Sustainable Cities and Society 71 (2021) 102937

Contents lists available at ScienceDirect

Sustainable Cities and Society

journal homepage: www.elsevier.com/locate/scs

Hierarchical parameter optimization based support vector regression for power load forecasting

Zeyu Wang [a], Xiaojun Zhou [a,c,*], Jituo Tian [a], Tingwen Huang [b]

[a] School of Automation, Central South University, Changsha 410083, China
[b] Texas A&M University at Qatar, Doha 23874, Qatar
[c] Peng Cheng Laboratory, Shenzhen 518000, China

ARTICLE INFO

Keywords:
Power load forecasting
Hybrid support vector regression
Hierarchical parameter optimization method
State transition algorithm

ABSTRACT

Power load forecasting is an important task of smart grid, which is of great significance to the sustainable development of society. In this paper, a hybrid support vector regression (HSVR) is raised for the medium and long term load forecasting. To further improve prediction accuracy, the coupling and interdependent relationship between hyperparameters and model parameters in the optimization process is focused. A hierarchical optimization method based on nested strategy and state transition algorithm (STA) is proposed to find optimal parameters. The effectiveness of the proposed hierarchical optimization method is confirmed on several benchmarks, and the resulting hierarchical optimization method based SVR is also successfully applied to a real industrial power load forecasting problem in China.

1. Introduction

Nowadays, the external economy and environment have an increasing impact on the power grid, and the construction is facing unprecedented challenges. Studies on intelligent prediction methods and optimization algorithms with a certain adaptive deterministic model can not only improve the accuracy of prediction but improve intelligence and efficiency of smart grid (Li, Yu, Huang, & He, 2018; Li, Yu, Yu, Chen, & Wang, 2017; Silva, Khan, & Han, 2018). As non-storable energy, electrical energy must be generated as soon as there is a demand, so it is imperative to estimate the system load for power companies ahead. Taking the nature of historical data and natural environment into account, load data is non-stationary, nonlinear and dynamic, but at the same time, some influencing factors are regular that can be used to effectively predict the power.

In general, load forecasting can be divided into very short term (1~7 days ahead), short term (1~4 weeks ahead), medium term (1~12 months ahead) and long term (1~20 years ahead) with different time horizons (Abu-Shikhah & Elkarmi, 2011; Ghiassi, Zimbra, & Saidane, 2006; Han et al., 2018). Very short term load forecasting (VSTLF) and short term load forecasting (STLF) are critical to daily scheduling, economic and secure operation of power systems, which have been intensively studied in recent decades. However, medium term load forecasting (MTLF) and long term load forecasting (LTLF) offer useful information for the planning of demand side management, and provide significant benefits for firms operating (Ghiassi et al., 2006). Therefore, it is meaningful to pay more attention to medium and long term load forecasting.

The key to achieving accurate prediction is to establish a reasonable mathematical model, then choose an appropriate solution method and implement with a specific algorithm. Accurate load forecasting is a much more difficult problem. Especially changes within medium to long term load often attribute to external factors such as temperature, fuel prices, other economic variables and so forth. Classic statistical forecasting methods, including regression analysis and gray forecasting were applied on short-term load and energy consumption forecasting in Huang and Shih (2003) and Yuan, Liu, and Fang (2016). These approaches are commonly employed to handle sequences with linear characteristics, but render inapplicable for complex nonlinear systems in real life. In recent decades, big data analytics and machine learning methods have been widely used and achieved better performance on various regression issues in energy sector. Nowadays, complex and nonlinear relationships between the load and external factors for medium and long term load forecasting can be modeled based on machine learning methods. In Chen, Chang, and Lin (2004), to predict the maximum daily load of the next month, a support vector machine (SVM) model was raised. A hybrid model based on least-square support vector machine (LSSVM) and an autoregressive integrated moving average (ARIMA) was proposed in Khalid and Javaid (2020) for a long term electricity consumption forecasting in Turkey. Neural networks

* Corresponding author.
E-mail address: michael.x.zhou@csu.edu.cn (X. Zhou).

https://doi.org/10.1016/j.scs.2021.102937
Received 12 November 2020; Received in revised form 4 February 2021; Accepted 11 April 2021
Available online 16 April 2021

have been commonly conducted on combination with other methods (evolution or fuzzy methods). In Ghiassi et al. (2006), a dynamic artificial neural network (DAN2) system was presented for medium term load modeling and forecasting. Various artificial neural network (ANN) models were employed to quantify the required energy of the existing building in Ilbeigi, Ghomeishi, and Dehghanbanadaki (2020). In Chang, Fan, and Lin (2011), monthly load data and various external factors air pressure, temperature, humidity and daylight time were considered, and a weighted fuzzy neural network (FNN) was put forward for medium term forecasting. Data collection, modeling considerations and forecasting results of the MTLF based on radial basis function neural networks (RBFNN) were presented in Xia, Wang, and Mcmenemy (2010).

In energy systems, machine learning models are significant for predictive analytics of power distribution, usage and demand due to the reliability, performance, accuracy, less computational complexity and speed. Yet, the main shortcomings of neural networks are obvious, albeit with the strong learning and generalization capability, neural networks requires relatively large amount of training data for learning the data pattern. The high time complexity and the convergence problem when dealing with data of a long period of time are also drawbacks of the model. Moreover, its structure such as the number of hidden layers, learning rate and so on is also much depending on experience, which may effectively limit the interpretability. Accompanying the mature nonlinear mapping capabilities and data processing characteristics, SVR has received wide successful applications in improving load forecasting accuracy (Ajmera, Singh, & Chauhan, 2016; Chen et al., 2004; Khalid & Javaid, 2020). For a complex nonlinear regression problem, SVR performs linear regression in the high-dimensional feature space created by a kernel function using insensitive loss. Kernel functions can reduce computational complexity by mapping original space points to another high dimensional characteristic space (Cherkassky, 1997). From that point, SVR contains a certain geometric interpretation that offers several advantages over other approaches so that it provides broad application prospects in prediction (Suykens, De Brabanter, Lukas, & Vandewalle, 2002).

As the last step of model design and the first step of training, it is essential to find an appropriate parameter optimization method and strategy. In SVR, inappropriate selections of adjustable kernel parameters, regular term, and the error of regression function may cause underfitting and overfitting to a certain extent. Cross-validation and grid search are used to find best values in the entire parameter space (Jimenez, Lazaro, & Dorronsoro, 2009; Maunder & Harley, 2011). Theoretically, if sufficient prior knowledge is provided, these parameters can be easily and effectively determined according to the appropriate scale of training datasets (Cherkassky & Ma, 2004). Nevertheless, for large scale datasets and high dimensional spaces, it is not sufficient to perform the selection of hyperparameters through empirical values, which is also the main obstacle of parameters tuning. Until recently, some parameters optimization procedures using evolutionary algorithms arise frequently (Khalid & Javaid, 2020). Particle swarm optimization (PSO) was used to optimize the kernel parameter and regularization parameter of LSSVR in Li and Li (2019). In Wu, Tzeng, and Lin (2009), a novel genetic algorithm (GA) was adapted to find optimal type and parameter values of kernel function of SVR to increase the accuracy of SVR. Chaos theory is combined with firefly algorithm to optimize SVR hyperparameters in Kazem, Sharifi, Hussain, Saberi, and Hussain (2013). Despite different SVR parameter setting methods have been proposed in many studies, there are some drawbacks. On the one hand, there are actually two types of parameters often require tuning, namely, hyperparameters that need to be adjusted manually and model general parameters that are continuously adjusted with training. Under such a scenario, existing optimization methods are many focusing on hyperparameters tuning or cast different optimization problems as an entirety that gloss over an important issue: coupling relationship. From the standpoint of this interdependent relationship, diversities between optimization problems is also ignored. On the other hand, parameters tuning of SVR is a complex nonlinear multi-modal coupling optimization problem, and it is necessary to increase the diversity of solutions and reduce the possibility of falling into local optimum. It is worth mentioning that state transition algorithm (STA) has designed different state transformation operators including rotation, translation, expansion and axesion and each operator in the algorithm can generate geometric neighborhoods with regular shapes and controllable sizes. From that point, this algorithm can guarantee the effectiveness and diversity of candidate solutions, and finally converge to the global optimal solution.

The entire parameter optimization is hierarchical. In some kernel-based methods, through the 'kernel trick' and regularized convex loss function, the error minimization is finally transformed into a convex optimization problem that is not difficult to handle (Bennett, Kunapuli, Hu, & Pang, 2008). But it typically contains several hyperparameters that should be specified in advance. In SVR, it is necessary to pick the proper kernel function, regular term C, and ϵ-tube before weight and threshold which are general parameters to be optimized. Subsequently, reducing the error of in-sample testing becomes the goal of hyperparameter tuning. Therefore, the choice of hyperparameters affects the general parameters optimization and vice versa. The hyperparameter tuning task is constructed as a parameter optimization problem which contains general parameters optimization as a constraint. There are hierarchical differences between the two that are interrelated as a whole. In doing so, parameters tuning is viewed as a hierarchical optimization problem.

Hierarchical optimization is inspired by the bilevel optimization which can be perceived as a static version of a non-cooperative two-person game that is proposed in Stackelberg, Peacock, and Boulding (1952) by Von Stackelberg. Both upper and lower layers have their own objective functions and constraints. The objective function and constraints of upper layer are not only related to its decision variables, but also affected the optimal solutions of the follower and vice versa. In this paper, a hierarchical optimization method in a nested manner is proposed and a hierarchical optimization algorithm based on STA is developed. The mathematical algorithm is used to reduce the enormous computational expense in lower layer and a new evolutionary based optimization technique STA is developed to adjusted hyperparameters of the SVR model. We highlight contributions of this study as follows:

- The hybrid SVR model is established for electric load forecasting. The parameters optimization of SVR is formulated as a hierarchical optimization problem, and a structured hierarchical optimization method is put forward.
- A nested optimization strategy based on STA is constructed for hierarchical parameter optimization.
- The proposed method is extended to the optimization process of parameters in SVR, which is successfully applied to industrial load forecasting.

The remainder of this paper is organized as follows: In Section 2, hybrid SVR model is established for electric load forecasting with respect to characteristics of load datasets. Parameter tuning is casted as a hierarchical optimization problem and the corresponding hierarchical optimization model is established. In Section 3, the hierarchical parameter optimization framework based on nested strategy and STA are put forward and applied to SVR's parameters search. The proposed strategy and method are further supported by results on several benchmarks and real datasets in Section 4. The conclusion is drawn in Section 5.

2. The hierarchical optimization problem of hybrid SVR for power load forecasting

2.1. Power load forecasting

Power load refers to the total electric energy drawn from power system by user's electrical equipment at a specific moment. On the basis

of different entities, electric load can be divided into industrial load, agricultural load, transportation load, and household consumption. The industrial power load consumption mainly exists in light industry, high energy-consuming industry, advanced manufacturing, and mining industry. At present, China's industrial electricity consumption has accounted for 69% of total electric load that is still increasing with continuous expansion of production. In addition, the production and consumption of industrial power should be carried out simultaneously on account of the power generation cannot be stored in large quantity.

Accordingly, forecasting in the smart grid plays an essential role in power dispatch, efficient energy management and maintains the balance between demand and supply of electricity (Bibri & Krogstie, 2017). The power consumption behaviors can be divided into two types, one is independent of external variables (termed factors-independent models), and the other is parameterized by external factors, such as weather, economic, time index (i.e. the month of year) and random events (termed factors-dependent models) (Han et al., 2018). Nevertheless, there are several shortcomings in factors-independent models. For instance, due to factors-independent models that cannot accurately reveal how external factors influence the variation of electric load, these models are lack of interpretation, and factors-dependent models may be superior to factors-independent because available external variables can be used to provide prior knowledge. The prediction is also not robust not to include external variables as inputs. Historical power consumption data and multiple external factors which are known to have a significant impact on the use of load are fully considered in this study, and a reasonable forecasting model of system to determine power load at a specific time in the future is used for undermentioned load forecasting. A variety of factors contribute to the following characteristics of datasets: (1) time series characteristics, (2) non-linearity, (3) strong correlation, and (4) scale difference. Moreover, with the different time scales, load data can be divided into short, medium and long term, whereas senior managers are more focusing on business strategy and development trends of industry that makes medium and long term load forecasting more meaningful. Fig. 1 shows two different time-scale datasets in industry. In the next section, we introduce the way to deal with the problem.

2.2. Hybrid SVR

According to the time series characteristics, determine the original space points $\boldsymbol{x}(t)$. The sliding window size k is set and $\boldsymbol{x}(t+1)$ load can be predicted from $\boldsymbol{x}(t), \boldsymbol{x}(t-1), \ldots, \boldsymbol{x}(t-k+1)$. Then, the new training set is defined as $D = \left\{\left(\boldsymbol{x}_i, y_i\right), i = 1, \ldots, N\right\}$, where N is the samples size, $\boldsymbol{x}_i \in R^n$ is the n-dimensional input vector and $y_i \in R^1$ is the corresponding one-dimensional output value. SVR aims to find a hyperplane and minimize the total deviation of all sample points from this plane.

From that point, the expected hyperplane $f(\boldsymbol{x}) = \omega^T \varphi(\boldsymbol{x}) + b$ should be as close to y as possible, including coefficient ω and b that need to be estimated according to datasets. The nonlinear function $\varphi(\boldsymbol{x})$ is used to map original samples to higher-dimensional space. ξ_i and ξ_i^* are adopted to further avoid noise from datasets. Finally, the SVR model is converted to an optimization problem:

$$\min R\left(\omega, b, \xi_i, \xi_i^*\right) = C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right) + \frac{1}{2}\|\omega\|^2$$
$$\text{s.t.}\begin{cases} y_i - f\left(\boldsymbol{x}_i\right) \le \varepsilon + \xi_i^* \\ f\left(\boldsymbol{x}_i\right) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0 \\ C > 0 \end{cases} \tag{1}$$

In addention to ε-SVR, ν-SVR that proposed by Scholkopf, Smola, Williamson, and Bartlett (2000) solves regression problems by employing $\nu \in (0, 1)$ to regulate the number of support vectors as follows:

$$\min R\left(\omega, b, \xi_i, \xi_i^*, \varepsilon\right) = C\left(\nu\varepsilon + \sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)\right) + \frac{1}{2}\|\omega\|^2$$
$$\text{s.t.}\begin{cases} y_i - f\left(\boldsymbol{x}_i\right) \le \varepsilon + \xi_i^* \\ f\left(\boldsymbol{x}_i\right) - y_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0 \\ C > 0 \end{cases} \tag{2}$$

Both SVR models have been successfully applied to predictions. However, different time-scale datasets may bring forecasting difficulties in this study. To enhance objectivity and accuracy, hybrid SVR (HSVR) that is the weighted linear combination of ε-SVR and ν-SVR has been proposed in Huang, Yang, Zhou, and Yang (2019).

In addition, following parameters are critical to HSVR:

Hyperparameters:

- The regularization coefficient C. It can be viewed as a tradeoff parameter between accuracy and model complexity.
- The width of ε-tube. Regression model tolerates ε deviation between $f(\boldsymbol{x})$ and y at most, consequently, a proper ε value can effectively reduce noise effect.
- Kernel functional parameter γ. It is noted that the radial basis function (RBF) which is widespread with high performance in developing forecasting models will be used as the kernel function in this study. Parameter γ is related to the function width, in charge of determining individual impact on the rest of training set.
- Parameter of ν-SVR. ν of ν-SVR can regulate the number of support vectors through varying from 0 towards 1 in the optimization process.
- Weights of ε-SVR and ν-SVR. Weight value k controls the weights of two types of regression models.

Model parameters:

- The regression model's linear combination weight vector w and bias b.

These parameters and corresponding combinations affect model's complexity and forecasting accuracy. Moreover, the mutual influence between these parameters may introduce uncertainties alike. Therefore, it is essential to find a proper solution approach for parameter tuning.

2.3. Hierarchical optimization problem formulation

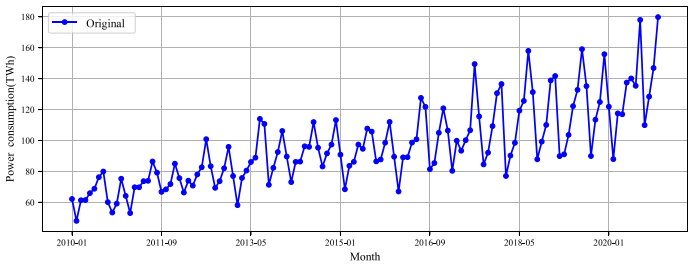
Aforementioned difficulties in parameters tuning can be settled to some extent by casting a hierarchical optimization formulation. The prerequisite for updating general parameters of this model is that relevant hyperparameters are known, thereafter, the error obtained by in-sample testing becomes the target of hyperparameter tuning. General parameters optimization problem acts as a parameterized constraint to hyperparameter tuning, and their hierarchy and relevance post this problem as a typical hierarchical structure optimization problem.

HSVR is the weighted linear combination of the two SVRs. To facilitate understanding, ε-SVR is taken as an example, the original regression problem can be defined as:

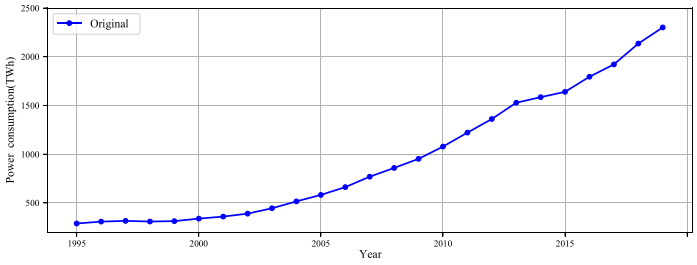
$$\min R = \frac{C}{N}\sum_{i=1}^{N} J\left(y_i, f\left(\boldsymbol{x}_i\right)\right) + \frac{1}{2}\|\omega\|^2 \tag{3}$$

Lower layer convex optimization is parameterized by minimizing $\frac{1}{2}\|\omega\|^2$ for maximizing the interval. To make majority of samples fall into ε-tube, let J as the loss function where C and ξ are regularization coefficient and relaxation variable respectively.

$$J_\varepsilon(y, f(\boldsymbol{x})) = \max\{|f(\boldsymbol{x}) - y| - \varepsilon, 0\} = \xi \tag{4}$$



(a) The Monthly dataset contains annual industrial electricity consumption data of a province in China from January 2010 to December 2020.



(b) The Yearly dataset contains annual industrial electricity consumption data of a province in China from 1995 to 2019.

Fig. 1. Medium and long term power load datasets.

To guarantee the minimum error of regression problem, the upper level optimization is defined as follows:

$$\min_{C,\varepsilon,\gamma} F = \frac{1}{N}\sum_{i=1}^{N}\left(\omega^T\varphi\left(\boldsymbol{x}_i\right)+b-y_i\right)^2 \tag{5}$$

With respect to previous analysis, upper layer hyperparameters C, ε, γ, and lower layer general parameters ω, b typically require tuning, so that corresponding objective functions and constraints are listed to complete the optimization formulation. The hierarchical SVR is defined as follows:

$$\begin{cases} \min_{C,\varepsilon,\gamma} F = \frac{1}{N}\sum_{i=1}^{N}(\omega^T\varphi(\boldsymbol{x}_i)+b-y_i)^2 \\ \text{s.t. } C,\varepsilon,\gamma \geq 0 \\ \min_{\omega,b} f = C\sum_{j=1}^{M}(\xi_j)+\frac{1}{2}\|\omega\|^2 \\ \text{s.t.} \begin{cases} y_j-\omega^T\varphi(\boldsymbol{x}_j)-b-\varepsilon \leq \xi_j \\ \omega^T\varphi(\boldsymbol{x}_j)+b-y_j-\varepsilon \leq \xi_j \\ \xi_j \geq 0 \end{cases} \end{cases} \tag{6}$$

The above formulation requires to minimize the loss function F in the upper layer while maximize the interval in the lower layer which is instantiated as f.

In the optimization process, the upper layer C, ε and γ act as parameters which are passed to the lower layer. After finding optimal solutions for the lower layer ω and b, the optimal lower layer vectors are fed back to F as parameters. Then, the leader determines the search mode through evolutionary algorithms to obtain satisfactory solutions. Acting iteratively by passing and feeding back, the final lower layer optimal solutions and corresponding upper layer vectors constitute the feasible solution set.

3. The proposed hierarchical parameter optimization method

In this study, the proposed hierarchical optimization method contains two important parts. Firstly, it attempts to handle the upper and lower parameter optimization problems in a nested manner. Meanwhile, many new challenges have been encountered so that a hierarchical optimization strategy cooperated with STA is proposed. The conceptual framework of the proposed method is shown in Fig. 2.

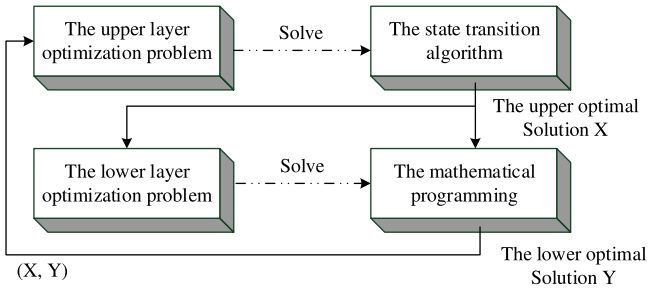


Fig. 2. The hierarchical parameter optimization method.

3.1. Nested strategy

In this case, tuning different types of parameters is a hierarchical optimization problem with the nested characteristic, that is, different layers of issues involve one layer nested in the other. For a brief overview, a set of feasible candidates of lower layer optimization task are affected by the given upper layer decisions while the feasible vectors available to either layer is interdependent.

Hierarchical optimization formulation reveals the “nested” connection in parameters tuning problems, which may cause some difficulties: Firstly, the incomplete understanding of the upper layer task by the lower layer may lead to the uncertainty of results. Besides, the upper layer optimization problem is non-convex, nonlinear and easy to yield a local optimal solution, which plays the leading role in the entire optimization process. In addition, even if constituent functions of either layer satisfies the convexity assumption, the task may still act as

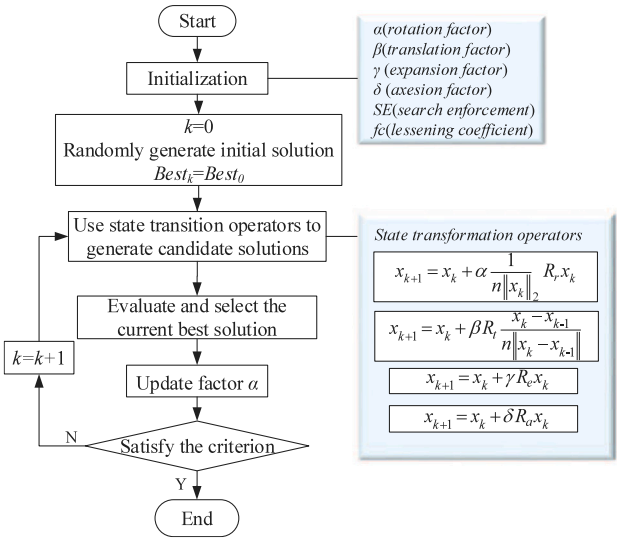


Fig. 3. The flowchart of STA.

the non-convex optimization. In brief, due to asymmetry, uncertainty and non-convexity, the hierarchical optimization problem is difficult to solve. Moreover, the leader–follower hierarchical relationship and cyclical iteration of optimization also involve high time complexity in the solution process.

Apparently, classical approaches show great limitations in dealing with these difficulties, while kinds of evolutionary approaches have been employed to tackle these problems. At present, nested evolutionary strategies are presented for achieving the hierarchical optimal performance (Sinha, Malo, Frantsev, & Deb, 2014), which are implemented in primarily two ways in terms of evolutionary algorithms: The evolutionary algorithm was applied at upper layer and the classical algorithm was used on the parameters of lower layer. In Mathieu, Pittard, and Anandalingam (1994), the upper layer used genetic algorithm, and the lower layer used linear programming. Another nested strategy was proposed in Li and Wang (2007) where simplex-based crossover strategy was used at upper layer and the lower layer used one of the classic algorithms. Finally, it is proved that the idea can be effectively used to solve the optimization task. The other one is implementing evolutionary algorithms on both layers. Wang, Ma, and Chen (2017) proposed a new algorithm which is constructed by combining two sole improved fruit fly optimization algorithms, and Li, Tian, and Min (2006) used a nested PSO to solve the problem. The effectiveness of this technique has been demonstrated on some standard test problems with a small number of variables.

In this study, an evolutionary algorithm is selected to optimize the upper layer objective function while the mathematical programming is employed to optimize the lower layer. According to the mechanism of hierarchical optimization, the upper layer optimization task takes the dominant place while the lower layer is secondary. Instead of spending time and efforts to find optimal solutions of the lower layer, we rather prefer to get better optimization results for the upper layer based on satisfactory solutions of lower layer. Otherwise, although evolutionary algorithms that allow a wide exploration in feasible regions may find global optimal solutions, the underlying computational expense is huge. The lower layer calculation speed should be accelerated while the upper evaluation time should be appropriately decreased to effectively reduce the time complexity. Therefore, we develop a hierarchical method relying on a nested strategy where evolutionary algorithm is developed to optimize the upper layer and obtain the global optimal solutions leading the entire optimization problem while lower layer applies the mathematical algorithm to improve computational efficiency.

3.2. Hierarchical parameter optimization strategy based on STA

3.2.1. A brief description of STA

A structured intelligent algorithm—state transition algorithm (STA) (Zhou, Yang, & Gui, 2012) is used in this study. STA can achieve optimality, rapidity and convergence in optimization process for global optimization. Due to its effective global search capability, stability and flexibility, STA shows fantastic performance in many practical applications (Zhou, Huang, Huang, Yang, & Gui, 2020; Zhou, Wang, Huang, & Yang, 2020; Zhou, Yang, & Gui, 2018; Zhou, Yang, Xie, Yang, & Huang, 2019).

In view of STA, candidate solutions to a specific optimization problem can be described as different states, and the procedure of updating solutions can be treated as state transition. By evaluating and updating candidates, the current best solution can be reserved. With iteratively update over time, solutions are gradually transferred to the optimal state by particular transformation operators. In the continuous STA, the unified form of solution generation is shown as follows:

$$\begin{cases} \boldsymbol{x}_{k+1} = A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k \\ y_{k+1} = f(\boldsymbol{x}_{k+1}) \end{cases} \tag{7}$$

where A_k and B_k are the specific state transformation operators, $\boldsymbol{x}_k$ is a state that can be interpreted as a solution to the optimization task and the fitness function is f.

Four special state transformation operators are employed to generate the candidate solution set for both local and global search. These operators shown in Fig. 3 are the core of STA, which can generate candidates in the solution space. The first two operators represent rotation and translation transformation, which can realize local search, while the remains operators act as the expansion and axesion transformation in charge of global search. Corresponding to above operators, α (rotation factor), β (translation factor), γ (expansion factor) and δ (axesion factor) are positive.

It is worth mentioning that parameters tuning of SVR is a complex nonlinear multi-modal coupling optimization problem, and each operator in this algorithm can generate geometric neighborhoods with regular shapes and controllable sizes. From that point, this algorithm can guarantee the effectiveness and diversity of candidate solutions. In addition, the STA uses different operators alternate rotation in a timely manner so that can quickly find the global optimal solution in the sense of probability.

In brief, an optimization problem can be solved by STA through several kinds of state transformation operations. Based on the given initial solutions, candidates are generated by sampling and running certain state operators. Comparing with previous solutions, the 'greedy criterion' is used to select and update current solutions. This process is repeated for several times until the specified termination condition is met.

3.2.2. The proposed hierarchical parameter optimization algorithm

The main idea of this hierarchical optimization is combining convergence and globality of STA and rapidity of mathematical programming, which can handle difficulties of non-convexity and computational complexity theoretically. The pseudocode of this algorithm is shown in Algorithm 1.

Algorithm 1 The proposed hierarchical parameter optimization algorithm

Require:
Initialize parameters of STA: α, β, γ, δ and SE;
Initialize iteration N;
Ensure:
Optimal solutions;
while Iteration N is not met **do**
  Generate SE candidate states X by a certain state transformation operator;
  while The lower layer optimization objective is not met **do**
    Select X as parameters to update Y by mathematical programming;
    Obtain and fix optimal parameter values of Y;
  end while
  Get the solution set (X, Y);
end while

3.3. Hierarchical parameter optimization method for SVR

In Section 2, different types of parameters optimization in HSVR is a typical hierarchical optimization task. Next, we will use the proposed method to optimize different types of parameters.

Step1: Choose parameters for STA. Parameters of this algorithm are chosen in this stage, such as transformation factors $\alpha = 1$, $\beta = 1$, $\gamma = 1$, $\delta = 1$ and $SE = 20$. The maximal number of iterations is 100.

Step2: Initialization. Set upper layer decision variables $X \triangleq \{\mathbf{C}, \varepsilon, \gamma\}$.

Step3: Train and optimize lower layer parameters. According to the goal of maximizing the interval, upper layer decision variables act as parameters to optimize $Y \triangleq \{\omega, \mathbf{b}\}$ in lower layer.

Step4: Train and optimize upper layer parameters. Consider Y as upper layer parameters.

> **Step4a**: Generate neighborhoods for the upper layer through state transition operators.
> **Step4b**: Update current solutions. Comparing current solutions with collected samples of optimal solutions, gather a certain number of candidates from neighborhoods through a sampling mechanism. The 'greedy criterion' is posed as the update mechanism to replace current solutions.

Step5: Check the stop condition. Values of X are fixed and passed as the best to the lower layer, if the stop condition is satisfied, optimization process is terminated and the final optimal solution (X, Y) is output. Otherwise, return to **Step 3**. Finally, this method optimizes parameters of the HSVR for industrial power load forecasting and the framework is shown in Fig. 4.

4. Experiment and discussion

4.1. Benchmark experiments

In this section, the proposed hierarchical optimization algorithm based on STA is compared with the hierarchical strategy with other evolutionary algorithms(PSO and GA). In Table 1, four benchmark test problems shown in Sinha, Lu, Deb, and Malo (2020) and Oduguwa and Roy (2002) are selected. The experiment is performed with the fixed number of iterations(50) in 10 independent runs and the results are presented in Fig. 5 and Table 2.

The test problem 1 in Oduguwa and Roy (2002) is used as an example. At the best solution $X = (0.4022, 0.8011)$ and $Y = (1.9996, 0.0000)$, the upper layer objective function is $F = -3.9193$ as well as the lower layer objective function is $f = -2.0087$. Comparing with the reference results and results of hybrid model with GA or PSO, solutions found by these algorithms are not the global optimal solution while the proposed algorithm can solve this problem to some extend.

From Table 2 and Fig. 5, we can see that problems 1~4 solved by the proposed algorithm are equal to or better than the results in references and other evolutionary algorithms. Table 3 indicates the comparison between the best and worst results found by the proposed algorithm in 10 independent runs. We find that the worst solution is the same or very close to the optimal solution, which shows the stability of this algorithm. The numerical experiments presented here are preliminary while additional testing on practical and synthetic problems are carried out in the next section.

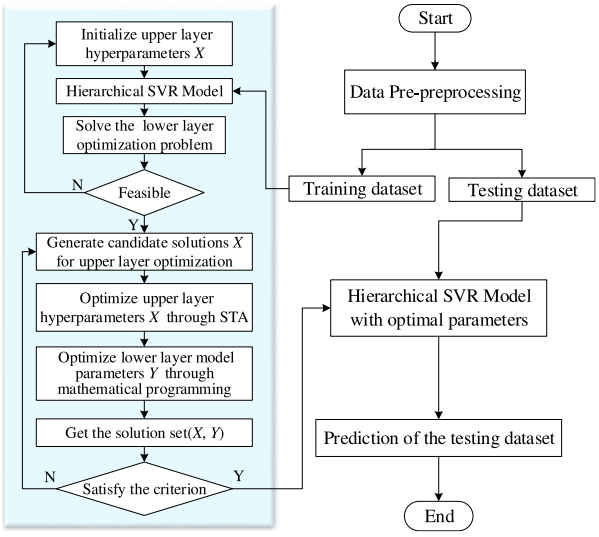


Fig. 4. The flowchart of HSVR-STA.

Table 1
Test problems.

Problem	Formulation	Setting
TP1	$\min_{x\geq 0} F(x,y) = -8x_1 - 4x_2 + 4y_1 - 40y_2 - 4y_3$ $\min_{y\geq 0} f(x,y) = x_1 + 2x_2 + y_1 + y_2 + 2y_3$ s.t. $\begin{cases} y_2 + y_3 - y_1 \leq 1 \\ 2x_1 - y_1 + 2y_2 - 0.5y_3 \leq 1 \\ 2x_2 + 2y_1 - y_2 - 0.5y_3 \leq 1 \end{cases}$	$n = 2, m = 3$
TP2	$\min_{x} F(x,y) = rx^{\top}x - 3y_1 - 4y_2 + 0.5y^{\top}y$ $\min_{y\geq 0} f(x,y) = 0.5y^{\top}Hy - b(x)^{\top}y$ s.t. $\begin{cases} -0.333y_1 + y_2 - 2 \leq 0 \\ y_1 - 0.333y_2 - 2 \leq 0 \end{cases}$ $r = 0.1 \quad H = \begin{bmatrix} 1 & 3 \\ 3 & 10 \end{bmatrix} \quad b(x) = \begin{bmatrix} -1 & 2 \\ 3 & -3 \end{bmatrix} x$	$n = 2, m = 2$
TP3	$\min_{x} F(x,y) = (x-1)^2 + (y-1)^2$ $\min_{y} f(x,y) = 0.5y^2 + 500y - 50xy$	$n = 1, m = 1$
TP4	$\min_{x\geq 0} F(x,y) = (x-1)^2 + 2y_1 - 2x$ $\min_{y_1,y_2\geq 0} f(x,y) = (2y_1 - 4)^2 + (2y_2 - 1)^2 + xy_1$ s.t. $\begin{cases} 4x + 5y_1 + 4y_2 \leq 12 \\ 4y_2 - 4x - 5y_1 \leq -4 \\ 4x - 4y_1 + 5y_2 \leq 4 \\ 4y_1 - 4x + 5y_2 \leq 4 \end{cases}$	$n = 1, m = 2$

4.2. Industrial power load forecasting

To validate the advantage of HSVR model and the proposed parameter optimization method, in this section, this method is conducted

Table 2
Comparison of the best results found by proposed algorithm and the reference results.

No.	$F(x^*, y^*)$				$f(x^*, y^*)$			
	GA	PSO	STA	Ref	GA	PSO	STA	Ref
TP1	−28.4129	−28.7481	−29.2	−29.2	3.0477	3.2130	3.2	3.2
TP2	−3.8310	−3.9182	−3.9193	−3.6	−0.3739	−1.9048	−2.0087	−2
TP3	81.3283	81.3283	81.3279	82.44	−0.3532	−0.3123	−0.3360	0.271
TP4	−1.1976	−1.0895	−1.2099	−1.2091	7.5738	7.2104	7.6173	7.6145

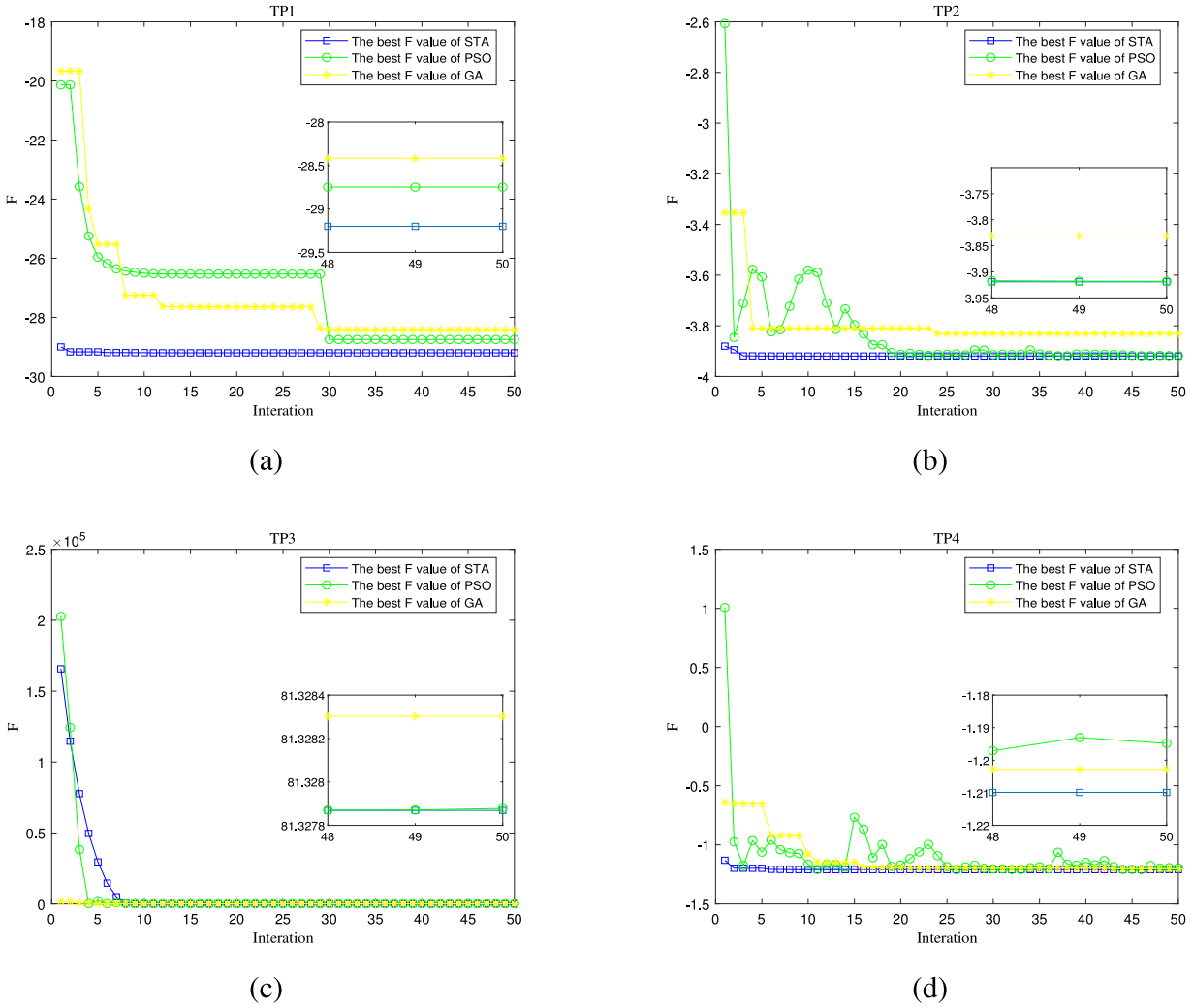


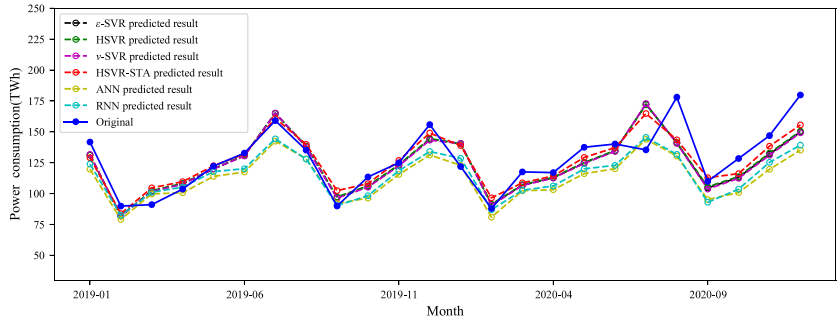
Fig. 5. The upper fitness value changes with the number of iterations.

Table 3
The best and worst results found by proposed algorithm.

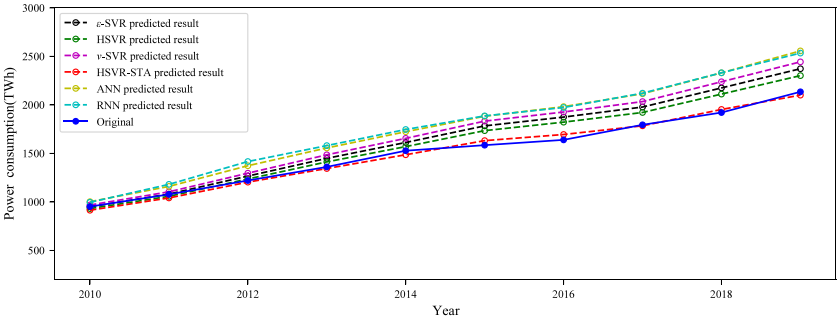
No.	$F(x^*, y^*)$		$f(x^*, y^*)$	
	STA-Best	STA-Worst	STA-Best	STA-Worst
TP1	−29.2	−29.1984	3.2	3.1999
TP2	−3.9193	−3.9193	−2.0087	−2.0087
TP3	81.3279	81.3279	−0.3360	−0.3359
TP4	−1.2099	−1.2095	7.6173	7.6172

on solving real-world prediction tasks, namely, two cases of power consumption forecasting from industry in a province of China as shown in Fig. 1 from Section 2. Historical load data and multiple external factors that are known to have a significant impact on power usage are used to train this method.

To be specific, the monthly data set contains data of more than 10 years (132 months) from January 2010 to December 2020. Each sample includes total monthly electrical load, monthly mean of monthly maximum and minimum temperature. Moreover, under the action of several months of the year, the pseudo-periodicity of electric load is 12 months. Due to external factors, the pseudo-periodicity changes causing a random variation of pseudo-periodicity in electric load. Although the load patterns are different, most of the patterns follow similar trends. Therefore, exploiting the characteristic of pseudo-periodicity, the month of the year is used as the independent variable in this study to increase the input dimension of datasets and the complexity of prediction models. The data set is divided into two parts: training set and test set. The training set is made up of 108 months of data from January 2010 to December 2018 and the test set is made up of 2 years (24months) of data from January 2019 to December 2020. The yearly data set contains data of 25 years from 1995 to 2019. Each sample includes 3 dimensional attributes, namely total yearly electrical load, gross domestic product (GDP) and average retail price of fuel



(a) Prediction results on the Monthly test dataset.



(b) Prediction results on the Yearly test dataset.

Fig. 6. Prediction results on the Monthly and Yearly test datasets.

Table 4
Datasets classification for training and testing.

Dataset	Data samples	Input features variables	Training data duration	Test data duration
Month	132	15	January 2010~December 2018	January 2019~December 2020
Year	25	3	1995~2009	2010~2019

commodity. The data set is also divided into training set (1995~2009) and test set (2010~2019)(see Table 4).

To assess the performance of forecasting values, the following evaluation indicators are commonly used:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} \mid y_i - \hat{y}_i \mid \tag{9}$$

$$\mathrm{R}^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(\overline{y_i} - \hat{y}_i)^2} \tag{10}$$

where y_i and $\hat{y}_i$ represent the actual and predicted values respectively, N is the test dataset size and i is the index value of test instances. Furthermore, 20 repeated experiments are conducted on a personal computer with 2.4 GHz CPU and 8GB RAM, and the average of evaluation results are used as the final result. Operators' factors α, β, γ and δ are given as 1 while k is defined as 0.5 to control weights of ε-SVR and ν-SVR.

The HSVR model is compared with ε-SVR, ν-SVR and other deep-learning models (such as ANN and RNN), and the proposed optimization method (HSVR-STA) is compared with the single layer parameter optimization approach. During the test phase, the following comparative curves of actual and forecast values in monthly and yearly datasets are presented on Figs. 6(a) and 6(b). The comprehensive evaluation results of the experiment is shown in Table 5.

Table 5
Comparison results of different methods on datasets.

Dataset	Method	MAE	RMSE	R^2
Month	ε-SVR	9.3891	14.3541	0.6814
	ν-SVR	9.5776	13.7917	0.6913
	HSVR	8.4027	13.6916	0.6959
	ANN	8.6778	16.7256	0.6184
	RNN	9.8771	15.2949	0.6130
	HSVR-STA	8.2319	12.6010	0.7598
Year	ε-SVR	96.7963	93.8935	0.6985
	ν-SVR	85.9889	109.4589	0.6561
	HSVR	60.0047	70.3142	0.8573
	ANN	108.1154	127.6541	0.6348
	RNN	114.0184	128.7730	0.7298
	HSVR-STA	28.3358	31.6916	0.9931

Firstly, we can see that the weighted linear combination model of ε-SVR and ν-SVR is similar to or better than each of them in these two cases, and the MAE obtained by the proposed HSVR model for monthly and yearly test datasets are 8.4027 and 60.0047, which are much lower than that obtained by ANN and RNN models. Hence, the proposed hybrid model is very effective for these two cases. Secondly, in HSVR, forecasting results obtained by the hierarchical parameter optimization method are closer to the original data according to Fig. 6. As shown in Table 5, RMSE and MAE obtained by the proposed optimization method for monthly and yearly test datasets are lower than other models, which are also much lower than that obtained by the hybrid model with single

layer optimization approach, and the correlation index R^2 are 0.7598 and 0.9931 respectively, which reveals a significant improvement in accuracy. Noted that, the hierarchical parameter optimization method based on HSVR can provide more stable and accurate forecasting using less training data in comparison with ANN and RNN. In comparison to the single layer optimization procedure, employing STA to optimize upper layer parameters and referring the hierarchical process to realize parameter optimization of the model can obtain better prediction results. The proposed hierarchical optimization approach might thus be possible a specialized model parameter optimization method that has significantly influence on power load forecasting. The comprehensive results show that the proposed parameter optimization method significantly improves the model performance.

5. Conclusion

Electrical load forecasting is an important part of power system which is of great significance to the national economy and sustainable development of society. In this study, a hybrid SVR is constructed for industrial power load forecasting and parameter tuning task of SVR is formulated as a hierarchical optimization problem. A hierarchical parameter optimization approach based on nested strategy and STA is proposed to find optimal hyperparameters and model parameters. Numerical results on several benchmark functions have demonstrated the convergence and effectiveness of the proposed algorithm. Finally, the proposed method based on SVR has offered significant improvement in the industrial power load forecasting.

Declaration of competing interest

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

Acknowledgments

This study was funded by the National Natural Science Foundation of China (Grant No. 61988101, 61873285, 61860206014), the Research Foundation of Science and Technology of Hunan Province, China, under Grant 2019RS1003.

References

Abu-Shikhah, Nazih, & Elkarmi, Fawwaz (2011). Medium-term electric load forecasting using singular value decomposition. *Energy*, *36*(7), 4259–4271.

Ajmera, Siddharth, Singh, Alok, & Chauhan, Vibhor (2016). An approach towards medium term forecasting based on support vector regression. In *2016 IEEE 7th power India international conference* (pp. 1–6).

Bennett, Kristin P., Kunapuli, Gautam, Hu, Jing, & Pang, Jong-Shi (2008). Bilevel optimization and machine learning. In *IEEE world congress on computational intelligence* (pp. 25–47). Springer.

Bibri, Simon Elias, & Krogstie, John (2017). Smart sustainable cities of the future: An extensive interdisciplinary literature review. *Sustainable Cities and Society*, *31*, 183–212.

Chang, Pei Chann, Fan, Chin Yuan, & Lin, Jyun Jie (2011). Monthly electricity demand forecasting based on a weighted evolving fuzzy neural network approach. *International Journal of Electrical Power & Energy Systems*, *33*(1), 17–27.

Chen, Bo Juen, Chang, Ming Wei, & Lin, Chih Jen (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. *IEEE Transactions on Power Systems*, *19*(4), 1821–1830.

Cherkassky, Vladimir (1997). The nature of statistical learning theory. *IEEE Transactions on Neural Networks*, *8*(6), 1564.

Cherkassky, Vladimir, & Ma, Yunqian (2004). Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, *17*(1), 113–126.

Ghiassi, M., Zimbra, David K., & Saidane, H. (2006). Medium term system load forecasting with a dynamic artificial neural network model. *Electric Power Systems Research*, *76*(5), 302–316.

Han, Lingyi, Peng, Yuexing, Li, Yonghui, Yong, Binbin, Zhou, Qingguo, & Shu, Lei (2018). Enhanced deep networks for short-term and medium-term load forecasting. *IEEE Access*, *7*, 4045–4055.

Huang, Shyh-Jier, & Shih, Kuang-Rong (2003). Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Transactions on Power Systems*, *18*(2), 673–679.

Huang, Zhaoke, Yang, Chunhua, Zhou, Xiaojun, & Yang, Shengxiang (2019). Energy consumption forecasting for the nonferrous metallurgy industry using hybrid support vector regression with an adaptive state transition algorithm. *Cognitive Computation*, *12*, 1–12.

Ilbeigi, By Marjan, Ghomeishi, Mohammad, & Dehghanbanadaki, Ali (2020). Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. *Sustainable Cities and Society*, *61*, Article 102325.

Jimenez, Alvaro Barbero, Lazaro, Jorge Lopez, & Dorronsoro, Jose R. (2009). Finding optimal model parameters by deterministic and annealed focused grid search. *Neurocomputing*, *72*(13), 2824–2832.

Kazem, Ahmad, Sharifi, Ebrahim, Hussain, Farookh Khadeer, Saberi, Morteza, & Hussain, Omar Khadeer (2013). Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing Journal*, *13*(2), 947–958.

Khalid, Rabiya, & Javaid, Nadeem (2020). A survey on hyperparameters optimization algorithms of forecasting models in smart grid. *Sustainable Cities and Society*, *61*, Article 102275.

Li, Zirong, & Li, Lian (2019). A hybrid model of least squares support vector regression optimized by particle swarm optimization for electricity demand prediction. In *Proceedings of the 2019 11th international conference on machine learning and computing* (pp. 91–103).

Li, Xiangyong, Tian, Peng, & Min, Xiaoping (2006). A hierarchical particle swarm optimization for solving bilevel programming problems. *Lecture Notes in Computer Science*, 1169–1178.

Li, Hecheng, & Wang, Yuping (2007). A hybrid genetic algorithm for solving nonlinear bilevel programming problems based on the simplex method. In *Third international conference on natural computation* (vol. 4) (pp. 91–95).

Li, Chaojie, Yu, Xinghuo, Huang, Tingwen, & He, Xing (2018). Distributed optimal consensus over resource allocation network and its application to dynamical economic dispatch. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(6), 2407–2418.

Li, Chaojie, Yu, Xinghuo, Yu, Wenwu, Chen, Guo, & Wang, Jianhui (2017). Efficient computation for sparse load shifting in demand side management. *IEEE Transactions on Smart Grid*, *8*(1), 250–261.

Mathieu, Richard G., Pittard, L., & Anandalingam, G. (1994). Genetic algorithm based approach to bi-level linear programming. *Rairo-Operations Research*, *28*(1), 1–21.

Maunder, Mark N., & Harley, Shelton J. (2011). Using cross validation model selection to determine the shape of nonparametric selectivity curves in fisheries stock assessment models. *Fisheries Research*, *110*(2), 283–288.

Oduguwa, V., & Roy, R. (2002). Bi-level optimisation using genetic algorithm. In *Proceedings 2002 IEEE international conference on artificial intelligence systems* (pp. 322–327). IEEE.

Scholkopf, Bernhard, Smola, Alexander J., Williamson, Robert C., & Bartlett, Peter L. (2000). New support vector algorithms. *Neural Computation*, *12*(5), 1207–1245.

Silva, Bhagya Nathali, Khan, Murad, & Han, Kijun (2018). Towards sustainable smart cities: A review of trends, architectures, components, and open challenges in smart cities. *Sustainable Cities and Society*, *38*, 697–713.

Sinha, Ankur, Lu, Zhichao, Deb, Kalyanmoy, & Malo, Pekka (2020). Bilevel optimization based on iterative approximation of multiple mappings. *Journal of Heuristics*, *26*(2), 151–185.

Sinha, Ankur, Malo, Pekka, Frantsev, Anton, & Deb, Kalyanmoy (2014). Finding optimal strategies in a multi-period multi-leader-follower Stackelberg game using an evolutionary algorithm. *Computers & Operations Research*, *41*(1), 374–385.

Stackelberg, Heinrich Von, Peacock, Alan T., & Boulding, K. E. (1952). The theory of the market economy. *Economica*, *20*(80), 384.

Suykens, Johan A. K., De Brabanter, J., Lukas, L., & Vandewalle, Joos (2002). Weighted least squares support vector machines : Robustness and sparse approximation. *Neurocomputing*, *48*(1), 85–105.

Wang, Guangmin, Ma, Linmao, & Chen, Jiawei (2017). A bilevel improved fruit fly optimization algorithm for the nonlinear bilevel programming problem. *Knowledge Based Systems*, *138*, 113–123.

Wu, Chih Hung, Tzeng, Gwo Hshiung, & Lin, Rong Ho (2009). A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications*, *36*(3p1), 4725–4735.

Xia, Changhao, Wang, Jian, & Mcmenemy, Karen (2010). Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks. *International Journal of Electrical Power & Energy Systems*, *32*(7), 743–750.

Yuan, Chaoqing, Liu, Sifeng, & Fang, Zhigeng (2016). Comparison of China's primary energy consumption forecasting by using ARIMA (the autoregressive integrated moving average) model and GM (1, 1) model. *Energy*, *100*, 384–390.

Zhou, Xiaojun, Huang, Miao, Huang, Tingwen, Yang, Chunhua, & Gui, Weihua (2020). Dynamic optimization for copper removal process with continuous production constraints. *IEEE Transactions on Industrial Informatics*, *16*(12), 7255–7263.

Zhou, Xiaojun, Wang, Xiangyue, Huang, Tingwen, & Yang, Chunhua (2020). Hybrid intelligence assisted sample average approximation method for chance constrained dynamic optimization. *IEEE Transactions on Industrial Informatics*, *PP*(99), 1.

Zhou, Xiaojun, Yang, Chunhua, & Gui, Weihua (2012). State transition algorithm. *Journal of Industrial and Management Optimization*, *8*(4), 1039–1056.

Zhou, Xiaojun, Yang, Chunhua, & Gui, Weihua (2018). A statistical study on parameter selection of operators in continuous state transition algorithm. *IEEE Transactions on Cybernetics*, *49*(10), 3722–3730.

Zhou, Xiaojun, Yang, Ke, Xie, Yongfang, Yang, Chunhua, & Huang, Tingwen (2019). A novel modularity-based discrete state transition algorithm for community detection in networks. *Neurocomputing*, *334*, 89–99.