

Article

# Hybridizing Chaotic and Quantum Mechanisms and Fruit Fly Optimization Algorithm with Least Squares Support Vector Regression Model in Electric Load Forecasting

## Ming-Wei Li<sup>1</sup>, Jing Geng<sup>1</sup>, Wei-Chiang Hong<sup>2,\*</sup> <sup>D</sup> and Yang Zhang<sup>1</sup>

- <sup>1</sup> College of shipbuilding engineering, Harbin Engineering University, Harbin 150001, Heilongjiang, China; limingwei@hrbeu.edu.cn (M.-W.L.); gengjing@hrbeu.edu.cn (J.G.); zhangyang@hrbeu.edu.cn (Y.Z.)
- <sup>2</sup> School of Education Intelligent Technology, Jiangsu Normal University/No. 101, Shanghai Rd., Tongshan District, Xuzhou 221116, Jiangsu, China
- \* Correspondence: samuelsonhong@gmail.com; Tel.: +86-516-83500307

Received: 13 August 2018; Accepted: 22 August 2018; Published: 24 August 2018



Abstract: Compared with a large power grid, a microgrid electric load (MEL) has the characteristics of strong nonlinearity, multiple factors, and large fluctuation, which lead to it being difficult to receive more accurate forecasting performances. To solve the abovementioned characteristics of a MEL time series, the least squares support vector machine (LS-SVR) hybridizing with meta-heuristic algorithms is applied to simulate the nonlinear system of a MEL time series. As it is known that the fruit fly optimization algorithm (FOA) has several embedded drawbacks that lead to problems, this paper applies a quantum computing mechanism (QCM) to empower each fruit fly to possess quantum behavior during the searching processes, i.e., a QFOA algorithm. Eventually, the cat chaotic mapping function is introduced into the QFOA algorithm, namely CQFOA, to implement the chaotic global perturbation strategy to help fruit flies to escape from the local optima while the population's diversity is poor. Finally, a new MEL forecasting method, namely the LS-SVR-CQFOA model, is established by hybridizing the LS-SVR model with CQFOA. The experimental results illustrate that, in three datasets, the proposed LS-SVR-CQFOA model is superior to other alternative models, including BPNN (back-propagation neural networks), LS-SVR-CQPSO (LS-SVR with chaotic quantum particle swarm optimization algorithm), LS-SVR-CQTS (LS-SVR with chaotic quantum tabu search algorithm), LS-SVR-CQGA (LS-SVR with chaotic quantum genetic algorithm), LS-SVR-CQBA (LS-SVR with chaotic quantum bat algorithm), LS-SVR-FOA, and LS-SVR-QFOA models, in terms of forecasting accuracy indexes. In addition, it passes the significance test at a 97.5% confidence level.

**Keywords:** least squares support vector regression (LS-SVR); chaos theory; quantum computing mechanism (QCM); fruit fly optimization algorithm (FOA); microgrid electric load forecasting (MEL)

## 1. Introduction

## 1.1. Motivation

MEL forecasting is the basis of microgrid operation scheduling and energy management. It is an important prerequisite for the intelligent management of distributed energy. The forecasting performance would directly affect the microgrid system's energy trading, power supply planning, and power supply quality. However, the MEL forecasting accuracy is not only influenced by the mathematical model, but also by the associated historical dataset. In addition, compared with the large power grid, microgrid electric load (MEL) has the characteristics of strong nonlinearity, multiple factors,



and large fluctuation, which lead to it being difficult to achieve more accurate forecasting performances. Along with the development of artificial intelligent technologies, load forecasting methods have been continuously applied to load forecasting. Furthermore, the hybridization or combination of the intelligent algorithms also provides new models to improve the load forecasting performances. These hybrid or combined models either employ a novel intelligent algorithm or framework to improve the embedded drawbacks or apply the advantages of two of the above models to achieve more satisfactory results. The models apply a wide range of load forecasting approaches and are mainly divided into two categories, traditional forecasting models and intelligent forecasting models.

## 1.2. Relevant Literature Reviews

Conventional load forecasting models include exponential smoothing models [1], time series models [2], and regression analysis models [3]. An exponential smoothing model is a curve fitting method that defines different coefficients for the historical load data. It can be understood that a series with the forecasted load time has a large influence on the future load, while a series with the long time from the forecasted load has a small influence on the future load [1]. The time series model is applied to load forecasting, which is characterized by a fast forecasting speed and can reflect the continuity of load forecasting, but requires the stability of the time series. The disadvantage is that it cannot reflect the impact of external environmental factors on load forecasting [2]. The regression model seeks a causal relationship between the independent variable and the dependent variables according to the historical load change law, determining the regression equation, and the model parameters. The disadvantage of this model is that there are too many factors affecting the forecasting accuracy. It is not only affected by the parameters of the model itself, but also by the quality of the data. When the external influence factors are too many or the relevant influent factor data are difficult to analyze, the regression forecasting model will result in huge errors [3].

Intelligent forecasting models include the wavelet analysis method [4,5], grey forecasting theory [6,7], the neural network model [8,9], and the support vector regression (SVR) model [10]. In load forecasting, the wavelet analysis method is combined with external factors to establish a suitable load forecasting model by decomposing the load data into sequences on different scales [4,5]. The advantages of the grey model are easy to implement and there are fewer influencing factors employed. However, the disadvantage is that the processed data sequence has more grayscale, which results in large forecasting error [6,7]. Therefore, when this model is applied to load forecasting, only a few recent data points would be accurately forecasted; more distant data could only be reflected as trend values and planned values [7]. Due to the superior nonlinear performances, many models based on artificial neural networks (ANNs) have been applied to improve the load forecasting accuracy [8,9]. To achieve more accurate forecasting performance, these models and other new or novel forecasting approaches have been hybridized or combined [9]. For example, an adaptive network-based fuzzy inference system is combined with an RBF neural network [11], the Monte Carlo algorithm is combined with the Bayesian neural network [12], fuzzy behavior is hybridized with a neural network (WFNN) [13], a knowledge-based feedback tuning fuzzy system is hybridized with a multi-layer perceptron artificial neural network (MLPANN) [14], and so on. However, these ANNs-based models suffer from some serious problems, such as trapping into local optimum easily, it being time-consuming to achieve a functional approximation, and the difficulty of selecting the structural parameters of a network [15,16], which limits its application in load forecasting to a large extent.

The SVR model is based on statistical learning theory, as proposed by Vapnik [17]. It has a solid mathematical foundation, a better generalization ability, a relatively faster convergence rate, and can find global optimal solutions [18]. Because the basic theory of the SVR model is perfect and the model is also easy to establish, it has attracted extensive attention from scholars in the load forecasting fields. In recent years, some scholars have applied the SVR model to the research of load forecasting [18] and achieved superior results. One study [19] proposes the EMD-PSO-GA-SVR model to improve the forecasting accuracy, by hybridizing the empirical mode decomposition (EMD)

3 of 22

with two particle swarm optimization (PSO) and the genetic algorithm (GA). In addition, a modified version of the SVR model, namely the LS-SVR model, only considers equality constraints instead of inequalities [20,21]. Focused on the advantages of the LS-SVR model to deal with such problems, this paper tries to simulate the nonlinear system of the MEL time series to receive the forecasting values and improve the forecasting accuracy. However, the disadvantages of the SVR-based models in load forecasting are that when the sample size of the load is large, the time of system learning and training is highly time-consuming, and the determination of parameters mainly depends on the experience of the researchers. This has a certain degree of influence on the accuracy in load forecasting. Therefore, exploring more suitable parameter determination methods has always been an effective way to improve the forecasting accuracy of the SVR-based models. To determine more appropriate parameters of the SVR-based models, Hong and his colleagues have conducted research using different evolutionary algorithms hybridized with an SVR model [22–24]. In the meantime, Hong and his successors have also applied different chaotic mapping functions (including the logistic function [22,23] and the cat mapping function [10]) to diversify the population during modeling processes, and the cloud theory to make sure the temperature continuously decreases during the annealing process, eventually determining the most appropriate parameters to receive more satisfactory forecasting accuracy [10].

The fruit fly optimization algorithm (FOA) is a new swarm intelligent optimization algorithm proposed in 2011, it searches for global optimization based on fruit fly foraging behavior [25,26]. The algorithm has only four control parameters [27]. Compared with other algorithms, FOA has the advantages of being easy to program and having fewer parameters, less computation, and high accuracy [28,29]. FOA belongs to the domain of evolutionary computation; it realizes the optimization of complex problems by simulating fruit flies to search for food sources by using olfaction and vision. It has been successfully applied to the predictive control fields [30,31]. However, similar to those swarm intelligent optimization algorithms with iterative searching mechanisms, the standard FOA also has drawbacks such as a premature convergent tendency, a slow convergent rate in the later searching stage, and poor local search performance [32].

Quantum computing has become one of the leading branches of science in the modern era due to its powerful computing ability. This not only prompted us to study new quantum algorithms, but also inspired us to re-examine some traditional optimization algorithms from the quantum computing mechanism. The quantum computing mechanism (QCM) makes full use of the superposition and coherence of quantum states. Compared with other evolutionary algorithms, the QCM uses a novel encoding method-quantum bit encoding. Through the encoding of qubits, an individual can characterize any linear superposition state, whereas traditional encoding methods can only represent one specific one. As a result, with QCM it is easier to maintain population diversity than with other traditional evolutionary algorithms. Nowadays, it has become a hot topic of research that QCM is able to hybridize with evolutionary algorithms to receive more satisfactory searching results. The literature [33] introduced QCM into genetic algorithms and proposed quantum derived genetic algorithm (QIGA). From the point of view of algorithmic mechanism, it is very similar to the isolated niches genetic algorithm. Han and Kim [34] proposed a genetic quantum algorithm (GQA) based on QCM. Compared with traditional evolutionary algorithms, its greatest advantage is its better ability to maintain population diversity. Han and Kim [35] further introduced the population migration mechanism based on ure [34], and renamed the algorithm a quantum evolution algorithm (QEA). Huang [36], Lee and Lin [37,38], and Li et al. [39] hybridized the particle swam optimization (PSO) algorithm, Tabu search (TS) algorithm, genetic algorithm (GA), and bat algorithm (BA) with the QCM and the cat mapping function, and proposed the CQPSO, CQTS, CQGA, and CQBA algorithms, which were employed to select the appropriate parameters of an SVR model. The results of the application indicate that the improved algorithms obtain more appropriate parameters, and higher forecasting accuracy is achieved. The above applications also reveal that the improved algorithm, by hybridizing with QCM, could effectively avoid local optimal position and premature convergence.

#### 1.3. Contributions

Considering the inherent drawback of the FOA, i.e., suffering from premature convergence, this paper tries to hybridize the FOA with QCM and the cat chaotic mapping function to solve the premature problem of FOA. Eventually, determine more appropriate parameters of an LS-SVR model. The major contributions are as follows:

- (1) QCM is employed to empower the search ability of each fruit fly during the searching processes of QFOA. The cat chaotic mapping function is introduced into QFOA and implements the chaotic global perturbation strategy to help a fruit fly escape from the local optima when the population's diversity is poor.
- (2) We propose a novel hybrid optimization algorithm, namely CQFOA, to be hybridized with an LS-SVR model, namely the LS-SVR-CQFOA model, to conduct the MEL forecasting. Other similar alternative hybrid algorithms (hybridizing chaotic mapping function, QCM, and evolutionary algorithms) in existing papers, such as the CQPSO algorithm used by Huang [36], the CQTS and CQGA algorithms used by Lee and Lin [37,38], and the CQBA algorithm used by Li et al. [39], are selected as alternative models to test the superiority of the LS-SVR-CQFOA model in terms of forecasting accuracy.
- (3) The forecasting results illustrate that, in three datasets, the proposed LS-SVR-CQFOA model is superior to other alternative models in terms of forecasting accuracy indexes; in addition, it passes the significance test at a 97.5% confidence level.

#### 1.4. The Organization of This Paper

The rest of this paper is organized as follows. The modeling details of an LS-SVR model, the proposed CQFOA, and the proposed LS-SVR-CQFOA model are introduced in Section 2. Section 3 presents a numerical example and a comparison of the proposed LS-SVR-CQFOA model with other alternative models. Some insight discussions are provided in Section 4. Finally, the conclusions are given in Section 5.

#### 2. Materials and Methods

### 2.1. Least Squares Support Vector Regression (LS-SVR)

The SVR model is an algorithm based on pattern recognition of statistical learning theory. It is a novel machine learning approach proposed by Vapnik in the mid-1990s [17]. The LS-SVR model was put forward by Suykens [20]. It is an improvement and an extension of the standard SVR model, which replaces the inequality constraints of an SVR model with equality constraint [21]. The LS-SVR model converts quadratic programming problem into linear programming solving, reduces the computational complexity, and improves the convergent speed. It can solve the load forecasting problems due to its characteristics of nonlinearity, high dimension, and local minima.

## 2.1.1. Principle of the Standard SVR Model

Set a dataset as  $\{(x_i, y_i)\}_{i=1}^N$ ,  $x_i \in \mathbb{R}^n$  is the input vector of n-dimensional system,  $y_i \in \mathbb{R}$  is the output (not a single real value, but a *n*-dimensional vector) of system. The basic idea of the SVR model can be summarized as follows: *n*-dimensional input samples are mapped from the original space to the high-dimensional feature space *F* by nonlinear transformation  $\varphi(\cdot)$ , and the optimal linear regression function is constructed in this space, as shown in Equation (1) [17]:

$$f(x) = w^T \varphi(x) + b, \tag{1}$$

where f(x) represents the forecasting values; the weight, *w*, and the coefficient, *b*, would be determined during the SVR modeling processes.

The standard SVR model takes the  $\varepsilon$  insensitive loss function as an estimation problem for risk minimization, thus the optimization objective can be expressed as in Equation (2) [17]:

$$\min \frac{1}{2} w^T w + c \sum_{i=1}^{N} (\xi_i + \xi_i^*)$$

$$s.t. \begin{cases} y_i - w^T \varphi(x_i) - b \le \varepsilon + \xi_i \\ w^T(x_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i \ge 0, \ \xi_i^* \ge 0 \\ i = 1, \cdots, N \end{cases}$$
(2)

where c is the balance factor, usually set to 1, and  $\xi_i$  and  $\xi_i^*$  are the error of introducing the training set, which can represent the extent to which the sample point exceeds the fitting precision  $\varepsilon$ .

Equation (2) could be solved according to quadratic programming processes; the solution of the weight, w, in Equation (2) is calculated as in Equation (3) [17]:

$$w^* = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \varphi(x),$$
 (3)

where  $\alpha_i$  and  $\alpha_i^*$  are Lagrange multipliers.

The SVR function is eventually constructed as in Equation (4) [17]:

$$y(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \Psi(x_i, x) + b,$$
(4)

where  $\Psi(x_i, x)$ , the so-called kernel function, is introduced to replace the nonlinear mapping function,  $\varphi(\cdot)$ , as shown in Equation (5) [15]:

$$\Psi(x_i, x_j) = \varphi(x_i)^T \varphi(x_j).$$
<sup>(5)</sup>

2.1.2. Principle of the LS-SVR Model

The LS-SVR model is an extension of the standard SVR model. It selects the binomial of error  $\xi_t$  as the loss function; then the optimization problem can be described as in Equation (6) [20]:

$$\min \frac{1}{2}w^T w + \frac{1}{2}\gamma \sum_{i=1}^N \xi_i^2$$
s.t.  $y_i = w^T \varphi(x_i) + b + \xi_i, \ i = 1, 2, \cdots, N$ 
(6)

where the bigger the positive real number  $\gamma$  is, the smaller the regression error of the model is.

The LS-SVR model defines the loss function different from the standard SVR model, and changes its inequality constraint into an equality constraint so that w can be obtained in the dual space. After obtaining parameters  $\alpha$  and b by quadratic programming processes, the LS-SVR model is described as in Equation (7) [20]:

$$y(x) = \sum_{i=1}^{N} \alpha_i \Psi(x_i, x) + b.$$
 (7)

It can be seen that an LS-SVR model contains two parameters, the regularization parameter  $\gamma$  and the radial basis kernel function,  $\sigma^2$ . The forecasting performance of an LS-SVR model is related to the selection of  $\gamma$  and  $\sigma^2$ . The role of  $\gamma$  is to balance the confidence range and experience risk of learning machines. If  $\gamma$  is too large, the goal is only to minimize the experience risk. On the contrary, when the value of  $\gamma$  is too small, the penalty for the experience error will be small, thus increasing the value of experience risk  $\sigma$  controls the width of the Gaussian kernel function and the distribution range of the training data. The smaller  $\sigma$  is, the greater the structural risk there is, which leads to

overfitting. Therefore, the parameter selection of an LS-SVR model has always been the key to improve the forecasting accuracy.

#### 2.2. Chaotic Quantum Fruit Fly Algorithm (CQFOA)

FOA is a population intelligent evolutionary algorithm that simulates the foraging behavior of fruit flies [26]. Fruit flies are superior to other species in smell and vision. In the process of foraging, firstly, fruit flies rely on smell to find the food source. Secondly, they visually locate the specific location of food and the current position of other fruit flies, and then fly to the location of food through population interaction. At present, FOA has been applied to the forecasting of traffic accidents, export trade, and other fields [40].

#### 2.2.1. Fruit Fly Optimization Algorithm (FOA)

According to the characteristics of fruit flies searching for food, FOA includes the following main steps.

**Step 1.** Initialize randomly the fruit flies' location ( $X_0$  and  $Y_0$ ) of population.

**Step 2.** Give individual fruit flies the random direction and distance for searching for food by smell, as in Equations (8) and (9) [26]:

$$X_i = X_0 + \text{Random Value} \tag{8}$$

$$Y_i = Y_0 + \text{Random Value.} \tag{9}$$

**Step 3.** Due to the location of food being unknown, firstly, the distance from the origin (*Dist*) is estimated as in Equation (10) [25], then the determination value of taste concentration (*S*) is calculated as in Equation (11) [25], i.e., the value is the inverse of the distance.

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \tag{10}$$

$$S_i = 1/Dist_i \tag{11}$$

**Step 4.** The determination value of taste concentration (*S*) is substituted into the determination function of taste concentration (or Fitness function) to determine the individual position of the fruit fly (*Smell*<sub>*i*</sub>), as shown in Equation (12) [26]:

$$Smell_i = Function(S_i).$$
 (12)

**Step 5.** Find the *Drosophila* species (*Best index* and *Best Smell* values) with the highest odor concentrations in this population, as in Equation (13) [26]:

$$\max(Smell_i) \to (Best\_Smell_i) \text{ and } (Best\_index).$$
 (13)

**Step 6.** The optimal flavor concentration value (*Optimal\_Smell*) is retained along with the *x* and *y* coordinates (with *Best\_index*) as in Equations (14)–(16) [25], then the *Drosophila* population uses vision to fly to this position.

$$Optimal\_Smell = Best\_Smell_{i=current}$$
(14)

$$X_0 = X_{Best \ index} \tag{15}$$

$$Y_0 = Y_{Best \ index} \tag{16}$$

**Step 7.** Enter the iterative optimization, repeat **Steps 2** to **5** and judge whether the flavor concentration is better than that of the previous iteration; if so, go back to **Step 6**.

The FOA algorithm is highly adaptable, so it can efficiently search without calculating partial derivatives of the target function. It overcomes the disadvantage of trapping into local optima easily.

However, as a swarm intelligence optimization algorithm, FOA still tends to fall into a local optimal solution, due to the declining diversity in the late evolutionary population.

It is noticed that there are some significant differences between the FOA and PSO algorithms. For FOA, the taste concentration (*S*) is used to determine the individual position of each fruit fly, and the highest odor concentration in this population is retained along with the *x* and *y* coordinates; eventually, the *Drosophila* population uses vision to fly to this position. Therefore, it is based on the taste concentration to control the searching direction to find out the optimal solution. For the PSO algorithm, the inertia weight controls the impact of the previous velocity of the particle on its current one by using two positive constants called acceleration coefficients and two independent uniformly distributed random variables. Therefore, it is based on the inertia weight to control the velocity to find out the optimal solution.

Thus, aiming to deal with the inherent drawback of FOA, i.e., suffering from premature convergence or trapping into local optima easily, this paper tries to use the QCM to empower each fruit fly to possess quantum behavior (namely QFOA) during the modeling processes. At the same time, the cat mapping function is introduced into QFOA (namely CQFOA) to implement the chaotic global perturbation strategy to help a fruit fly escape from the local optima when the population's diversity is poor. Eventually, the proposed CQFOA is employed to determine the appropriate parameters of an LS-SVR model and increase the forecasting accuracy.

#### 2.2.2. Quantum Computing Mechanism for FOA

#### (1) Quantization of Fruit Flies

In the quantum computing process, a sequence consisting of quantum bits is replaced by a traditional sequence. The quantum fruit fly is a linear combination of state  $|0\rangle$  and state  $|1\rangle$ , which can be expressed as in Equation (17) [34,35]:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{17}$$

where  $\alpha^2$  and  $\beta^2$  are the probability of states,  $|0\rangle$  and  $|1\rangle$ , respectively, satisfying  $\alpha^2 + \beta^2 = 1$ , and  $(\alpha, \beta)$  are qubits composed of quantum bits.

A quantum sequence, i.e., a feasible solution, can be expressed as an arrangement of *l* qubits, as shown in Equation (18) [34,35]:

$$q_i = \left\{ \begin{array}{ccc} \alpha_1 & \alpha_2 & \cdots & \alpha_l \\ \beta_1 & \beta_2 & \cdots & \beta_l \end{array} \right\},\tag{18}$$

where the initial values of  $\alpha_j$  and  $\beta_j$  are all set as  $1/\sqrt{2}$  to meet the equity principle,  $\alpha_j^2 + \beta_j^2 = 1$  (j = 1, 2, ..., l), which is updated through the quantum revolving door during the iteration.

Conversion between quantum sequence and binary sequence is the key to convert FOA to QFOA. Randomly generate a random number of [0,1],  $rand_j$ , if  $rand_j \ge \alpha_j^2$ , the corresponding binary quantum bit value is 1, otherwise, 0, as shown in Equation (19):

$$x_j = \begin{cases} 1 & rand_j \ge \alpha_j^2 \\ 0 & else \end{cases}.$$
 (19)

Using the above method, the quantum sequence, *q*, can be transformed into a binary sequence, *x*; then the optimal parameter problem of an LS-SVR model can be determined using QFOA.

### (2) Quantum Fruit Fly Position Update Strategy

In the QFOA process, the position of quantum fruit flies represented by a quantum sequence is updated to find more feasible solutions and the best parameters. This paper uses quantum rotation to update the position of quantum fruit flies. The quantum position of individual *i* (there are in total *N* quantum fruit flies) can be extended from Equation (18) and is expressed as in Equation (20):

$$q_i = \left\{ \begin{array}{ccc} \alpha_{i1} & \alpha_{i2} & \cdots & \alpha_{il} \\ \beta_{i1} & \beta_{i2} & \cdots & \beta_{il} \end{array} \right\}, \tag{20}$$

where  $\alpha_{ij}^2 + \beta_{ij}^2 = 1$ ; i = 1, 2, ..., N; j = 1, 2, ..., l; and  $0 \le \alpha_{ij} \le 1, 0 \le \beta_{ij} \le 1$ .

Quantum rotation is a quantum revolving door determined by the quantum rotation angle, which updates the quantum sequence and conducts a random search around the position of quantum fruit flies to explore the local optimal solution. The  $\theta_{ij}^g$  is the *j*th quantum rotation angle of the population iterated to the *i*th fruit fly of generation, *g*, and the quantum bit  $q_{ij}^g$  (due to the nonnegative position constraint of  $q_{ij}^g$ , the absolute function, abs() is used to take the absolute value of each element in the calculation result) is updated according to the quantum revolving gate  $U(\theta_{ij}^g)$ , as shown in Equations (21) and (22) [34,35]:

$$q_{ij}^{g+1} = \operatorname{abs}\left(U\left(\theta_{ij}^{g+1}\right) \times q_{ij}^{g}\right)$$
(21)

$$U\left(\theta_{ij}^{g}\right) = \begin{bmatrix} \cos\theta_{ij}^{g} & -\sin\theta_{ij}^{g} \\ \sin\theta_{ij}^{g} & \cos\theta_{ij}^{g} \end{bmatrix}.$$
 (22)

In special cases, when the quantum rotation angle,  $\theta_{ij}^{g+1}$ , is equal to 0, the quantum bit,  $q_{ij}^{g+1}$ , uses quantum non-gate  $\overline{N}$  to update with some small probability, as indicated in Equation (23) [35]:

$$q_{ij}^{t+1} = \overline{N} \times q_{ij}^t = \begin{bmatrix} 0 & 1\\ 1 & 0 \end{bmatrix} \times q_{ij}^t.$$
(23)

#### 2.2.3. Chaotic Quantum Global Perturbation

For a bionic evolutionary algorithm, it is a general phenomenon that the population's diversity would be poor, along with the increased iterations. This phenomenon would also lead to being trapped into local optima during modeling processes. As mentioned, the chaotic mapping function can be employed to maintain the population's diversity to avoid trapping into local optima. Many studies have applied chaotic theory to improve the performances of these bionic evolutionary algorithms, such as the artificial bee colony (ABC) algorithm [41], and the particle swarm optimization (PSO) algorithm [42]. The authors have also employed the cat chaotic mapping function to improve the genetic algorithm (GA) [43], the PSO algorithm [44], and the bat algorithm [39], the results of which demonstrate that the searching quality of GA, PSO, ABC, and BA algorithms could be improved by employing chaotic mapping functions. Hence, the cat chaotic mapping function is once again used as the global chaotic perturbation strategy (GCPS) in this paper, and is hybridized with QFOA, namely CQFOA, which hybridizes GCPS with the QFOA while suffering from the problem of being trapped into local optima during the iterative modeling processes.

The two-dimensional cat mapping function is shown as in Equation (24) [39]:

$$\begin{cases} y^{t+1} = frac(y^t + z^t) \\ z^{t+1} = frac(y^t + 2z^t) \end{cases}$$
(24)

where *frac* function is used to calculate the fractional parts of a real number, *y*, by subtracting an approached integer.

The global chaotic perturbation strategy (GCPS) is illustrated as follows.

(1) **Generate 2popsize chaotic disturbance fruit flies.** For each *Fruit fly*<sup>*i*</sup> (I = 1, 2, ..., 2popsize), Equation (24) is applied to generate *d* random numbers,  $z_j$ , j = 1, 2, ..., d. Then, the qubit (with quantum state,  $|0\rangle$ ) amplitude,  $\cos \theta_i^i$ , of *Fruit fly*<sup>*i*</sup> is shown in Equation (25):

$$\cos\theta_i^i = y_j = 2z_j - 1. \tag{25}$$

- (2) **Select 0.5 popsize better chaotic disturbance fruit flies**. Compute the fitness value of each *Fruit fly* from 2 popsize chaotic disturbance fruit flies, and arrange these fruit flies to be a sequence based on the order of fitness values. Then, select the fruit flies with 0.5 popsize ranking ahead in the fitness values; as a result, the 0.5 popsize better chaotic disturbance fruit flies are obtained.
- (3) **Determine 0.5 popsize current fruit flies with better fitness**. Compute the fitness value of each *Fruit fly* from current QFOA, and arrange these fruit flies to be a sequence based on the order of fitness values. Then, select the fruit flies with 0.5 popsize ranking ahead in the fitness values.
- (4) **Form the new CQFOA population**. Mix the 0.5 popsize better chaotic disturbance fruit flies with 0.5 popsize current fruit flies with better fitness from current QFOA, and form a new population that contains new 1popsize fruit flies, and name it the new CQFOA population.
- (5) **Complete global chaotic perturbation**. After obtaining the new population of CQFOA, take it as the new population of QFOA and continue to execute the QFOA process.

#### 2.2.4. Implementation Steps of CQFOA

The steps of the proposed CQFOA for parameter optimization of an LS-SVR model are as follows as shown in Figure 1.

**Step 1. Initialization**. The population size of quantum *Drosophila* is 1 popsize; the maximum number of iterations is Gen-max; the random search radius is R; and the chaos disturbance control coefficient is  $N_{GCP}$ .

**Step 2. Random searching**. For quantum rotation angle,  $\theta_{ij}$ , of a random search, according to the quantum rotation angle, fruit fly locations on each dimension are updated, and then, a quantum revolving door is applied to update the quantum sequence, as shown in Equations (26) and (27) [34,35]:

$$\theta_{ij} = \theta(j) + R \times \operatorname{rand}(1) \tag{26}$$

$$q_{ij} = abs\left(\left[\begin{array}{cc}\cos\theta_{ij} & -\sin\theta_{ij}\\\sin\theta_{ij} & \cos\theta_{ij}\end{array}\right] \times Q(j)\right),\tag{27}$$

where *i* is an individual of quantum fruit flies, i = 1, 2, ..., 1 popsize; *j* is the position dimension of quantum fruit flies, j = 1, 2, ..., l. As mentioned above, the position of  $q_{ij}$  is non-negative constrained, thus, the absolute function, abs() is used to take the absolute value of each element in the calculation result.

**Step 3.** Calculating fitness. Mapping each Drosophila location,  $q_i$ , to the feasible domain of an LS-SVR model parameters to receive the parameters,  $(\gamma_i, \sigma_i)$ . The training data are used to complete the training processes of the LS – SVR<sub>i</sub> model and calculate the forecasting value in the training stage corresponding to each set of parameters. Then, the forecasting error is calculated as in Equation (12) of CQFOA by the mean absolute percentage error (MAPE), as shown in Equation (28):

$$MAPE = \frac{1}{N} \sum_{i=1}^{N} \left| \frac{f_i(x) - \hat{f}_i(x)}{f_i(x)} \right| \times 100\%,$$
(28)

where *N* is the total number of data points;  $f_i(x)$  is the actual load value at point *i*; and  $\hat{f}_i(x)$  is the forecasted load value at point *i*.

**Step 4.** Choosing the current optimum. Calculate the taste concentration of fruit fly, *Smell*<sub>*i*</sub>, by using Equation (12), and find the best flavor concentration of individual, *Best\_Smell*<sub>*i*</sub>, by Equation (13), as the optimal fitness value.

**Step 5.** Updating global optimization. Compare whether the contemporary odor concentration,  $Best\_Smell_{i=current}$ , is better than the global optima,  $Best\_Smell_i$ . If so, update the global value by Equation (14), and enable the individual quantum fruit fly to fly to the optimal position with vision, as in Equations (29) and (30), then go to **Step 6**. Otherwise, go to **Step 6** directly.

$$\theta_0 = \theta_{Best\_index} \tag{29}$$

$$q_0 = q_{Best\_index} \tag{30}$$

**Step 6. Global chaos perturbation judgment**. If the distance from the last disturbance is equal to  $N_{GCP}$ , go to **Step 7**; otherwise, go to **Step 8**.

**Step 7. Global chaos perturbation operations**. Based on the current population, conduct the global chaos perturbation algorithm to obtain the new CQFOA population. Then, take the new CQFOA population as the new population of QFOA, and continue to execute the QFOA process.

**Step 8. Iterative refinements**. Determine whether the current population satisfies the condition of evolutionary termination. If so, stop the optimization process and output the optimal results. Otherwise, repeat **Steps 2** to **8**.



Figure 1. Chaotic quantum FOA algorithm flowchart.

## 3. Forecasting Results

#### 3.1. Dataset of Experimental Examples

To test the performance of the proposed LS-SVR-CQFOA model, this paper employs the MEL data from an island data acquisition system in 2014 (IDAS 2014) [45] and the data of GEFCom2014-E [46] to carry out a numerical forecast. Taking the whole time of 24 h as the sampling interval, the load data contains 168-hour load values in total, i.e., from 01:00 14 July 2014 to 24:00 20 July 2014 in IDAS 2014 (namely IDAS 2014), and another two load datasets with the same 168-hour load values, i.e., from 01:00 1 January 2014 to 24:00 7 January 2014 (namely GEFCom2014 (Jan.)) and from 01:00 1 July 2014 to 24:00 7 July 2014 (namely GEFCom2014 (July)) in GEFCom2014-E, respectively.

The preciseness and integrity of historical data directly impact the forecasting accuracy. The data of the historical load are collected and obtained by electrical equipment. To some extent, the data

transmission and measurement will lead to some "bad data" in the data of historical load, which mainly includes missing and abnormal data. If these data are used for modeling, the establishment of load forecasting model and the forecasting will bring adverse effects. Thus, the preprocessing of historical data is essential to load forecasting. In this paper, before the numerical test, the data of the MEL are preprocessed, including: completing the missing data; identifying abnormal data; eliminating and replacing unreasonable data; and normalizing data. When the input of an LS-SVR model is multidimensional with a large data size (e.g., several orders of magnitude), it may lead to problems when using the raw data to implement model training directly. Therefore, it is essential that the sample data are normalized for processing, to keep all the sample data values in a certain interval (this topic limits [0,1]), ensuring that all of the data have the same order of magnitude.

The normalization of load data is converted according to Equation (31), where i = 1, 2, ..., N (*N* is the number of samples);  $x_i$  and  $y_i$  represent the values of before and after the normalization of sample data, respectively; and min( $x_i$ ) and max( $x_i$ ) represent the minimal and maximal values of sample data, respectively.

$$y_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$
(31)

After the end of the forecasting, it is necessary to use the inverse normalization equation to calculate the actual load value, as shown in Equation (32):

$$x_i = (\max(x_i) - \min(x_i))y_i + \min(x_i).$$
(32)

The normalized data of the values in IDAS 2014, GEFCom2014 (Jan.) and GEFCom2014 (July) are collected and shown in Tables 1–3, respectively.

Time	14 July	15 July	16 July	17 July	18 July	19 July	20 July
01:00	0.1617	0.1245	0.1526	0.2246	0.1870	0.3354	0.3669
02:00	0.0742	0.0000	0.0826	0.1590	0.1386	0.1924	0.1878
03:00	0.0000	0.0109	0.0000	0.0395	0.0381	0.1022	0.0919
04:00	0.0071	0.1278	0.0937	0.0000	0.0000	0.0000	0.0000
05:00	0.0531	0.1944	0.1419	0.1106	0.1218	0.1570	0.1770
06:00	0.0786	0.0611	0.0920	0.1428	0.1728	0.2558	0.2497
07:00	0.2636	0.1786	0.2724	0.3096	0.3788	0.4038	0.3943
08:00	0.3709	0.4417	0.3464	0.3586	0.4361	0.5129	0.4692
09:00	0.6872	0.5894	0.6549	0.7426	0.7970	0.6051	0.5829
10:00	0.9520	0.8746	0.9028	0.9055	0.9842	0.7632	0.7530
11:00	1.0000	0.9342	0.9650	0.9683	1.0000	0.8130	0.8332
12:00	0.9632	0.9730	0.9087	0.9217	0.9450	0.8935	0.8803
13:00	0.8552	1.0000	0.8135	0.8256	0.8821	0.8077	0.8122
14:00	0.8288	0.9152	0.9257	0.7377	0.8370	0.7185	0.7410
15:00	0.8224	0.8104	0.7663	0.7468	0.7961	0.6037	0.6882
16:00	0.8655	0.9448	0.8542	0.8099	0.8420	0.7347	0.7567
17:00	0.8552	0.7966	0.8340	0.8104	0.8323	0.7593	0.8439
18:00	0.9440	0.8809	0.9155	0.8976	0.9567	0.9286	0.9539
19:00	0.9574	0.8677	1.0000	0.9779	0.9694	0.9734	0.9741
20:00	0.9746	0.9693	0.9657	1.0000	0.9808	1.0000	1.0000
21:00	0.9372	0.8784	0.9236	0.9419	0.9546	0.9575	0.9664
22:00	0.8704	0.7697	0.7977	0.7889	0.8417	0.8634	0.8824
23:00	0.6328	0.5519	0.7193	0.6425	0.6655	0.5858	0.6035
24:00	0.3127	0.2114	0.2794	0.2559	0.3357	0.1080	0.0975

Table 1. Normalization values of load data for IDAS 2014.

Time	1 January	2 January	3 January	4 January	5 January	6 January	7 January
01:00	0.1769	0.0568	0.1127	0.1314	0.1648	0.0769	0.0532
02:00	0.0877	0.0206	0.0338	0.0480	0.0765	0.0222	0.0123
03:00	0.0234	0.0000	0.0000	0.0000	0.0087	0.0000	0.0000
04:00	0.0000	0.0084	0.0035	0.0044	0.0063	0.0076	0.0140
05:00	0.0175	0.0746	0.0634	0.0497	0.0268	0.0565	0.0862
06:00	0.0863	0.2155	0.2134	0.1368	0.0938	0.2122	0.2569
07:00	0.1835	0.4382	0.4345	0.3082	0.2090	0.4740	0.5389
08:00	0.2763	0.5802	0.5894	0.4813	0.3517	0.6277	0.6503
09:00	0.4028	0.6453	0.6972	0.6705	0.5039	0.6849	0.6581
10:00	0.5212	0.7110	0.7683	0.7860	0.6136	0.7300	0.6693
11:00	0.5819	0.7455	0.8106	0.8073	0.6333	0.7446	0.6861
12:00	0.6016	0.7751	0.8042	0.7726	0.6080	0.7573	0.6900
13:00	0.6089	0.7684	0.7592	0.6936	0.5623	0.7300	0.6788
14:00	0.5789	0.7712	0.7176	0.5950	0.5221	0.7078	0.6754
15:00	0.5563	0.7634	0.6887	0.5400	0.4937	0.6842	0.6676
16:00	0.5768	0.7556	0.6852	0.5560	0.5560	0.7109	0.6928
17:00	0.8165	0.8836	0.8479	0.7913	0.8060	0.8558	0.8411
18:00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
19:00	0.9810	0.9605	0.9845	0.9423	0.9416	0.9778	0.9955
20:00	0.8984	0.8686	0.8859	0.8188	0.8036	0.8920	0.9379
21:00	0.7807	0.7723	0.7908	0.7087	0.6672	0.7903	0.8489
22:00	0.5885	0.6114	0.6289	0.4982	0.4219	0.6112	0.6933
23:00	0.3596	0.4399	0.4303	0.2860	0.1774	0.4180	0.4980
24:00	0.1923	0.2957	0.2542	0.0719	0.0000	0.2764	0.3553

Table 2. Normalization values of load data for GEFCom2014 (Jan.).

Table 3. Normalization values of load data for GEFCom2014 (July).

Time	1 July	2 July	3 July	4 July	5 July	6 July	7 July
01:00	0.1562	0.1612	0.1583	0.2747	0.2636	0.1699	0.1063
02:00	0.0728	0.0882	0.0763	0.1302	0.1266	0.0857	0.0394
03:00	0.0238	0.0348	0.0232	0.0456	0.0554	0.0302	0.0054
04:00	0.0000	0.0000	0.0000	0.0000	0.0063	0.0000	0.0000
05:00	0.0222	0.0186	0.0181	0.0190	0.0000	0.0021	0.0302
06:00	0.0945	0.0957	0.1040	0.0589	0.0554	0.0154	0.1187
07:00	0.2811	0.2781	0.3143	0.2091	0.1872	0.0955	0.2972
08:00	0.4692	0.4736	0.5172	0.4316	0.4153	0.2521	0.4903
09:00	0.6244	0.6212	0.6637	0.6873	0.7008	0.4459	0.6424
10:00	0.7396	0.7516	0.7733	0.8878	0.9017	0.6131	0.7476
11:00	0.8306	0.8479	0.8722	0.9734	0.9561	0.7163	0.8425
12:00	0.8979	0.9209	0.9389	1.0000	0.9561	0.7570	0.9051
13:00	0.9378	0.9673	0.9678	0.9876	0.9111	0.7809	0.9434
14:00	0.9737	1.0000	0.9938	0.9287	0.8515	0.7928	0.9865
15:00	0.9879	0.9829	1.0000	0.8546	0.8243	0.8111	0.9995
16:00	0.9970	0.9290	0.9881	0.8032	0.8462	0.8574	1.0000
17:00	1.0000	0.8564	0.9423	0.8004	0.9195	0.9199	0.9962
18:00	0.9960	0.8101	0.9005	0.8279	0.9937	0.9853	0.9833
19:00	0.9687	0.7567	0.8672	0.8203	1.0000	1.0000	0.9579
20:00	0.9176	0.6907	0.7756	0.7386	0.9435	0.9579	0.9213
21:00	0.9044	0.6489	0.7377	0.6787	0.9362	0.9417	0.8975
22:00	0.8291	0.5461	0.6354	0.5428	0.8692	0.8687	0.7875
23:00	0.6138	0.3572	0.4262	0.3279	0.6883	0.6426	0.5701
24:00	0.4095	0.1678	0.2272	0.0913	0.4341	0.4213	0.3927

During the modeling processes, the load data are divided into three parts: the training set with the former 120 h, the validation set with the middle 24 h, and the testing set with the latter 24 h. Then, the rolling-based modeling procedure, proposed by Hong [18,47], is applied to assist CQFOA to look for appropriate parameters, ( $\gamma$ ,  $\sigma$ ), of an LS-SVR model during the training stage. Repeat this modeling procedure until all forecasting loads are received. The training error and the validation error can be

calculated simultaneously. The adjusted parameters, ( $\gamma$ ,  $\sigma$ ), would be selected as the most suitable parameters only with both the smallest validation and testing errors. The testing dataset is never used during the training and validation stages; it will only be used to calculate the forecasting accuracy. Eventually, the 24 h's load data are forecasted by the proposed LS-SVR-CQFOA model.

## 3.2. Forecasting Accuracy Indexes and Performance Tests

#### 3.2.1. Forecasting Accuracy Index

This study uses the *MAPE* (mentioned in Equation (28)), the root mean square error (RMSE), and the mean absolute error (MAE) as forecasting accuracy indexes. The RMSE and MAE are defined as in Equations (33) and (34), respectively:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} \left(f_i(x) - \hat{f}_i(x)\right)^2}{N}}$$
(33)

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| f_i(x) - \hat{f}_i(x) \right|,$$
(34)

where *N* is the total number of data points;  $f_i(x)$  is the actual value at point *i*; and  $\hat{f}_i(x)$  is the forecasting value at point *i*.

#### 3.2.2. Forecasting Performance Improvement Tests

To demonstrate the significant forecasting performances of the proposed model, Diebold and Mariano [48] and Derrac et al. [49] suggest that, for a small data size (24-h load forecasting) test, a Wilcoxon signed-rank test [50] is suitable. Thus, we decided to apply the Wilcoxon signed-rank test. For the same data size, a Wilcoxon test detects the significance of the difference (i.e., the forecasting errors from two forecasting models) in the central tendency. Therefore, let  $d_i$  be the absolute forecasting errors from any two models on ith forecasting value:  $R^+$  be the sum of ranks that  $d_i > 0$ ;  $R^-$  the sum of ranks that  $d_i < 0$ . If  $d_i = 0$ , then, remove this comparison and decrease the sample size. The statistics of Wilcoxon test, W, is calculated as in Equation (37):

$$W = \min\{R^+, R^-\}.$$
 (35)

If *W* is smaller than or equal to the critical value, based on the Wilcoxon distribution under *n* degrees of freedom, then the null hypothesis (i.e., equal performance from the two compared forecasting models) could not be accepted, i.e., the proposed model achieves significance.

#### 3.3. The Forecasting Results of the LS-SVR-CQFOA Model

#### 3.3.1. Parameter Setting of the CQFOA Algorithm

The parameters of the proposed CQFOA algorithm for the numerical example are set as follows: the population size, *popsize*, is set to 200; the maximal iteration, *gen-max*, is set to 1000; and the control coefficient of chaotic disturbance,  $N_{GCP}$ , is set to 15. These two parameters of the LS-SVR model are set as,  $\gamma \in [0, 1000]$ , and  $\sigma \in [0, 500]$ , respectively. The iterative time of each algorithm is set as the same to ensure the reliability of the forecasting results.

## 3.3.2. Results and Analysis

Considering the CQPSO, CQTS, and CQGA algorithms have been used to determine the parameters of an SVR-based load forecasting model in [36–39], those existing algorithms are also hybridized with an LS-SVR model to provide forecasting values to compare with the proposed

model here. These alternative models include LS-SVR-FOA, LS-SVR-QFOA, LS-SVR-CQPSO (LS-SVR hybridized with chaotic quantum particle swarm optimization algorithm [36]), LS-SVR-CQTS (LS-SVR hybridized with chaotic quantum Tabu search algorithm [37]), LS-SVR-CQGA (LS-SVR hybridized with chaotic quantum genetic algorithm [38]), and LS-SVR-CQBA (LS-SVR hybridized with chaotic quantum bat algorithm [39]), in order to compare the forecasting performance of LS-SVR-based models comprehensively, this article also selects BPNN method as a contrast model. The parameters of an LS-SVR model are selected by the CQPSO, CQTS, CQGA, CQBA, FOA, QFOA, and CQFOA algorithms, respectively. The details of the suitable parameters of all models for the IDAS 2014, the GEFCom2014 (Jan.) and the GEFCom2014 (July) data are shown in Tables 4–6, respectively.

**Table 4.** LS-SVR parameters, MAPE, and computing times of CQFOA and other algorithms for IDAS 2014.

Ontimization Algorithms	LS-SVR Parameters		- MARE of Validation (%)	Computing Times (s)	
Optimization Aigorithins	$\gamma$	σ		computing rimes (s)	
LS-SVR-CQPSO [36]	685	125	1.17	129	
LS-SVR-CQTS [37]	357	118	1.13	113	
LS-SVR-CQGA [38]	623	137	1.11	152	
LS-SVR-CQBA [39]	469	116	1.07	227	
LS-SVR-FOA	581	109	1.29	87	
LS-SVR-QFOA	638	124	1.32	202	
LS-SVR-CQFOA,	734	104	1.02	136	

**Table 5.** Parameters combination of LS-SVR determined by CQFOA and other algorithms for GEFCom2014 (Jan.).

Ontimization Algorithms	Parameters		- MARE of Validation (%)	Computation Times (s)	
Optimization Aigorithms	γ	σ		computation miles (3)	
LS-SVR-CQPSO [36]	574	87	0.98	134	
LS-SVR-CQTS [37]	426	68	1.02	109	
LS-SVR-CQGA [38]	653	98	0.95	155	
LS-SVR-CQBA [39]	501	82	0.9	231	
LS-SVR-FOA	482	94	1.54	82	
LS-SVR-QFOA	387	79	1.13	205	
LS-SVR-CQFOA,	688	88	0.86	132	

**Table 6.** Parameters combination of LS-SVR determined by CQFOA and other algorithms for GEFCom2014 (July).

Ontimization Algorithms	Parameters		- MAPE of Validation (%)	Computation Times (s)	
Optimization Aigorithms	γ	σ		companion miles (5)	
LS-SVR-CQPSO [36]	375	92	0.96	139	
LS-SVR-CQTS [37]	543	59	1.04	107	
LS-SVR-CQGA [38]	684	62	0.98	159	
LS-SVR-CQBA [39]	498	90	0.95	239	
LS-SVR-FOA	413	48	1.51	79	
LS-SVR-QFOA	384	83	1.07	212	
LS-SVR-CQFOA,	482	79	0.79	147	

Based on the same training settings, another representative model, the back-propagation neural network (BPNN) is compared with the proposed model. The forecasting results of these models mentioned above and the actual values for IDAS 2014, GEFCom2014 (Jan.) and GEFCom2014 (July) are given in Figures 2–4, respectively. This indicates that the proposed LS-SVR-CQFOA model achieves a better performance than the other alternative models, i.e., closer to the actual load values.

750

700

650

550

500

450

22:00

00:00

02:00

Value 009

Actualvalue LSSVRCQTS LSSVRCQBA



18:00

14:00

22:00

22:00

00:00

20:00

20:00

18:00

Time

16:00



00 17:00 Time

Value Value

06:00

04:00

08:00

650└ 00:91

10:00



Figure 3. Forecasting values of LS-SVR-CQFOA and other models for GEFCom2014 (Jan.).

02:00



Figure 4. Forecasting values of LS-SVR-CQFOA and other models for GEFCom2014 (July).

Tables 7–9 indicate the evaluation results from different forecasting accuracy indexes for IDAS 2014, GEFCom2014 (Jan.) and GEFCom2014 (July), respectively. For Table 7, the proposed LS-SVR-CQFOA model achieves smaller values for all employed accuracy indexes than the seven other models: RMSE (14.10), MAPE (2.21%), and MAE (13.88), respectively. For Table 8, similarly, the proposed LS-SVR-CQFOA model also achieves smaller values for all employed accuracy indexes compared to the seven other models: RMSE (40.62), MAPE (1.02%), and MAE (39.76), respectively. Similarly in Table 9, the proposed LS-SVR-CQFOA model also achieves smaller values for all employed accuracy indexes for all employed accuracy indexes than the other seven models: RMSE (38.70), MAPE (1.01%), and MAE (37.48), respectively. The details of the analysis results are as follows.

<b>Compared Models</b>	RMSE	<b>MAPE (%)</b>	MAE
BPNN	24.89	3.92	24.55
LS-SVR-CQPSO [36]	14.40	2.27	14.21
LS-SVR-CQTS [37]	14.50	2.26	14.24
LS-SVR-CQGA [38]	14.41	2.24	14.13
LS-SVR-CQBA [39]	14.45	2.25	14.18
LS-SVR-FOA	15.90	2.48	15.62
LS-SVR-QFOA	15.03	2.32	14.69
LS-SVR-CQFOA	14.10	2.21	13.88

Table 7. Forecasting indexes of LS-SVR-CQFOA and other models for IDAS 2014.

Table 8. Forecasting indexes of LS-SVR-CQFOA and other models for GEFCom2014 (Jan.).

Compared Models	RMSE	<b>MAPE (%)</b>	MAE
BPNN	92.30	2.34	90.74
LS-SVR-CQPSO [36]	51.46	1.31	50.69
LS-SVR-CQTS [37]	50.85	1.27	49.70
LS-SVR-CQGA [38]	46.36	1.16	45.31
LS-SVR-CQBA [39]	42.76	1.07	41.80
LS-SVR-FOA	75.55	1.89	73.88
LS-SVR-QFOA	59.74	1.47	57.96
LS-SVR-CQFOA	40.62	1.02	39.76

Table 9. Forecasting indexes of LS-SVR-CQFOA and other models for GEFCom2014 (July).

Finally, to test the significance in terms of forecasting accuracy improvements from the proposed LS-SVR-CQFOA model, the Wilcoxon signed-rank test is conducted under two significant levels,  $\alpha = 0.025$  and  $\alpha = 0.05$ , by one-tail test. The test results for the IDAS 2014, the GEFCom2014 (Jan.), and the GEFCom2014 (July) datasets are described in Tables 10–12, respectively. In these three tables, the results demonstrate that the proposed LS-SVR-CQFOA model achieved significantly better forecasting performance than the other alternative models. For example, in the IDAS 2014 dataset, for LS-SVR-CQFOA vs. LS-SVR-CQPSO, the statistic of Wilcoxon test, W = 72, is smaller than the critical statistics, W\*\* = 81 (under  $\alpha = 0.025$ ) and W\* = 91 (under  $\alpha = 0.05$ ), thus we could conclude that the proposed LS-SVR-CQFOA model is significantly outperform the LS-SVR-CQPSO model. In addition, the *p*-value = 0.022 is also smaller than the critical  $\alpha = 0.025$  and  $\alpha = 0.05$ , which support the conclusion.

Table 10. Results of Wilcoxon signed-rank test for IDAS 2014.

Compared Models	Wilcoxon Signed-Rank Test			
Compared Woders	$T_{0.025} = 81$	$T_{0.05} = 91$	<i>p</i> -Value	
LS-SVR-CQFOA vs. BPNN	0 <sup>T</sup>	0 <sup>T</sup>	0.000 **	
LS-SVR-CQFOA vs. LS-SVR-CQPSO	72 <sup>T</sup>	72 <sup>T</sup>	0.022 **	
LS-SVR-CQFOA vs. LS-SVR-CQTS	64 <sup>T</sup>	64 <sup>T</sup>	0.017 **	
LS-SVR-CQFOA vs. LS-SVR-CQGA	67 <sup>T</sup>	67 <sup>T</sup>	0.018 **	
LS-SVR-CQFOA vs. LS-SVR-CQBA	60 <sup>T</sup>	60 <sup>T</sup>	0.012 **	
LS-SVR-CQFOA vs. LS-SVR-FOA	50 <sup>T</sup>	50 <sup>T</sup>	0.009 **	
LS-SVR-CQFOA vs. LS-SVR-QFOA	68 <sup>T</sup>	68 <sup>T</sup>	0.019 **	

<sup>T</sup> Denotes that the LS-SVR-CQGA model significantly outperforms the other models. \*\* implies the *p*-value is lower than  $\alpha = 0.025$ ; \* implies the *p*-value is lower than  $\alpha = 0.05$ .

Compared Models	Wilcoxon Signed-Rank Test				
Compared Models	$T_{0.025} = 81$	$T_{0.05} = 91$	<i>p</i> -Value		
LS-SVR-CQFOA vs. BPNN	0 <sup>T</sup>	0 <sup>T</sup>	0.000 **		
LS-SVR-CQFOA vs. LS-SVR-CQPSO	74 <sup>T</sup>	74 <sup>T</sup>	0.023 **		
LS-SVR-CQFOA vs. LS-SVR-CQTS	75 <sup>T</sup>	75 <sup>T</sup>	0.024 **		
LS-SVR-CQFOA vs. LS-SVR-CQGA	78 <sup>T</sup>	78 <sup>T</sup>	0.026 **		
LS-SVR-CQFOA vs. LS-SVR-CQBA	80 <sup>T</sup>	80 <sup>T</sup>	0.027 **		
LS-SVR-CQFOA vs. LS-SVR-FOA	65 <sup>T</sup>	65 <sup>T</sup>	0.018 **		
LS-SVR-CQFOA vs. LS-SVR-QFOA	72 <sup>T</sup>	72 <sup>T</sup>	0.022 **		

Table 11. Results of Wilcoxon signed-rank test for GEFCom2014 (Jan.).

<sup>T</sup> Denotes that the LS-SVR-CQGA model significantly outperforms the other models. \*\* implies the *p*-value is lower than  $\alpha = 0.025$ ; \* implies the *p*-value is lower than  $\alpha = 0.05$ .

Compared Models	Wilcoxon Signed-Rank Test			
Compared Woders	$T_{0.025} = 81$	$T_{0.05} = 91$	<i>p</i> -Value	
LS-SVR-CQFOA vs. BPNN	0 <sup>T</sup>	0 <sup>T</sup>	0.000 **	
LS-SVR-CQFOA vs. LS-SVR-CQPSO	73 <sup>T</sup>	73 <sup>T</sup>	0.023 **	
LS-SVR-CQFOA vs. LS-SVR-CQTS	76 <sup>T</sup>	76 <sup>T</sup>	0.024 **	
LS-SVR-CQFOA vs. LS-SVR-CQGA	77 <sup>T</sup>	77 <sup>T</sup>	0.026 **	
LS-SVR-CQFOA vs. LS-SVR-CQBA	79 <sup>T</sup>	79 <sup>T</sup>	0.027 **	
LS-SVR-CQFOA vs. LS-SVR-FOA	65 <sup>T</sup>	65 <sup>T</sup>	0.018 **	
LS-SVR-CQFOA vs. LS-SVR-QFOA	71 <sup>T</sup>	71 <sup>T</sup>	0.022 **	

Table 12. Results of Wilcoxon signed-rank test for GEFCom2014 (July).

<sup>T</sup> Denotes that the LS-SVR-CQGA model significantly outperforms the other models. \*\* implies the *p*-value is lower than  $\alpha = 0.025$ ; \* implies the *p*-value is lower than  $\alpha = 0.05$ .

#### 4. Discussion

Taking the IDAS 2014 dataset as an example, firstly, the forecasting results of these LS-SVR-based models are all closer to the actual load values than the BPNN model. This shows that LS-SVR-based models can simulate nonlinear systems of microgrid load more accurately than the BPNN model, due to its advantages in dealing with nonlinear problems.

Secondly, in Table 4, the selected FOA and QFOA algorithms could achieve the best solution,  $(\gamma, \sigma) = (581, 109)$  and  $(\gamma, \sigma) = (638, 124)$ , with forecasting error, (RMSE = 15.93, MAPE = 2.48%, MAE = 15.63) and (RMSE = 14.87, MAPE = 2.32%, MAE = 14.61), respectively. However, the solution can be further improved by the proposed CQFOA algorithm to  $(\gamma, \sigma) = (734, 104)$  with more accurate forecasting performance, (RMSE = 14.10, MAPE = 2.21%, MAE = 13.88). Similar results could also be learned in the GEFCom2014 (Jan.) and the GEFCom2014 (July) from Tables 5 and 6, respectively. This illustrates that the proposed approach is feasible, i.e., hybridizing the FOA with QCM and chaotic mapping function to determine more appropriate parameters of an LS-SVR model to improve the forecasting accuracy.

Comparing the LS-SVR-QFOA model with the LS-SVR-FOA model, the forecasting accuracy of the LS-SVR-QFOA model is superior to that of the LS-SVR-FOA model. This demonstrates that the QCM empowers the fruit fly to have quantum behaviors, i.e., the QFOA find more appropriate parameters of an LS-SVR model, which improves the forecasting accuracy of the LS-SVR-FOA model in which the FOA is hybridized with an LS-SVR model. For example, in Table 4, the usage of the QCM in FOA changes the forecasting performances (RMSE = 15.93, MAPE = 2.48%, MAE = 15.63) of the LS-SVR-FOA model to the much better performance (RMSE = 14.87, MAPE = 2.32%, MAE = 14.61) of the LS-SVR-QFOA model. Similar results are demonstrated in the GEFCom2014 (Jan.) and the GEFCom2014 (July) from Tables 5 and 6, respectively.

For forecasting performance comparison between the LS-SVR-CQFOA and LS-SVR-QFOA models, the values of RMSE, MAPE, and MAE for the LS-SVR-CQFOA model are smaller than those of the LS-SVR-QFOA model. This reveals that the introduction of cat chaotic mapping function into QFOA plays a positive role in searching appropriate parameters when the population of QFOA algorithm is trapped into local optima. Then, the CQFOA finds more appropriate parameters. As a result, as shown in Table 4, employing CQFOA to select the parameters for an LS-SVR model markedly improves the performance (RMSE = 14.87, MAPE = 2.32%, MAE = 14.61) of the LS-SVR-QFOA model to the much better one (RMSE = 14.10, MAPE = 2.21%, MAE = 13.88) of the LS-SVR-CQFOA model. Similar results are illustrated in the GEFCom2014 (Jan.) and the GEFCom2014 (July) from Tables 5 and 6, respectively.

Comparing the time-consuming problem during the parameter searching processes in all the IDAS 2014, the GEFCom2014 (Jan.), and the GEFCom2014 (July) datasets, the proposed CQFOA is less than that of the CQGA and CQBA algorithms, but more than that of the CQPSO and CQTS algorithms. However, considering the time requirements of the actual application, the increase in time compared with CQPSO (more than 7 s) and CQTS (more than 23 s) is acceptable.

Finally, some limitations should be noticed. This paper only employs an existing dataset to establish the proposed model; thus, for different seasons, months, weeks, and dates, the electricity load patterns should be changed season by season, month by month, and week by week. For real-world applications, this paper should be a good beginning to guide planners and decision-makers to establish electricity load forecasting models overlapping the seasons, months, and weeks to achieve more comprehensive results. Thus, our planned future research direction is to explore the feasibility of hybridizing more powerful novel optimization frameworks (e.g., chaotic mapping functions, quantum computing mechanism, and hourly, daily, weekly, monthly adjusted mechanism) and novel meta-heuristic algorithms with an LS-SVR model to overcome the drawbacks of evolutionary algorithms to achieve excellent forecasting accuracy.

## 5. Conclusions

This paper proposes a novel hybrid forecasting model by hybridizing an LS-SVR model with the QCM, the cat chaotic mapping function, and the FOA. The forecasting results show that the proposed model achieves better performance than the alternative forecasting models, by hybridizing chaotic mapping function, QCM, and other evolutionary algorithms with an LS-SVR-based model. Employing the cat chaotic mapping function to enrich the diversity of searching scope and enhance the ergodicity of the population could successfully avoid trapping into local optima, and, also proves that applying QCM to overcome the limitations of the fruit fly's searching behaviors empowers the fruit fly to undertake quantum searching behaviors, thereby achieving more satisfactory results for MEL forecasting. The global chaotic perturbation strategy based on the cat mapping function is employed to jump out of local minima while the population of QFOA suffers from premature convergence, and also helps to improve the forecasting performance.

**Author Contributions:** M.-W.L. and W.-C.H. conceived and designed the experiments; G.J. and Z.Y. performed the experiments; M.-W.L. and W.-C.H. analyzed the data and wrote the paper.

**Funding:** Funding: This research was funded by the National Natural Science Foundation of China (51509056); the Heilongjiang Province Natural Science Fund (E2017028); the Fundamental Research Funds for the Central Universities (HEUCFG201813); the Open Fund of the State Key Laboratory of Coastal and Offshore Engineering (LP1610); Heilongjiang Sanjiang Project Administration Scientific Research and Experiments (SGZL/KY-08); and the Jiangsu Distinguished Professor Project (no. 9213618401), Jiangsu Normal University, Jiangsu Provincial Department of Education, China.

Acknowledgments: Ming-Wei Li, Jing Geng, and Yang Zhang acknowledge the support from the project grants: the National Natural Science Foundation of China (51509056); the Heilongjiang Province Natural Science Fund (E2017028); the Fundamental Research Funds for the Central Universities (HEUCFG201813); the Open Fund of the State Key Laboratory of Coastal and Offshore Engineering (LP1610); and Heilongjiang Sanjiang Project Administration Scientific Research and Experiments (SGZL/KY-08). Wei-Chiang Hong acknowledges the support from the Jiangsu Distinguished Professor Project (no. 9213618401) of Jiangsu Normal University, Jiangsu Provincial Department of Education, China.

Conflicts of Interest: The authors declare no conflict of interest.

#### References

- 1. Maçaira, P.M.; Souza, R.C.; Oliveira, F.L.C. Modelling and forecasting the residential electricity consumption in Brazil with pegels exponential smoothing techniques. *Procedia Comput. Sci.* 2015, *55*, 328–335. [CrossRef]
- Pappas, S.S.; Ekonomou, L.; Karampelas, P.; Karamousantas, D.C.; Katsikas, S.K.; Chatzarakis, G.E.; Skafidas, P.D. Electricity demand load forecasting of the Hellenic power system using an ARMA model. *Electr. Power Syst. Res.* 2010, *80*, 256–264. [CrossRef]
- Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* 2016, 130, 139–147. [CrossRef]
- Chen, Y.; Luh, P.B.; Guan, C.; Zhao, Y.; Michel, L.D.; Coolbeth, M.A.; Friedland, P.B.; Rourke, S.J. Short-term load forecasting: Similar day-based wavelet neural networks. *IEEE Trans. Power Syst.* 2010, 25, 322–330. [CrossRef]

- 5. Li, S.; Wang, P.; Goel, L. Short-term load forecasting by wavelet transform and evolutionary extreme learning machine. *Electr. Power Syst. Res.* 2015, 122, 96–103. [CrossRef]
- Fan, G.F.; Wang, A.; Hong, W.C. Combining grey model and self-adapting intelligent grey model with genetic algorithm and annual share changes in natural gas demand forecasting. *Energies* 2018, 11, 1625. [CrossRef]
- 7. Ma, X.; Liu, Z. Application of a novel time-delayed polynomial grey model to predict the natural gas consumption in China. *J. Comput. Appl. Math.* **2017**, *324*, 17–24. [CrossRef]
- 8. Lou, C.W.; Dong, M.C. A novel random fuzzy neural networks for tackling uncertainties of electric load forecasting. *Int. J. Electr. Power Energy Syst.* **2015**, *73*, 34–44. [CrossRef]
- 9. Ertugrul, Ö.F. Forecasting electricity load by a novel recurrent extreme learning machines approach. *Int. J. Electr. Power Energy Syst.* 2016, *78*, 429–435. [CrossRef]
- 10. Geng, J.; Huang, M.L.; Li, M.W.; Hong, W.C. Hybridization of seasonal chaotic cloud simulated annealing algorithm in a SVR-based load forecasting model. *Neurocomputing* **2015**, *151*, 1362–1373. [CrossRef]
- 11. Hooshmand, R.A.; Amooshahi, H.; Parastegari, M. A hybrid intelligent algorithm Based short-term load forecasting approach. *Int. J. Electr. Power Energy Syst.* **2013**, *45*, 313–324. [CrossRef]
- 12. Niu, D.X.; Shi, H.; Wu, D.D. Short-term load forecasting using Bayesian neural networks learned by hybrid Monte Carlo algorithm. *Appl. Soft Comput.* **2012**, *12*, 1822–1827. [CrossRef]
- 13. Hanmandlu, M.; Chauhan, B.K. Load forecasting using hybrid models. *IEEE Trans. Power Syst.* 2011, 26, 20–29. [CrossRef]
- Mahmoud, T.S.; Habibi, D.; Hassan, M.Y.; Bass, O. Modelling self-optimised short term load forecasting for medium voltage loads using tunning fuzzy systems and artificial neural networks. *Energy Convers. Manag.* 2015, 106, 1396–1408. [CrossRef]
- 15. Suykens, J.A.K.; Vandewalle, J.; De Moor, B. Optimal control by least squares support vector machines. *Neural Netw.* **2001**, *14*, 23–35. [CrossRef]
- 16. Sankar, R.; Sapankevych, N.I. Time series prediction using support vector machines: A survey. *IEEE Comput. Intell. Mag.* **2009**, *4*, 24–38.
- 17. Vapnik, V.N. The Nature of Statistical Learning Theory; Springer: New York, NY, USA, 1995.
- 18. Hong, W.C. Electric load forecasting by seasonal recurrent LS-SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy* **2011**, *36*, 5568–5578. [CrossRef]
- 19. Fan, G.F.; Peng, L.L.; Zhao, X.; Hong, W.C. Applications of hybrid EMD with PSO and GA for an SVR-based load forecasting model. *Energies* **2017**, *10*, 1713. [CrossRef]
- 20. Suykens, J.A.K.; Vanddewalle, J. Least squares support vector machines classifiers. *Neural Netw. Lett.* **1999**, 19, 293–300. [CrossRef]
- Wang, J.; Hu, J. A robust combination approach for short-term wind speed forecasting and analysis—Combination of the ARIMA (Autoregressive Integrated Moving Average), ELM (Extreme Learning Machine), SVM (Support Vector Machine) and LSSVM (Least Square SVM) forecasts using a GPR (Gaussian Process Regression) model. *Energy* 2015, *93*, 41–56.
- 22. Hong, W.C.; Dong, Y.; Zhang, W.; Chen, L.Y.; Panigrahi, B.K. Cyclic electric load forecasting by seasonal LS-SVR with chaotic genetic algorithm. *Int. J. Electr. Power Energy Syst.* **2013**, *44*, 604–614. [CrossRef]
- 23. Ju, F.Y.; Hong, W.C. Application of seasonal SVR with chaotic gravitational search algorithm in electricity forecasting. *Appl. Math. Model.* **2013**, *37*, 9643–9651. [CrossRef]
- 24. Fan, G.; Peng, L.L.; Hong, W.C.; Sun, F. Electric load forecasting by the SVR model with differential empirical mode decomposition and auto regression. *Neurocomputing* **2016**, *173*, 958–970. [CrossRef]
- 25. Pan, W.T. Fruit Fly Optimization Algorithm; Tsanghai Publishing: Taipei, Taiwan, China, 2011.
- 26. Pan, W.T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowl.-Based Syst.* **2012**, *26*, 69–74. [CrossRef]
- Mitić, M.; Vuković, N.; Petrović, M.; Miljković, Z. Chaotic fruit fly optimization algorithm. *Knowl.-Based Syst.* 2015, *89*, 446–458. [CrossRef]
- 28. Wu, L.; Liu, Q.; Tian, X.; Zhang, J.; Xiao, W. A new improved fruit fly optimization algorithm IAFOA and its application to solve engineering optimization problems. *Knowl.-Based Syst.* **2018**, *144*, 153–173. [CrossRef]
- 29. Han, X.; Liu, Q.; Wang, H.; Wang, L. Novel fruit fly optimization algorithm with trend search and co-evolution. *Knowl.-Based Syst.* 2018, 141, 1–17. [CrossRef]

- 30. Zhang, X.; Lu, X.; Jia, S.; Li, X. A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning. *Appl. Soft Comput.* **2018**, *70*, 371–388. [CrossRef]
- Han, S.Z.; Pan, W.T.; Zhou, Y.Y.; Liu, Z.L. Construct the prediction model for China agricultural output value based on the optimization neural network of fruit fly optimization algorithm. *Future Gener. Comput. Syst.* 2018, *86*, 663–669. [CrossRef]
- Yang, X.S.; Gandomi, A.H. Bat algorithm: A novel approach for global engineering optimization. *Eng. Comput.* 2012, 29, 464–483. [CrossRef]
- Narayanan, A.; Moore, M. Quantum-inspired genetic algorithms. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 61–66.
- Han, K.H.; Kim, J.H. Genetic quantum algorithm and its application to combinatorial optimization problem. In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; pp. 1354–1360.
- 35. Han, K.H.; Kim, J.H. Quantum-inspired evolutionary algorithm for a class of combinatorial optimization. *IEEE Trans. Evol. Comput.* **2002**, *6*, 580–593. [CrossRef]
- 36. Huang, M.L. Hybridization of chaotic quantum particle swarm optimization with SVR in electric demand forecasting. *Energies* **2016**, *9*, 426. [CrossRef]
- 37. Lee, C.W.; Lin, B.Y. Application of hybrid quantum tabu search with support vector regression for load forecasting. *Energies* **2016**, *9*, 873. [CrossRef]
- 38. Lee, C.W.; Lin, B.Y. Applications of the chaotic quantum genetic algorithm with support vector regression in load forecasting. *Energies* **2017**, *10*, 1832. [CrossRef]
- 39. Li, M.W.; Geng, J.; Wang, S.; Hong, W.C. Hybrid chaotic quantum bat algorithm with SVR in electric load forecasting. *Energies* **2017**, *10*, 2180. [CrossRef]
- 40. Shi, D.Y.; Lu, L.J. A judge model of the impact of lane closure incident on individual vehicles on freeways based on RFID technology and FOA-GRNN method. *J. Wuhan Univ. Technol.* **2012**, *34*, 63–68.
- 41. Yuan, X.; Wang, P.; Yuan, Y.; Huang, Y.; Zhang, X. A new quantum inspired chaotic artificial bee colony algorithm for optimal power flow problem. *Energy Convers. Manag.* **2015**, *100*, 1–9. [CrossRef]
- 42. Peng, A.N. Particle swarm optimization algorithm based on chaotic theory and adaptive inertia weight. *J. Nanoelectron. Optoelectron.* **2017**, *12*, 404–408. [CrossRef]
- 43. Li, M.W.; Geng, J.; Hong, W.C.; Chen, Z.Y. A novel approach based on the Gauss-vLS-SVR with a new hybrid evolutionary algorithm and input vector decision method for port throughput forecasting. *Neural Comput. Appl.* **2017**, *28*, S621–S640. [CrossRef]
- 44. Li, M.W.; Hong, W.C.; Geng, J.; Wang, J. Berth and quay crane coordinated scheduling using chaos cloud particle swarm optimization algorithm. *Neural Comput. Appl.* **2017**, *28*, 3163–3182. [CrossRef]
- Xiong, Y. Study on Short-Term Micro-Grid Load Forecasting Based on IGA-PSO RBF Neural Network. Master's Thesis, South China University of Technology, Guangzhou, China, 2016.
- 46. Hong, T.; Pinson, P.; Fan, S.; Zareipour, H.; Troccoli, A.; Hyndman, R.J. Probabilistic energy forecasting: Global Energy Forecasting Competition 2014 and beyond. *Int. J. Forecast.* **2016**, *32*, 896–913. [CrossRef]
- 47. Hong, W.C. Application of seasonal SVR with chaotic immune algorithm in traffic flow forecasting. *Neural Comput. Appl.* **2012**, *21*, 583–593. [CrossRef]
- 48. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. J. Bus. Econ. Stat. 1995, 13, 134–144.
- Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* 2011, 1, 3–18. [CrossRef]
- 50. Wilcoxon, F. Individual comparisons by ranking methods. Biom. Bull. 1945, 1, 80–83. [CrossRef]



© 2018 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 (http://creativecommons.org/licenses/by/4.0/).