, and fast convergence to the optimal solution. In addition, these three optimization methods can reduce the number of hidden nodes of the ELM model.

- Finally, the presented metaheuristic algorithms (the JSO, the HHO, and the FPA) have the characteristic of being self-adaptive to tune the weight parameter of the ELM without trapping in the local optima. This characteristic can increase the forecasting stability of the traditional model. Furthermore, this method can reduce the cause of sensitivity to outliers, which leads the forecasting process to be more stable, and the standard deviation was used to calculate the forecasting stability of the proposed models.

This paper is organized as follows. Section 2 indicates the basic principles of the algorithms used in this paper. Section 3 indicates the methodology of the datasets' preparation and the proposed models. Section 4 indicates the overall experimental results using the datasets and the proposed models in Section 3. Lastly, the conclusion and future work of this work are described in Section 5.

## 2. Basic Principles

This section describes the basic principles and materials used in this research, which consists of forecasting methodology, ELM, JSO, HHO, FPA, a summary of selected metaheuristic optimization, and related works as shown below.

### 2.1. Forecasting Methodology

The basic methodology for forecasting (Figure 1) can be described in five steps [26] as follows. Firstly, the forecasting task problem is defined; the aim of this step is to study the problem and factors that can affect the outcome of forecasting. Secondly, the information is gathered for forecasting; the aim of this step is to collect and analyze the significant data needed for forecasting and finding the expected result [27]. Thirdly, the exploratory analysis of the overall forecasting is conducted; the aim of this step is to actively analyze the consistency of data to so that they can be implement without noise or elements missing in the dataset such as the imbalance or inconsistency of data [28], missing values of data [29], and so on. Fourthly, the fitting models are selected; this step is the key of this research, which proposes the new model to increase the performance of the forecasting. Fifthly, the evaluation of the forecasting model is conducted, this step aims to consider and evaluate the experimental results obtained from all models and then select the best model for forecasting. The performances of all models [30] can be compared by considering error metrics such as the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE), and so on.

### 2.2. Extreme Learning Machine

The Extreme Learning Machine (ELM) was proposed by Guang-Bin Huang et al. [18]. Due to the non-iterative tuning model, the training time of the ELM model was faster than other machine learning models in the hidden layer.

Figure 2 is the architecture of the ELM model. The structure of this model has three layers. The first layer is the input layer, which imports the sample dataset  $(a_j, t_j)$  of input data  $(a_j = [a_{j1}, a_{j2}, a_{j3}, \dots, a_{jn}]^T \in R^n)$  and target data  $(t_j = [t_{j1}, t_{j2}, t_{j3}, \dots, t_{jm}]^T \in R^m)$  where  $n$  is the number of instances of data with  $j = 1, 2, \dots, n$ . After the input layer step is completed, the second hidden layer is calculated as shown as in (1).

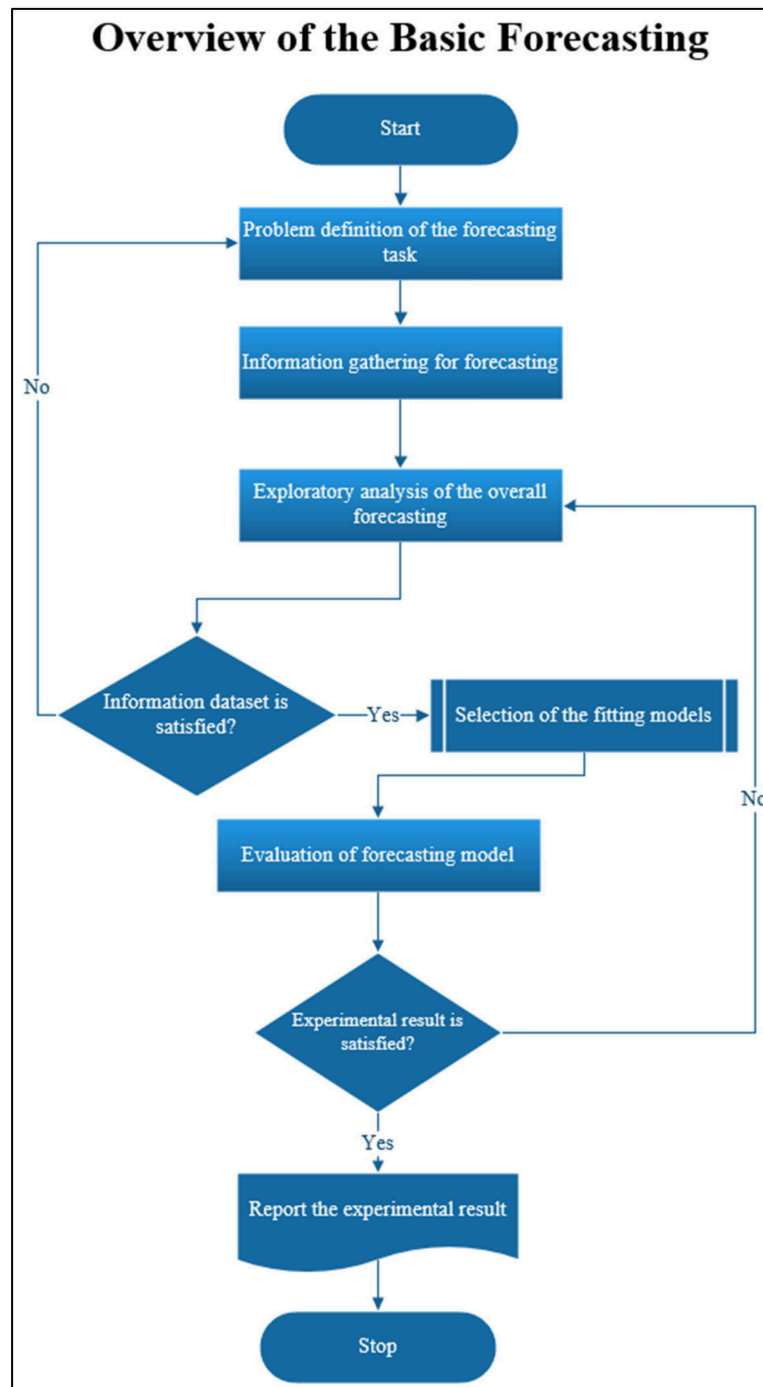
$$\sum_{i=1}^L \beta_i g_i(a_j) = \sum_{i=1}^L \beta_i g(w_i \cdot a_j + b_i) = e_j \quad (1)$$

where  $w_i$  is the initial weight randomization,  $\beta_i$  is the connecting weight of hidden nodes and output nodes,  $L$  is the random number of hidden nodes, and  $g(a)$  is the activation function. Equation (1) can be revised to a short equation defined as the matrix, as in (2).

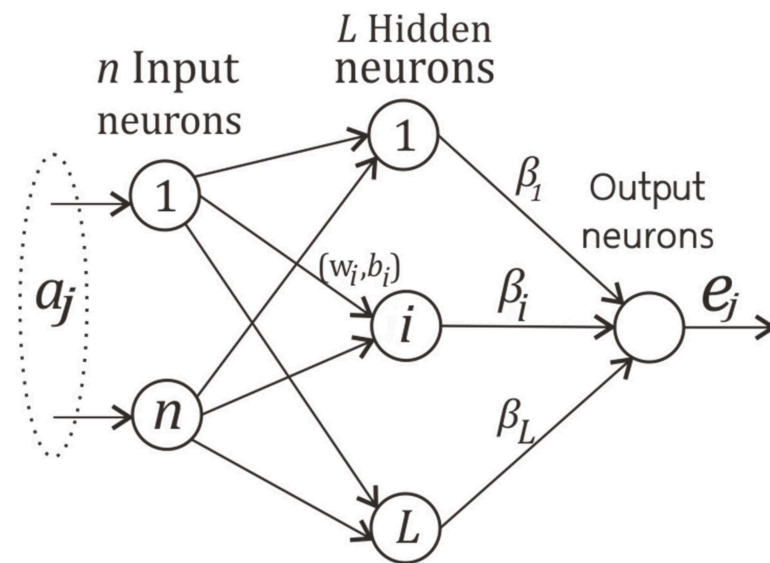
$$H\beta = T \quad (2)$$

$$\text{where } H = \begin{bmatrix} h(a_1) \\ \vdots \\ h(a_N) \end{bmatrix} = \begin{bmatrix} G(w_1, b_1, a_1) & \cdots & G(w_L, b_L, a_1) \\ \vdots & \ddots & \vdots \\ G(w_1, b_1, a_N) & \cdots & G(w_L, b_L, a_N) \end{bmatrix}_{N \times L} \quad \beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m}$$

$$\text{and } T = \begin{bmatrix} t_1^T \\ \vdots \\ t_L^T \end{bmatrix}_{N \times m}.$$



**Figure 1.** Overview of the Basic Forecasting Process [26].



**Figure 2.** The architecture of the Extreme Learning Machine Model [18,31].

After the hidden layer step is completed, the aim of the third output layer step is to find the  $\beta$  matrix; Equation (2) is inverted to Equation (3) as shown below.

$$\beta = H^+ T \quad (3)$$

where  $H^+$  is the Moore–Penrose pseudo-inverse [19] of the  $H$  matrix.

Due to the problem of overfitting of the ELM model [20], this paper proposes a method to find the optimal weight parameter with the metaheuristic optimization that can reduce the cause of the overfitting.

### 2.3. Jellyfish Search Optimization

Jellyfish Search Optimization (JSO) was proposed by Jui-Sheng Chou and Dinh-Nhat Truong [23]. The main idea of this optimization was inspired by the nature of jellyfish behavior in the ocean when hunting prey for their food [32]. Figure 3 is the JSO algorithm flowchart that can be described as follows:

1. Define the objective function ( $f(X)$ ) in terms of  $X = (x_1, \dots, x_d)^T$ , the best location parameter ( $X^*$ ), the number of search space, the number of population ( $N$ ), the max iteration ( $T$ ), the iteration cycle time starting from 1 to max iteration ( $t$ ). Jellyfish population ( $X_i$ ) is initialized with a logistic chaotic map [33].
2. Calculate the control time ( $c(t)$ ) as presented in (4).

$$c(t) = \left| \left( 1 - \frac{1}{T} \right) \times (2 \times \text{rand}(0,1) - 1) \right| \quad (4)$$

where  $\text{rand}(0,1)$  is the number of randomization from 0 to 1, which changes in every iteration.

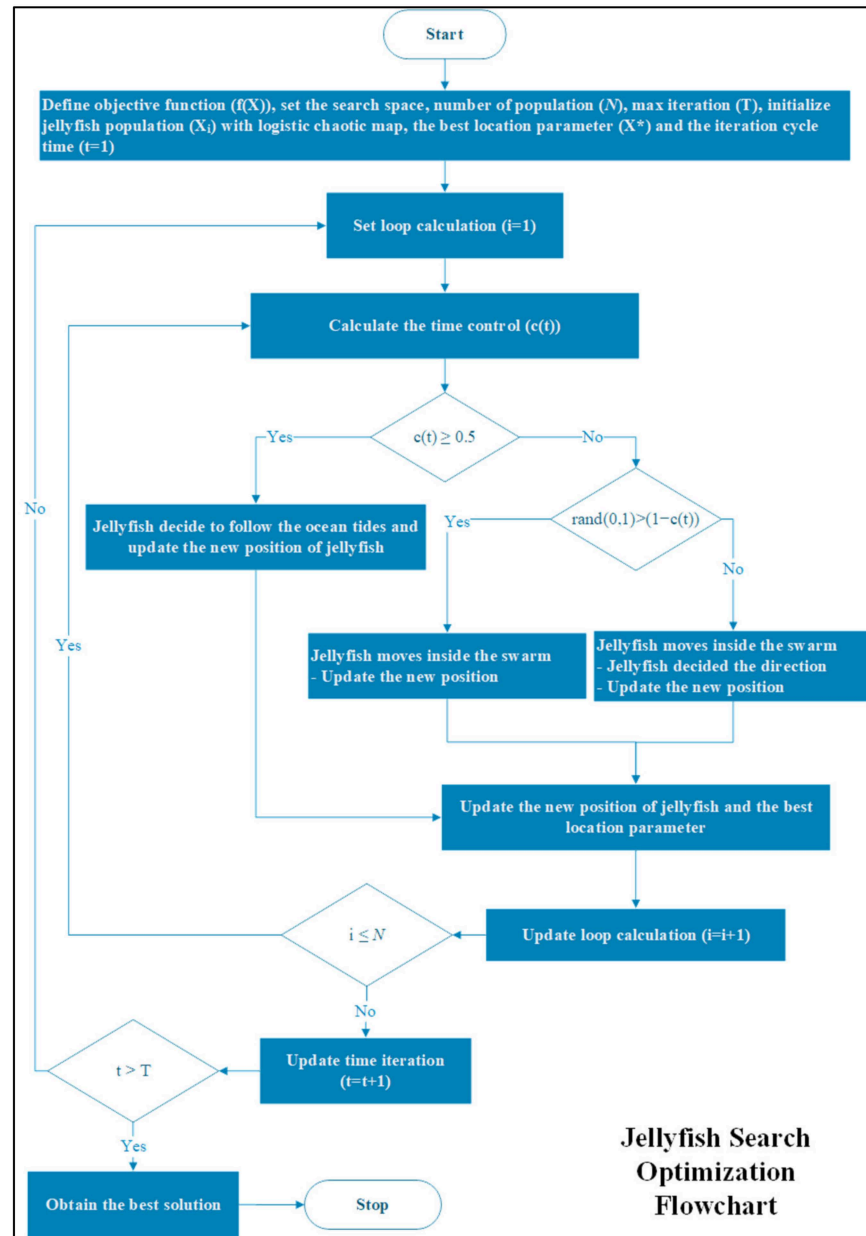
If the control time is greater than or equal to 0.5 ( $c(t) \geq 0.5$ ), then the jellyfish decides to follow the ocean tides as shown in (5). Otherwise, the movement of jellyfish to the swarm is calculated by Equation (7).

$$\overrightarrow{\text{trend}} = X^* - \beta \times \text{rand}(0,1) \times \mu \quad (5)$$

where  $\overrightarrow{\text{trend}}$  is the direction of the ocean tides,  $\beta$  is a distribution coefficient that is greater than 0 ( $\beta > 0$ ), and  $\mu$  is the mean location of all jellyfishes.

After that, jellyfish live at the new position that can be calculated as shown as in (6).

$$X_i(t+1) = X_i(t) + rand(0,1) \times (X^* - \beta \times rand(0,1)) \times \mu \quad (6)$$



**Figure 3.** Flowchart of the Jellyfish Search Optimization [23].

If  $rand(0,1)$  is greater than  $1 - c(t)$ , then jellyfish show the passive motions, which can be calculated by Equation (7). Otherwise, jellyfish show the active motions by deciding the direction as shown by Equation (8).

$$X_i(t+1) = X_i(t) + \gamma \times rand(0,1) \times (U_b - L_b) \quad (7)$$

where  $\gamma$  is the motion coefficients, and  $U_b$  and  $L_b$  are upper bound and lower bound, respectively.

$$\overrightarrow{Step} = X_i(t+1) - X_i(t) = rand(0,1) \times \overrightarrow{Direction} \quad (8)$$

where  $\overrightarrow{Step}$  is the direction of the movement of jellyfish and  $\overrightarrow{Direction}$  can be decided by Equation (9).

$$\overrightarrow{Direction} = \begin{cases} X_j(t) - X_i(t) & \text{if } f(X_i) \geq f(X_j) \\ X_i(t) - X_j(t) & \text{if } f(X_i) < f(X_j) \end{cases} \quad (9)$$

3. Update the new position of the jellyfish and the best location parameter.
4. Repeat step 2 until the iteration reaches the max iteration criterion.

The JSO algorithm is used to optimize the weight parameter in the ELM model that is described in Section 3.3.

#### 2.4. Harris Hawk Optimization

The Harris Hawk Optimization (HHO) was proposed by Ali Asghar Heidari et al. [24]. The main idea of this optimization model came from the cooperative behavior and hunting of prey of Harris Hawk [34]. Figure 4 presents a flowchart of the HHO algorithm that can be described as follows:

1. Define the objective function ( $f(X)$ ) in terms of  $X = (x_1, \dots, x_d)^T$ , the location of rabbit (best location parameter) ( $X^*$ ), the number of population ( $N$ ), the max iteration ( $T$ ), the iteration cycle time starting from 1 to max iteration ( $t$ ), and initialize hawk population ( $X_i$ ).
2. Specify the initial energy ( $E_0$ ) as in (10).

$$E_0 = 2 \times rand(0, 1) - 1 \quad (10)$$

3. Specify the initial jump strength ( $J$ ) as in (11).

$$J = 2 \times (1 - rand(0, 1)) \quad (11)$$

4. Adjust the escaping energy of the prey ( $E$ ) as in (12).

$$E = 2E_0 \left( 1 - \frac{t}{T} \right) \quad (12)$$

5. If  $|E| \geq 1$ , then go to the exploration phase as in (13).

If  $q \geq 0.5$ :

$$X(t+1) = X_{rand}(t) - rand_1(0, 1) |X_{rand}(t) - 2rand_2(0, 1)X(t)| \quad (13)$$

If  $q < 0.5$ :

$$X(t+1) = (X^* - X_m(t)) - rand_3(0, 1)(L_b + rand_4(0, 1)(U_b - L_b))$$

where  $X_{rand}(t)$  is the random position of the currently selected hawk,  $X_m(t)$  is the average position of the current population, and  $q$  is random numbers between 0 and 1.

6. If  $|E| < 1$ , then consider with four conditions of the exploitation phase as follows:
  - 6.1. If the instantly randomized number ( $r$ ) is greater than or equal to 0.5 ( $r \geq 0.5$ ) and  $|E| \geq 0.5$ , then go to the soft besiege as in (14).

$$X(t+1) = (X^* - X(t)) - E|X^* - X(t)| \quad (14)$$

- 6.2. If  $r \geq 0.5$  and  $|E| < 0.5$ , then go to the hard besiege as in (15).

$$X(t+1) = X^* - E|X^* - X(t)| \quad (15)$$

- 6.3. If  $r < 0.5$  and  $|E| \geq 0.5$ , then go to the soft besiege with progressive rapid dives as in (16).

$$X(t+1) = \begin{cases} Y & \text{if } f(Y) < f(X(t)) \\ Z & \text{if } f(Z) < f(X(t)) \end{cases} \quad (16)$$

where  $Y$  is the next movement of the hawk as in (17).

$$Y = X^* - E|JX^* - X(t)| \quad (17)$$

and  $Z$  is the random movement of the hawk with the levy flight concept as in (18).

$$Z = (X^* - E|JX^* - X(t)|) S \times LF(D) \quad (18)$$

where  $D$  is the dimension of the problem,  $S$  is the  $1 \times D$  size of random vector and  $LF$  is the levy flight function [35] as in (19).

$$LF = 0.01 \times \frac{\text{rand}(0,1) \times \left( \frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}}}{|\text{rand}(0,1)|^{\frac{1}{\beta}}} \quad (19)$$

where  $\beta$  is the default constant set to 1.5.

- 6.4. If  $r < 0.5$  and  $|E| < 0.5$ , then go to the hard besiege with progressive rapid dives as in (20).

$$X(t+1) = \begin{cases} Y & \text{if } f(Y) < f(X(t)) \\ Z & \text{if } f(Z) < f(X(t)) \end{cases} \quad (20)$$

where  $Y$  is the next movement of the hawk as in (21).

$$Y = X^* - E|JX^* - X_m(t)| \quad (21)$$

and  $Z$  is the random movement of the hawk with levy flight concept as in (22).

$$Z = (X^* - E|JX^* - X_m(t)|) S \times LF(D) \quad (22)$$

7. Update the new position of the hawk and the location of the rabbit (best location parameter).
8. Repeat step 4 until the iteration reaches the max iteration criterion.

The HHO algorithm is used to optimize the weight parameter in the ELM model that is described in Section 3.3.

### 2.5. Flower Pollination Algorithm

The Flower Pollination Algorithm (FPA) was proposed by Xin-She Yang [25]. The main idea of this optimization model was inspired by the nature of the pollination process of flowers. Figure 5 presents the FPA algorithm flowchart that can be described as follows:

1. Define the objective function ( $f(X)$ ) in terms of  $X = (x_1, \dots, x_d)^T$ , define the best location parameter ( $X^*$ ), set the number of population ( $N$ ), set the max iteration ( $T$ ), set the iteration cycle time starting from 1 to max iteration ( $t$ ), and initialize flower population ( $X_i$ ) and fixed switch probability set to 0.8 ( $p = 0.8$ ).
2. Define the random number between 0 and 1 and compare it with switch probability. If  $\text{rand}(0,1) < p$ , then go to the global pollination phase as in (23). Otherwise, go to the local pollination phase as in (24).

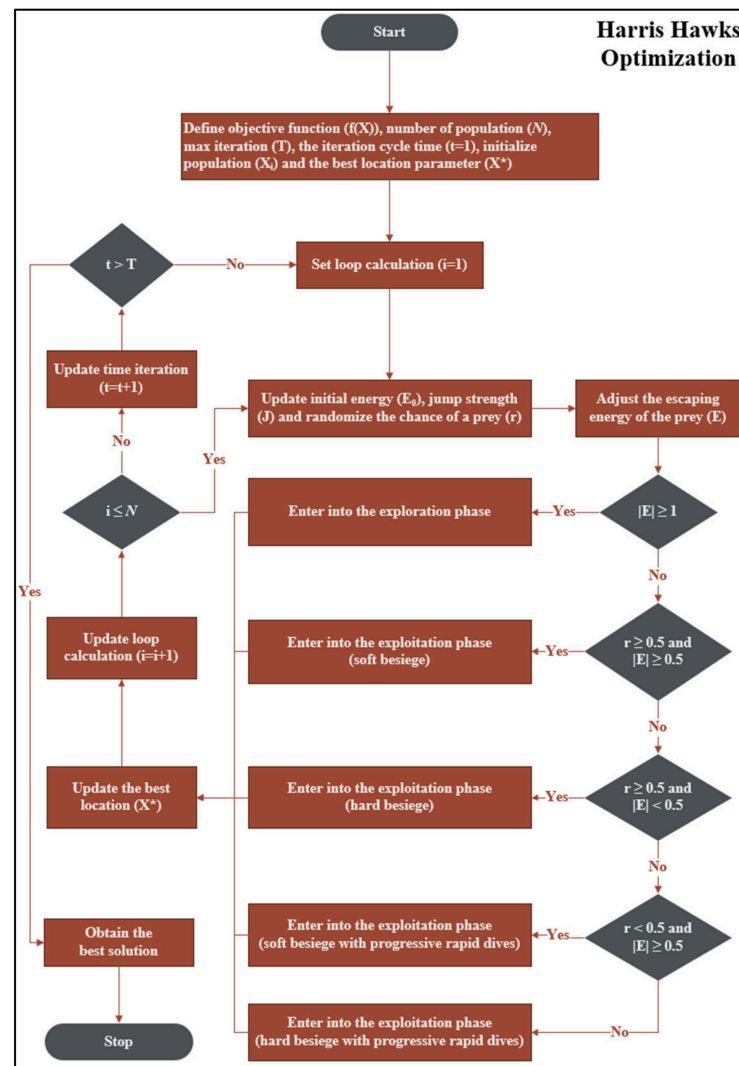
$$X_i(t+1) = X_i(t) + LF(X_i(t) - X^*) \quad (23)$$

where  $LF$  is the levy flight function [35], which is defined as in (19).

$$X_i(t+1) = X_i(t) + \text{rand}(0,1) \times (X_j(t) - X_k(t)) \quad (24)$$

where  $X_j(t)$  and  $X_k(t)$  are other populations obtained from a random position in terms of  $t$ .

3. Update the new position of flowers and the best location parameter.
4. Repeat step 2 until the iteration reaches the max iteration criterion.



**Figure 4.** Flowchart of the Harris Hawk Optimization [24].

The FPA algorithm is used to optimize the weight parameter in the ELM model that is described in Section 3.3.

## 2.6. Summary of the JSO, the HHO, and the FPA

As aforementioned, the Jellyfish Search Optimization, the Harris Hawk Optimization, and the Flower Pollination Algorithm are the metaheuristic algorithms inspired by animals' and plants' behavior in nature [36,37]. According to Table 1, the benefits and drawbacks of the three metaheuristics are summarized. The benefits of JSO [23] include its fast speed to converge calculation, stability in the processing of the problem, and being hard to trap in local optima. However, the drawback of JSO is that there are quite a minor of related works because this optimization was a novel metaheuristics model in 2020. The benefit of HHO [38] is there are many steps of calculation that are hard to trap in local optima; however, the drawback of HHO is that it is slow to converge calculation. Finally, the benefits of FPA [39] are it is fast to converge calculation because the steps of this optimization are simple and easy to implement; however, the drawback of FPA is that it is easy to trap in local optima.

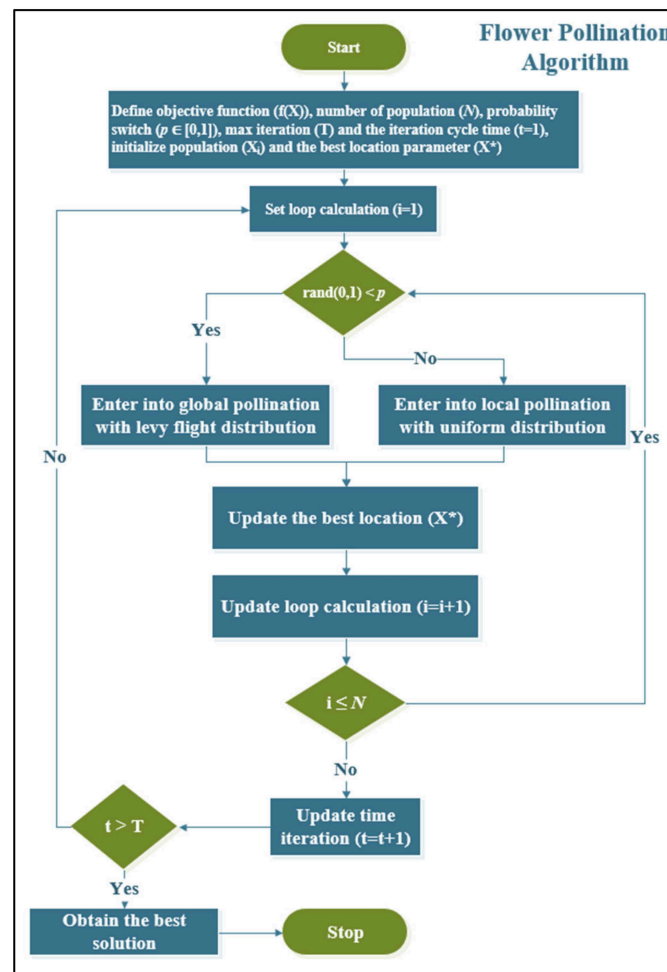


Figure 5. Flowchart of the Flower Pollination Algorithm [25].

Table 1. Benefits and Drawbacks of the JSO, the HHO, and the FPA.

Metaheuristic Algorithms	Benefits	Drawbacks
JSO	<ul style="list-style-type: none"> <li>Fast calculation.</li> <li>Stability of the processing.</li> <li>Hard to trap in local optima.</li> </ul>	<ul style="list-style-type: none"> <li>There are quite a minor of related works due to the novel metaheuristic model (2020).</li> </ul>
HHO	<ul style="list-style-type: none"> <li>Hard to trap in local optima.</li> </ul>	<ul style="list-style-type: none"> <li>Slow calculation due to the complex steps of the algorithm.</li> </ul>
FPA	<ul style="list-style-type: none"> <li>Fast calculation due to the simple steps of the algorithm</li> </ul>	<ul style="list-style-type: none"> <li>Easy to trap in local optima.</li> </ul>

In conclusion, the three metaheuristic optimizations were merged with the ELM model to optimize the weight parameter, compare the three proposed models with the forecasting dataset, and evaluate the model with the error metrics objective function.

### 3. Data Preparation and Proposed Models

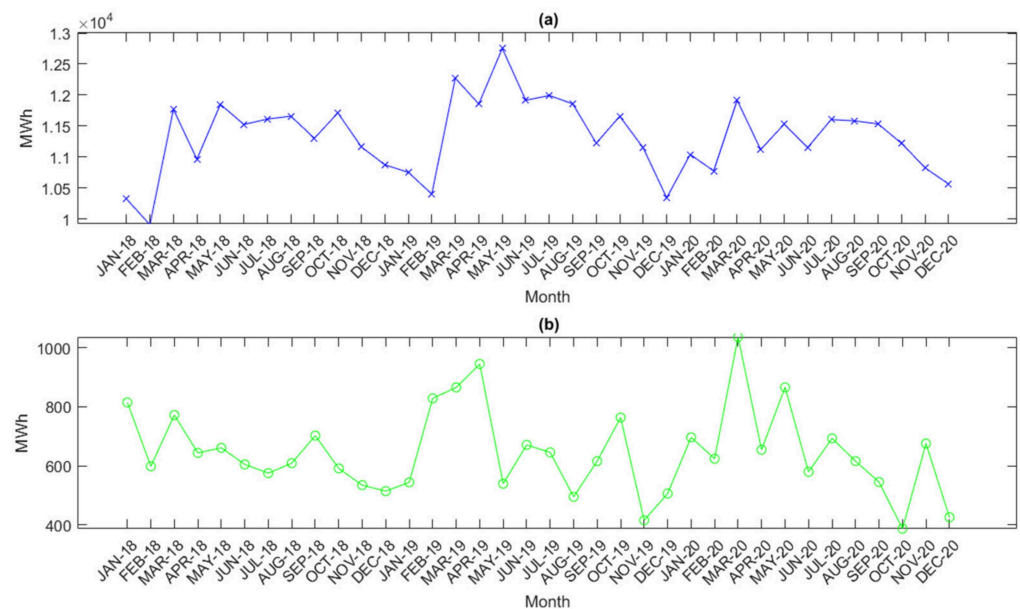
This section describes the technique of this research work, which consists of an overview of the electric energy demand datasets in Thailand, proposed data, and models used in the experiment to achieve the best result, the experimental setup and hyper-

parameter settings for all proposed and state-of-the-art models, and performance evaluation of each model.

### 3.1. Preparation of Electric Energy Demand Data

Electric energy demand datasets were collected from Provincial Electricity Authority in Thailand (PEA), which can be accessed from references [40,41]. The scope of the electric energy demand datasets in this research was set from 2018 to 2020.

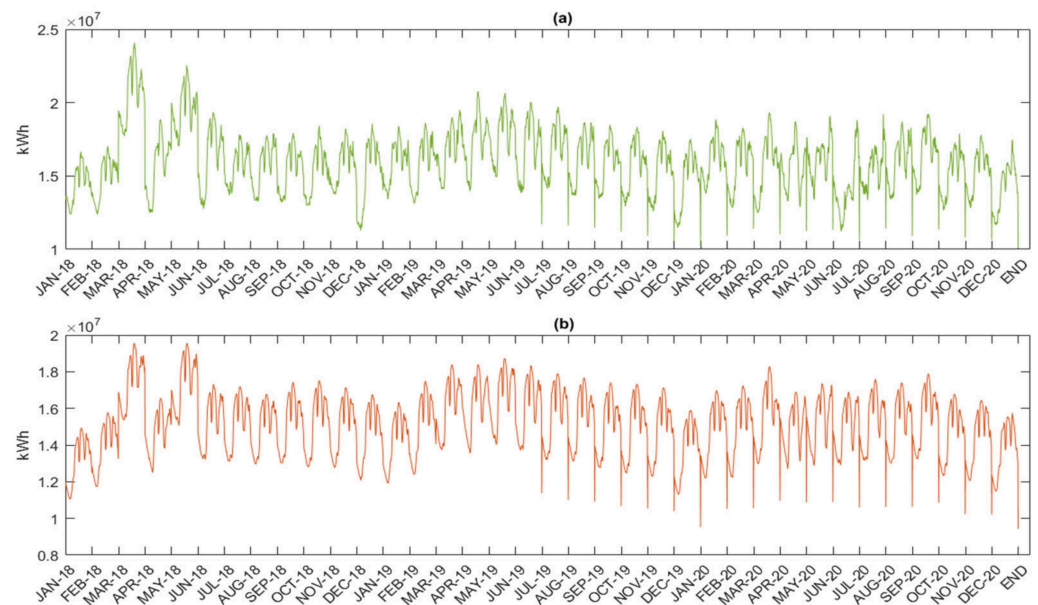
Figure 6a is the pattern of the monthly demand data of electric energy from January 2018 to December 2020 in terms of MegaWatt-Hour (MWh). The minimum electric energy demand is 9909 MWh in February 2018, the maximum electric energy is 12,752 MWh in May 2019, and the average electric energy demands in 2018, 2019, and 2020 are 11,222 MWh, 11,514 MWh, and 11,239 MWh, respectively. Figure 6b is the pattern of the monthly loss data of electric energy in Thailand from January 2018 to December 2020. The minimum loss is 387 MWh in October 2020, the maximum loss is 1036 MWh in March 2020, and the average losses in 2018, 2019, and 2020 are 635 MWh, 653 MWh, and 650 MWh, respectively.



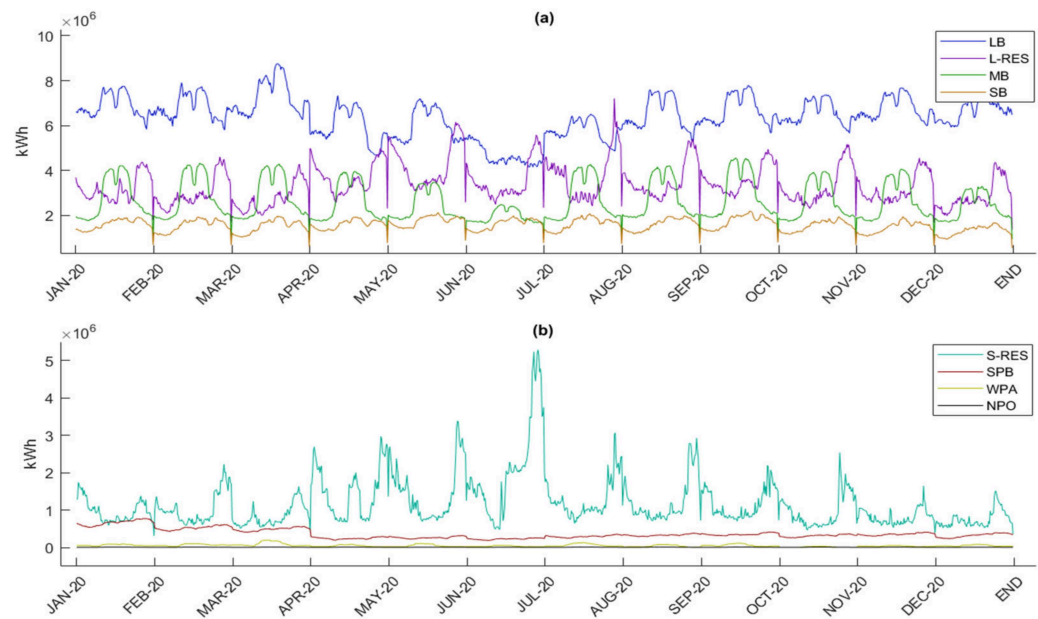
**Figure 6.** (a) The pattern of electric energy demand data from 2018 to 2020. (b) The pattern of electric energy loss data from 2018 to 2020.

Figure 7a,b are the patterns of the peak demand data of electric energy and the pattern of the workday demand data of electric energy, respectively. Both peak demand data and workday demand data were collected in 15 min intervals from January 2018 to December 2020 in terms of a kiloWatt-Hour (kWh). The peak demand data were collected from the days that use the highest electricity for each month, and the workday demand data were collected from Monday to Friday for each month.

Figure 8 is the group pattern of the peak demand data of electric energy collected in 15 min intervals from January 2018 to December 2020 in terms of kWh. This dataset can be separated into 8 clusters when sorting from the highest total electric energy demand to the lowest total electric energy demand. The 8 clusters consist of Large Business (LB), Large Residential or Residential (L-RES) demand, which consumes energy greater than or equal to 150 kWh per month, Medium Business (MB), Small Business (SB), Small Residential or Residential (S-RES) demand, which consumes energy less than 150 kWh per month, Specific Business (SPB), Water Pumping for Agriculture (WPA), and Nonprofit Organization (NPO).



**Figure 7.** (a) Pattern of peak day electric energy demand data from 2018 to 2020. (b) Pattern of workday electric energy demand data from 2018 to 2020.



**Figure 8.** (a) Pattern of high electric energy demand cluster, which consists of Large Business (LB), Large Residential (L-RES), Medium Business (MB), and Small Business (SB). (b) Pattern of low electric energy demand cluster, which consists of Small Residential (S-RES), Specific Business (SPB), Water Pumping for Agriculture (WPA), and Nonprofit Organization (NPO).

### 3.2. Dataset

When the preparation of electric energy demand data was completed, the experimental datasets were built to be tested in proposed models to forecast data with minimum error. In this research, the datasets were classified into seven groups for testing the proposed models. Table 2 describes the detail of each dataset, which is separated in terms of train and test data.

**Table 2.** Details of Datasets.

Datasets	Dataset Type	Forecasting Type	Train Data	Test Data
1	Monthly electric energy demand data	Long-term forecasting	JAN 2018–DEC 2019	JAN 2020–DEC 2020
2	Monthly electric energy loss data	Long-term forecasting	JAN 2018–DEC 2019	JAN 2020–DEC 2020
3	Peak day 15 min interval electric energy demand data	Long-term forecasting	JAN 2018–DEC 2019	JAN 2020–DEC 2020
4	Workday 15 min interval electric energy demand data	Long-term forecasting	JAN 2018–DEC 2019	JAN 2020–DEC 2020
5	Peak day 15 min interval electric energy demand data	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
6	Workday 15 min interval electric energy demand data	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7A	Cluster of 15 min interval peak day electric energy demand data—S-RES	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7B	Cluster of 15 min interval peak day electric energy demand data—L-RES	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7C	Cluster of 15 min interval peak day electric energy demand data—SB	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7D	Cluster of 15 min interval peak day electric energy demand data—MB	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7E	Cluster of 15 min interval peak day electric energy demand data—LB	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7F	Cluster of 15 min interval peak day electric energy demand data—SPB	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7G	Cluster of 15 min interval peak day electric energy demand data—NPO	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020
7H	Cluster of 15 min interval peak day electric energy demand data—WPA	Short-term forecasting	JAN 2020–NOV 2020	DEC 2020

Figure 9 is the electric energy pattern data from dataset 1 to dataset 6, where the blue line is the train data for machine learning, and the red line is the test data for machine learning. Therefore, the detail from dataset 1 to dataset 6 can be described as follows:

Dataset 1 is the monthly electric energy demand data that is separated from January 2018 to December 2019 as train data (24 instances) and from January 2020 to December 2020 as test data (12 instances).

Dataset 2 is the monthly electric energy loss data that is separated from January 2018 to December 2019 as train data (24 instances) and from January 2020 to December 2020 as test data (12 instances).

Dataset 3 is the peak day 15 min interval electric energy demand data that is separated from January 2018 to December 2019 as train data (2304 instances) and from January 2020 to December 2020 as test data (1151 instances).

Dataset 4 is the workday 15 min interval electric energy demand data that is separated from January 2018 to December 2019 as train data (2304 instances) and from January 2020 to December 2020 as test data (1151 instances).

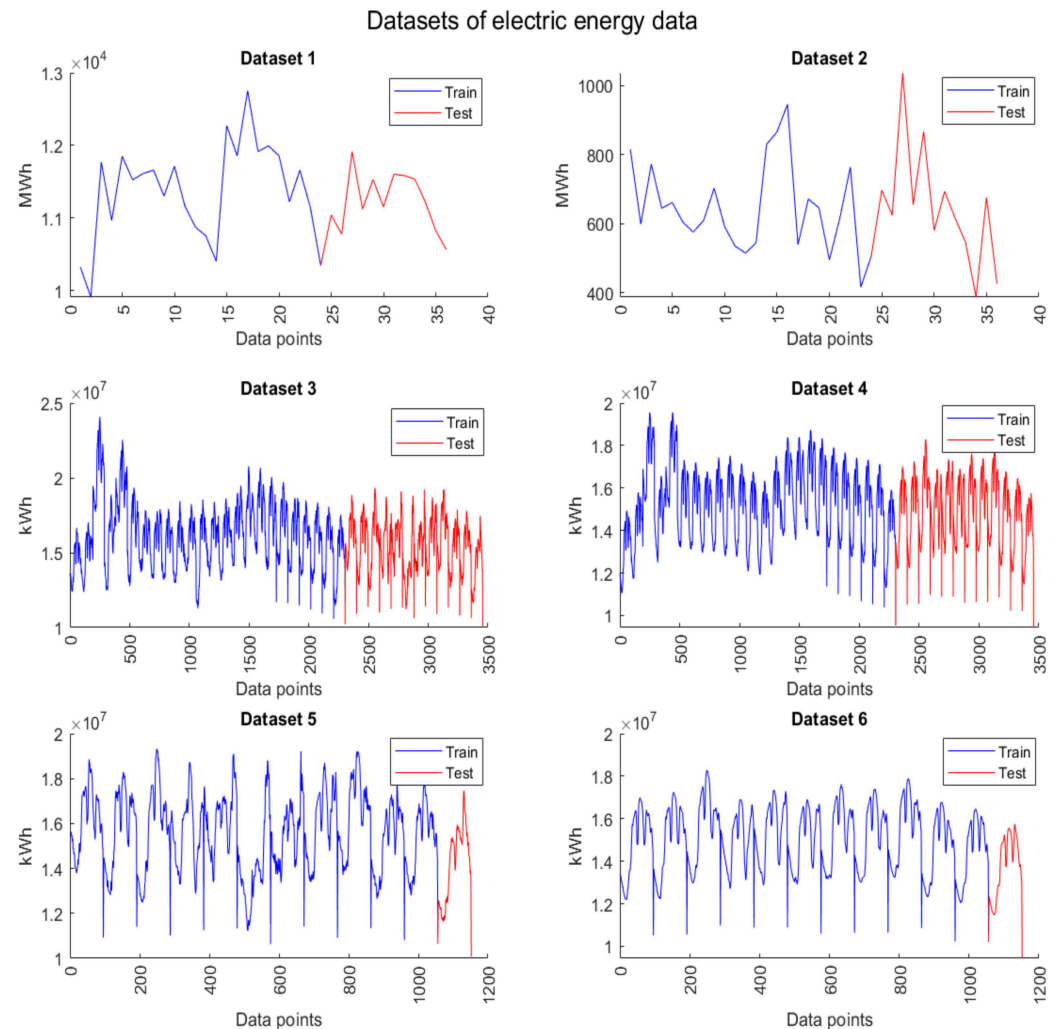
Dataset 5 is the peak day 15 min interval electric energy demand data that is separated from January 2020 to December 2020 as train data (1506 instances) and only December 2020 as test data (96 instances).

Dataset 6 is the workday 15 min interval electric energy demand data that is separated from January 2020 to December 2020 as train data (1506 instances) and only December 2020 as test data (96 instances).

Figure 10 is the electric energy pattern data of dataset 7, where the blue line is the train data for machine learning, and the red line is the test data for machine learning. Therefore, the detail of dataset 7 can be described as follows:

Dataset 7 is the cluster of 15 min interval peak day electric energy demand data that are separated into 8 subcases (7A to 7H). Subcase 7A is the cluster of S-RES electric energy demand data, subcase 7B is the cluster of L-RES electricity demand data, subcase 7C is the

cluster of SB electricity demand data, subcase 7D is the cluster of MB electricity demand data, subcase 7E is the cluster of LB electricity demand data, subcase 7F is the cluster of SPB electricity demand data, subcase 7G is the cluster of NPO electricity demand data, and subcase 7H is the cluster of WPA electricity demand data. All subcases are separated from January 2020 to December 2020 as train data (1506 instances) and only December 2020 as test data (96 instances).



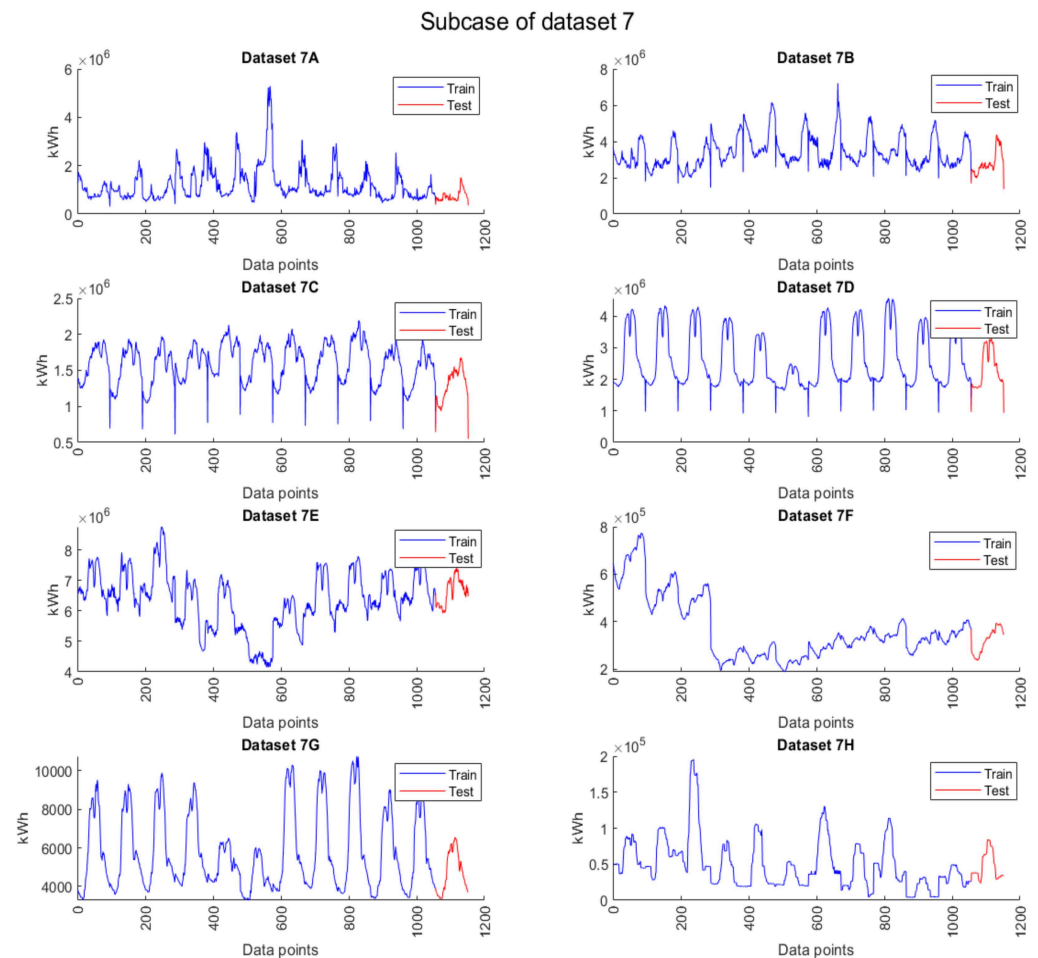
**Figure 9.** Train data (blue line) and test data (red line) of datasets 1 to 6 [40,41].

### 3.3. Proposed Models

The concept of the proposed models is using metaheuristic algorithms to optimize the weight parameter to reduce the cause of overfitting. In this research, three metaheuristic algorithms, which consist of JSO, HHO, and FPA, modified the randomization of weight parameter for searching the best weight parameter to improve the performance of the forecasting. The algorithm of the proposed models is described in Algorithm 1.

First of all, based on the ELM model, all setting parameters were defined in the model, which consisted of a number of hidden nodes, input and target data, and initial weight parameter. Secondly, based on metaheuristic models, all setting parameters of each model were defined, which consisted of the number of populations, switch probability for the FPA model, and the best solution of the population. Thirdly, the weight parameter was defined based on metaheuristic models and then evaluated the new weight parameter for calculating in the ELM model. Fourthly, the activation function was calculated in the ELM model to receive the last weight parameter from the hidden layer. All processes

were calculated until the iteration reached the max iteration criterion and the best weight parameter was received for calculating in the testing phase.



**Figure 10.** Train data (blue line) and test data (red line) of each subcase of dataset 7 [40,41].

### 3.4. Experimental Setup and Hyper-Parameters Setting

The hardware specification of the computer used in this research work was CPU Intel Core i7-7700HQ 2.80 GHz (up to 3.40 GHz), RAM 32 GB, and SSD NVMe. The software was the MATLAB version R2020a. The major hyper-parameters [42,43] of proposed models consisted of the number of hidden nodes and the number of populations and the major hyper-parameters [44] of state-of-the-art models consisting of the number of hidden nodes that were analyzed through the electric energy demand datasets. The objective function to find the best hyper-parameters was the Root Mean Square Error (RMSE).

The hyper-parameter tuning [45,46] of the proposed and state-of-the-art models is shown in Figure 11. In this experiment, the average of RMSE from all models was obtained by taking all the datasets (7 datasets) used to find the RMSE value, then averaging RMSE value in terms of the number of hidden nodes and population. In the proposed models, the boundary of hidden nodes was set from 20 to 200 (increasing 20 nodes), the boundary of populations was set from 20 to 100 (increasing 10 populations), and the number of iterations for metaheuristics was set to 100 due to the fast computation. In the state-of-the-art models, the boundary of hidden nodes was set from 20 to 300 (increasing 20 nodes). For fast computing in the long short-term memory (LSTM) model, the fixed setting of hyper-parameters in the LSTM model consisted of 2 hidden layers, 1000 epochs, 64 batch size, and optimizer was set to Adam.

**Algorithm 1.** Pseudo-code of the Proposed Models.

---

```

1: Define objective function  $f(x)$ ,  $(x_i, t_i) \in \mathbb{R}^n \times \mathbb{R}^m$  by  $i = 1, 2, \dots, N$ 
2: Define the initialize n number of population in metaheuristic models (JSO, HHO, or FPA)
3: Define a switch probability  $p \in [0,1]$  (FPA only)
4: Define the best solution  $g_*$  in the initial population
5: Define L hidden nodes in ELM
6: Define  $G(w_i, b_i, x_i)$  Hidden nodes activation function in ELM (sigmoidal function)
7: Define  $\beta$  output weight vector
8: Define T as the max iteration for metaheuristic models
9: while (t < T)
10:   for i = 1: n
11:     Adjust  $w_i^{t+1}$  as the population with the metaheuristic models (JSO, HHO, or FPA)
12:     Evaluate new solution  $w_i^{t+1}$  to ELM
13:     for i = 1:L
14:       for j = 1:N
15:          $H(i,j) = G(w_i^{t+1}, b_i, x_i) * \beta$ 
16:       end
17:     end
18:      $\beta = H^+T$ 
19:     If new solutions are better, update new  $g_*$  in the population
20:   end for
21: Find the current best solution  $g_*$  & best output weight  $\beta$ 
22: end while

```

---

According to Figure 11, the hyper-parameter experiment results show that the best average RMSE of JS-ELM is 0.0838, and the locations of hidden nodes and populations are 40 and 50, respectively. The best average RMSE of HH-ELM is 0.0927, and the locations of hidden nodes and populations are 40 and 50, respectively. The best average RMSE of FP-ELM is 0.1384, and the locations of hidden nodes and populations are 40 and 50, respectively. The best average RMSE of LSTM is 0.0880, and the location of hidden nodes is 200. The best average RMSE of ELM is 0.0907, and the location of hidden nodes is 200.

For comparing with the metaheuristic-based proposed models, the Particle Swarm Optimization Extreme Learning Machine (PSO-ELM) [47,48] model was considered to forecast in electric energy demand datasets. The best hyper-parameters of PSO-ELM model were received from [48], which consist of 150 hidden nodes, 30 populations (swarm size), acceleration coefficients  $C_1 = 1$  and  $C_2 = 2$ , and inertia weight was set to 0.9.

To summarize, the best hyper-parameters of JS-ELM, HH-ELM, and FP-ELM are 40 hidden nodes and 50 populations. The best hyper-parameters of LSTM and ELM are 200 hidden nodes. All suitable hyper-parameters were defined as presented in Table 3.

### 3.5. Performance Evaluation

In this research, all datasets were processed by min–max normalization [49] as calculated in (25).

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (25)$$

where  $x$  is a data point and  $x'$  is normalized data.

To evaluate the performance of proposed models, three error metrics, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE), were used in this experiment as presented in (26)–(28), respectively.

$$MAE(t) = \frac{1}{n} \sum_{i=1}^n |e_i - t_i| \quad (26)$$

$$MAPE(t) = \frac{100}{n} \sum_{i=1}^n \left| \frac{e_i - t_i}{t_i} \right| \quad (27)$$

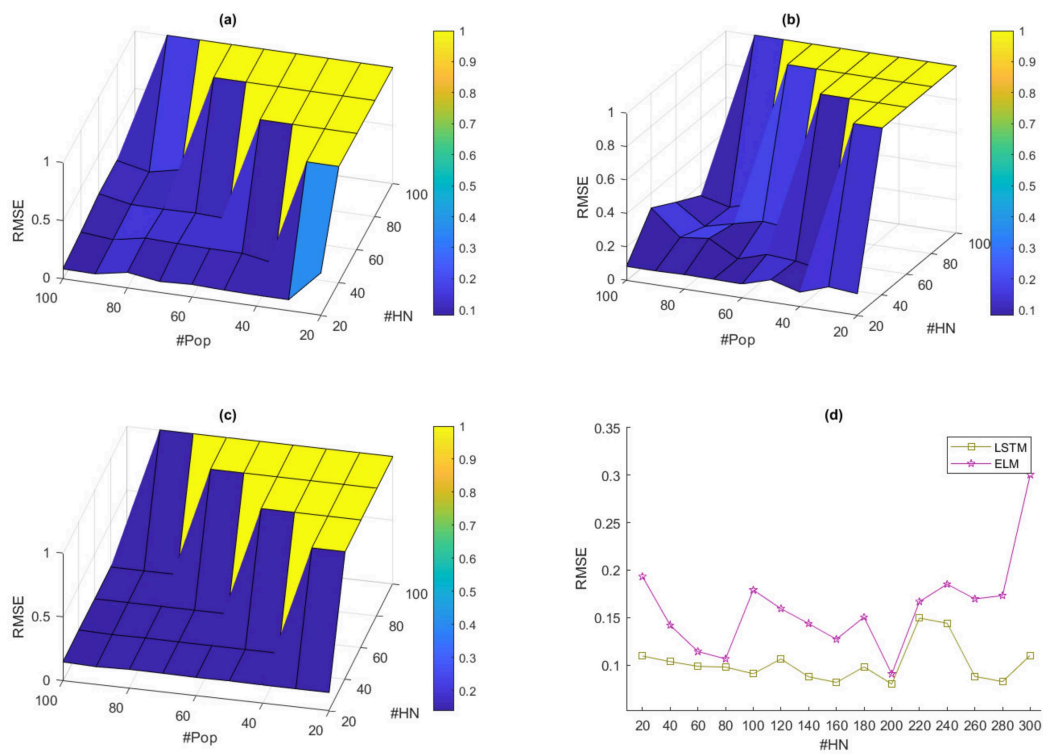
$$RMSE(t) = \sqrt{\frac{\sum_{i=1}^n (e_i - t_i)^2}{n}} \quad (28)$$

where  $i$  is the present time variable,  $n$  is the number of input data,  $e_i$  is the expected data at the time  $i$ , and  $t_i$  is the actual data at the time  $i$ .

To receive the best solution from the performance evaluation, which is suggested from [50,51], all error metrics were combined to the Cumulative Weighted Error (CWE) for evaluating the final result of the experiment as presented in (29).

$$CWE = \frac{MAE + \left(\frac{MAPE}{100}\right) + RMSE}{3} \quad (29)$$

After error metrics were completely calculated, the result of error metrics was used to evaluate the performance of the proposed models as discussed in Section 4.



**Figure 11.** Hyper-parameter tuning of the proposed and state-of-the-art models where #pop is the number of populations and #HN is the number of hidden nodes. (a) The average RMSE of JS-ELM. (b) The average RMSE of HH-ELM. (c) The average RMSE of FP-ELM. (d) The average RMSE of LSTM and ELM.

**Table 3.** Initial Parameters of All Models.

Models	Number of Hidden Nodes	Number of Populations	Number of Iterations for Metaheuristic Optimization	Number of Global Iterations for Machine Learning Model
JS-ELM	40	50	100	20
HH-ELM	40	50	100	20
FP-ELM	40	50	100	20
LSTM	200	-	-	20
PSO-ELM	150	30	100	20
ELM	200	-	-	20

#### 4. Experimental Results

This section describes the experimental results of the proposed models and ELM using electric energy demand datasets. This section is separated into three subsections consisting of the experimental results overview, forecasting time consumption for each model, and the summary of the experimental results.

##### 4.1. Discussion of the Experimental Results

In the experimental results, the datasets mentioned in Section 3.2 were set. The forecasting error of the proposed models was presented and all error metrics as mentioned in Section 3.5 were compared. According to Table 4, the detail of the experimental result is described in terms of the dataset with the minimum error, minimum mean error, and standard deviation (S.D.) error result when using the number of global iterations defined in Table 3.

**Table 4.** Experimental results of all models (bold text is the best error result).

Datasets	Forecasting Error Metric		Proposed Models			State-of-the-Art Models		
			JS-ELM	HH-ELM	FP-ELM	LSTM	PSO-ELM	ELM
1	MAE	$E_*$	<b>0.1122</b>	0.1491	0.2010	0.1868	0.2277	0.1571
		$\mu$	0.1829	0.2268	0.3039	0.2326	0.2683	<b>0.1740</b>
		$\sigma$	0.0390	0.0387	0.0455	0.0224	0.0277	<b>0.0115</b>
	MAPE	$E_*$	<b>2.4418</b>	2.6958	4.7795	3.2880	2.6200	3.5355
		$\mu$	3.8853	3.9704	7.9162	<b>3.5178</b>	3.4023	4.7433
		$\sigma$	1.6551	0.7713	2.2491	<b>0.1998</b>	0.4985	0.7377
	RMSE	$E_*$	<b>0.1717</b>	0.2026	0.2761	0.2307	0.2729	0.2000
		$\mu$	0.2255	0.2712	0.3733	0.2776	0.3181	<b>0.2184</b>
		$\sigma$	0.0408	0.0426	0.0494	0.0237	0.0310	<b>0.0106</b>
	CWE	$E_*$	<b>0.1028</b>	0.1262	0.1750	0.1501	0.1756	0.1308
		$\mu$	0.1491	0.1792	0.2521	0.1818	0.2068	<b>0.1466</b>
		$\sigma$	0.0321	0.0297	0.0391	0.0160	0.0212	<b>0.0098</b>
	MAE	$E_*$	<b>0.1662</b>	0.1845	0.1866	0.3406	0.3014	0.3075
		$\mu$	0.2803	0.2826	<b>0.2657</b>	0.4095	0.3366	0.3192
		$\sigma$	0.0448	0.0399	0.0510	0.0299	0.0308	<b>0.0077</b>
2	MAPE	$E_*$	<b>16.8109</b>	21.8086	20.0015	24.3841	39.6875	17.0482
		$\mu$	101.3614	87.9551	110.2814	32.0613	324.3600	<b>19.8431</b>
		$\sigma$	237.7910	114.8906	167.0535	4.2338	366.7532	<b>2.7022</b>
	RMSE	$E_*$	<b>0.1918</b>	0.2513	0.2323	0.4140	0.3291	0.3472
		$\mu$	0.3361	0.3346	<b>0.3291</b>	0.4852	0.4008	0.3592
		$\sigma$	0.0456	0.0376	0.0600	0.0329	0.0358	<b>0.0064</b>
	CWE	$E_*$	<b>0.1754</b>	0.2179	0.2063	0.3328	0.3424	0.2751
		$\mu$	0.5433	0.4989	0.5659	0.4051	1.3269	<b>0.2923</b>
		$\sigma$	0.8228	0.4088	0.5938	0.0350	1.2447	<b>0.0137</b>
	MAE	$E_*$	0.0493	0.0496	0.0370	<b>0.0356</b>	0.0441	0.1371
		$\mu$	0.0854	0.0922	<b>0.0524</b>	0.0720	0.0553	0.1644
		$\sigma$	0.0298	0.0332	0.0092	0.0359	0.0532	<b>0.0263</b>
	MAPE	$E_*$	2.0761	2.2084	2.2287	<b>2.0124</b>	2.2983	6.7634
		$\mu$	<b>2.2080</b>	2.3103	2.2734	2.2679	2.2789	12.5276
		$\sigma$	0.1040	0.0610	<b>0.0811</b>	0.1488	0.0542	6.5947
3	RMSE	$E_*$	0.0832	0.0843	0.0784	<b>0.0688</b>	0.0818	0.1668
		$\mu$	0.1087	0.1141	<b>0.0862</b>	0.0985	0.0945	0.1996
		$\sigma$	0.0225	0.0251	<b>0.0063</b>	0.0301	0.0212	0.0289
	CWE	$E_*$	0.0511	0.0520	0.0459	<b>0.0415</b>	0.0499	0.1238
		$\mu$	0.0721	0.0765	<b>0.0538</b>	0.0644	0.0678	0.1631
		$\sigma$	0.0178	0.0196	<b>0.0054</b>	0.0225	0.0031	0.0404
	MAE	$E_*$	0.0276	0.0262	0.0259	<b>0.0180</b>	0.0298	0.1221
		$\mu$	0.0579	0.0519	<b>0.0294</b>	0.0725	0.0578	0.1555
		$\sigma$	0.0176	0.0220	<b>0.0047</b>	0.0302	0.0019	0.0323

Table 4. Cont.

Datasets	Forecasting Error Metric		Proposed Models			State-of-the-Art Models		
			JS-ELM	HH-ELM	FP-ELM	LSTM	PSO-ELM	ELM
5	MAPE	$E_*$	2.1786	2.2010	2.3628	<b>0.8040</b>	2.4519	7.3798
		$\mu$	2.2888	2.3608	2.4075	<b>0.9159</b>	2.4599	15.2733
		$\sigma$	0.0945	0.0773	<b>0.0391</b>	0.0711	0.9825	4.8535
	RMSE	$E_*$	0.0639	0.0668	0.0674	<b>0.0503</b>	0.0698	0.1504
		$\mu$	0.0830	0.0812	<b>0.0692</b>	0.0900	0.0782	0.1861
		$\sigma$	0.0120	0.0132	<b>0.0019</b>	0.0271	0.0429	0.0349
	CWE	$E_*$	0.0377	0.0383	0.0390	<b>0.0254</b>	0.0399	0.1154
		$\mu$	0.0546	0.0522	<b>0.0409</b>	0.0572	0.0569	0.1648
		$\sigma$	0.0102	0.0120	<b>0.0023</b>	0.0193	0.0341	0.0386
	MAE	$E_*$	<b>0.0464</b>	0.0596	0.0994	0.0807	0.1078	0.1509
		$\mu$	0.1444	<b>0.1044</b>	0.1103	0.1377	0.1985	0.1955
		$\sigma$	0.0928	0.0234	<b>0.0057</b>	0.0372	0.0231	0.0321
	MAPE	$E_*$	2.6149	2.7385	2.7791	<b>1.9792</b>	2.8974	7.8708
		$\mu$	7.1240	2.9822	<b>2.8590</b>	3.7442	2.9987	15.5761
		$\sigma$	11.7593	0.1493	<b>0.0493</b>	0.8836	0.1293	6.4519
	RMSE	$E_*$	<b>0.0706</b>	0.0896	0.1221	0.1308	0.1298	0.2092
		$\mu$	0.1792	<b>0.1235</b>	0.1300	0.1793	0.1984	0.2460
		$\sigma$	0.0924	0.0184	<b>0.0043</b>	0.0313	0.0231	0.0308
	CWE	$E_*$	<b>0.0477</b>	0.0589	0.0831	0.0771	0.0868	0.1463
		$\mu$	0.1316	<b>0.0859</b>	0.0896	0.1182	0.0981	0.1991
		$\sigma$	0.1009	0.0144	<b>0.0035</b>	0.0258	0.0412	0.0425
	MAE	$E_*$	<b>0.0838</b>	0.0828	0.1194	0.1356	0.1427	0.2016
		$\mu$	0.1451	<b>0.1352</b>	0.1383	0.1538	0.1549	0.2878
		$\sigma$	0.0317	0.0189	<b>0.0064</b>	0.0125	0.0542	0.0669
6	MAPE	$E_*$	2.7064	2.6429	2.7679	<b>0.8133</b>	2.9145	9.4805
		$\mu$	3.0684	2.9169	2.8812	<b>1.0700</b>	2.8992	48.6885
		$\sigma$	0.3358	0.2019	<b>0.0497</b>	0.1153	0.0341	57.6711
	RMSE	$E_*$	<b>0.1068</b>	0.1139	0.1648	0.1767	0.1809	0.2347
		$\mu$	0.1712	<b>0.1631</b>	0.1723	0.1885	0.1874	0.3268
		$\sigma$	0.0286	0.0173	<b>0.0036</b>	0.0087	0.0241	0.0692
	CWE	$E_*$	<b>0.0726</b>	0.0744	0.1040	0.1068	0.1175	0.1770
		$\mu$	0.1157	<b>0.1092</b>	0.1131	0.1177	0.1237	0.3671
		$\sigma$	0.0212	0.0127	<b>0.0035</b>	0.0075	0.0412	0.2376
	MAE	$E_*$	0.0820	0.0772	<b>0.0741</b>	0.1178	0.0863	0.1682
		$\mu$	0.1153	0.1218	<b>0.1005</b>	0.1652	0.1562	0.2885
		$\sigma$	0.0249	0.0177	<b>0.0146</b>	0.0339	0.0369	0.0791
	MAPE	$E_*$	<b>10.3068</b>	10.5220	10.5318	17.3387	15.1356	159.3181
		$\mu$	12.4254	11.8836	<b>11.1192</b>	55.7781	15.3368	921.8174
		$\sigma$	1.8623	0.9537	<b>0.3495</b>	66.0648	0.8156	1800.9530
	RMSE	$E_*$	0.1115	0.1052	<b>0.1022</b>	0.1691	0.1934	0.2211
		$\mu$	0.1427	0.1471	<b>0.1260</b>	0.2181	0.1996	0.3286
		$\sigma$	0.0225	0.0172	<b>0.0133</b>	0.0380	0.058	0.0755
	CWE	$E_*$	0.0989	0.0959	<b>0.0939</b>	0.1534	0.1867	0.6608
		$\mu$	0.1274	0.1292	<b>0.1126</b>	0.3137	0.1923	3.2784
		$\sigma$	0.0220	0.0148	<b>0.0105</b>	0.2442	0.0265	6.0547
	MAE	$E_*$	0.0704	0.1335	0.1179	<b>0.0571</b>	0.1394	0.1583
		$\mu$	0.1643	0.1804	0.3152	<b>0.1337</b>	0.1557	0.2125
		$\sigma$	0.0577	0.0170	0.0943	<b>0.0381</b>	0.0878	0.0498
7B	MAPE	$E_*$	10.6971	10.3719	9.0110	<b>4.1967</b>	9.1245	29.5542
		$\mu$	11.4928	11.6090	13.0535	<b>6.4049</b>	9.6878	77.1914
		$\sigma$	0.7465	<b>0.6466</b>	2.4811	1.4669	5.7795	46.7933
	RMSE	$E_*$	<b>0.0964</b>	0.1525	0.1368	0.1027	0.1475	0.2282
		$\mu$	0.1830	0.2001	0.3414	<b>0.1594</b>	0.1969	0.2732
		$\sigma$	0.0579	0.0159	0.1032	<b>0.0294</b>	0.0677	0.0451
	CWE	$E_*$	0.0912	0.1299	0.1149	<b>0.0673</b>	0.1264	0.2273
		$\mu$	0.1541	0.1655	0.2624	<b>0.1191</b>	0.1377	0.4192
		$\sigma$	0.0410	0.0131	0.0741	<b>0.0274</b>	0.0547	0.1876

Table 4. Cont.

Datasets	Forecasting Error Metric		Proposed Models			State-of-the-Art Models		
			JS-ELM	HH-ELM	FP-ELM	LSTM	PSO-ELM	ELM
7C	MAE	$E_*$	<b>0.0614</b>	0.1533	0.1706	0.0795	0.1677	0.1620
		$\mu$	<b>0.1514</b>	0.1892	0.1811	0.1548	0.1814	0.3231
		$\sigma$	0.0345	0.0520	<b>0.0058</b>	0.0281	0.0611	0.1019
	MAPE	$E_*$	4.0018	4.3382	4.2618	<b>3.2813</b>	4.4435	20.5478
		$\mu$	4.6888	4.6820	<b>4.4270</b>	7.9141	4.6989	59.7436
		$\sigma$	0.8142	0.2128	<b>0.0764</b>	12.3156	0.0896	62.1757
	RMSE	$E_*$	<b>0.1025</b>	0.1810	0.1910	0.1308	0.1973	0.2197
		$\mu$	<b>0.1825</b>	0.2136	0.2042	0.1924	0.2137	0.3597
		$\sigma$	0.0274	0.0467	<b>0.0064</b>	0.0243	0.0631	0.0983
	CWE	$E_*$	<b>0.0680</b>	0.1259	0.1347	0.0810	0.1361	0.1957
		$\mu$	<b>0.1269</b>	0.1499	0.1432	0.1421	0.1468	0.4268
		$\sigma$	0.0233	0.0336	<b>0.0043</b>	0.0585	0.0435	0.2740
7D	MAE	$E_*$	0.0672	0.0389	0.0317	<b>0.0210</b>	0.0379	0.2040
		$\mu$	0.1192	0.1007	<b>0.0423</b>	0.0623	0.0576	0.2310
		$\sigma$	0.0323	0.0499	<b>0.0157</b>	0.0611	0.0193	0.0241
	MAPE	$E_*$	3.5799	3.4621	3.7425	<b>1.8334</b>	3.7989	23.2984
		$\mu$	3.9516	4.4777	3.9223	<b>2.3698</b>	3.9155	42.9784
		$\sigma$	0.2343	1.3263	<b>0.1612</b>	0.4682	0.1789	19.6851
	RMSE	$E_*$	0.0897	0.0712	0.0676	<b>0.0285</b>	0.0797	0.2336
		$\mu$	0.1368	0.1191	0.0734	<b>0.0726</b>	0.0895	0.2838
		$\sigma$	0.0333	0.0438	<b>0.0126</b>	0.0611	0.0567	0.0356
	CWE	$E_*$	0.0642	0.0482	0.0456	<b>0.0226</b>	0.0489	0.2235
		$\mu$	0.0985	0.0882	<b>0.0516</b>	0.0528	0.0597	0.3149
		$\sigma$	0.0227	0.0357	<b>0.0100</b>	0.0423	0.0935	0.0855
7E	MAE	$E_*$	0.0612	0.0612	0.0608	<b>0.0591</b>	0.0688	0.2469
		$\mu$	0.0647	0.0642	<b>0.0634</b>	0.0769	0.0712	0.2797
		$\sigma$	0.0035	0.0021	<b>0.0037</b>	0.0119	0.0121	0.0255
	MAPE	$E_*$	1.3713	<b>1.3617</b>	1.3800	1.4003	1.3966	7.4999
		$\mu$	1.4018	<b>1.3958</b>	1.4350	1.7621	1.4579	17.5561
		$\sigma$	<b>0.0176</b>	0.0217	0.0767	0.2355	0.2899	23.2809
	RMSE	$E_*$	0.0818	0.0818	0.0817	<b>0.0756</b>	0.0799	0.2915
		$\mu$	0.0859	0.0853	<b>0.0836</b>	0.1019	0.0897	0.3304
		$\sigma$	0.0035	<b>0.0019</b>	0.0031	0.0163	0.0048	0.0291
	CWE	$E_*$	0.0523	0.0522	0.0521	<b>0.0496</b>	0.0587	0.2045
		$\mu$	0.0549	0.0545	<b>0.0538</b>	0.0655	0.0574	0.2619
		$\sigma$	0.0024	<b>0.0014</b>	0.0025	0.0102	0.0040	0.0958
7F	MAE	$E_*$	0.0370	0.0364	0.0351	<b>0.0298</b>	0.0389	0.2810
		$\mu$	0.0445	0.0410	0.0384	<b>0.0368</b>	0.0415	0.3115
		$\sigma$	0.0049	0.0050	<b>0.0016</b>	0.0055	0.0063	0.0290
	MAPE	$E_*$	1.5979	1.518	1.4372	<b>1.3786</b>	1.4889	23.3061
		$\mu$	1.7345	1.6883	1.5431	<b>1.4887</b>	1.7562	34.2174
		$\sigma$	0.1305	0.1385	<b>0.0765</b>	0.1271	0.1015	18.2012
	RMSE	$E_*$	<b>0.0657</b>	0.0664	0.0659	0.0659	0.0678	0.3297
		$\mu$	0.0718	0.0691	<b>0.0675</b>	0.0686	0.0782	0.3637
		$\sigma$	0.0032	0.0030	<b>0.0009</b>	0.0030	0.0087	0.0312
	CWE	$E_*$	0.0396	0.0393	0.0384	<b>0.0365</b>	0.0399	0.2813
		$\mu$	0.0445	0.0423	<b>0.0404</b>	0.0401	0.0484	0.3391
		$\sigma$	0.0032	0.0031	<b>0.0011</b>	0.0033	0.0087	0.0807
7G	MAE	$E_*$	0.0285	0.0281	<b>0.0276</b>	0.0298	0.0379	0.1073
		$\mu$	0.0300	<b>0.0296</b>	0.0297	0.0621	0.0485	0.1729
		$\sigma$	<b>0.0008</b>	0.0009	0.0015	0.0427	0.0042	0.0370
	MAPE	$E_*$	1.8554	1.8418	<b>1.7774</b>	1.9445	1.7884	13.4419
		$\mu$	1.8964	1.8908	<b>1.8846</b>	4.0213	1.8993	14.7832
		$\sigma$	<b>0.0239</b>	0.0268	0.0934	5.0059	0.0367	0.6157
	RMSE	$E_*$	0.0380	<b>0.0379</b>	0.0380	0.0461	0.0388	0.1363
		$\mu$	0.0395	<b>0.0393</b>	0.0401	0.0827	0.0399	0.2280
		$\sigma$	0.0008	<b>0.0007</b>	0.0023	0.0558	0.0010	0.0538

Table 4. Cont.

Datasets	Forecasting Error Metric		Proposed Models			State-of-the-Art Models		
			JS-ELM	HH-ELM	FP-ELM	LSTM	PSO-ELM	ELM
7H	CWE	$E_*$	0.0284	0.0281	<b>0.0278</b>	0.0318	0.0288	0.1260
		$\mu$	0.0295	<b>0.0293</b>	0.0296	0.0617	0.0298	0.1829
		$\sigma$	<b>0.0005</b>	0.0006	0.0016	0.0495	0.0007	0.0323
	MAE	$E_*$	0.0296	0.0298	<b>0.0293</b>	0.0482	0.0311	0.2217
		$\mu$	0.0352	0.0348	<b>0.0327</b>	0.0934	0.0324	0.2857
		$\sigma$	0.0033	0.0029	<b>0.0021</b>	0.0443	0.0115	0.0465
	MAPE	$E_*$	<b>3.7510</b>	4.0816	3.9517	6.1304	3.8854	53.3827
		$\mu$	4.4249	4.3978	<b>4.2475</b>	12.0408	4.5691	153.4755
		$\sigma$	0.3165	0.1781	<b>0.1753</b>	8.1731	0.5362	107.6369
	RMSE	$E_*$	0.0491	0.0498	<b>0.0476</b>	0.0684	0.0562	0.2875
		$\mu$	0.0550	0.0534	<b>0.0513</b>	0.1252	0.0556	0.3252
		$\sigma$	0.0032	0.0023	<b>0.0022</b>	0.0535	0.0135	0.0448
	CWE	$E_*$	<b>0.0387</b>	0.0401	0.0388	0.0593	0.0398	0.3477
		$\mu$	0.0448	0.0441	<b>0.0421</b>	0.1130	0.0455	0.7152
		$\sigma$	0.0032	0.0023	<b>0.0020</b>	0.0599	0.0068	0.3892

The minimum error was obtained by finding the lowest forecasting error result from the global iterations as presented in (30). The minimum mean error was obtained by averaging all results from the global iterations as presented in (31). The minimum S.D. error was obtained by using the forecasting error result and the minimum mean error result to indicate the standard deviation of the error solution from global iterations as presented in (32).

$$E_* = \min_i \{E_i\}_{i=1}^N \quad (30)$$

$$\mu = \frac{1}{N} \sum_{i=1}^N E_i \quad (31)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N |E_i - \mu|^2} \quad (32)$$

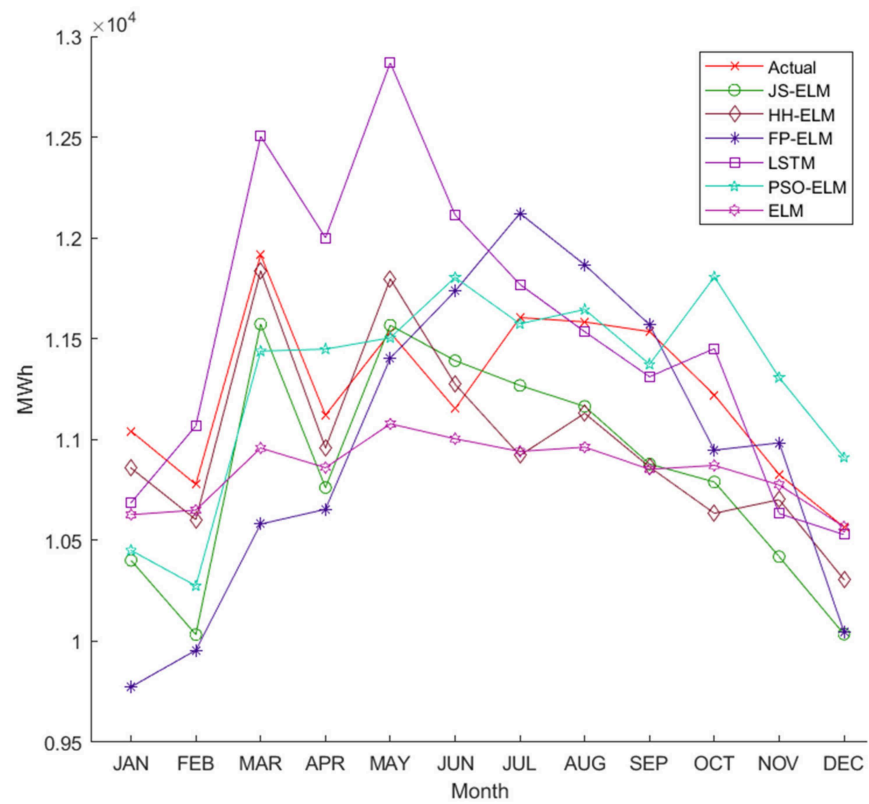
where  $E_*$  is the minimum error,  $\mu$  is the minimum mean error,  $\sigma$  is the minimum S.D. error,  $i$  is the present time variable,  $N$  is the number of global iterations, and  $E_i$  is the result of the forecasting error metric for each global iteration.

The minimum forecasting error result of dataset 1 is shown in Figure 12. The result shows that all models provided results near the actual target value. Furthermore, according to the CWE error metric in Table 4, JS-ELM provided the best minimum error result when compared to the other models' results. However, LSTM, PSO-ELM, and ELM provided higher stability of forecasting when considered on the minimum mean error and minimum S.D. error. HH-ELM provided the minimum error lower than that of the ELM model, but the minimum mean error was higher than that of the ELM model. The overall error metrics of FP-ELM were higher than those of other models.

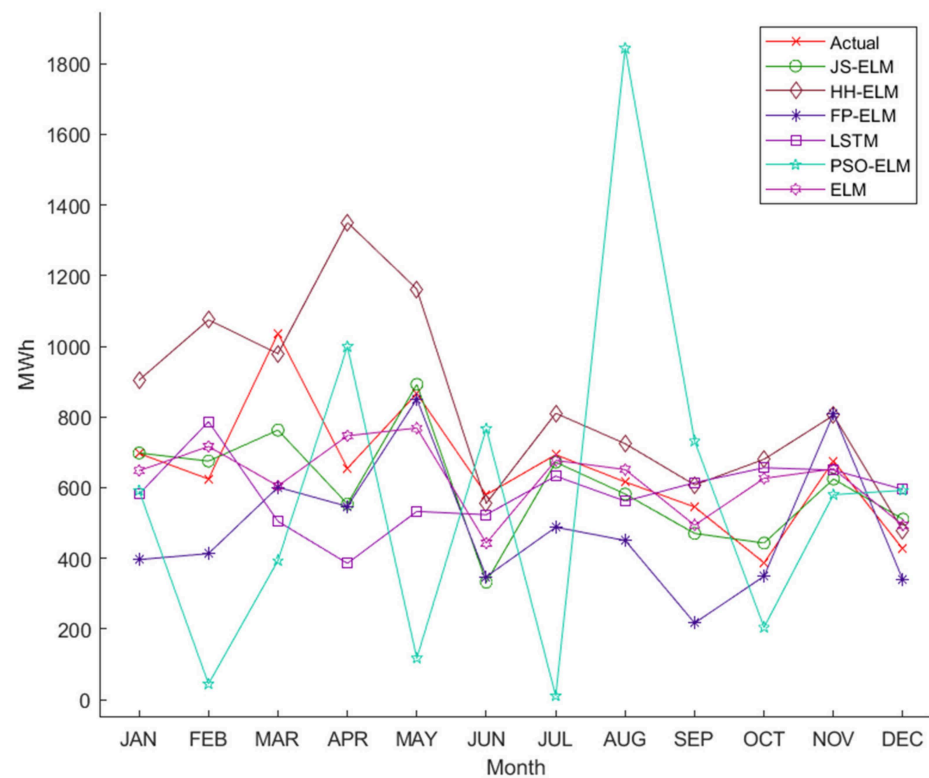
The minimum forecasting error result of dataset 2 is shown in Figure 13. The result shows that all models provided results near the actual target value. Furthermore, according to the CWE error metric in Table 4, JS-ELM provided the best minimum error result when compared to the other models' results. However, the minimum mean error and minimum standard deviation of JS-ELM, HH-ELM, FP-ELM, and PSO-ELM were higher than those of the LSTM and ELM models, which means the proposed models could provide the instability of the attempt to dataset 2.

The minimum forecasting error result of dataset 3 is shown in Figure 14. All proposed models and LSTM provided results near the actual target value. The ELM model was already overfitted because the training RMSE result was 0.0207, but the forecasting RMSE result was 0.1668. Furthermore, the minimum mean error and minimum standard deviation

of ELM were higher than those of the proposed models, which means the ELM could provide the instability of the attempt to dataset 3.

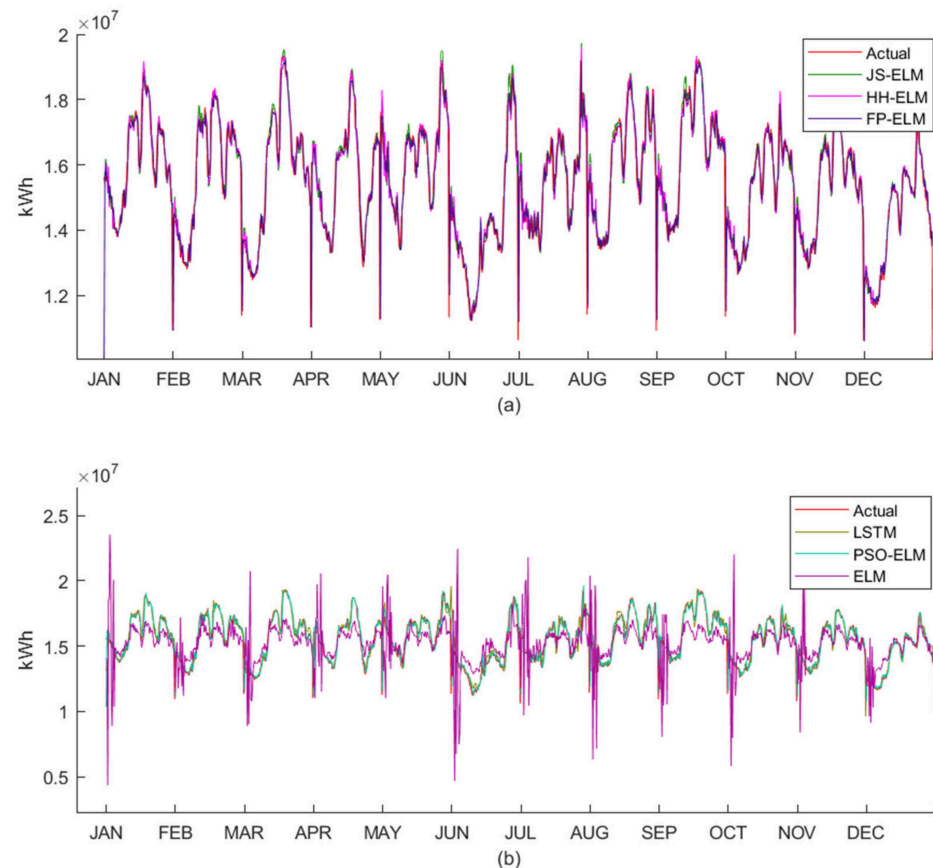


**Figure 12.** Forecasting result of dataset 1 (minimum error result).



**Figure 13.** Forecasting result of dataset 2 (minimum error result).

The minimum forecasting error result of dataset 4 is shown in Figure 15. All proposed models and LSTM provided results near the actual target value. The ELM model was already overfitted because the training RMSE result was 0.0253, but the forecasting RMSE result was 0.1504. Furthermore, the minimum mean error and minimum standard deviation of ELM were higher than those of proposed models, which means the ELM could provide the instability of the attempt to dataset 4.



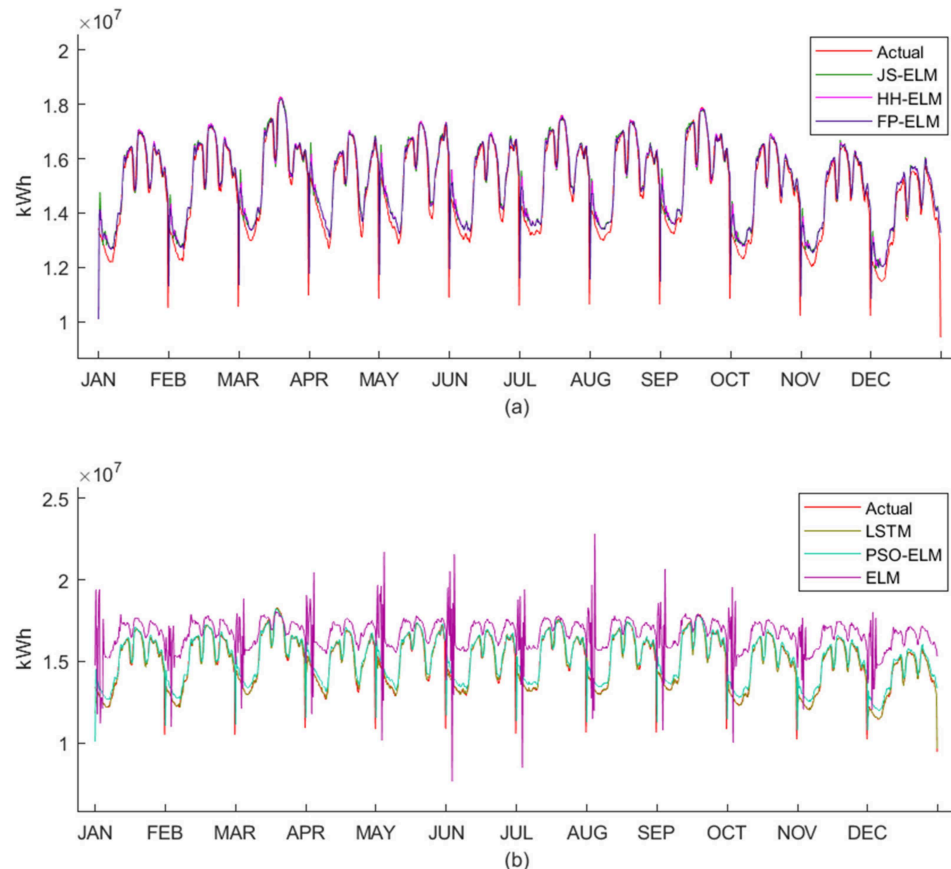
**Figure 14.** Forecasting result of dataset 3 (minimum error result). (a) Forecasted energy of each proposed model. (b) Forecasted energy of each state-of-the-art model.

The minimum forecasting error result of dataset 5 is shown in Figure 16. The result shows that the expected target value of JS-ELM was close to the actual target value. The expected target values of HH-ELM and FP-ELM slightly swung from 02:00 to 04:00 when compared to the actual target value. The expected target values of LSTM slightly swung from 00:00 to 01:00 when compared to the actual target value. The expected target value of ELM was clearly different from the actual target value.

The minimum forecasting error result of dataset 6 is shown in Figure 17. The result shows that all models provided results near the actual target value. However, the ELM model was already overfitted because the training RMSE result was 0.0151, but the forecasting RMSE result was 0.2347. Furthermore, the minimum mean error and minimum standard deviation of ELM were higher than those of the proposed models, which means ELM could provide the instability of the attempt to dataset 6.

The minimum forecasting error results of dataset 7 are shown in Figures 18–20. Dataset 7 contained eight sub-datasets to attempt on this experiment, which consists of sub-dataset 7A (the S-RES profile), sub-dataset 7B (the L-RES profile), sub-dataset 7C (the SB profile), sub-dataset 7D (the MB profile), sub-dataset 7E (the LB profile), sub-dataset 7F (the SPB profile), sub-dataset 7G (the NPO profile), and sub-dataset 7H (the WPA profile). The overall results of dataset 7 show that the proposed models provided the results

nearest to the actual target value. However, the expected target value of the ELM model was instability, which means the ELM model was already overfitted because the training accuracy between the expected target value and actual target value was overrated, thus, the forecasting result was unacceptable. Furthermore, the minimum mean error and minimum standard deviation of ELM were higher than those of the proposed models, which means the ELM could provide the instability of the attempt to dataset 7.



**Figure 15.** Forecasting result of dataset 4 (minimum error result). (a) Forecasted energy of each proposed model. (b) Forecasted energy of each state-of-the-art model.

#### 4.2. Discussion on the Time Consumption

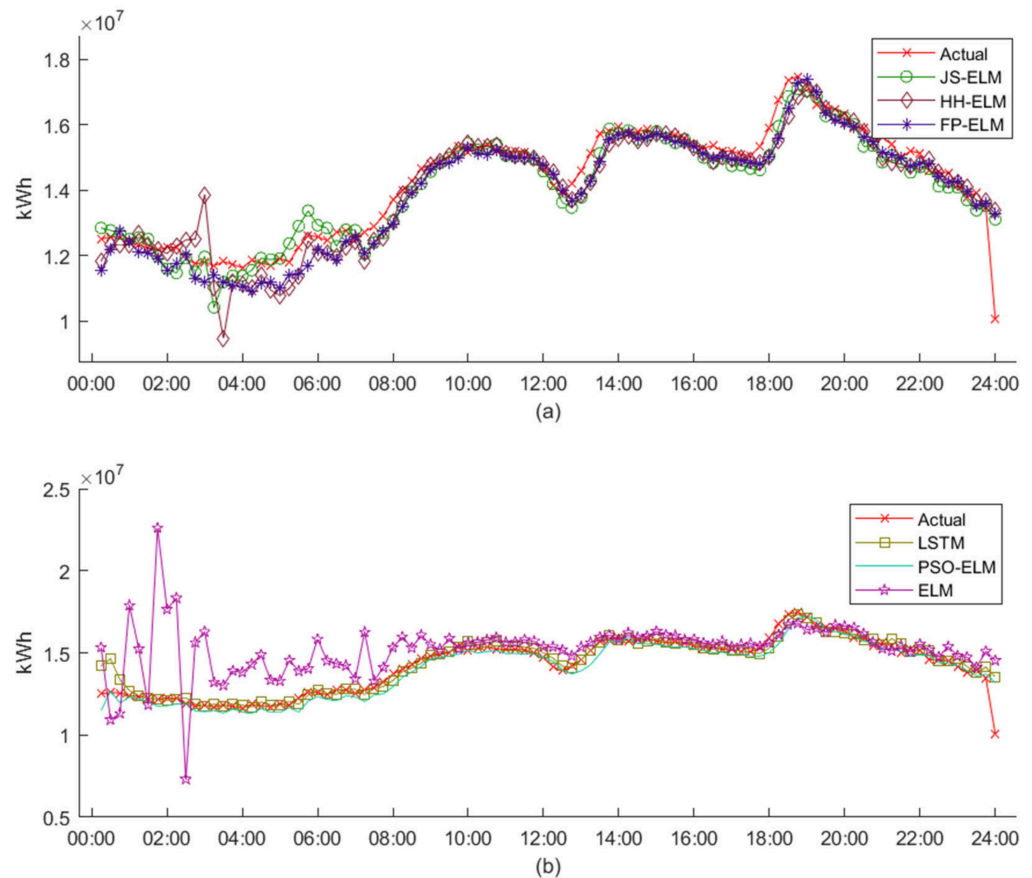
In this research, the training times of each forecasting model were calculated and compared. According to Table 5, the result of the training time showed that ELM consumed the least time compared with other models. Due to the metaheuristic calculation, JS-ELM and FP-ELM consumed a lower amount of time to calculate the data and provide the best result. HH-ELM consumed more time than the other models because the number of time steps of the HHO calculation are greater than JSO and FPA. However, the LSTM model consumed more time than all other models in the experiment. The forecasting time was not focused upon because all models consumed a low forecasting time (in milliseconds).

#### 4.3. Summary of the Experimental Results

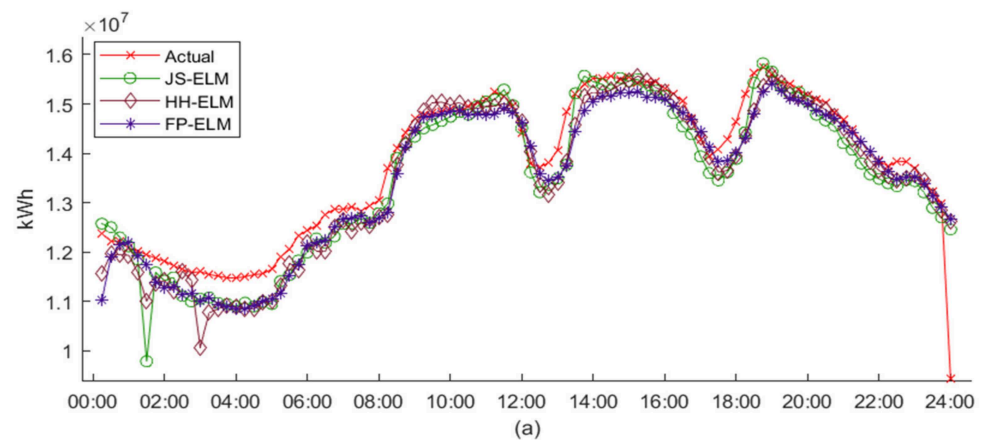
In this research, three main factors, the minimum error, minimum mean error, and minimum standard deviation, were analyzed by error metrics (MAE, MAPE, RMSE, and CWE) to evaluate the performance of all models that fitted each dataset.

The CWE comparison for all models from datasets 1 to 6 is shown in Figure 21. For datasets 1 and 2, the JS-ELM model was the most suitable in terms of the minimum error result, while the ELM was the most suitable in terms of the minimum mean error and S.D. error. For dataset 3, the FP-ELM model was suitable for all three factors. For dataset 4,

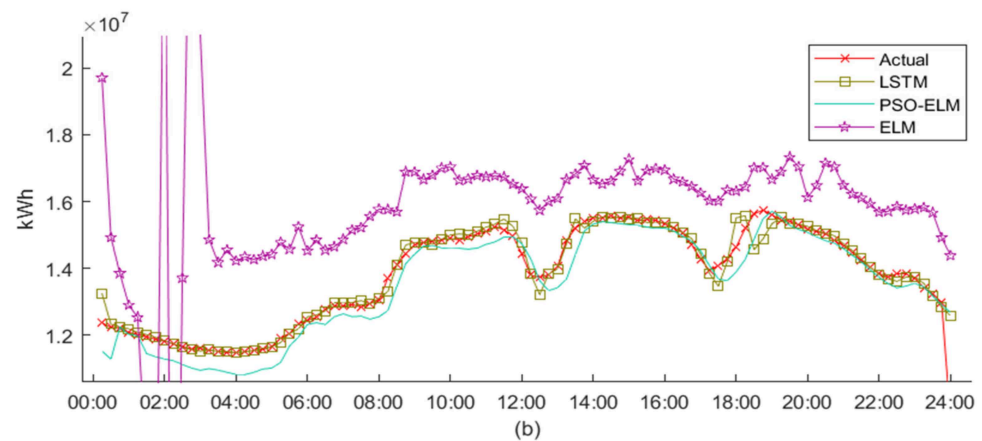
the LSTM model was the most suitable in terms of the minimum error result, while the FP-ELM was the most suitable in terms of the minimum mean error and S.D. error. For datasets 5 and 6, the JS-ELM model was the most suitable in terms of the minimum error result, while HH-ELM and FP-ELM were the most suitable in terms of the minimum mean error and S.D. error, respectively.



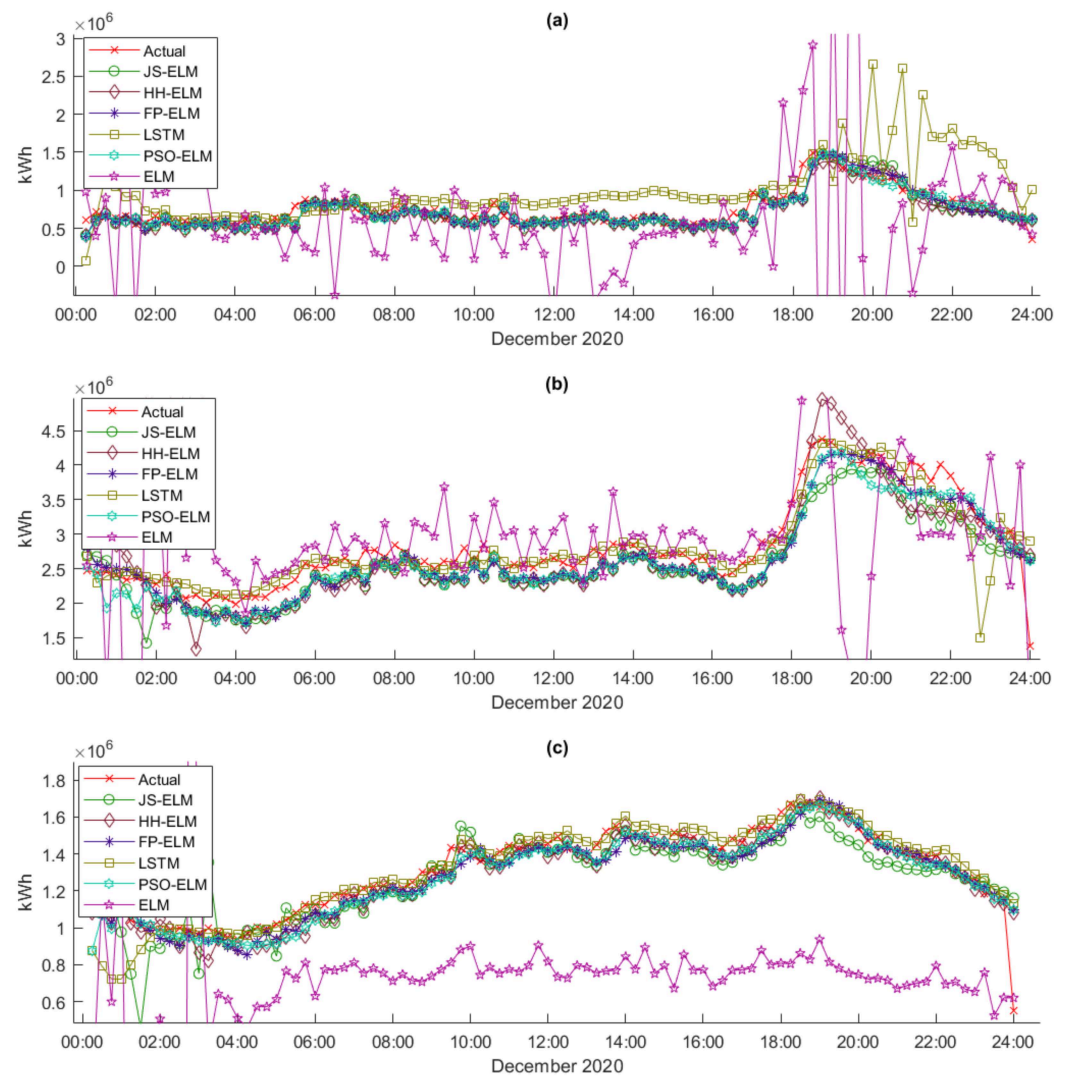
**Figure 16.** Forecasting result of dataset 5 (minimum error result). (a) Forecasted energy of each proposed model. (b) Forecasted energy of each state-of-the-art model.



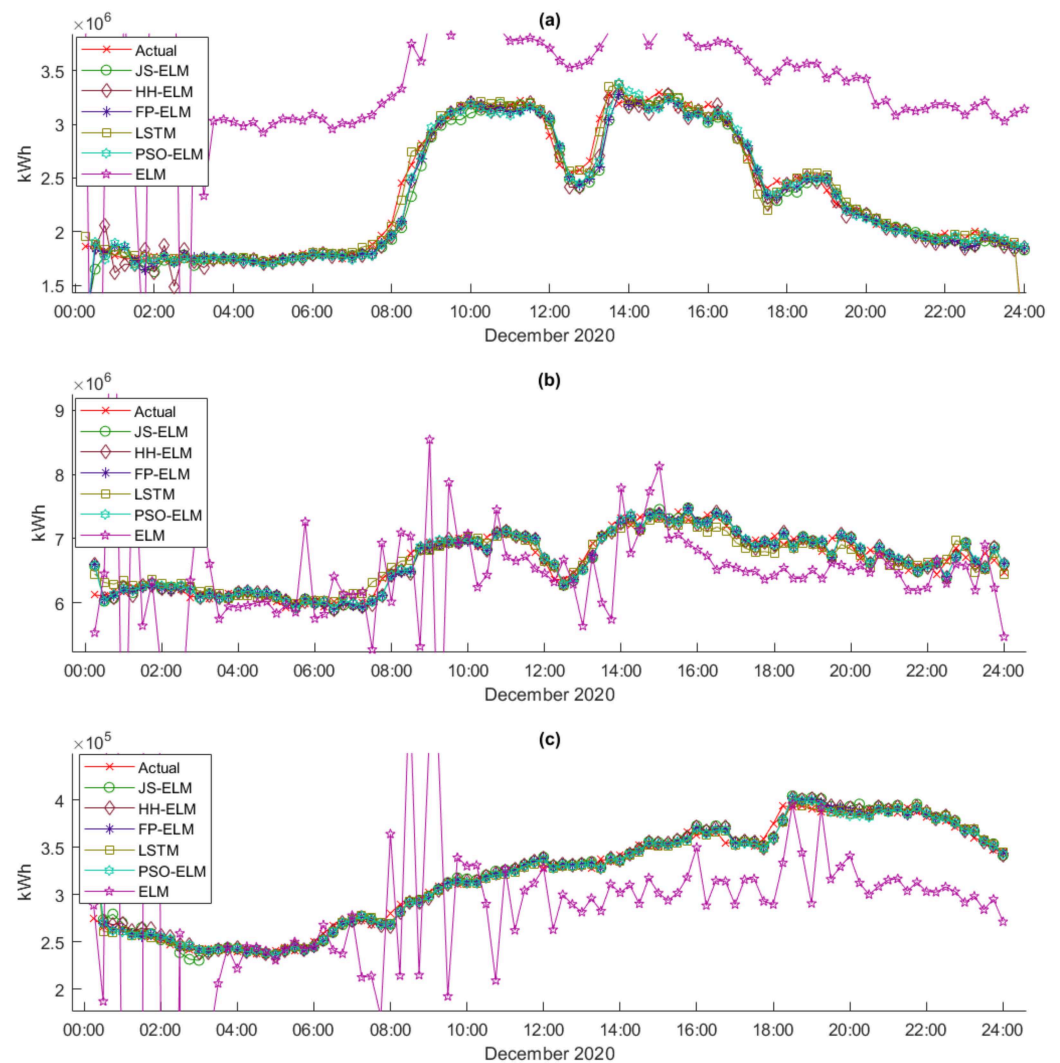
**Figure 17.** Cont.



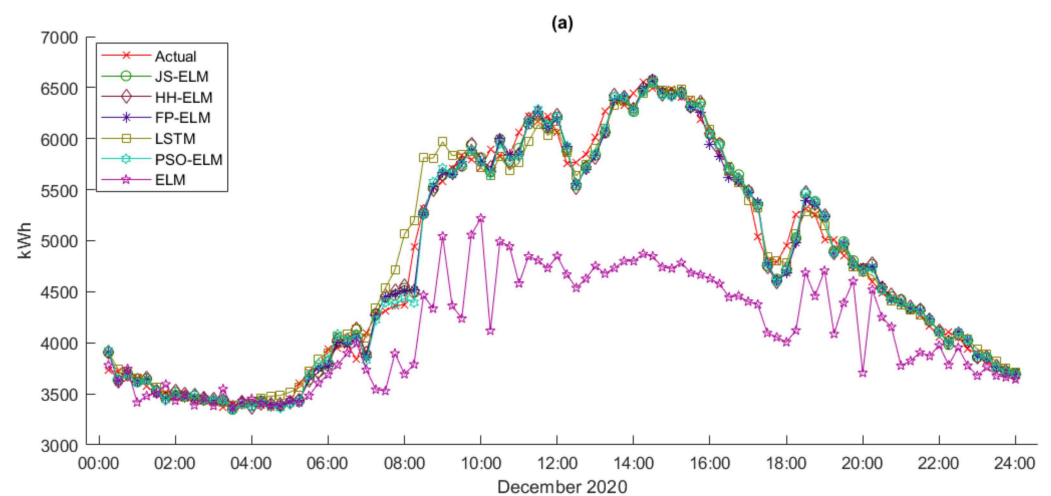
**Figure 17.** Forecasting result of dataset 6 (minimum error result). (a) Forecasted energy of each proposed model. (b) Forecasted energy of each state-of-the-art model.



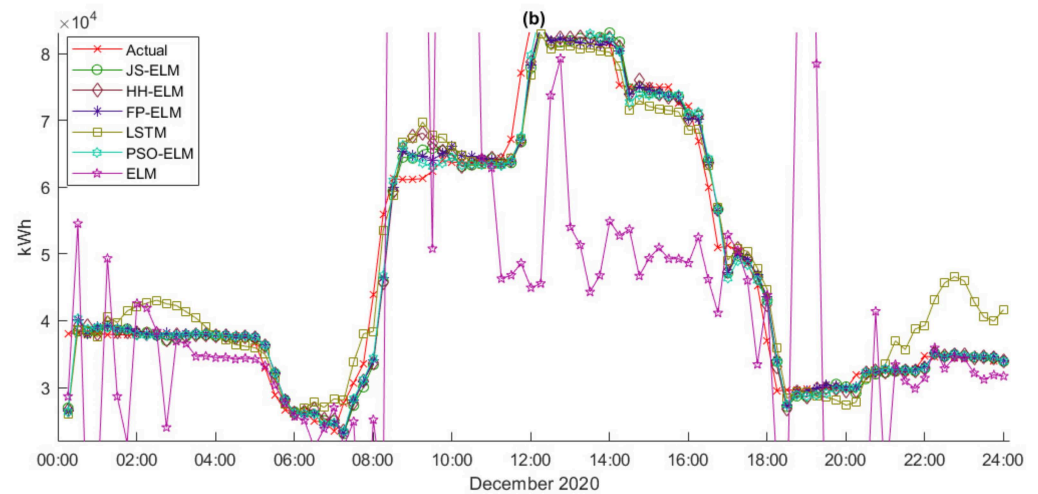
**Figure 18.** Forecasting result of dataset 7A–7C (minimum error result). (a) Forecasted energy of each proposed model in dataset 7A. (b) Forecasted energy of each proposed model in dataset 7B. (c) Forecasted energy of each proposed model in dataset 7C.



**Figure 19.** Forecasting result of dataset 7D–7F (minimum error result). (a) Forecasted energy of each proposed model in dataset 7D. (b) Forecasted energy of each proposed model in dataset 7E. (c) Forecasted energy of each proposed model in dataset 7F.



**Figure 20.** Cont.



**Figure 20.** Forecasting result of dataset 7G–7H (minimum error result). (a) Forecasted energy of each proposed model in dataset 7G. (b) Forecasted energy of each proposed model in dataset 7H.

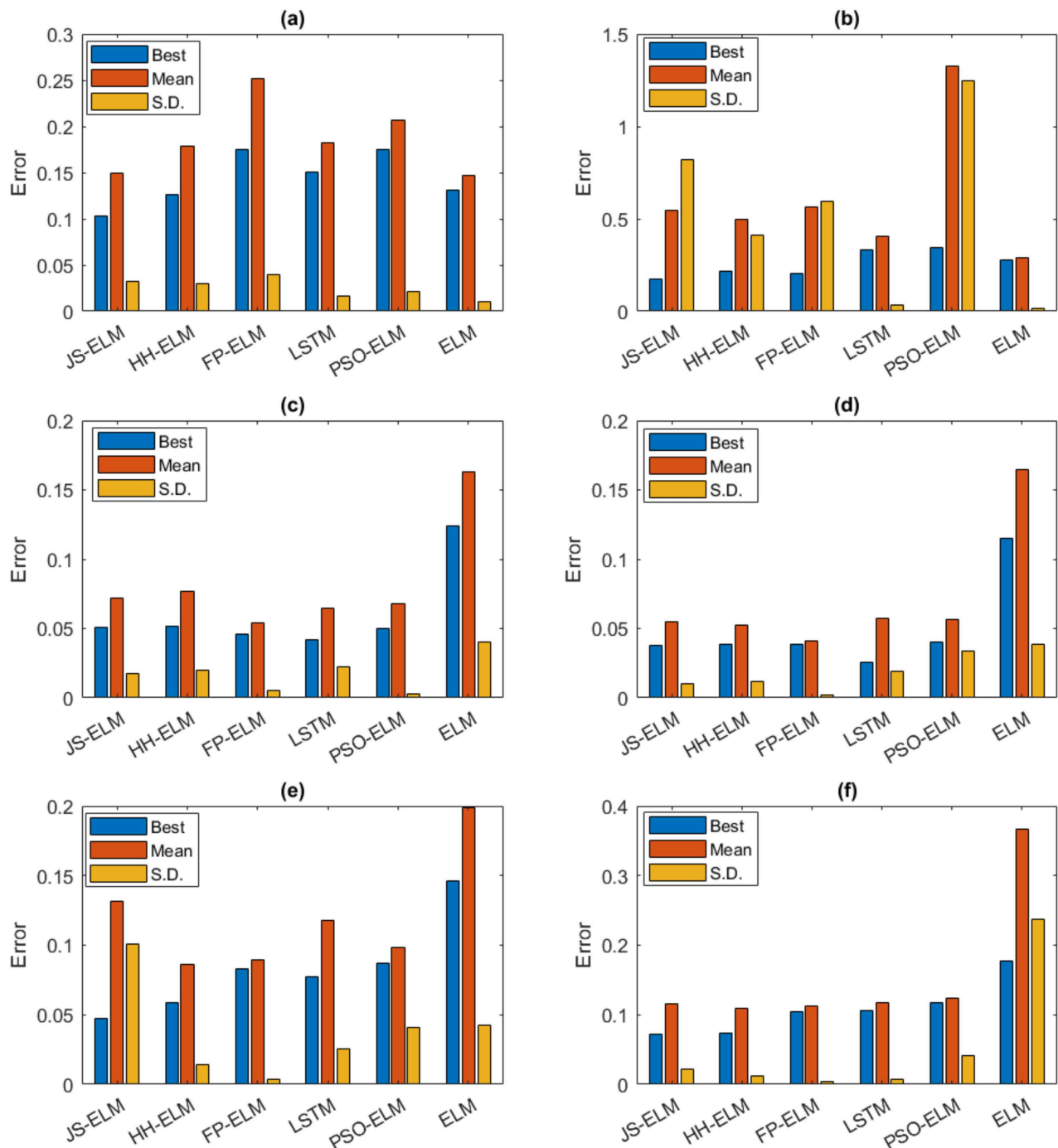
**Table 5.** Result of the Training Time Experiment (Bold text is the suitable time processing result).

Datasets	The Training Time of Models (Seconds)					
	JS-ELM	HH-ELM	FP-ELM	LSTM	PSO-ELM	ELM
1	<b>11.7813</b>	20.0781	<b>9.5625</b>	43.7802	17.3698	0.0001
2	<b>9.1094</b>	17.2500	<b>8.4844</b>	35.4268	14.2356	0.0001
3	<b>102.8281</b>	228.6250	<b>96.1563</b>	397.0562	135.3756	0.7188
4	<b>89.3438</b>	183.2188	<b>89.6563</b>	393.6320	129.8451	0.5781
5	<b>74.5469</b>	146.0156	<b>70.4844</b>	386.5398	107.6406	0.4844
6	<b>69.6563</b>	137.0156	<b>67.0469</b>	375.6900	108.0342	0.2500
7A	<b>61.0313</b>	124.6563	<b>60.5781</b>	345.5692	109.0312	0.2344
7B	<b>62.1719</b>	126.6250	<b>61.1875</b>	358.3612	108.0781	0.2969
7C	<b>64.2031</b>	132.9688	<b>64.2344</b>	362.5597	107.9843	0.3281
7D	<b>70.8438</b>	138.5625	<b>66.0625</b>	372.6623	106.2031	0.3594
7E	<b>64.2188</b>	138.9375	<b>65.7656</b>	351.1258	108.9218	0.2500
7F	<b>65.5625</b>	132.3281	<b>64.3594</b>	363.6690	106.5625	0.4219
7G	<b>62.6719</b>	127.1406	<b>67.9063</b>	349.9562	108.9531	0.3906
7H	<b>74.6094</b>	145.0469	<b>75.4219</b>	383.3356	104.0156	0.2188

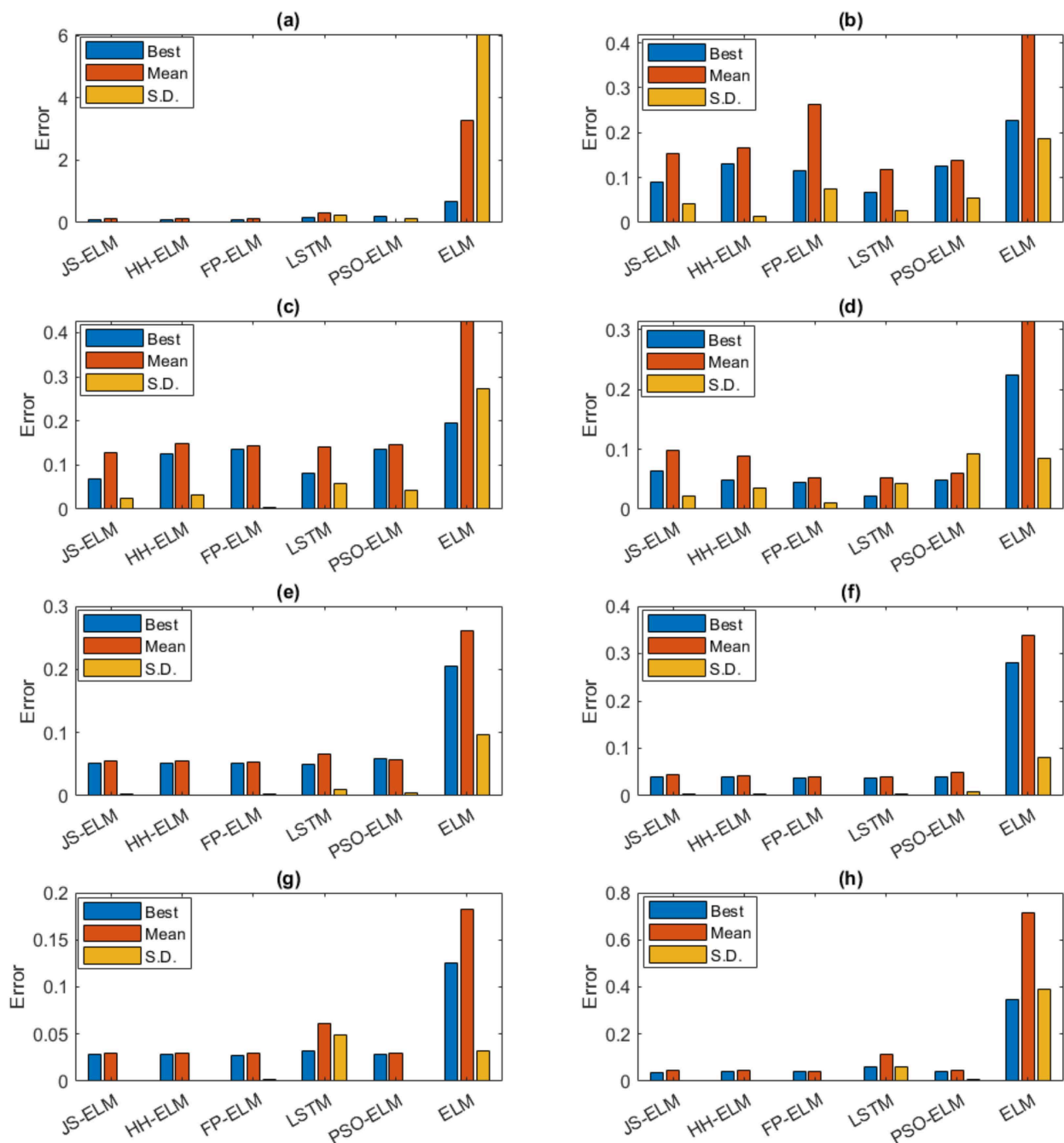
The CWE comparison of all models for dataset 7 is shown in Figure 22. All proposed models were the most suitable for all three factors, while the ELM model provided the highest CWE rate, which means the ELM model was already overfitted, as referred to in Section 4.1.

According to Section 4.2, the time consumptions of the proposed models were higher than the time consumption of the ELM model because the proposed models were tuned by metaheuristics optimization. Major factors that directly affected the high time consumption of the proposed models consist of the time step calculation for each metaheuristic algorithm, number of populations, iterations of the metaheuristics algorithms, and number of hidden nodes of the ELM model. In conclusion, the proposed models slightly consumed more time in calculation than the time consumption of the ELM model, while the experiment results for all datasets were highly accurate and more stable than the result of the ELM model.

In comparison, the overall error results of the proposed models and LSTM were close. However, according to Table 5, the training time of the proposed models was faster than the LSTM model.



**Figure 21.** CWE comparison for all models from datasets 1 to 6 (lower is better). (a) CWE comparison of each proposed model in dataset 1. (b) CWE comparison of each proposed model in dataset 2. (c) CWE comparison of each proposed model in dataset 3. (d) CWE comparison of each proposed model in dataset 4. (e) CWE comparison of each proposed model in dataset 5. (f) CWE comparison of each proposed model in dataset 6.



**Figure 22.** CWE comparison for all models from datasets 7 (lower is better). (a) CWE comparison of each proposed model in dataset 7A. (b) CWE comparison of each proposed model in dataset 7B. (c) CWE comparison of each proposed model in dataset 7C. (d) CWE comparison of each proposed model in dataset 7D. (e) CWE comparison of each proposed model in dataset 7E. (f) CWE comparison of each proposed model in dataset 7F. (g) CWE comparison of each proposed model in dataset 7G. (h) CWE comparison of each proposed model in dataset 7H.

## 5. Conclusions and Future Work

This research proposed the novel ELM model optimized by metaheuristic algorithms, namely JS-ELM, HH-ELM, and FP-ELM, to forecast the electric energy dataset. The characteristic of the metaheuristic optimizations, namely JSO, HHO, and FPA, is that they are self-adaptive to tune the weight parameter of the ELM model without being trapped in the local optimum. Therefore, the models mentioned earlier were applied to the ELM

model by tuning the weight parameter of the ELM model instead of using the traditional randomization of the weight parameter. In addition, these three optimization methods can reduce the number of hidden nodes of the ELM model. To demonstrate the performance of the proposed method, all models were applied to forecast seven real-life electric energy datasets. The dataset of electric energy demand consists of monthly electric energy demand data from 2018 to 2020, monthly electric energy loss data from 2018 to 2020, peak day 15 min interval electric energy demand data from 2018 to 2020, workday 15 min interval electric energy demand data from 2018 to 2020, peak day 15 min interval electric energy demand data in 2020, workday 15 min interval electric energy demand data in 2020, and cluster of 15 min interval peak day electric energy demand data. Consequently, the result showed that the proposed models could improve the forecasting accuracy, provide forecasting stability, and reduce the cause of overfitting from the traditional model.

According to Table 6, the JS-ELM model was the most suitable for the minimum error result. The overall forecasting results of the HH-ELM model were similar to the JS-ELM model results, but the time consumption was higher than the time consumption of other models. The FP-ELM model was the most suitable in terms of the minimum mean error and minimum S.D. The time consumption of the proposed models depended on the number of populations, iterations criterion of metaheuristics algorithms, and the number of hidden nodes of the ELM model. The HH-ELM model used twice the time consumption compared with the time consumptions of the JS-ELM and the FP-ELM models. Furthermore, the time consumptions of the proposed models were lower than the LSTM model, while the overall error results of the proposed models and LSTM were close.

**Table 6.** Summary of the Outperformance models (ranking by CWE).

Datasets	Outperformance Models (Ranking by CWE)		
	Best	Mean	S.D.
1	JS-ELM	JS-ELM	ELM
2	JS-ELM	ELM	ELM
3	JS-ELM	FP-ELM	FP-ELM
4	LSTM	FP-ELM	FP-ELM
5	JS-ELM	HH-ELM	FP-ELM
6	JS-ELM	HH-ELM	FP-ELM
7A	FP-ELM	FP-ELM	FP-ELM
7B	LSTM	LSTM	LSTM
7C	FP-ELM	JS-ELM	JS-ELM
7D	LSTM	FP-ELM	FP-ELM
7E	LSTM	FP-ELM	HH-ELM
7F	LSTM	FP-ELM	FP-ELM
7G	FP-ELM	HH-ELM	JS-ELM
7H	JS-ELM	FP-ELM	FP-ELM

For suggestions for future works, the proposed models can improve the stability of forecasting, for instance, according to Table 6, JS-ELM was the most suitable when considering the minimum error result, and FP-ELM was the most suitable when considering the S.D. result. Therefore, future work may propose a novel model that applies the hybrids of JSO and FPA to the ELM model. Due to the benefits of JSO and FPA, the expected results of the aforementioned novel model can be achieved with a more stable accuracy of forecasting and provide the best forecasting result. Moreover, the time consumption of forecasting is the major topic to discuss. According to Table 5, the time consumption of JS-ELM and FP-ELM were close and less than other models except for ELM. The time

consumption of the proposed models and the future novel model can be reduced by using the step of tuning the hyper-parameters with a suitable algorithm and the methodology of an ensemble learning algorithm [52–54] that splits the appropriate sub-datasets from the primary dataset, forecasts the sub-datasets, and aggregates the forecasted model results. Finally, due to the variety of metaheuristics algorithms [47] that are continuously being developed, an alternative metaheuristics algorithm can be discussed to implement with the ELM model or other machine learning models to improve the accuracy of the forecasted electric energy demand.

**Author Contributions:** S.B.: writing—original draft preparation, conceptualization, methodology, software, data curation, simulation. C.S.: supervision, investigation, conceptualization, writing—review and editing. P.F.: supervision, investigation, conceptualization, data curation, resources. R.C.: supervision, writing—review and editing, formal analysis, methodology, investigation, validation. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Provincial Electricity Authority of Thailand, which provided funding and useful information for the experiment in this research, and the Faculty of Engineering, Khon Kaen University, under Grant Ph.D.Ee-1/2564.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Datasets from Section 3 (electric energy demand data) can be found and downloaded at <http://peaoc.pea.co.th/loadprofile/en/>. (accessed on 8 October 2021).

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Pritima, D.; Krishnan, P.G.; Padmanabhan, P.; Stalin, B. Performance of electrical energy based unconventional machining processes—Review. *Mater. Today Proc.* **2021**, in press. [CrossRef]
2. Hyndman, R.J. Forecasting: An Overview. In *International Encyclopedia of Statistical Science*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 536–539. [CrossRef]
3. Mas-Machuca, M.; Sainz, M.; Martinez-Costa, C. A review of forecasting models for new products. *Intang. Cap.* **2014**, *10*, 1–25. [CrossRef]
4. Palm, F.C.; Nijman, T.E. Missing Observations in the Dynamic Regression Model. *Econometrica* **1984**, *52*, 1415. [CrossRef]
5. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons: Hoboken, NJ, USA, 2015.
6. Brown, R.G. *Smoothing, Forecasting and Prediction of Discrete Time Series*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1963.
7. McCulloch, W.S.; Pitts, W.H. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
8. Memarzadeh, G.; Keynia, F. Short-term electricity load and price forecasting by a new optimal LSTM-NN based prediction algorithm. *Electr. Power Syst. Res.* **2020**, *192*, 106995. [CrossRef]
9. Liu, Z.; Loo, C.K.; Pasupa, K. A novel error-output recurrent two-layer extreme learning machine for multi-step time series prediction. *Sustain. Cities Soc.* **2020**, *66*, 102613. [CrossRef]
10. Larrea, M.; Porto, A.; Irigoyen, E.; Barragán, A.J.; Andújar, J.M. Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem. *Neurocomputing* **2020**, *452*, 465–472. [CrossRef]
11. Chen, Y.; Kloft, M.; Yang, Y.; Li, C.; Li, L. Mixed kernel based extreme learning machine for electric load forecasting. *Neurocomputing* **2018**, *312*, 90–106. [CrossRef]
12. Chen, Q.; Xia, M.; Lu, T.; Jiang, X.; Liu, W.; Sun, Q. Short-Term Load Forecasting Based on Deep Learning for End-User Transformer Subject to Volatile Electric Heating Loads. *IEEE Access* **2019**, *7*, 162697–162707. [CrossRef]
13. Rafi, S.H.; Masood, N.A.; Deeba, S.R.; Hossain, E. A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network. *IEEE Access* **2021**, *9*, 32436–32448. [CrossRef]
14. Sajjad, M.; Khan, Z.A.; Ullah, A.; Hussain, T.; Ullah, W.; Lee, M.Y.; Baik, S.W. A Novel CNN-GRU-Based Hybrid Approach for Short-Term Residential Load Forecasting. *IEEE Access* **2020**, *8*, 143759–143768. [CrossRef]
15. Hu, Y.; Li, J.; Hong, M.; Ren, J.; Lin, R.; Liu, Y.; Liu, M.; Man, Y. Short term electric load forecasting model and its verification for process industrial enterprises based on hybrid GA-PSO-BPNN algorithm—A case study of papermaking process. *Energy* **2019**, *170*, 1215–1227. [CrossRef]
16. Kazemzadeh, M.-R.; Amjadi, A.; Amraee, T. A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting. *Energy* **2020**, *204*, 117948. [CrossRef]

17. Hafeez, G.; Alimgeer, K.S.; Khan, I. Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. *Appl. Energy* **2020**, *269*, 114915. [CrossRef]
18. Huang, G.-B.; Zhu, Q.-Y.; Siew, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [CrossRef]
19. Penrose, R. A generalized inverse for matrices. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [CrossRef]
20. Wang, J.; Lu, S.; Wang, S.-H.; Zhang, Y.-D. A review on extreme learning machine. *Multimedia Tools Appl.* **2021**, *80*, 1–50. [CrossRef]
21. Horata, P.; Chiewchanwattana, S.; Sunat, K. Robust extreme learning machine. *Neurocomputing* **2013**, *102*, 31–44. [CrossRef]
22. Song, S.; Wang, M.; Lin, Y. An improved algorithm for incremental extreme learning machine. *Syst. Sci. Control Eng.* **2020**, *8*, 308–317. [CrossRef]
23. Chou, J.-S.; Truong, D.-N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2020**, *389*, 125535. [CrossRef]
24. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **2019**, *97*, 849–872. [CrossRef]
25. Yang, X.-S. Flower Pollination Algorithm for Global Optimization. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7445, pp. 240–249. Available online: [https://doi.org/10.1007/978-3-642-32894-7\\_27](https://doi.org/10.1007/978-3-642-32894-7_27) (accessed on 16 June 2021).
26. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2021; Available online: <https://otexts.com/fpp3/index.html> (accessed on 10 February 2021).
27. Armstrong, J.S.; Green, K.C. Forecasting methods and principles: Evidence-based checklists. *J. Glob. Sch. Mark. Sci.* **2018**, *28*, 103–159. [CrossRef]
28. Mohamad, M.; Selamat, A.; Subroto, I.M.; Krejcar, O. Improving the classification performance on imbalanced data sets via new hybrid parameterisation model. *J. King Saud Univ. Comput. Inf. Sci.* **2019**, *33*, 787–797. [CrossRef]
29. Dinh, D.-T.; Huynh, V.-N.; Sriboonchitta, S. Clustering mixed numerical and categorical data with missing values. *Inf. Sci.* **2021**, *571*, 418–442. [CrossRef]
30. Millsap, R.E.; Chase, C.W.J. R: A Language and Environment for Statistical Computing. *J. Bus. Forecast. Methods Syst.* **1995**, *14*, 461–473. Available online: <http://www.r-project.org> (accessed on 12 July 2021).
31. Boriratr, S.; Tepsiri, W.; Krobnopparat, A.; Khunsaeng, N. Forecasting and Evaluation Electricity Loss in Thailand via Flower Pollination Extreme Learning Machine Model. In Proceedings of the 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 12–15 August 2018; pp. 160–164. [CrossRef]
32. Bastian, T.; Lilley, M.K.; Beggs, S.E.; Hays, G.C.; Doyle, T.K. Ecosystem relevance of variable jellyfish biomass in the Irish Sea between years, regions and water types. *Estuar. Coast. Shelf Sci.* **2014**, *149*, 302–312. [CrossRef]
33. May, R.M. Simple mathematical models with very complicated dynamics. *Nature* **1976**, *261*, 459–467. [CrossRef]
34. Bednarz, J.C. Cooperative Hunting Harris’ Hawks (*Parabuteo unicinctus*). *Science* **1988**, *239*, 1525–1527. [CrossRef]
35. Kleinberg, J.M. Navigation in a small world. *Nature* **2000**, *406*, 845. [CrossRef]
36. Gomes, W.; Beck, A.; Lopez, R.H.; Miguel, L.F. A probabilistic metric for comparing metaheuristic optimization algorithms. *Struct. Saf.* **2018**, *70*, 59–70. [CrossRef]
37. Kaul, S.; Kumar, Y. Nature-Inspired Metaheuristic Algorithms for Constraint Handling: Challenges, Issues, and Research Perspective. In *Constraint Handling in Metaheuristics and Applications*; Springer: Singapore, 2021; pp. 55–80. [CrossRef]
38. Alabool, H.M.; Alarabiat, D.; Abualigah, L.; Heidari, A.A. Harris hawks optimization: A comprehensive review of recent variants and applications. *Neural Comput. Appl.* **2021**, *33*, 8939–8980. [CrossRef]
39. Lei, M.; Zhou, Y.; Luo, Q. Enhanced Metaheuristic Optimization: Wind-Driven Flower Pollination Algorithm. *IEEE Access* **2019**, *7*, 111439–111465. [CrossRef]
40. PEA. Load Profile: Load Research of PEA. Available online: <http://peaoc.pea.co.th/loadprofile/en/> (accessed on 2 August 2021).
41. PEA. Electricity Sales Report & Load Research of PEA. *Provincial Electricity Authority of Thailand*. 2020. Available online: <http://peaoc.pea.co.th/> (accessed on 8 October 2021).
42. Shahhosseini, M.; Hu, G.; Pham, H. Optimizing ensemble weights and hyperparameters of machine learning models for regression problems. *Mach. Learn. Appl.* **2022**, *7*, 100251. [CrossRef]
43. Agrawal, T. *Hyperparameter Optimization in Machine Learning*; Apress: Berkeley, CA, USA, 2021; volume XIX, p. 166. Available online: <https://doi.org/10.1007/978-1-4842-6579-6> (accessed on 29 November 2021).
44. Nakisa, B.; Rastgoo, M.N.; Rakotonirainy, A.; Maire, F.; Chandran, V. Long Short Term Memory Hyperparameter Optimization for a Neural Network Based Emotion Recognition Framework. *IEEE Access* **2018**, *6*, 49325–49338. [CrossRef]
45. Xie, Y.; Li, C.; Tang, G.; Liu, F. A novel deep interval prediction model with adaptive interval construction strategy and automatic hyperparameter tuning for wind speed forecasting. *Energy* **2020**, *216*, 119179. [CrossRef]
46. Neshat, M.; Nezhad, M.M.; Abbasnejad, E.; Mirjalili, S.; Groppi, D.; Heydari, A.; Tjernberg, L.B.; Garcia, D.A.; Alexander, B.; Shi, Q.; et al. Wind turbine power output prediction using a new hybrid neuro-evolutionary method. *Energy* **2021**, *229*, 120617. [CrossRef]
47. Mirjalili, S. Projects, Seyedali Mirjalili Project. 2022. Available online: <https://seyedalimirjalili.com/projects> (accessed on 15 March 2022).

48. Kaloop, M.R.; Kumar, D.; Samui, P.; Gabr, A.R.; Hu, J.W.; Jin, X.; Roy, B. Particle Swarm Optimization Algorithm-Extreme Learning Machine (PSO-ELM) Model for Predicting Resilient Modulus of Stabilized Aggregate Bases. *Appl. Sci.* **2019**, *9*, 3221. [\[CrossRef\]](#)
49. Patro, S.G.K.; Sahu, K.K. Normalization: A preprocessing stage. *Int. Adv. Res. J. Sci. Eng. Technol.* **2015**, *2*, 20–22. [\[CrossRef\]](#)
50. Pirbazari, A.M.; Farmanbar, M.; Chakravorty, A.; Rong, C. Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis. *Processes* **2020**, *8*, 484. [\[CrossRef\]](#)
51. Asare-Bediako, B.; Kling, W.L.; Ribeiro, P.F. Day-ahead residential load forecasting with artificial neural networks using smart meter data. In Proceedings of the 2013 IEEE Grenoble Conference, Grenoble, France, 16–20 June 2013; pp. 1–6. [\[CrossRef\]](#)
52. Wijaya, D.R.; Afianti, F.; Arifianto, A.; Rahmawati, D.; Kodogiannis, V.S. Ensemble machine learning approach for electronic nose signal processing. *Sens. Bio-Sens. Res.* **2022**, *36*, 100495. [\[CrossRef\]](#)
53. Carneiro, T.C.; Rocha, P.A.; Carvalho, P.C.; Fernández-Ramírez, L.M. Ridge regression ensemble of machine learning models applied to solar and wind forecasting in Brazil and Spain. *Appl. Energy* **2022**, *314*, 118936. [\[CrossRef\]](#)
54. Matrenin, P.; Safaraliev, M.; Dmitriev, S.; Kokin, S.; Ghulomzoda, A.; Mitrofanov, S. Medium-term load forecasting in isolated power systems based on ensemble machine learning models. *Energy Rep.* **2021**, *8*, 612–618. [\[CrossRef\]](#)