


Article

A Hybrid Seasonal Mechanism with a Chaotic Cuckoo Search Algorithm with a Support Vector Regression Model for Electric Load Forecasting

Yongquan Dong, Zichen Zhang and Wei-Chiang Hong * 

School of Computer Science and Technology (School of Education Intelligent Technology), Jiangsu Normal University/101, Shanghai Rd., Tongshan District, Xuzhou 221116, Jiangsu, China; tomдық@jnsu.edu.cn (Y.D.); zzcckzzw@126.com (Z.Z.)

* Correspondence: samuelsonhong@gmail.com; Tel.: +86-516-8350-0307

Received: 24 March 2018; Accepted: 18 April 2018; Published: 20 April 2018



Abstract: Providing accurate electric load forecasting results plays a crucial role in daily energy management of the power supply system. Due to superior forecasting performance, the hybridizing support vector regression (SVR) model with evolutionary algorithms has received attention and deserves to continue being explored widely. The cuckoo search (CS) algorithm has the potential to contribute more satisfactory electric load forecasting results. However, the original CS algorithm suffers from its inherent drawbacks, such as parameters that require accurate setting, loss of population diversity, and easy trapping in local optima (i.e., premature convergence). Therefore, proposing some critical improvement mechanisms and employing an improved CS algorithm to determine suitable parameter combinations for an SVR model is essential. This paper proposes the SVR with chaotic cuckoo search (SVRCCS) model based on using a tent chaotic mapping function to enrich the cuckoo search space and diversify the population to avoid trapping in local optima. In addition, to deal with the cyclic nature of electric loads, a seasonal mechanism is combined with the SVRCCS model, namely giving a seasonal SVR with chaotic cuckoo search (SSVRCCS) model, to produce more accurate forecasting performances. The numerical results, tested by using the datasets from the National Electricity Market (NEM, Queensland, Australia) and the New York Independent System Operator (NYISO, NY, USA), show that the proposed SSVRCCS model outperforms other alternative models.

Keywords: support vector regression; tent chaotic mapping function; cuckoo search algorithm; seasonal mechanism; load forecasting

1. Introduction

Accurate electric load forecasting is important to facilitate the decision-making process for power unit commitment, economic load dispatch, power system operation and security, contingency scheduling, and so on [1,2]. As indicated in existing papers, a 1% electric load forecasting error increase would lead to a £10 million additional operational cost [3], on the contrary, decreasing forecasting errors by 1% would produce appreciable operation benefits [2]. Therefore, looking for more accurate forecasting models or applying novel intelligent algorithms to achieve satisfactory load forecasting results, to optimize the decisions of electricity supplies and load plans, to improve the efficiency of the power system operations, eventually, reduces the system risks to within a controllable range. However, due to lots of factors, such as energy policy, urban population, socio-economical activities, weather conditions, holidays, and so on [4], the electric load data display seasonality, non-linearity, and a chaotic nature, which complicates electric load forecasting work [5].

Lots of electric load forecasting models have been proposed to continue improving forecasting performances. These forecasting models can be of two types, the first one is based on the statistical methodology, and the other one involves applications of artificial intelligence technology. The statistical models, which include the ARIMA models [6,7], regression models [8,9], exponential smoothing models [10], Kalman filtering models [11,12], Bayesian estimation models [13,14], and so on use historical data to find out the linear relationships among time periods. However, due to their theoretical definitions, these statistical models can only deal well with linear relationships among electric loads and the other factors mentioned above. Therefore, these models could only produce unsatisfactory forecasting performances [15].

Due to its superior nonlinear processing capability, artificial intelligence technology methods such as artificial neural networks (ANNs) [16,17], expert system models [18,19], and fuzzy inference systems [20,21] have been widely applied to improve the performance of electric load forecasting. To overcome the inherent shortcomings of these artificial intelligent models, hybrid models (hybridizing two artificial intelligent models with each other) and combined models (combining two models with each other) have been the research hotspots recently. For example, hybridized or combined with each other models [22] and with evolutionary algorithms [23]. However, these artificial intelligence models (including hybrid or combined models) also have shortcomings themselves, such as being time consuming, difficult to determine structural parameters, and trapping into local minima. Readers may refer to [24] for more discussions regarding load forecasting.

With outstanding nonlinear processing capability, composed of high dimensional mapping ability and kernel computing technology, the support vector regression (SVR) model [25–27] has already produced superior abundant application results in many fields. The application experience demonstrates that an SVR model with well-computed parameters by any evolutionary algorithm could provide significant satisfactory forecasting performance, and overcome the shortcomings of evolutionary algorithms to compute appropriate parameters. For applications in electric load forecasting, Hong and his successors [28,29] have used two types of chaotic mapping functions (i.e., logistic function and cat mapping function) to keep the diversity of population during the search process to avoid trapping into local optima, to significantly improve the forecasting accuracy level.

The cuckoo search (CS) algorithm [30] is a novel meta-heuristic optimization algorithm inspired by the brood reproductive strategy of cuckoo birds via an interesting brood parasitic mechanism, i.e., mimicking the pattern and color of the host's eggs, throwing the eggs out or not, or building a new nest, etc. In [31], the authors demonstrate that, by applying various test functions, it is superior to other algorithms, such as genetic algorithm (GA), differential evolution (DE), simulated annealing (SA) algorithm, and particle swarm optimization (PSO) algorithm in searching for a global optimum. Nowadays, the CS algorithm is widely applied in engineering applications, such as unit maintenance scheduling [32], data clustering optimization [33], medical image recognition [34], manufacturing engineering optimization [35], and software cost estimation [36], etc. However, as mentioned in [37], the original CS algorithm has some inherent limitations, such as its initialization settings of the host nest location, Lévy flight parameter, and boundary handling problem. In addition, because it is a population-based optimization algorithm, the original CS algorithm also suffers from slow convergence rate in the later searching period, homogeneous searching behaviors (low diversity of population), and a premature convergence tendency [33,38,39].

Due to its easy implementation and ability to enrich the cuckoo search space and diversify the population to avoid trapping into local optima, this paper would like to apply a chaotic mapping function to overcome the core shortcomings of the original CS algorithm, to produce more accurate electric load forecasting results. Thus, a tent chaotic mapping function, demonstrating a range of dynamical behavior ranging from predictable to chaos, is hybridized with a CS algorithm to determine three parameters of an SVR model. A new electric load forecasting model, obtained by hybridizing a tent chaotic mapping function and CS algorithm with an SVR model, namely the SVR with chaotic cuckoo search (SVRCCS) model, is thus proposed. In the meanwhile, as mentioned in existing

papers [5,28,29], electric load data, particularly short term load data, illustrate an obvious cyclic tendency, thus, the seasonal mechanism proposed in the authors' previous papers [5,28,29] would be further improved and combined with the SVRCCS model. Finally, the proposed seasonal SVR with CCS, namely the SVR with chaotic cuckoo search (SSVRCCS) model, is employed to improve the forecasting accuracy level by sufficiently capturing the non-linear and cyclic tendency of electric load changes. Furthermore, the forecasting results of the SSVRCCS model are used to compare them with other alternative models, such as the SARIMA, GRNN, SVRCCS, and SVRCS models, to test the forecasting accuracy improvements achieved. The principal contribution of this paper is in continuing to hybridize the SVR model with a tent chaotic computing mechanism, CS algorithm, and eventually, combine a seasonal mechanism, to widely explore the electric load forecasting model to produce higher accuracy performances.

The remainder of this article is organized as follows: the basic formulation of an SVR model, the proposed CCS algorithm, seasonal mechanism, and the modeling details of the proposed SSVRCCS model are described in Section 2. A numerical example and forecasting accuracy comparisons among the proposed model and other alternative models are presented in Section 3. Finally, conclusions are given in Section 4.

2. The Proposed SVR with Chaotic Cuckoo Search (SSVRCCS) Model

2.1. Support Vector Regression (SVR) Model

The modeling details of an SVR model are presented briefly as follows. The training data set, $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, is mapped into a high dimensional feature space by a non-linear mapping function, $\varphi(\mathbf{x})$. Then, in the high dimensional feature space, the SVR function, f , is theoretically used to formulate the nonlinear relationships between the input training data (\mathbf{x}_i) and the output data (y_i). This can be shown as Equation (1):

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

where $f(\mathbf{x})$ represents the forecasted values; the weight, \mathbf{w} , and the coefficient, b , are computed along with minimizing the empirical risk, as shown in Equation (2):

$$R(f) = C \frac{1}{N} \sum_{i=1}^N \Theta_{\varepsilon}(y_i, f(\mathbf{x}_i)) + \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (2)$$

$$\Theta_{\varepsilon}(\mathbf{y}, f(\mathbf{x})) = \begin{cases} 0, & \text{if } |f(\mathbf{x}) - \mathbf{y}| \leq \varepsilon \\ |f(\mathbf{x}) - \mathbf{y}| - \varepsilon, & \text{otherwise} \end{cases} \quad (3)$$

where $\Theta_{\varepsilon}(\mathbf{y}, f(\mathbf{x}))$ is so-called ε -insensitive loss function, as shown in Equation (3). It is used to determine the optimal hyperplane to separate the training data into two subsets with maximal distance, i.e., minimizing the training errors between these two separated training data subsets and $\Theta_{\varepsilon}(\mathbf{y}, f(\mathbf{x}))$, respectively. C is a parameter to penalize the training errors. The second term, $\frac{1}{2} \mathbf{w}^T \mathbf{w}$, is then used to represent the maximal distance between mentioned two separated data subsets, meanwhile, it also determines the steepness and the flatness of $f(\mathbf{x})$.

Then, the SVR modeling problem could be demonstrated as minimizing the total training errors. It is a quadratic programming problem with two slack variables, ξ and ξ^* , to measure the distance between the training data values and the edge values of ε -tube. Training errors under ε are denoted as ξ^* , whereas training errors above ε are denoted as ξ , as shown in Equation (4):

$$\text{Min}_{\mathbf{w}, \xi, \xi^*} R(\mathbf{w}, \xi, \xi^*) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*) \quad (4)$$

with the constraints:

$$\begin{aligned}
y_i - f(\mathbf{x}_i) &\leq \varepsilon + \zeta_i^*, \\
-y_i - f(\mathbf{x}_i) &\leq \varepsilon + \zeta_i, \\
\zeta_i^* &\geq 0 \\
\zeta_i &\geq 0 \\
i &= 1, 2, \dots, N
\end{aligned}$$

The solution of Equation (4) is optimized by using Lagrange multipliers, β_i^* , and β_i , the weight vector, \mathbf{w} , in Equation (1) is computed as Equation (5):

$$\mathbf{w}^* = \sum_{i=1}^N (\beta_i^* - \beta_i) \varphi(\mathbf{x}_i) \quad (5)$$

Eventually, the SVR forecasting function is calculated as Equation (6):

$$f(\mathbf{x}) = \sum_{i=1}^N (\beta_i^* - \beta_i) K(\mathbf{x}_i, \mathbf{x}_j) + b \quad (6)$$

where $K(\mathbf{x}_i, \mathbf{x}_j)$ is the so-called kernel function, and its value could be computed by the inner product of $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \times \varphi(\mathbf{x}_j)$. There are several kinds of kernel function, such as Gaussian function (Equation (7)) and the polynomial kernel function. Due to its superior ability to map nonlinear data into high dimensional space, a Gaussian function is used in this paper:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right) \quad (7)$$

Therefore, determining the three parameters, σ , C , and ε of an SVR model would play the critical role to achieve more accurate forecasting performances [5,28,29]. The parameter ε decides the number of support vectors. If ε is large enough, it implies few support vectors with low forecasting accuracy; if ε has a value that is too small, it would increase the forecasting accuracy but be too complex to adopt. Parameter C , as mentioned, penalizes the training errors. If C is large enough, it would increase the forecasting accuracy but suffer from being difficult to adopt; if C has a too small value, the model would suffer from large training errors. Parameter σ represents the relationships among data and the correlations among support vectors. If σ is large enough, the correlations among support vectors are strong and we can obtain accurate forecasting results, but if the value of σ is small, the correlations among support vectors are weak, and adoption is difficult.

However, structural methods to determine the SVR parameters are lacking. Hong and his colleagues have pointed out the advanced exploration way by hybridizing chaotic mapping functions with evolutionary algorithms to overcome the embedded premature convergence problem, to select suitable parameter combination, to achieve highly accurate forecasting performances. To continue this valuable exploration, the chaotic cuckoo search algorithm, the CCS algorithm, is proposed to be hybridized with an SVR model to determine an appropriate parameter combination.

2.2. Chaotic Cuckoo Search (CCS) Algorithm

2.2.1. Tent Chaotic Mapping Function

The chaotic mapping function is an optimization technique to map the original data series to show sensitive dependence on the initial conditions and infinite different periodic responses (chaotic ergodicity), to maintain the diversity of population in the whole optimization procedures, to enrich the search behavior, and to avoid premature convergence. The most popular chaotic mapping function is the logistic function, however, based on the analysis on the chaotic characteristics of the different mapping functions, a tent chaotic mapping function [39] demonstrates a range of dynamical behavior

ranging from predictable to chaos, i.e., with good ergodic uniformity [40]. This paper thus applies the tent chaotic mapping function to be hybridized with the CS algorithm to determine the three parameters of an SVR model.

The tent chaotic mapping function is shown as Equation (8):

$$x_{n+1} = \begin{cases} 2x_n & x \in [0, 0.5] \\ 2(1 - x_n) & x \in (0.5, 1] \end{cases} \quad (8)$$

where x_n is the iterative value of the variable x in the n th step, and n is the number of iteration steps.

2.2.2. Cuckoo Search (CS) Algorithm

The CS algorithm is a novel meta-heuristic optimization algorithm, inspired by cuckoo birds' obligate brood parasitic behavior of laying their eggs in the nests of other host birds. Meanwhile, by applying Lévy flight behaviors, the search speed is much faster than that of the normal random walk. Therefore, cuckoo birds can reduce the number of iterations and thus speed up the local search efficiency. For CS algorithm implementation, each egg in a nest represents a potential solution. The cuckoo birds could choose, by Lévy flight behaviors, recently-spawned nests to lay their eggs in the host nests to ensure their eggs could hatch first due to the natural phenomenon that cuckoo eggs usually hatch before the host birds' eggs. It takes times for the host birds to discover that the eggs in their nests do not belong to them, based on the probability, p_a . When these "stranger" eggs are discovered, they either throw out those eggs or abandon the whole nest to build a new nest in a new location. The cuckoo birds would continuously lay new eggs (solutions), and they would choose the nest, by Lévy flight behaviors, around the current best solutions.

The CS algorithm contains three famous idealized rules [31]: (1) each cuckoo lays one egg at a time in a randomly selected host; (2) high-quality eggs and their host nests would survive to the next generation; (3) the number of available host nests is fixed, and the host bird detects the "stranger" egg with a probability $p_a \in [0, 1]$. In this case, the host bird can either throw away the egg or abandon the nest, and build a completely new nest. The last rule can be approximated by a fraction (p_a) of the n host nests that are replaced by new nests (with new random solutions). The value of p_a is often set as 0.25 [37].

The CS algorithm could maintain the balance between two kinds of search (random walks), the local search and the global search, by a switching parameter, p_a . The switching parameter p_a determines the cuckoo birds to abandon a fraction of the worst nests and build new ones for discovering new and more promising regions in the search space. These two random walks are defined by Equations (9) and (10), respectively:

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \delta) \otimes (x_j^t - x_k^t) s \quad (9)$$

$$x_i^{t+1} = x_i^t + \alpha \mathcal{L}(s, \lambda) \quad (10)$$

where x_j^t and x_k^t are current positions randomly selected; α is the positive Lévy flight step size scaling factor; s is the step size; $H(\cdot)$ is the Heavy-side function; δ is a random number from uniform distribution; \otimes represents the entry-wise product of two vectors; $\mathcal{L}(s, \lambda)$ is the Lévy distribution and is used to define the step size of random walk, it is defined as Equation (11):

$$\mathcal{L}(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}} \quad (11)$$

where λ is the standard deviation of step size; the gamma function, $\Gamma(\lambda)$, is defined as $\Gamma(\lambda) = \int_0^\infty t^{\lambda-1} e^{-t} dt$, and represents an extension of factorial function, if λ is a positive integer, then, $\Gamma(\lambda) = (\lambda - 1)!$. Lévy flight distribution enables a series of straight jumps chosen from any flight

movements, it is also capable to find out the global optimum, i.e., it could ensure that the system will not be trapped in a local optimum [41].

2.2.3. Implementation Steps of CCS Algorithm

The procedure of the hybrid CCS algorithm with an SVR model is illustrated as followings. The relevant flowchart is shown in Figure 1.

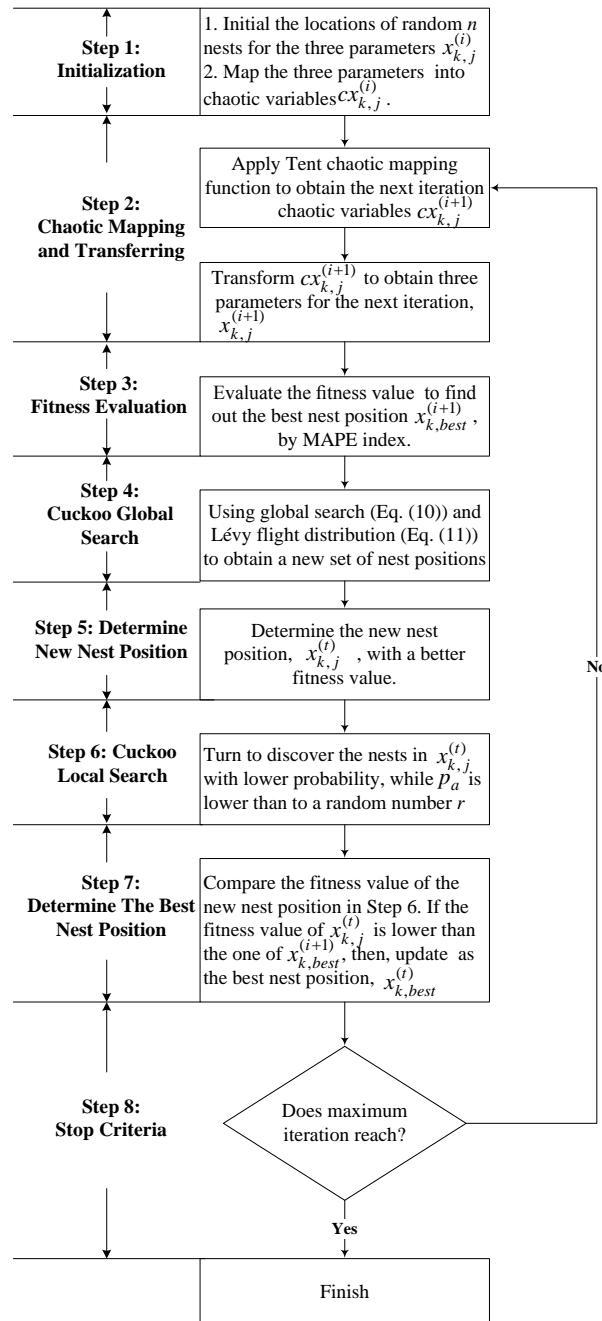


Figure 1. Chaotic cuckoo search algorithm flowchart.

Step 1: Initialization.

The locations of random n nests for the three parameters of an SVR model as $x_{k,j}^{(i)} = [x_{k,1}^{(i)}, x_{k,2}^{(i)}, \dots, x_{k,n}^{(i)}]^T$, $k = C, \sigma, \varepsilon$; i represents the iteration number; j represents the number of

nests. Let $i = 0$, and normalize the parameters as chaotic variables, $cx_{k,j}^{(i)}$, within the interval $[0, 1]$ by Equation (12):

$$cx_{k,j}^{(i)} = \frac{x_{k,j}^{(i)} - Min_k}{Max_k - Min_k} \quad (12)$$

where Min_k and Max_k are the minima and the maxima of the three parameters, respectively.

Step 2: Chaotic Mapping and Transferring.

Apply the tent chaotic mapping function, defined as Equation (8), to obtain the next iteration of chaotic variables, $cx_{k,j}^{(i+1)}$, as shown in Equation (13):

$$cx_{k,j}^{(i+1)} = \begin{cases} 2cx_{k,j}^{(i)} & cx_{k,j}^{(i)} \in [0, 0.5] \\ 2(1 - cx_{k,j}^{(i)}) & cx_{k,j}^{(i)} \in (0.5, 1] \end{cases} \quad (13)$$

Then, transform $cx_{k,j}^{(i+1)}$ to obtain three parameters for the next iteration, $x_{k,j}^{(i+1)}$, by the following Equation (14):

$$x_{k,j}^{(i+1)} = Min_k + cx_{k,j}^{(i+1)}(Max_k - Min_k) \quad (14)$$

Step 3: Fitness Evaluation.

Evaluate the fitness value with $x_{k,j}^{(i+1)}$ for all nests to find out the best nest position, $x_{k,best}^{(i+1)}$, in terms of smaller forecasting accuracy index value. In this paper, the forecasting error is calculated as the fitness value by the mean absolute percentage error (MAPE), as shown in Equation (15):

$$MAPE = \frac{1}{N} \sum_{i=1}^N \left| \frac{a_i - f_i}{a_i} \right| \times 100\% \quad (15)$$

where N is the total number of data; a_i is the actual electric load value at point i ; f_i is the forecasted electric load value at point i .

Step 4: Cuckoo Global Search.

Implement a cuckoo global search, i.e., Equation (10), by using the best nest position, $x_{k,best}^{(i+1)}$, and update other nest positions by Lévy flight distribution (Equation (11)) to obtain a new set of nest positions, then, compute the fitness value.

Step 5: Determine New Nest Position.

Compare the fitness value of the new nest positions with the fitness value of the previous iteration, and update the nest position with a better one. Then determine the new nest position as $x_{k,j}^{(t)} = [x_{k,1}^{(t)}, x_{k,2}^{(t)}, \dots, x_{k,n}^{(t)}]^T$.

Step 6: Cuckoo Local Search.

If p_a is lower than to a random number r , then turn to discover the nests in $x_{k,j}^{(t)}$ with lower probability instead of the higher one. Then, compute the fitting value of the new nests and continue updating the nest position $x_{k,j}^{(t)}$ with smaller MAPE value by comparing it with the previous fitness value.

Step 7: Determine The Best Nest Position.

Compare the fitness value of the new nest position, $x_{k,j}^{(t)}$, in Step 6, with the fitness value of the best nest position, $x_{k,best}^{(i+1)}$. If the fitness value of $x_{k,j}^{(t)}$ is lower than the one of $x_{k,best}^{(i+1)}$, then, update $x_{k,j}^{(t)}$ as the best nest position, $x_{k,best}^{(t)}$.

Step 8: Stop Criteria.

If the number of search iterations are greater than a given maximum search iterations, then, the best nest position, $x_{k,best}^{(t)}$, among the current population is determined as parameters (C, σ, ϵ) of an SVR model; otherwise, go back to Step 2 and continue searching the next iteration.

2.3. Seasonal Mechanism

As indicated in existing papers [5,28,29] the short term electric load data often display cyclic tendencies due to the cyclic nature of economic activities (production, transportation, operation, etc.) or the seasonal climate in Nature (air conditioners and heaters in summer and winter, respectively). It is useful to increase the forecasting accuracy by calculating these seasonal effects (or seasonal indexes) to adjust the seasonal biases. Several researchers have proposed seasonal adjustment approaches to determine the seasonal effects, such as Koc and Altinay [42], Goh and Law [43], and Wang et al. [44], who all apply regression models to decompose the seasonal component. Martens et al. [45] apply a flexible Fourier transform to estimate the daily variation of the stock exchange, and compute a seasonal estimator. Deo et al. [46] composed two Fourier transforms in a cyclic period to further identify the seasonal estimator. Comparing these seasonal adjustment models, Deo’s model extends Martens’s model for application to general cycle-length data, particularly for hour-based or other shorter cycle-length data. Considering that this paper deals with half-hour based short term electric load data, this paper would like to employ the seasonal mechanism proposed by Hong and his colleagues in [5,28,29]. That is, firstly apply the ARIMA model to identify the seasonal length of the target time series data set; secondly, calculate these seasonal indexes to adjust cyclic effects to receive more satisfied forecasting performances, as shown in Equation (16):

$$Seasonratio_q = \ln\left(\frac{a_q}{f_q}\right)^2 = 2(\ln a_q - \ln f_q) \tag{16}$$

where $q = j, l + j, 2l + j, \dots, (m - 1)l + j$ with m seasonal (cyclic) periods and l seasonal length in each period. Thirdly, the seasonal index (SI) for each seasonal point j in each period is calculated as Equation (17):

$$SI_j = \exp\left(\frac{1}{m} \sum_{q=j}^{(m-1)l+j} Seasonratio_q\right) / 2 \tag{17}$$

where $j = 1, 2, \dots, l$. The seasonal mechanism is demonstrated in Figure 2.

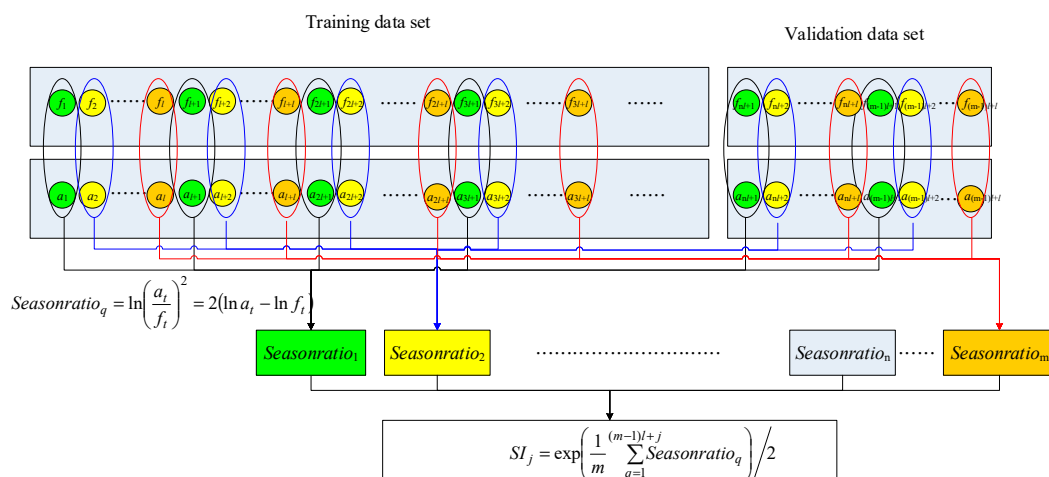


Figure 2. Seasonal mechanism.

3. Numerical Examples of the Proposed SSVRCCS Model

3.1. Data Set of Numerical Examples

To demonstrate the superiorities of the tent chaotic mapping function and seasonal mechanism of the proposed SSVRCCS model, this paper uses the half-hour electric load data from the Queensland regional market of the National Electricity Market (NEM, Queensland, Australia) [47], named Example 1, and the New York Independent System Operator (NYISO, New York, NY, USA) [48], named Example 2. The employed electric load data contains a total of 768 half-hour electric load values in Example 1, i.e., from 00:30 01 October 2017 to 00:00 17 October 2017. Based on Schalkoff's [49] recommendation that the ratio of validation data set to training data set should be approximately one to four, therefore, the electric load data set is divided into three sub-sets. The training set has 432 half-hour electric load values (i.e., from 00:30 01 October 2017 to 00:00 09 October 2017). The validation set contains 144 half-hour electric load values (i.e., from 00:30 09 October 2017 to 00:00 13 October 2017). The testing set has 192 half-hour electric load values (i.e., from 00:30 13 October 2017 to 00:00 17 October 2017). Similarly, in Example 2, the used electric load data also contains a total of 768 hourly electric load values, i.e., from 00:00 01 January 2018 to 23:00 1 February 2018. The electric load data set is also divided into three sub-sets. The training set has 432 hourly electric load values (i.e., from 00:00 01 January 2018 to 23:00 18 January 2018). The validation set has 144 hourly electric load values (i.e., from 00:00 19 January 2018 to 23:00 24 January 2018). The testing set has 192 hourly electric load values (i.e., from 00:00 25 January 2018 to 23:00 1 February 2018). To be based on the same comparison conditions, all compared models thus have the same data division sets.

During the modeling processes, in the training stage, the rolling-based procedure, proposed by Hong [28], is also applied to assist CCS algorithm to implement well searching for an appropriate parameter combination (σ , C , ϵ) of an SVR model. Specifically, the CCS algorithm minimizes the empirical risk, as shown in Equation (4), to obtain the potential parameter combination by employing the first n electric load data in the training set; then, it receives the first forecasted electric load by the SVR model with these potential parameter combination, i.e., the $(n + 1)$ th forecasting electric load. For the second round, the next n electric load data, from 2nd to $(n + 1)$ th electric load values, are then used by the SVR model to obtain new potential parameter combination, then, similarly, the $(n + 2)$ th forecasting electric load is receive. This procedure would never be stopped till the totally 432 forecasting electric load are computed. The training error and the validation error are also calculated in each iteration.

Only with the smallest validation and testing errors, a potential parameter combination could be finalized as the determined parameter combination of an SVR model. Then, the never used testing data set would be employed to demonstrate the forecasting performances, i.e., eventually, the 192 half-hour/hourly electric load would be forecasted by the proposed SSVRCCS model.

3.2. The SVR with Chaotic Cuckoo Search (SSVRCCS) Electric Load Forecasting Model

3.2.1. Embedded Parameter Settings of the CCS Algorithm

The embedded parameters of CCS algorithm for modeling are set as follows: the number of host nests is set to be 50; the maximum number of iterations is set as 500; the initial probability parameter p_a is set as 0.25. During the parameter optimizing process of an SVR model, the searching feasible ranges of the three parameters are set as following, $\sigma \in [0.01, 5]$, $\epsilon \in [0.01, 1]$, and $C \in [0.01, 60,000]$. In addition, considering that the iteration time would affect the performance of each model, the given optimization time for each model with an evolutionary algorithm is set at the same inasmuch as possible.

3.2.2. Forecasting Accuracy Indexes

Three forecasting accuracy evaluation indexes are used to compare the forecasting performances for each model: (1) the MAPE mentioned in Equation (5); (2) the root mean square error (RMSE); and (3) the mean absolute error (MAE). The latter two indexes could be calculated by Equations (18) and (19), respectively:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N (a_i - f_i)^2}{N}} \quad (18)$$

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |a_i - f_i| \quad (19)$$

where N is the total number of data; a_i is the actual electric load value at point i ; f_i is the forecasted electric load value at point i .

3.2.3. Forecasting Accuracy Significance Tests

To demonstrate the significant superiority of the proposed SSVRCCS model in terms of forecasting accuracy, some famous statistical tests are implemented. Based on Diebold and Mariano's [50] and Derrac et al. [51] research suggestions, the Wilcoxon signed-rank test [52] and Friedman test [53] are simultaneously applied in this paper.

The Wilcoxon signed-rank test is used to compare the significant differences in terms of central tendency between two data set with the same size. Let d_i represent the i -th pair difference of the i -th forecasting errors from any two forecasting models, the differences are ranked according to their absolute values. Let r^+ represent the sum of ranks that the first model larger than the second one; r^- represent the sum of ranks that the second model larger than the first one. In case of $d_j = 0$, then, exclude the j -th pair and reduce sample size. The statistic W of the Wilcoxon signed-rank test is shown as Equation (20):

$$W = \min\{r^+, r^-\} \quad (20)$$

If W meets the criterion of the Wilcoxon distribution under N degrees of freedom, then, the null hypothesis of equal performance of these two compared models cannot be accepted. It also implies that the proposed model is significantly superior to the other model. Of course, if the comparison size is larger than the critical size, the sampling distribution of W would approximate to the normal distribution instead of Wilcoxon distribution, and the associated p -value would also be provided.

On the other hand, due to the non-parametric statistical test in the ANOVA analysis procedure, the Friedman test is devoted to compare the significant differences among two or more models. The statistic F of the Friedman test is shown as Equation (21):

$$F = \frac{12N}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (21)$$

where N is the total number of forecasting results; k is the number of compared models; R_j is the average rank sum obtained in each forecasting value for each compared model as shown in Equation (22),

$$R_j = \frac{1}{N} \sum_{i=1}^N r_i^j \quad (22)$$

where r_i^j is the rank sum from 1 (the smallest forecasting error) to k (the worst forecasting error) for i th forecasting result, for j th compared model.

Similarly, if the associated p -value of F meets the criterion of not acceptance, the null hypothesis, equal performance among all compared models, could also not be held.

3.2.4. Forecasting Results and Analysis for Example 1

To compare the improved forecasting performance of the tent chaotic mapping function, a SVR with the original CS algorithm (without the tent chaotic mapping function), namely the SVRCS model, will also be taken into comparison. Therefore, according to the rolling-based procedure mentioned above, by using the training data set from Example 1 (mentioned in Section 3.1) to conduct the training work, and the parameters for SVRCS and SVRCCS models are eventually determined. These trained models are further used to forecast the electric load. Then, the forecasting results and the suitable parameters of SVRCS and SVRCCS models are listed in Table 1. It is clearly indicated that the proposed SVRCCS model has achieved smaller forecasting performances in terms of the forecasting accuracy indexes, MAPE, RMSE, and MAE.

Table 1. Three parameters of SVRCS and SVR with chaotic cuckoo search (SVRCCS) models for Example 1.

Evolutionary Algorithms	Parameters			MAPE of Testing (%)	RMSE of Testing	MAE of Testing
	σ	C	ϵ			
SVRCS	1.4744	17,877.54	0.3231	2.63	217.19	151.72
SVRCCS	0.5254	5,885.65	0.7358	1.51	126.92	87.94

As shown in Figure 3, the employed electric load data demonstrates seasonal/cyclic changing tendency in Example 1. In addition, the data recording frequency is on a half-hour basis, therefore, to comprehensively reveal the electric load changing tendency, the seasonal length is set as 48. Therefore, there are 48 seasonal indexes for the proposed SVRCCS and SVRCS models. The seasonal indexes for each half-hour are computed based on the 576 forecasting values of the SVRCCS and SVRCS models in the training (432 forecasting values) and validation (144 forecasting values) processes. The 48 seasonal indexes for the SVRCCS and SVRCS models are listed in Table 2, respectively.

Table 2. The 48 seasonal indexes for SVRCCS and SVRCS models for Example 1.

Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)	
	SVRCCS	SVRCS		SVRCCS	SVRCS		SVRCCS	SVRCS		SVRCCS	SVRCS
00:00	0.9615	0.9201	06:00	1.0360	1.0536	12:00	1.0025	1.0076	18:00	1.0071	1.0176
00:30	0.9881	0.9241	06:30	1.0518	1.0729	12:30	0.9960	1.0032	18:30	1.0034	1.0109
01:00	0.9893	0.9401	07:00	1.0671	1.0924	13:00	0.9935	0.9992	19:00	0.9694	0.9767
01:30	0.9922	0.9729	07:30	1.0394	1.0810	13:30	0.9975	1.0022	19:30	0.9913	0.9875
02:00	0.9919	0.9955	08:00	1.0088	1.0575	14:00	1.0026	1.0083	20:00	0.9820	0.9812
02:30	0.9948	0.9980	08:30	1.0076	1.0322	14:30	1.0015	1.0088	20:30	0.9789	0.9700
03:00	0.9950	0.9998	09:00	1.0004	1.0148	15:00	1.0000	1.0070	21:00	0.9830	0.9641
03:30	0.9915	0.9961	09:30	0.9903	0.9982	15:30	1.0022	1.0089	21:30	0.9780	0.9547
04:00	1.0082	1.0129	10:00	1.0031	1.0067	16:00	1.0033	1.0115	22:00	0.9906	0.9622
04:30	1.0075	1.0176	10:30	0.9912	0.9981	16:30	1.0097	1.0173	22:30	0.9932	0.9778
05:00	1.0124	1.0245	11:00	0.9928	0.9973	17:00	1.0098	1.0188	23:00	0.9659	0.9645
05:30	1.0139	1.0253	11:30	0.9967	1.0025	17:30	1.0053	1.0164	23:00	0.9601	0.9348

The forecasting comparison curves of six models, including the SARIMA_{(9,1,8) × (4,1,4)}, GRNN ($\sigma = 0.04$), SSVRCCS, SSVRCS, SVRCCS, and SVRCS models mentioned above and actual values are shown in Figure 4. It illustrates that the proposed SSVRCCS model is closer to the actual electric load values than other compared models. To further illustrate the tendency capturing capability of the proposed SSVRCCS model during the electric peak loads, Figures 5–8 are enlargements from four peaks in Figure 4 to clearly demonstrate how closer the SSVRCCS model matches to the actual electric load values than other alternative models. For example, for each peak, the red real line (SSVRCCS model) always follows closely with the black real line (actual electric load), whether climbing up the peak or climbing down the hill.

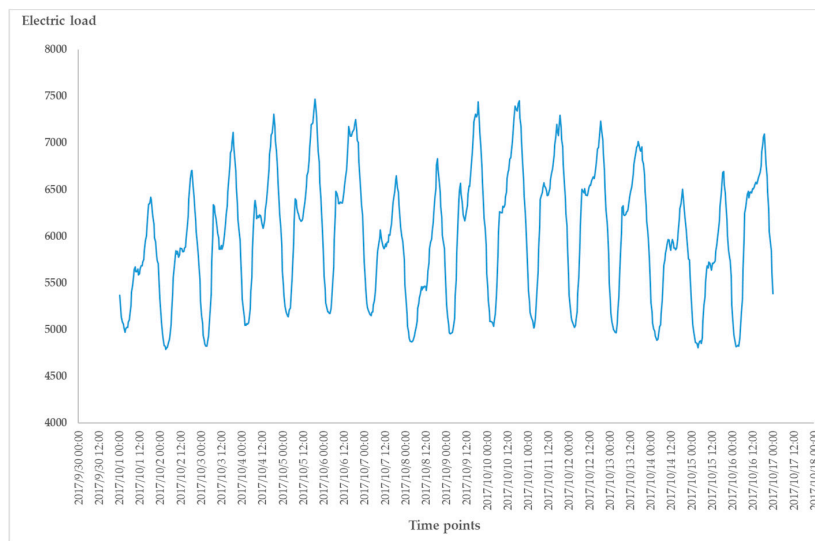


Figure 3. The seasonal tendency of actual half-hour electric load in Example 1.

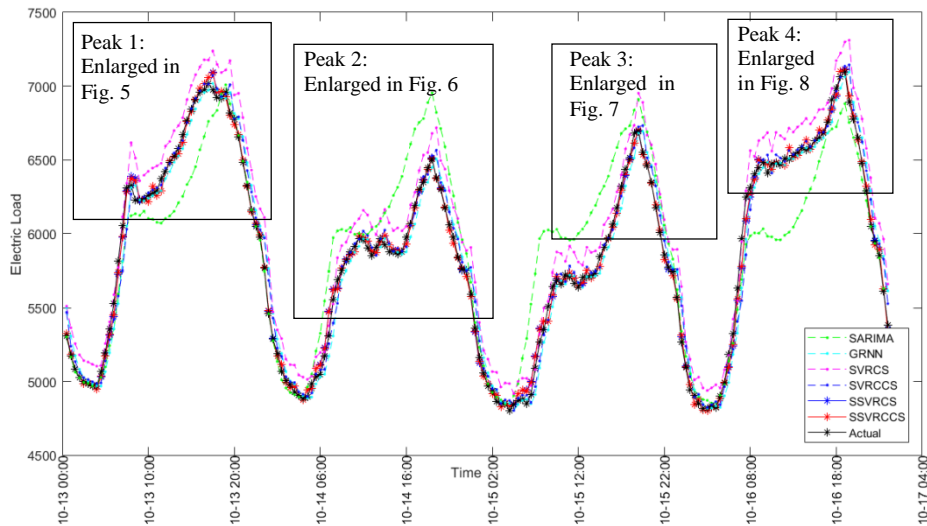


Figure 4. Forecasting values of SSVRCS model and other alternative models for Example 1.

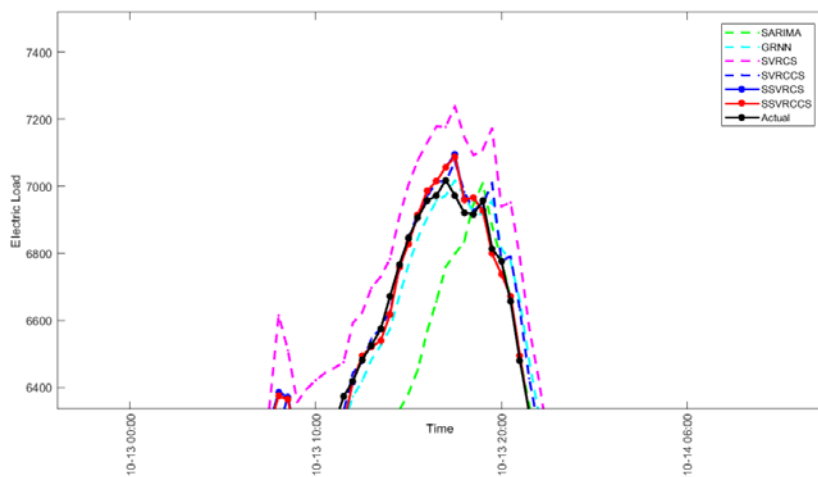


Figure 5. The enlargement comparison of Peak 1 from the compared models for Example 1.

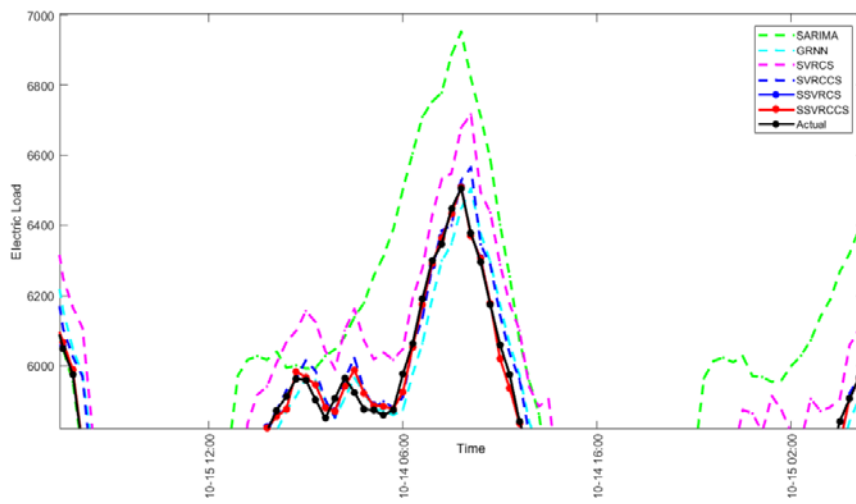


Figure 6. The enlargement comparison of Peak 2 from the compared models for Example 1.

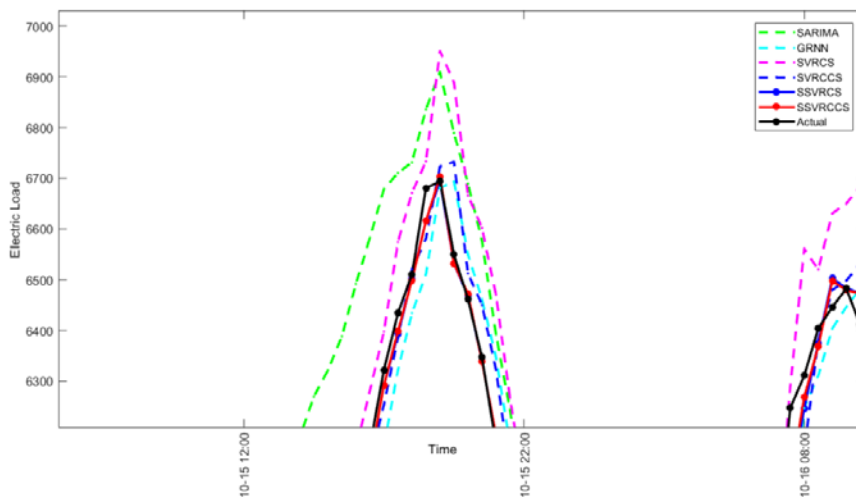


Figure 7. The enlargement comparison of Peak 3 from the compared models for Example 1.

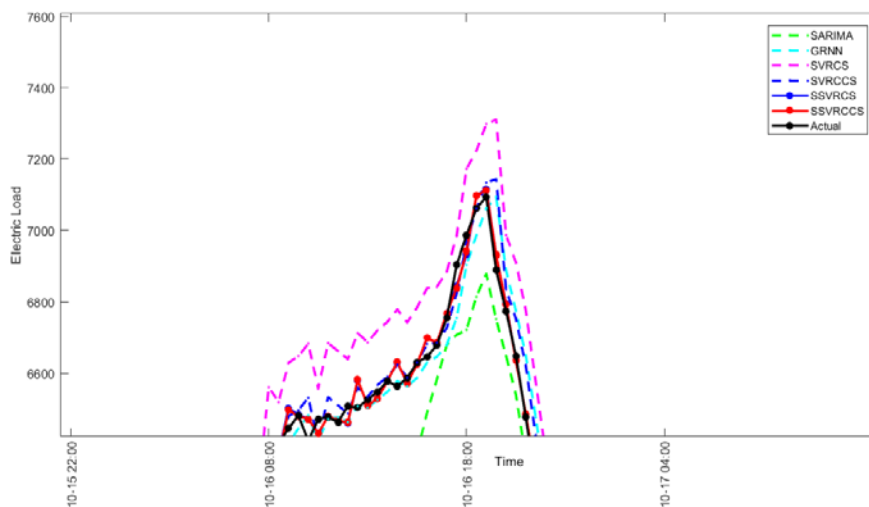


Figure 8. The enlargement comparison of Peak 4 from the compared models for Example 1.

Table 3 illustrates the forecasting accuracy indexes for the proposed SSVRCCS model and other alternative compared models. It is clearly to see that the MAPE, RMSE, and MAE of the proposed SSVRCCS model are 0.70%, 56.90, and 40.79, respectively, which are superior to the other five alternative models. It also implies that the proposed SSVRCCS model contributes great improvements in terms of load forecasting accuracy.

Table 3. Forecasting accuracy indexes of the compared models for Example 1.

Forecasting Accuracy Indexes	SARIMA _{(9,1,8)×(4,1,4)}	GRNN(α = 0.04)	SSVRCCS	SSVRCS	SVRCCS	SVRCS
MAPE (%)	3.62	1.53	0.70	0.99	1.51	2.63
RMSE	280.05	114.30	56.90	80.42	126.92	217.19
MAE	217.67	88.63	40.79	57.69	87.94	151.72

Finally, to ensure the significant contribution in terms of forecasting accuracy improvement for the proposed SSVRCCS model, the Wilcoxon signed-rank test and the Friedman test are conducted. Where Wilcoxon signed-rank test is implemented under two significance levels, α = 0.025 and α = 0.05, by two-tail test; the Friedman test is then implemented under only one significance level, α = 0.05. The test results in Table 4 show that the proposed SSVRCCS model almost reaches a significance level in terms of forecasting performance than other alternative compared models.

Table 4. Results of Wilcoxon signed-rank test and Friedman test for Example 1.

Compared Models	Wilcoxon Signed-Rank Test				Friedman Test
	α = 0.025; W = 9264	p-Value	α = 0.05; W = 9264	p-Value	α = 0.05;
SSVRCCS vs. SARIMA _{(9,1,8)×(4,1,4)}	842 ^a	0.00000 **	842 ^a	0.00000 **	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6$ $F = 23.49107$ $p = 0.000272$ (Reject H_0)
SSVRCCS vs. GRNN(σ = 0.04)	3025 ^a	0.00000 **	3025 ^a	0.00000 **	
SSVRCCS vs. SSVRCS	2159 ^a	0.00000 **	2159 ^a	0.00000 **	
SSVRCCS vs. SVRCCS	3539 ^a	0.00000 **	3539 ^a	0.00000 **	
SSVRCCS vs. SVRCS	4288 ^a	0.00000 **	4288 ^a	0.00000 **	

^a Denotes that the SSVRCCS model significantly outperforms the other alternative compared models; * represents that the test indicates not to accept the null hypothesis under α = 0.05. ** represents that the test indicates not to accept the null hypothesis under α = 0.025.

3.2.5. Forecasting Results and Analysis for Example 2

Similar to Example 1, SVRCS and SVRCCS models are also trained based on the rolling-based procedure by using the training data set from Example 2 (mentioned in Section 3.1). The forecasting results and the suitable parameters of SVRCS and SVRCCS models are shown in Table 5. It is also obviously that the proposed SVRCCS model has achieved a smaller forecasting performance in terms of forecasting accuracy indexes, MAPE, RMSE, and MAE.

Table 5. Three parameters of SVRCS and SVRCCS models for Example 2.

Evolutionary Algorithms	Parameters			MAPE of Testing (%)	RMSE of Testing	MAE of Testing
	σ	C	ε			
SVRCS	0.6628	36,844.57	0.2785	3.42	886.67	631.40
SVRCCS	0.3952	42,418.21	0.7546	2.30	515.10	426.42

Figure 9 also demonstrates the seasonal/cyclic changing tendency from the used electric load data in Example 2. Based on the hourly recording frequency, to completely address the changing tendency of the employed data, the seasonal length is set as 24. Therefore, there are 24 seasonal indexes for the proposed SVRCCS and SVRCS models. The seasonal indexes for each hour are computed based on the 576 forecasting values of the SVRCCS and SVRCS models in the training (432 forecasting values) and validation (144 forecasting values) processes. The 24 seasonal indexes for the SVRCCS and SVRCS models are listed in Table 6, respectively.

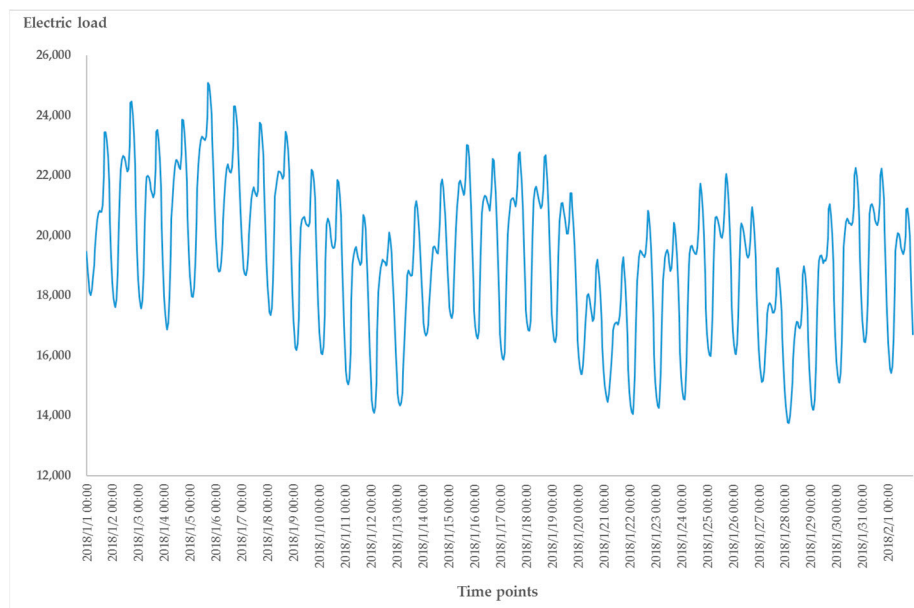


Figure 9. The seasonal tendency of actual hourly electric load in Example 2.

Table 6. The 24 seasonal indexes for SVRCCS and SVRCS models for Example 2.

Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)		Time Points	Seasonal Index (SI)	
	SVRCCS	SVRCS		SVRCCS	SVRCS		SVRCCS	SVRCS		SVRCCS	SVRCS
00:00	0.9718	0.9317	06:00	1.0545	1.1043	12:00	0.9848	0.9911	18:00	0.9753	1.0242
01:00	0.9848	0.9670	07:00	1.0383	1.1133	13:00	0.9896	0.9959	19:00	0.9707	0.9743
02:00	0.9894	0.9960	08:00	0.9854	1.0833	14:00	0.9898	0.9960	20:00	0.9711	0.9754
03:00	0.9937	1.0001	09:00	0.9913	1.0259	15:00	0.9994	1.0058	21:00	0.9610	0.9674
04:00	1.0076	1.0140	10:00	0.9860	0.9951	16:00	1.0144	1.0208	22:00	0.9519	0.9435
05:00	1.0343	1.0407	11:00	0.9841	0.9903	17:00	1.0252	1.0441	23:00	0.9567	0.9245

The forecasting comparison curves of six models in Example 2, including SARIMA_{(9,1,10) × (4,1,4)}, GRNN ($\sigma = 0.07$), SSVRCCS, SSVRCS, SVRCCS, and SVRCS models and actual values are shown as in Figure 10. It indicates that the proposed SSVRCCS model is closer to the actual electric load values than the other compared models. Similarly, the enlarged figures, Figures 11–14, from eight peaks in Figure 10 are provided to demonstrate the tendency capturing capability of the proposed SSVRCCS model and how closer the SSVRCCS model matches the actual electric load values than other alternative models. It is clear that for each peak, the red real line (SSVRCCS model) always follows closely with the black real line (actual electric load), whether climbing up the peak or climbing down the hill.

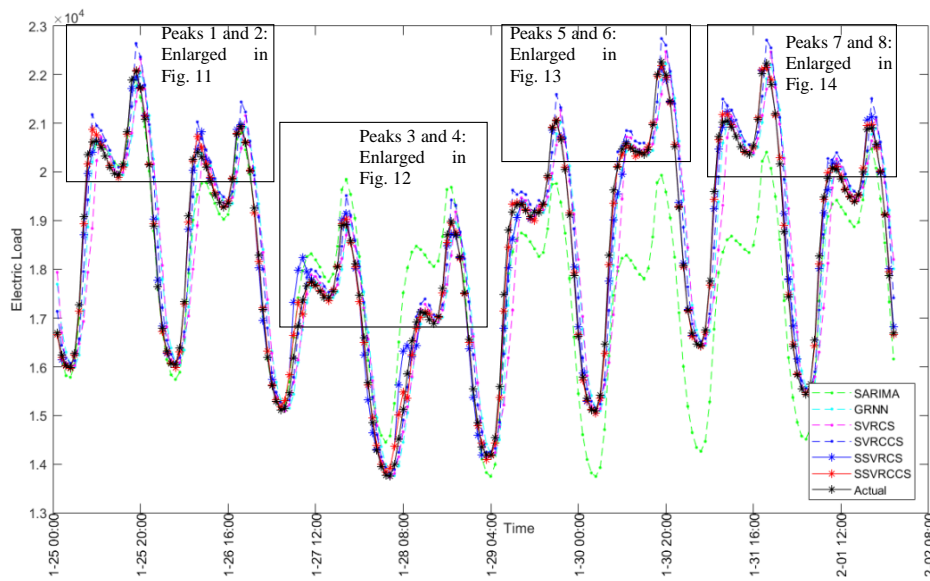


Figure 10. Forecasting values of SVR with chaotic cuckoo search (SSVRCCS) model and other alternative models for Example 2.

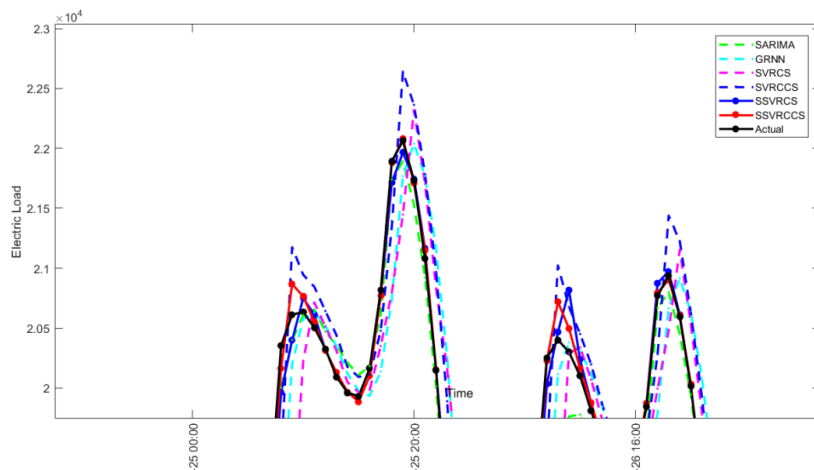


Figure 11. The enlargement comparison of Peaks 1 and 2 from the compared models for Example 2.

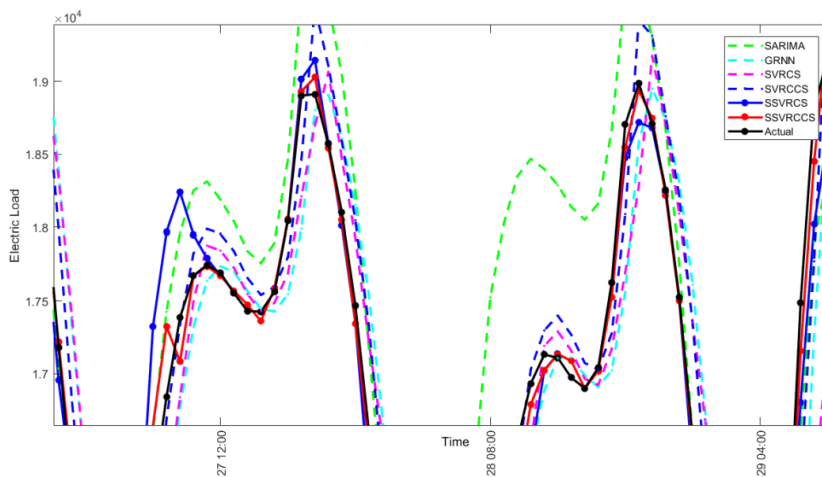


Figure 12. The enlargement comparison of Peaks 3 and 4 from the compared models for Example 2.

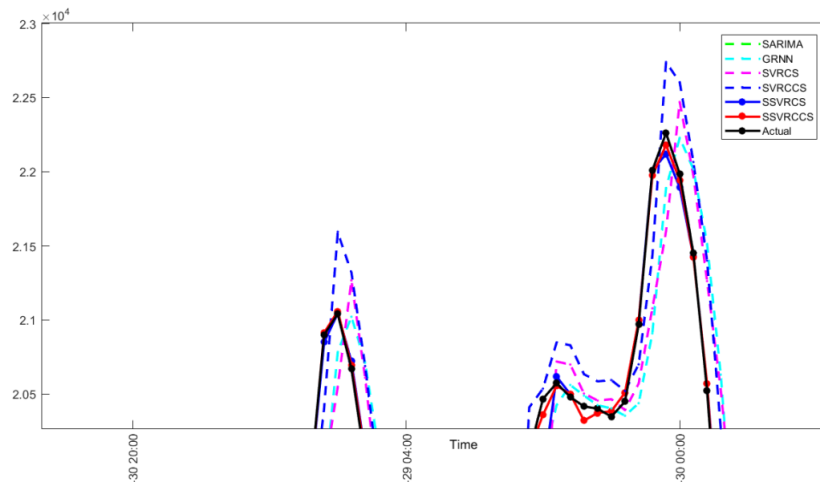


Figure 13. The enlargement comparison of Peaks 5 and 6 from the compared models for Example 2.

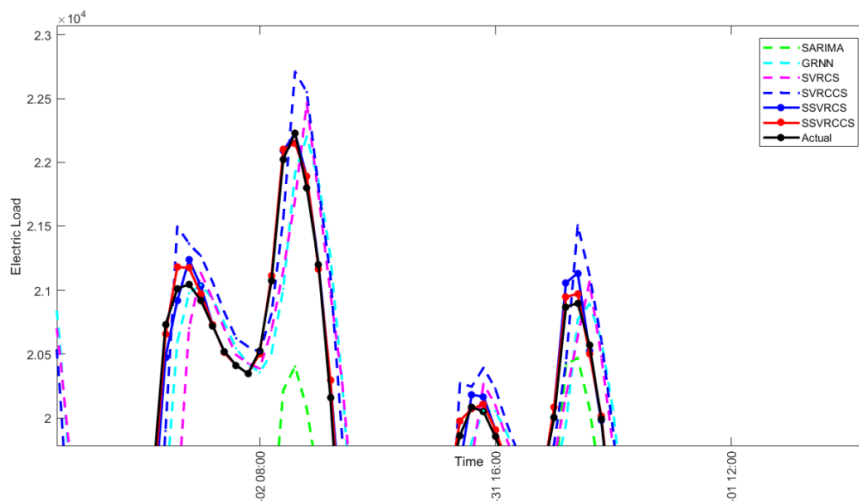


Figure 14. The enlargement comparison of Peaks 7 and 8 from the compared models for Example 2.

For comparison with other alternative models, Table 7 demonstrates the forecasting accuracy indexes for each compared model. Obviously, the proposed SSVRCCS model almost achieves the smallest index values in terms of the MAPE (0.46%), RMSE (126.10), and MAE (80.85), respectively. It is superior to the other five compared models. Once again, it indicates that the proposed SSVRCCS model could produce more accurate forecasting performances.

Table 7. Forecasting accuracy indexes of compared models for Example 2.

Forecasting Accuracy Indexes	SARIMA _{(9,1,10) × (4,1,4)}	GRNN(α = 0.07)	SSVRCCS	SSVRCS	SVRCCS	SVRCS
MAPE (%)	5.16	3.19	0.46	0.86	2.30	3.42
RMSE	1233.09	753.97	126.10	262.02	515.10	886.67
MAE	956.14	577.48	80.85	152.02	426.42	631.40

Finally, two statistical tests are also conducted to ensure the significant contribution in terms of forecasting accuracy improvement for the proposed SSVRCCS model. The test results are illustrated in Table 8 that the proposed SSVRCCS model almost reaches significance level in terms of forecasting performance than other alternative compared models.

Table 8. Results of Wilcoxon signed-rank test and Friedman test for Example 2.

Compared Models	Wilcoxon Signed-Rank Test				Friedman Test
	$\alpha = 0.025;$ $W = 9264$	p -Value	$\alpha = 0.05;$ $W = 9264$	p -Value	$\alpha = 0.05;$
SSVRCCS vs. <i>SARIMA</i> _{(9,1,10) × (4,1,4)}	152 ^a	0.00000 **	152 ^a	0.00000 **	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6$ $F = 149.8006$ $p = 0.0000$ (Reject H_0)
SSVRCCS vs. <i>GRNN</i> ($\sigma = 0.07$)	396 ^a	0.00000 **	396 ^a	0.00000 **	
SSVRCCS vs. SSVRCS	482 ^a	0.00000 **	482 ^a	0.00000 **	
SSVRCCS vs. SVRCCS	745 ^a	0.00000 **	745 ^a	0.00000 **	
SSVRCCS vs. SVRCS	5207 ^a	0.00000 **	5207 ^a	0.00000 **	

^a Denotes that the SSVRCCS model significantly outperforms the other alternative compared models; * represents that the test indicates not to accept the null hypothesis under $\alpha = 0.05$. ** represents that the test indicates not to accept the null hypothesis under $\alpha = 0.025$.

3.2.6. Discussions

To learn about the effects of the tent chaotic mapping function in both Examples 1 and 2, comparing the forecasting performances (the values of MAPE, RMSE, and MAE in Tables 3 and 7) between SVRCS and SVRCCS models, the forecasting accuracy of SVRCCS model is superior to that of SVRCS model. It reveals that the CCS algorithm could determine more appropriate parameter combinations for an SVR model by introducing the tent chaotic mapping function to enrich the cuckoo search space and the diversity of the population when the CS algorithm is going to be trapped in the local optima. In Example 1, as shown in Table 1, the parameter searching of an SVR model by CCS algorithm could be moved to a much better solution, $(\sigma, C, \epsilon) = (0.5254, 5885.65, 0.7358)$ with forecasting accuracy, $(MAPE, RMSE, MAE) = (1.51\%, 126.92, 87.94)$ from the local solution, $(\sigma, C, \epsilon) = (1.4744, 17877.54, 0.3231)$ with forecasting accuracy, $(MAPE, RMSE, MAE) = (2.63\%, 217.19, 151.72)$. It almost improves 1.12% (=2.63% – 1.51%) forecasting accuracy in terms of MAPE by employing Tent chaotic mapping function. The same in Example 2, as shown in Table 5, the CCS algorithm also helps to improve the result by 1.12% (=3.42% – 2.30%). These two examples both reveal the great contributions from the tent chaotic mapping function. In future research, it would be worth applying another chaotic mapping function to help to avoid trapping into local optima.

Furthermore, the seasonal mechanism can successfully help to deal with the seasonal/cyclic tendency changes of the electric load data to improve the forecasting accuracy, by determining seasonal length and calculating associate seasonal indexes (per half-hour for Example 1, and per hour for Example 2) from training and validation stages for each seasonal point. In this paper, authors hybridize the seasonal mechanism with SVRCS and SVRCCS models, namely SSVRCS and SSVRCCS models, respectively, by using their associate seasonal indexes, as shown in Tables 2 and 6, respectively. Based on these seasonal indexes, the forecasting results (in terms of MAPE) of the SVRCS and SVRCCS models for Example 1 are further revised from 2.63% and 1.51%, respectively, to achieve more acceptable forecasting accuracy, 0.99% and 0.70%, respectively. They almost improve 1.64% (=2.63% – 0.99%) and 0.81% (=1.51% – 0.70%) forecasting accuracy by applying seasonal mechanism. The same in Example 2, as shown in Table 7, the seasonal mechanism also improves 2.56% (=3.42% – 0.86%) and 1.84% (=2.30% – 0.46%) for SVRCS and SVRCCS models, respectively. In the meanwhile, based on Wilcoxon signed-rank test and Friedman test, as shown in Tables 4 and 8 for Examples 1 and 2, respectively, the SSVRCCS models also achieve statistical significance among other alternative models. Based on above discussions, this seasonal mechanism is also a considerable contribution, and it is worth the time cost to deal with the seasonal/cyclic information during modeling processes.

Therefore, it could be remarked that by hybridizing novel intelligent technologies, such as chaotic mapping functions, advanced searching mechanism, seasonal mechanism, and so on, to overcome some inherent drawbacks of the existing evolutionary algorithms could significantly improve forecasting accuracy. This kind of research paradigm also inspires some interesting future research.

4. Conclusions

This paper proposes a novel SVR-based hybrid electric load forecasting model, by hybridizing the seasonal mechanism, the tent chaotic mapping function, and the CS algorithm with an SVR model, namely the SSVRCCS model. The experimental results indicate that the proposed SSVRCCS model significantly outperforms other alternative compared forecasting models. This paper continues to overcome some inherent shortcomings of the CS algorithm, by actions such as enriching the search space and the diversity of the population by using the tent chaotic mapping function to avoid premature convergence problems and applying seasonal mechanism to provide useful adjustments caused from seasonal/cyclic effects of the employed data set. Eventually, the proposed SSVRCCS model achieves significant accurate forecasting performances.

This paper concludes some important findings. Firstly, by applying appropriate chaotic mapping functions it could help empower the search variables to possess ergodicity characteristics, to enrich the searching space, then, determine well appropriate parameter combinations of an SVR model, to eventually improve the forecasting accuracy. Therefore, any novel hybridizations of existed evolutionary algorithms with other optimization methods or mechanisms which could consider those actions mentioned above during modeling process are all deserving to take a trial to achieve more interesting results. Secondly, only hybridizing different single evolutionary algorithm with an SVR model could contribute minor forecasting accuracy improvements. It is more worthwhile to hybridize different novel intelligent technologies with single evolutionary algorithms to achieve more high forecasting accurate levels. This could be an interesting future research tendency in the SVR-based electric load forecasting field.

Acknowledgments: Yongquan Dong thanks the support from the project grants: National Natural Science Foundation of China (No. 61100167), Natural Science Foundation of Jiangsu Province, China (No. BK2011204), and Qing Lan Project, the National Training Program of Innovation and Entrepreneurship for Undergraduates (No. 201710320058); Zichen Zhang thanks the support from the project grant: Postgraduate Research & Practice Innovation Program of Jiangsu Province (No. 2017YXJ214); Wei-Chiang Hong thanks the support from Jiangsu Distinguished Professor Project by Jiangsu Provincial Department of Education.

Author Contributions: Yongquan Dong and Wei-Chiang Hong conceived, designed the experiments, and wrote the paper; Zichen Zhang collected the data, performed and analyzed the experiments.

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

References

1. Wan, C.; Zhao, J.; Member, S.; Song, Y. Photovoltaic and solar power forecasting for smart grid energy management. *CSEE J. Power Energy Syst.* **2015**, *1*, 38–46. [[CrossRef](#)]
2. Xiao, L.; Wang, J.; Hou, R.; Wu, J. A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. *Energy* **2015**, *82*, 524–549. [[CrossRef](#)]
3. Bunn, D.W.; Farmer, E.D. Comparative models for electrical load forecasting. *Int. J. Forecast.* **1986**, *2*, 241–242.
4. 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](#)]
5. 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](#)]
6. Hussain, A.; Rahman, M.; Memon, J.A. Forecasting electricity consumption in Pakistan: The way forward. *Energy Policy* **2016**, *90*, 73–80. [[CrossRef](#)]
7. 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](#)]
8. Vu, D.H.; Muttaqi, K.M.; Agalgaonkar, A.P. A variance inflation factor and backward elimination based robust regression model for forecasting monthly electricity demand using climatic variables. *Appl. Energy* **2015**, *140*, 385–394. [[CrossRef](#)]
9. Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **2016**, *130*, 139–147. [[CrossRef](#)]

10. 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](#)]
11. Al-Hamadi, H.M.; Soliman, S.A. Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model. *Electr. Power Syst. Res.* **2004**, *68*, 47–59. [[CrossRef](#)]
12. Zhang, M.; Bao, H.; Yan, L.; Cao, J.; Du, J. Research on processing of short-term historical data of daily load based on Kalman filter. *Power Syst. Technol.* **2003**, *9*, 39–42.
13. Hippert, H.S.; Taylor, J.W. An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. *Neural Netw.* **2010**, *23*, 386–395. [[CrossRef](#)] [[PubMed](#)]
14. Zhang, W.; Yang, J. Forecasting natural gas consumption in China by Bayesian model averaging. *Energy Rep.* **2015**, *1*, 216–220. [[CrossRef](#)]
15. Kelo, S.; Dudul, S. A wavelet Elman neural network for short-term electrical load prediction under the influence of temperature. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 1063–1071. [[CrossRef](#)]
16. Li, H.Z.; Guo, S.; Li, C.J.; Sun, J.Q. A hybrid annual power load forecasting model based on generalized regression neural network with fruit fly optimization algorithm. *Knowl.-Based Syst.* **2013**, *37*, 378–387. [[CrossRef](#)]
17. 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](#)]
18. Bennett, C.J.; Stewart, R.A.; Lu, J.W. Forecasting low voltage distribution network demand profiles using a pattern recognition based expert system. *Energy* **2014**, *67*, 200–212. [[CrossRef](#)]
19. Lahouar, A.; Slama, J.B.H. Day-ahead load forecast using random forest and expert input selection. *Energy Convers. Manag.* **2015**, *103*, 1040–1051. [[CrossRef](#)]
20. Akdemir, B.; Çetinkaya, N. Long-term load forecasting based on adaptive neural fuzzy inference system using real energy data. *Energy Procedia* **2012**, *14*, 794–799. [[CrossRef](#)]
21. Chaturvedi, D.K.; Sinha, A.P.; Malik, O.P. Short term load forecast using fuzzy logic and wavelet transform integrated generalized neural network. *Int. J. Electr. Power Energy Syst.* **2015**, *67*, 230–237. [[CrossRef](#)]
22. 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](#)]
23. Bahrami, S.; Hooshmand, R.-A.; Parastegari, M. Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. *Energy* **2014**, *72*, 434–442. [[CrossRef](#)]
24. Hahn, H.; Meyer-Nieberg, S.; Pickl, S. Electric load forecasting methods: Tools for decision making. *Eur. J. Oper. Res.* **2009**, *199*, 902–907. [[CrossRef](#)]
25. Vapnik, V.; Golowich, S.; Smola, A. Support vector machine for function approximation, regression estimation, and signal processing. *Adv. Neural Inf. Process. Syst.* **1996**, *9*, 281–287.
26. Zhang, X.; Ding, S.; Xue, Y. An improved multiple birth support vector machine for pattern classification. *Neurocomputing* **2017**, *225*, 119–128. [[CrossRef](#)]
27. Hua, X.; Ding, S. Weighted least squares projection twin support vector machines with local information. *Neurocomputing* **2015**, *160*, 228–237. [[CrossRef](#)]
28. Hong, W.C. Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy* **2011**, *36*, 5568–5578. [[CrossRef](#)]
29. Hong, W.-C.; Dong, Y.; Zhang, W.; Chen, L.-Y.; Panigrahi, B.K. Cyclic electric load forecasting by seasonal SVR with chaotic genetic algorithm. *Int. J. Electr. Power Energy Syst.* **2013**, *44*, 604–614. [[CrossRef](#)]
30. Gandomi, A.H.; Yang, X.S.; Alavi, A.H. Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Eng. Comput.* **2013**, *29*, 17–35. [[CrossRef](#)]
31. Yang, X.S.; Deb, S. Cuckoo search via Lévy flights. In Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBic), Coimbatore, India, 9–11 December 2009; IEEE Publications: Coimbatore, India, 2009; pp. 210–214.
32. Lakshminarayanan, S.; Kaur, D. Optimal maintenance scheduling of generator units using discrete integer cuckoo search optimization algorithm. *Swarm Evolut. Comput.* **2018**. [[CrossRef](#)]
33. Boushaki, S.I.; Kamel, N.; Bendjeghaba, O. A new quantum chaotic cuckoo search algorithm for data clustering. *Expert Syst. Appl.* **2018**, *96*, 358–372. [[CrossRef](#)]
34. Daniel, E.; Anitha, J.; Gnanaraj, J. Optimum laplacian wavelet mask based medical image using hybrid cuckoo search – grey wolf optimization algorithm. *Knowl.-Based Syst.* **2017**, *131*, 58–69. [[CrossRef](#)]

35. Dao, T.-P.; Huang, S.-C.; Thang, P.T. Hybrid Taguchi-cuckoo search algorithm for optimization of a compliant focus positioning platform. *Appl. Soft Comput.* **2017**, *57*, 526–538. [[CrossRef](#)]
36. Puspningrum, A.; Sarno, R. A hybrid cuckoo optimization and harmony search algorithm for software cost estimation. *Procedia Comput. Sci.* **2017**, *124*, 461–469. [[CrossRef](#)]
37. Huang, L.; Ding, S.; Yu, S.; Wang, J.; Lu, K. Chaos-enhanced Cuckoo search optimization algorithms for global optimization. *Appl. Math. Model.* **2016**, *40*, 3860–3875. [[CrossRef](#)]
38. Li, X.; Yin, M. A particle swarm inspired cuckoo search algorithm for real parameter optimization. *Soft Comput.* **2016**, *20*, 1389–1413. [[CrossRef](#)]
39. Sheng, Y.; Pan, H.; Xia, L.; Cai, Y.; Sun, X. Hybrid chaos particle swarm optimization algorithm and application in benzene-toluene flash vaporization. *J. Zhejiang Univ. Technol.* **2010**, *38*, 319–322.
40. Li, M.; Hong, W.-C.; Kang, H. Urban traffic flow forecasting using Gauss-SVR with cat mapping, cloud model and PSO hybrid algorithm. *Neurocomputing* **2013**, *99*, 230–240. [[CrossRef](#)]
41. Yang, X.S.; Deb, S. Cuckoo search: Recent advances and applications. *Neural Comput. Appl.* **2014**, *24*, 169–174. [[CrossRef](#)]
42. Koc, E.; Altinay, G. An analysis of seasonality in monthly per person tourist spending in Turkish inbound tourism from a market segmentation perspective. *Tour. Manag.* **2007**, *28*, 227–237. [[CrossRef](#)]
43. Goh, C.; Law, R. Modeling and forecasting tourism demand for arrivals with stochastic nonstationary seasonality and intervention. *Tour. Manag.* **2002**, *23*, 499–510. [[CrossRef](#)]
44. Wang, J.; Zhu, W.; Zhang, W.; Sun, D. A trend fixed on firstly and seasonal adjustment model combined with the ϵ -SVR for short-term forecasting of electricity demand. *Energy Policy* **2009**, *37*, 4901–4909. [[CrossRef](#)]
45. Martens, K.; Chang, Y.C.; Taylor, S. A comparison of seasonal adjustment methods when forecasting intraday volatility. *J. Financ. Res.* **2002**, *25*, 283–299. [[CrossRef](#)]
46. Deo, R.; Hurvich, C.; Lu, Y. Forecasting realized volatility using a long-memory stochastic volatility model: Estimation, prediction and seasonal adjustment. *J. Econom.* **2006**, *131*, 29–58. [[CrossRef](#)]
47. The Electricity Demand Data of National Electricity Market. Available online: <https://www.aemo.com.au/Electricity/National-Electricity-Market-NEM/Data-dashboard#aggregated-data> (accessed on 2 March 2018).
48. The Electricity Demand Data of the New York Independent System Operator (NYISO). Available online: http://www.nyiso.com/public/markets_operations/market_data/load_data/index.jsp (accessed on 2 April 2018).
49. Schalkoff, R.J. *Artificial Neural Networks*; McGraw-Hill: New York, USA, 1997.
50. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. *J. Bus. Econ. Stat.* **1995**, *13*, 134–144.
51. 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 Evolut. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
52. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]
53. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [[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/>).