Short-Term Load Forecasting Based on Integration of SVR and Stacking

ZHENQI TAN1, JING ZHANG1, (Member, IEEE), YU HE1, YING ZHANG2, GUOJIANG XIONG1, AND YING LIU3

1School of Electrical Engineering, Guizhou University, Guiyang 550025, China
2Electric Power Research Institute, Guizhou Power Grid Company Ltd., Guiyang 550002, China
3Power Grid Planning Research Center, Guizhou Power Grid Company Ltd., Guiyang 550002, China

Corresponding authors: Jing Zhang (zhangjing@gzu.edu.cn) and Yu He (yhe7@gzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51867005, and in part by the Guizhou Province Science and Technology Innovation Talent Team Project, China, under Grant [2018]5615.

ABSTRACT Selection of the kernel function by the support vector regression (SVR), for the purposes of load forecasting, is affected by the power load characteristics. The non-ideal SVR with a kernel function has low forecasting accuracy and poor generalization ability. A novel load forecasting method combining SVR and stacking is proposed in this paper. Base models are constructed based on SVRs with different kernel functions, then multiple base models are merged to obtain a base model layer via stacking algorithm. Finally, an SVR is connected as the meta-model layer. The stacking fusion model is composed of base model layer and meta-model layer. This model is trained with k-fold cross validation to enhance its generalization ability. An improved artificial fish swarm algorithm is employed to optimize the parameters to improve the forecasting accuracy of the stacking fusion model; speed variables are introduced to replace step lengths and improve the convergence speed and search ability. The forecasting accuracy and generalization ability of the proposed method are verified by comparative analysis.

INDEX TERMS SVR, load forecasting, kernel function, stacking fusion model, k-fold cross validation, improved artificial fish swarm algorithm.

I. INTRODUCTION
The forecasting of power demand is of crucial importance for the development of modern power systems. The stable and efficient management, scheduling and dispatch in power systems rely heavily on precise forecasting of future loads on various time horizons. In particular, short-term load forecasting (STLF) focuses on the forecasting of loads from several minutes up to one week into the future [1]. In the process of load forecasting, many factors such as weather and calendar rules make the load curve complicated and difficult to analyze, the effects of these uncertainties need to be explored in order to fully ensure system security [2], [3]. There is demand for innovative, highly accurate load forecasting methods to support today’s electric power systems.

Until the present, many load-forecasting methods such as the continuation method, regression analysis methods [4]–[6], time series methods [7]–[9], grey forecasting method [10], [11], artificial neural networks [12]–[14], Kalman filters [15], [16], and support vector machine (SVM) [17] are commonly used. In [18], a random search method was used to seek optimal hyper-parameters; the Long Short-Term Memory (LSTM) model with the best generalization ability was selected to improve the load forecasting effects. In [19], the weighted grey relation projection method was used to extract features and the improved particle swarm optimization algorithm was used to optimize model hyperparameters; finally, the support vector machine was employed for forecasting, which improved the forecasting accuracy. In [20], a variance-adjusted gradient boosting algorithm for approximating a Gaussian process regression (VAGR) was used to seek results of nonparametric regression, which could effectively avoid the cubic time complexity hindering the regression to the expansion of large data sets in the Gaussian process. In [21], a method was used to combine grey model with semi-parametric regression model by time-varying weight for load forecasting, which would make forecasting more accurate. In [18]–[21], different load forecasting models were compared. The results showed that fusion models and the method of hyperparameter...
optimization could effectively promote the forecasting accuracy of load forecasting. The SVR method has been widely investigated [22]–[24]. In [22], the method adopted the load features and the temperature features extracted by individual LSTM networks provides good forecasting results. In [23], a method based on EMD and SVR was used to forecast wind speed. In [24], a model based on affinity propagation SVR was used to forecast power load, and seek the optimal hyperparameters by particle swarm optimization. With regard to the recent research works, two problems arise in the use of SVR to forecast the data. First, the hyperparameter optimization algorithm used has problems with local optimal solutions and excessive iteration time, and secondly, the selection of the kernel function depends on experience to select the radial basis kernel function (RBF), and no comparison whether there is a more suitable kernel function.

To solve the problems, the improved artificial fish swarm algorithm and the model based on integration of SVR and stacking are proposed in this paper, respectively. In addition, the stacking fusion model can effectively improve the forecasting accuracy and generalization ability. The effectiveness of the proposed method is verified by comparing with other models.

The rest of the paper is organized as follows. Section II contains the introduction to principle of SVR and kernel functions. In section III, the principles of improved artificial fish swarm algorithm and stacking fusion model are introduced. In section IV, we present the experimental results and discussion. Finally, the conclusions are summarized in section V.

II. SVR PRINCIPLE BASED ON DIFFERENT KERNEL FUNCTIONS

In section II, the principle of the SVR model and analysis of the application of different kernel functions are introduced in detail, corresponding to section A and section B.

A. SVR PRINCIPLE

SVR is a branch of SVM applied in the field of function fitting. The set of training samples \( D = \{ (x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m) \} \), \( y_i < \varepsilon, y_i > -\varepsilon \). The form of the trained SVR model is \( f(x) = w^T x + b \). In other words, the corresponding forecasting value \( f(x_i) \) derived from a given \( x_i \) approximates the true value \( y_i \) to the greatest extent possible while \( w \) and \( b \) are the model parameters to be determined.

The type of hard-margin SVR optimization problem is expressed in the following form:

\[
\begin{align*}
\max_{w,b} & \quad \min_{x_i} \frac{1}{2} \|w\|^2 \\
\text{s.t.} & \quad y_i(w^T x_i + b) \leq 1 - \xi_i, \quad \forall i
\end{align*}
\]

Equation (1) is the objective function, that is, the maximum margin of the shortest distance between the samples contained in the space and the hyperplane. The constraint condition is that the distance from the farthest sample in the space to the hyperplane does not exceed \( \varepsilon \). After simplification, the following convex quadratic optimization formula can be obtained:

\[
\begin{align*}
\min_{w,b} & \quad 0.5 \|w\|^2 \\
\text{s.t.} & \quad y_i(w^T x_i + b) \leq 1 - \xi_i, \quad \forall i
\end{align*}
\]

By introducing slack variables \( \xi_i \) and penalty factors, Formula (2) can be rewritten as follows:

\[
\begin{align*}
\min_{w,b,\xi_i} & \quad 0.5 \|w\|^2 + C \sum_{i=1}^{N} (\xi_i + \tilde{\xi}_i) \\
\text{s.t.} & \quad y_i(w^T x_i + b) - \xi_i \geq 1 - \xi_i, \quad \forall i
\end{align*}
\]

where \( \xi_i, \tilde{\xi}_i \) are slack variables and \( C \) is the penalty factor in Formula (4). Lagrange’s Theorem is used to transform this into an unconstrained mathematical equation, then the Lagrange function is transformed into a duality. A mapping function is introduced here to map the input space to a high-dimensional linearly separable Hilbert space for seeking the solution to the nonlinear regression problem. Finally, the kernel function \( K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle \) eliminated tedious
calculations in high-dimensional space, where \( \varphi() \) represents the mapping function and \( \langle, \rangle \) represents the inner product. After simplification, the following dual problem can be obtained:

\[
\min_{\alpha^{(s)} \in \mathbb{R}^{2N}} \frac{1}{2} \sum_{i,j=1}^{N} (\alpha_i^* + \alpha_i)(\alpha_j^* + \alpha_j)K(x_i, x_j) + \varepsilon \sum_{i=1}^{N} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) \\
\text{s.t.} \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) = 0, \quad 0 \leq \alpha_i^{(s)} \leq C, \quad i = 1, \ldots, N
\]

where \( \alpha^{(s)} = (\alpha_1^{(s)}, \ldots, \alpha_N^{(s)}) \) is the Lagrange multiplier vector. Equation (7) provides \( \alpha^{(s)} = (\alpha_1^{(s)}, \ldots, \alpha_N^{(s)})^T \), and then \( b^\wedge = y_j - \sum_{i=1}^{N} (\alpha_i^{(s)} - \alpha_i^*)K(x_i, x_j) + \varepsilon \), and finally the decision function \( y = \sum_{i=1}^{N} (\alpha_i^{(s)} - \alpha_i^*)K(x_i, x_j) + b^\wedge \).

### B. KERNEL FUNCTION

The selection of the kernel function is the key to an effective SVR. Different kernel functions lead to different learning and generalization capabilities in the forecasting model [25]. The most common ones are as follows.

1. The linear kernel function can rapidly resolve linear problems with relatively few parameters. The inner product expression of the linear kernel function is as follows:
   \[ K(x_1, x_2) = x_1^T x_2 \]  

2. The RBF kernel function more stably processes nonlinear problems [26], and provides ideal fitting effects after selecting accurate parameters. The inner product expression of the RBF kernel function is as follows:
   \[ K(x_1, x_2) = \exp(-\frac{\|x_1 - x_2\|^2}{2\sigma^2}) \]  

3. The polynomial (Poly) kernel function solves nonlinear problems and is primarily suitable for orthogonally normalized data. The inner product expression of the Poly kernel function is as follows:
   \[ K(x_1, x_2) = (\alpha x_1^T x_2 + c)^d \]  

The sigmoid kernel function deals with nonlinear problems. In this case, the SVR is a type of multilayer perceptron neural network. In a convex quadratic optimization problem, it prevents falling into the local optimum. The inner product expression of the sigmoid kernel function is as follows:

\[ K(x_1, x_2) = \tanh(\alpha x_1^T y + c) \]

The SVR with linear, Poly or RBF are transformed into a linear problem-solving process after spatial mapping. Among them, the SVR with linear is a spatial mapping of the same dimension. Nonlinear problems can be transformed into linear problems through feature extraction. The SVR with Poly is limited by the order of polynomials and can only be mapped to a finite-dimensional space, which may have nonlinear problems. The SVR with RBF can be mapped to an infinite-dimensional space for solution, but improper selection of parameters can easily lead to over-fitting. The SVR with sigmoid is used as a multilayer perceptron neural network in this study to directly manage nonlinear problems.

### III. STACKING FUSION MODEL BASED ON IMPROVED ARTIFICIAL FISH SWARM ALGORITHM

The main contributions are described in section III. Firstly, in section A, the improved artificial fish swarm algorithm is proposed for the problems of local optimal solutions and long iteration time in the hyperparameter optimization algorithm. Secondly, in section B, the load forecasting method combining SVR and stacking is proposed to overcome the difficulty of selecting an appropriate kernel function in the SVR model.

#### A. IMPROVED ARTIFICIAL FISH SWARM ALGORITHM

As SVRs with different kernel functions observe the data differ from various space angles and structure angles, the various model parameters are inconsistent and thus difficult to adjust. An improved artificial fish swarm algorithm is used in this study instead to optimize the hyperparameters of SVRs with different kernel functions.

The artificial fish swarm algorithm simulates the behavior of swarms of fish foraging, swarming, chasing, and freedom to find food, corresponding to four behaviors in the algorithm, respectively: foraging, swarming, chasing, and freedom behavior [27]. The traditional artificial fish swarm algorithm uses fixed values in setting the hyperparameter step size. If the set step size is too large, it is prone to hovering around the optimal value, resulting in failure to obtain the optimal solution. If the set step size is too small, the convergence speed is low [28]. Seeking the optimal hyperparameters by setting the speed variable rather than the step size accelerates the convergence while securing the global optimal solution. The optimal solution on the bulletin board is updated by iteratively comparing the food concentration at the position of the artificial fish. The hyperparameters are...
deemed optimal when the satisfactory error bound or upper limit of the number of iterations is reached.

The individual state of the artificial fish can be defined as a vector $X = (x_1, x_2, \ldots, x_n)$, where $x_i$ ($i = 1, 2, \ldots, n$) is the variable for optimization. The current food concentration of the artificial fish is expressed as $Y = f(X)$, where $Y$ is the objective function value. The moving speed of the $i$th artificial fish is defined as $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ and the speed update hyperparameters as $c_1, c_2$ (the artificial fish’s knowledge of themselves and the artificial fish’s knowledge of the colonies, respectively). The optimal position found by the $i$th artificial fish is $P_i = (p_{i1}, p_{i2}, \ldots, p_{in})$, and the optimal position found by the whole fish colony is $G = (g_1, g_2, \ldots, g_n)$, and $\delta$ is the crowding degree factor.

(1) The foraging behavior: The artificial fish chooses the direction of movement through visual perception of the food concentration in the water. A state vector $X_j$ is randomly selected within the field of view. If $Y_j < Y_i$, the artificial fish moves in this direction at the speed $V_{i|next}$, otherwise, $X_j$ is reselected or random behavior is performed. This is expressed as:

$$\begin{align*}
V_{i|next} &= V_i + c_1 \xi (P_i - X_i) + c_2 \eta (G - X_i) \\
X_{i|next} &= X_i + \gamma \cdot V_{i|next} \cdot \frac{X_j - X_i}{\|X_j - X_i\|}
\end{align*}$$

(12)

where $\xi$, $\eta$, and $\gamma$ are random numbers within range of 0-1.

(2) The swarming behavior: A certain colony of artificial fish gathering in groups for collective foraging under two principles. Moving to the center of neighboring partners as much as possible and avoiding overcrowding. The number of artificial fish found by the $i$th artificial fish in its field of view is $n_f$ and the state of the artificial fish in the center position is $X_c$. When $Y_j/n_f < \delta Y_i$, the food concentration is sufficient and the artificial fish move in the direction at the speed $V_{i|next}$, otherwise, they will perform foraging behavior. This is expressed as:

$$\begin{align*}
V_{i|next} &= V_i + c_1 \xi (P_i - X_i) + c_2 \eta (G - X_i) \\
X_{i|next} &= X_i + \gamma \cdot V_{i|next} \cdot \frac{X_j - X_i}{\|X_j - X_i\|}
\end{align*}$$

(13)

(3) The chasing behavior: When a certain fish or several fish find food, the fish around them “follow” and swim towards the artificial fish $X_j$ in the best state in the field of view. Similar to swarming behavior, they avoid overcrowding, otherwise, they perform foraging behavior. The chasing behavior is expressed as:

$$\begin{align*}
V_{i|next} &= V_i + c_1 \xi (P_i - X_i) + c_2 \eta (G - X_i) \\
X_{i|next} &= X_i + \gamma \cdot V_{i|next} \cdot \frac{X_j - X_i}{\|X_j - X_i\|}
\end{align*}$$

(14)

(4) The freedom behavior: It is complementary behavior to foraging behavior. The state vector $X_j$ randomly selected by the artificial fish in the field of view always does not satisfy $Y_j < Y_i$, so a state is randomly selected to swim out toward and obtain a new state while effectively avoiding the local optimum:

$$\begin{align*}
V_{i|next} &= V_i + c_1 \xi (P_i - X_i) + c_2 \eta (G - X_i) \\
X_{i|next} &= X_i + \gamma \cdot V_{i|next} \cdot \frac{X_j - X_i}{\|X_j - X_i\|}
\end{align*}$$

(15)

The SVR hyperparameter is selected here as $x_j$ in $X = (x_1, x_2, \ldots, x_n)$. The artificial fish swarm performed the above four behaviors in the order of priority. The maximum number of iterations is set to obtain the optimal hyperparameters on the updated bulletin board. The process of the proposed improved artificial fish swarm algorithm is plotted as shown in Fig. 3.

**B. STACKING FUSION MODEL**

Based on the characteristics of the kernel function (Section I) and considering the insufficient learning ability and poor generalization ability they cause, a load forecasting method integrating SVR and stacking is developed in this study. The hypothesis space is, theoretically, a collective space composed of all features. The limited size of samples and massive size of the hypothesis space causes an incomplete mapping from input to output, resulting in low forecasting accuracy. However, SVRs with different kernel functions have different observation results from different data space angles and data structure angles. The stacking algorithm summarizes the different observation results to cover the entire hypothesis space (or as much of it as possible) and improve the forecasting accuracy.

A limited sample size leads to consistency between multiple hypotheses and the training set. A larger version space (set of such hypotheses) results in weaker generalization ability. The stacking algorithm used in this study learns the
version space of each SVR to minimize the version space of the overall model and improve the generalization ability. The stacking fusion model structure has upper and lower layers – a structure which strengthens the learning effect, reduces the redundancy and complexity of the forecasting model, ensures strong forecasting accuracy, and shortens the operation time. The upper layer is the base model layer, which is composed of base model layer and meta-model layer. When the model input is $X_i$, the nth basic model of the first layer is $M_n$ and the forecasting model of the second layer is $M$. The output of the nth basic model of the first layer is $M_n(X_i)$, which is used as the input of the forecasting model of the second layer. The final forecasting result $y_i$ is shown in equation (16):

$$y_i = F(F_1(X_i), \ldots, F_h(X_i), \ldots, F_N(X_i))$$ (16)

The two stages of the fusion process are analyzed separately here.

**step 1:** Based on the k-fold cross validation concept, the original data is divided into a training set $S$ and test set $T$. The training set is divided into k parts denoted as $S_1$-$S_k$. First, $S_2$-$S_k$ is used to train the base model 1, then the trained base model 1 is used to forecast the validation set $S_1$ and the test set. Next, $S_1$-$S_3$-$S_k$ are used to retrain the base model 1, then forecast the validation set $S_2$ and the test set. By analogy, $k$ sets ($S_1$-$S_k$) of forecasting values $a_1$, $a_2$, $a_k$ of the verification set and $k$ sets of forecasting values $b_1$, $b_2$, $b_k$ of the test set are obtained based on meta-model 1. $k$ sets of forecasting values of the verification set are combined to obtain the forecasting values $[a_1, a_2, \ldots, a_k]^T$ of the base model 1 as $A_1$. At the same time, the forecasting values $b_1$, $b_2$, $b_k$ of $k$ sets of the test set are averaged to obtain a set of forecasting values $B_1$. These steps are repeated for base models 2 to $N$ to obtain the forecasting values of each base model $[A_1, A_2, \ldots, A_N]$ and the forecasting values of the test set $[B_1, B_2, \ldots, B_N]$.

**step 2:** The second layer is composed of a meta model, which is trained on the forecasting values $[A_1, A_2, \ldots, A_N]$ of the base model derived in the first layer. The forecasting values $[B_1, B_2, \ldots, B_N]$ of the test set derived in the first layer are used as the test set of the second layer meta-model to obtain a final forecasting value.

The number of base models is strongly correlated with the fusion effect. A small number of models cannot achieve the complementary fusion of various models – an excessive number of models, conversely, results in redundancy and an unreasonable system parameter complexity and forecasting time. The preferable quantity of base models is between 3 and 5 [29], [30]. Four base models as the base model layer achieved the optimal fusion effect and highest learning efficiency in this study. Considering the different data spaces and data structures, the SVRs of four different kernel functions, linear, RBF, Poly, and sigmoid are used as the base model of the first layer of the stacking algorithm. According to the selection rules of the second layer of the stacking algorithm, the model with strong generalization ability or high forecasting accuracy is then selected as the second layer meta-model.
On the basis of Reference [31] and by comparison among the training results, the SVR of RBF with the highest forecasting accuracy is used as the second layer meta-model, which constitutes the entire stacking fusion model. The forecasting accuracy is highest in this case. Considering the inconsistency of various model hyperparameters and the difficulty with adjustment, the improved artificial fish swarm algorithm is employed to determine the SVR’s kernel coefficient and penalty parameter $C$. The hyperparameter values are adjusted repeatedly until satisfactory accuracy is obtained. The process of the proposed stacking fusion model is plotted as shown in Fig. 5.

IV. SIMULATION ANALYSIS

The example data and experimental results are analyzed in section IV. In section A, two sets of data are preprocessed. The input characteristics related to the load are listed in section B. In section C and section D, the effectiveness of the improved artificial fish swarm algorithm and stacking fusion model are verified by comparing with other models.

A. EXAMPLE DATA

Real-world load data from a certain area of Guizhou in 2018 and Spain in 2015 are preprocessed, respectively. The load data from a certain area of Spain is retrieved from https://www.kaggle.com/nicholasjhana/datasets. The data horizontal comparison method provided by Reference [32] is employed to identify continuously missing or continuously suddenly changed data and eliminate the impact of sudden changes caused by factors such as missing information or emergencies. The data size is constrained between $[0, 1]$ by the maximum-minimum criterion (min-max) standardization. The load from Guizhou is sampled once every 30 min over a total of 48 sampling points each day and the load from Spain is sampled once every 60 min over a total of 24 sampling points each day. The model is implemented in python. The code is executed in a desktop with Nvidia RTX2070 super graphics card and i7-8700K CPU. The predictive evaluation index is constructed with the mean absolute percentage error $e_{MAPE}$ and root mean square error $e_{RMSE}$, as shown below:

$$e_{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left( \frac{|x(i) - y(i)|}{x(i)} \right) \times 100\% \quad (17)$$

$$e_{RMSE} = \frac{1}{n} \sum_{i=1}^{n} (x(i) - y(i))^2 \quad (18)$$

where $x(i)$ and $y(i)$ denote the actual value and forecasting value at time point $i$, respectively; $n$ is the sample size.

B. INPUT AND FEATURES OF THE TRAINING SET

Taking into account various load patterns of different seasons of the year, the load data of the first 24 days of typical January, May, and August months are used as the training set corresponding to winter, spring, and summer load conditions. The test set is composed of the load data of the last 7 days in January, May, and August targeting the forecasting effect of the proposed method. The input variables are determined as the base model of the first layer is being trained. The optional input variables with higher correlation included historical information, weather information, and calendar rules, as shown in Table 1.

C. OPTIMIZATION OF MODEL HYPERPARAMETERS BASED ON IMPROVED ARTIFICIAL FISH SWARM ALGORITHM

As discussed in Section III, in the proposed method, the hyperparameters optimized by the improved artificial fish swarm algorithm include the penalty parameter $C$ and kernel coefficient $\gamma$ of each SVR. Based on the load data of different characteristics in January, May, and August, three sets of the optimized hyperparameters are obtained as shown in Table 2.

Table 2 shows significant differences in the hyperparameters of the base models. The stacking fusion model appears to reduce the forecasting error by selecting SVRs with different kernel functions. The observation results and version spaces
TABLE 1. Input variables.

| load from Guizhou | load from Spain |
|-------------------|-----------------|
| numbering         | characteristic attribute | numbering | characteristic attribute |
| 1<sub>i</sub>     | month            | 1<sub>i</sub> | month            |
| 2<sub>i</sub>     | week             | 2<sub>i</sub> | week             |
| 3<sub>i</sub>     | holidays         | 3<sub>i</sub> | holidays         |
| 4<sub>i</sub>     | temperature      | 4<sub>i</sub> | temperature      |
| 5<sub>i</sub>     | temperature one hour ago | 5<sub>i</sub> | temperature one hour ago |
| 6<sub>i</sub>     | temperature one day ago | 6<sub>i</sub> | temperature one day ago |
| 7<sub>i</sub>     | season           | 7<sub>i</sub> | season           |
| 8<sub>i</sub>     | power load one hour ago | 8<sub>i</sub> | power load one hour ago |
| 9<sub>i</sub>     | power load one day ago | 9<sub>i</sub> | power load one day ago |
| 10<sub>i</sub>    | power load two day ago | 10<sub>i</sub> | power load two day ago |
| 11<sub>i</sub>    | power load three day ago | 11<sub>i</sub> | power load three day ago |
| 12<sub>i</sub>    | power load four day ago | 12<sub>i</sub> | power load four day ago |
| 13<sub>i</sub>    | power load five day ago | 13<sub>i</sub> | power load five day ago |
| 14<sub>i</sub>    | power load six day ago | 14<sub>i</sub> | power load six day ago |

TABLE 2. Selection of model hyperparameters.

| model | kernel function | load from Guizhou | load from Spain |
|-------|-----------------|-------------------|-----------------|
|       |                 | January | May | August | January | May | August |
| Base  |                 | C | γ | C | γ | C | γ | C | γ | C | γ | C | γ |
| Model1 (SVR1) | RBF | 10 | 0.003 | 10 | 0.02 | 45 | 0.01 | 150 | 0.01 | 80 | 0.03 | 60 | 0.03 |
| Base  |                 | 1 | / | 0.1 | / | 0.1 | / | 1 | / | 1 | / | 1 | / |
| Model2 (SVR2) | linear | 0.5 | 0.01 | 1 | 0.008 | 0.1 | 0.008 | 5 | 0.01 | 5 | 0.008 | 10 | 0.005 |
| Base  |                 | Poly | 10 | 0.05 | 20 | 0.05 | 50 | 0.05 | 100 | 0.055 | 50 | 0.05 | 20 | 0.05 |
| Meta  |                 | RBF | 10 | 0.003 | 10 | 0.02 | 45 | 0.01 | 150 | 0.01 | 80 | 0.03 | 60 | 0.03 |

of different SVRs are learned to improve the forecasting accuracy and generalization performance.

D. COMPARATIVE ANALYSIS OF FORECASTING

The improved artificial fish swarm algorithm and traditional artificial fish swarm algorithm are compared on the error curve shown in Fig. 6.

Fig. 6 shows that the improved artificial fish swarm algorithm is significantly better than the original artificial fish swarm algorithm. Speed variables rather than step size is introduced to accelerate the convergence speed. The original artificial fish swarm algorithm reached the optimum after approximately 35 iterations and the improved artificial fish swarm algorithm did after approximately 17. As mentioned above, setting too small a step size results in the local optimum, setting the speed variable allows jumping out of the local optimum to approach the global optimum. The improved artificial fish swarm algorithm is selected as the hyperparameter optimization algorithm of the stacking fusion model to rapidly find the global optimal hyperparameters to ensure the best forecasting accuracy of the stacking fusion model.

The proposed method’s forecasting performance is also compared with LSTM, RF-stacking, XGBoost-stacking, and the SVR with RBF. To ensure that the stacking fusion model could find the optimal hyperparameters, the hyperparameters of the improved artificial fish swarm are set as follows: the
maximum number of iterations is 20, the number of populations is 10, the try number is 10, the crowding degree factor $\delta$ is 0.623, the speed parameters $c_1$ and $c_2$ are 2, 2, respectively. Six sets of load data for January, May, August in Guizhou and Spain are input into the stacking fusion model and other models for training and forecasting. Six sets of weekly load forecasting curves are obtained, as shown in Fig. 7.

As shown in Fig. 7, there are significant differences among the weekly load curves of different months in both regions. The weekly load curve of January 25-January 31 is the highest among them in Guizhou and Spain, which can be attributed to the load characteristics of different seasons. The three sets of weekly load curves for the weekend loads in Guizhou from 18:00 on Fridays to 24:00 on Sundays show a significant increase compared to work days. Saturday, August 25 shows the relatively low load in Guizhou due to the Zhongyuan Festival; the load reached its lowest level during this holiday as well. The load increased rapidly on the following Sunday. The forecasting accuracy of the five forecasting models are obtained, as shown in Table 3.

The forecasting accuracy as calculated (Table 3) suggests that the proposed stacking fusion model outperforms the other models. Among the three sets of forecasting indicators from Guizhou, the $\text{eMAPE}$ reaching 2.5% at minimum and 2.93% at maximum. The fitting effect of the SVR with RBF are most unsatisfactory, 0.67% higher than that
of the stacking fusion model on average, followed by that of the LSTM, RF-stacking and XGBoost-stacking, respectively. The proposed stacking fusion model shows a minimum $\text{eRMSE}$ of 160.82 MW and maximum of 242.14 MW, which are also better than the other model $\text{eRMSE}$ indicators. Among the three sets of weekly load forecasting indicators in Spain, the proposed stacking fusion model shows a minimum $\text{eMAPE}$ of 0.88% and maximum of 1.9%, the $\text{eRMSE}$ reaching 412.79 MW at minimum and 1195.81 MW at maximum.

It appears that each model achieves its respective ideal forecasting results at the smooth segment of the 66h-78h weekly load curve such as the end of May in Guizhou (Fig. 7, Table 3). When the load fluctuates drastically, for example, the SVR with RBF shows insufficient learning ability, and a high probability of over-fitting, which lead to considerable deviation from the actual data between the 78h and 102h time points. The LSTM could properly forecast the variation trend of load due to its long-term memory, but there still exist a wide gap between the forecasting values and actual values. The RF-stacking and XGBoost-stacking have the advantages of integrated learning, and the forecasting accuracy surpasses the SVR and LSTM, but they are not as effective as the integration of stacking and SVR. The proposed stacking fusion model shows excellent predictive performance and better fitting effect on the random volatility of load than the other models.

V. CONCLUSION

The stacking fusion model is proposed based on the SVR concept and utilized for short-term load forecasting with actual power load data in this paper. The proposed method is compared with other methods to confirm its feasibility and effectiveness. The conclusions can be summarized as follows. First, the stacking algorithm has higher forecasting accuracy and better generalization ability than other models because the differences observation results and version space are learned in SVRs with different kernel functions. Secondly, the generalization ability of the stacking fusion model can be improved by dividing the data set via k-fold cross validation. Third, the improved artificial fish swarm algorithm can be used to seek the optimal SVR hyperparameters, which improves the forecasting accuracy of the stacking fusion model while accelerating its convergence.

REFERENCES

[1] E. Ceperic, V. Ceperic, and A. Baric, “A strategy for short-term load forecasting by support vector regression machines,” *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.
[2] J. Zhang, L. Fan, Y. Zhang, G. Yao, P. Yu, G. Xiong, K. Meng, X. Chen, and Z. Dong, “A probabilistic assessment method for voltage stability considering large scale correlated stochastic variables,” *IEEE Access*, vol. 8, pp. 5407–5415, 2020.
[3] J. Zhang, G. Xiong, K. Meng, P. Yu, G. Yao, and Z. Dong, “An improved probabilistic load flow simulation method considering correlated stochastic variables,” *Int. J. Electr. Power Energy Syst.*, vol. 111, pp. 260–268, Oct. 2019.
[4] T. Zhao, J. Wang, and Y. Zhang, “Day-ahead hierarchical probabilistic load forecasting with linear quantile regression and empirical copulas,” *IEEE Access*, vol. 7, pp. 80699–80799, 2019.
[5] Y.-M. Wi, S.-K. Joo, and K.-B. Song, “Holiday load forecasting using fuzzy polynomial regression with weather feature selection and adjustment,” *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 596–603, May 2012.
[6] N. Tang, S. Mao, Y. Wang, and R. M. Nelms, “Solar power generation forecasting with a LASSO-based approach,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1000–1009, Apr. 2018.
[7] M. R. Haq and Z. Ni, “A new hybrid model for short-term electricity load forecasting,” *IEEE Access*, vol. 7, pp. 125413–125423, 2019.
[8] S. Jahandari, A. Kalhor, and B. N. Araabi, “Online forecasting of synchronous time series based on evolving linear models,” *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 5, pp. 1865–1876, May 2020.
[9] H. Dong, Y. Gao, X. Meng, and Y. Fang, “A multifactorial short-term load forecasting model combined with periodic and non-periodic Features--A case study of Qingdao, China,” *IEEE Access*, vol. 8, pp. 67416–67425, Oct. 2020.
[10] B. Li, J. Zhang, Y. He, and Y. Wang, “Short-term load-forecasting method based on wavelet decomposition with second-order gray neural network model combined with ADF test,” *IEEE Access*, vol. 5, pp. 16324–16331, 2017.
[11] A. Dejamkhooy, A. Dastfan, and A. Ahmadyfard, “Modeling and forecasting nonstationary voltage fluctuation based on grey system theory,” *IEEE Trans. Power Del.*, vol. 32, no. 3, pp. 1212–1219, Jun. 2017.
[12] N. K. Singh, A. K. Singh, and M. Tripathy, “A comparative study of BPNN, RBFNN and ELMNN neural network for short-term electric load forecasting: A case study of delhi region,” in Proc. 9th Int. Conf. Ind. Inf. Syst. (ICIIS), Dec. 2014, pp. 1–6.

[13] C. Cecati, J. Kolbusz, P. Rozycki, P. Siano, and B. M. Willamowski, “A novel RBF training algorithm for short-term electric load forecasting and comparative studies,” IEEE Trans. Ind. Electron., vol. 62, no. 10, pp. 6519–6529, Oct. 2015.

[14] H. Kebriaei, B. N. Araabi, and A. Rahimi-Kian, “Short-term load forecasting with a new nonsymmetric penalty function,” IEEE Trans. Power Syst., vol. 26, no. 4, pp. 1817–1825, Nov. 2011.

[15] H. M. Al-Hamadi and S. A. Soliman, “Fuzzy short-term electric load forecasting using Kalman filter,” IEEE Proc.-Gen. Transmiss. Distrib., vol. 153, no. 2, pp. 217–227, Mar. 2006.

[16] C. Guan, P. B. Luh, L. D. Michel, and Z. Chi, “Hybrid Kalman filters for very short-term load forecasting and prediction interval estimation,” IEEE Trans. Power Syst., vol. 28, no. 4, pp. 3806–3817, Nov. 2013.

[17] W.-M. Lin, R.-F. Yang, M.-T. Tsai, and C.-S. Tu, “Particle swarm optimisation aided least-square support vector machine for load forecast with spikes,” IET Gener. Transmiss. Distrib., vol. 10, no. 5, pp. 1145–1153, Apr. 2016.

[18] Y. Zhang, Q. Ai, and L. Lin, “A very short-term load forecasting method based on deep LSTM RNN at zone level,” Power Syst. Technol., vol. 43, no. 6, pp. 1889–1891, Jun. 2019.

[19] P. Zhao and Y. Dai, “Power load forecasting of SVM based on real-time price and weighted grey relational projection algorithm,” Power Syst. Technol., vol. 44, no. 04, pp. 1325–1332, Apr. 2020.

[20] H.-M. Lu, J.-S. Chen, and W.-C. Liao, “Nonparametric regression via variance-adjusted gradient boosting Gaussian process regression,” IEEE Trans. Knowl. Data Eng., early access, Nov. 15, 2020, doi: 10.1109/TKDE.2020.3053728.

[21] S. Ma, X. Chen, Y. Liao, G. Wang, X. Ding, and K. Chen, “The variable weight combination load forecasting based on grey model and semiparametric regression model,” in Proc. IEEE Int. Conf. IEEE Region 10 (TENCON), Oct. 2013, pp. 1–4.

[22] M. Imani, “Long short-term memory network and support vector regression for electrical load forecasting,” in Proc. Int. Conf. Power Gener. Syst. (TENCON), Oct. 2013, pp. 1–4.

[23] Y. Ren, P. N. Suganthan, and N. Srikanth, “A novel empirical mode decomposition with support vector regression for wind speed forecasting,” IEEE Trans. Neural Netw. Learn. Syst., vol. 27, no. 8, pp. 1793–1798, Aug. 2016.

[24] C. Li, X. Wang, Z. Cheng, and Y. Bai, “Forecasting bus passenger flows by using a clustering-based support vector regression approach,” IEEE Access, vol. 8, pp. 19717–19725, Jan. 2020.

[25] H. Lu, M. Azimi, and T. Iselye, “Short-term load forecasting of urban gas using a hybrid model based on improved fruit fly optimization algorithm and support vector machine,” Energy Rep., vol. 5, pp. 666–677, Nov. 2019.

[26] P. Jiang, F. Liu, and Y. Song, “A hybrid forecasting model based on date-framework strategy and improved feature selection technology for short-term load forecasting,” Energy, vol. 119, pp. 694–709, Jan. 2017.

[27] W. Guo, G. Qin, L. Wang, and R. Sun, “Robot path planning based on improved artificial fish swarm algorithm and MAKLINK graph,” Control Decis., vol. 35, no. 9, pp. 2145–2152, Sep. 2020.

[28] W. Yan, M. Li, X. Pan, G. Wu, and L. Liu, “Application of support vector regression cooperated with modified artificial fish swarm algorithm for wind tunnel performance prediction of automotive radiators,” Appl. Thermal Eng., vol. 164, Jan. 2020, Art. no. 114543.

[29] B. Liu, C. Qin, P. Ju, J. Zhao, Y. Chen, and J. Zhao, “Short-term bus load forecasting based on XGBoost and Stacking model fusion,” Electr. Power Autom. Eq., vol. 40, no. 3, pp. 147–153, Mar. 2020.

[30] Y. Li, Y. Wang, F. Liu, and Y. Wu, “Combination model of short-term wind speed prediction based on stacking fusion,” Power Syst. Technol., vol. 44, no. 8, pp. 2875–2882, Aug. 2020.

[31] A. Mohamed and M. E. El-Hawary, “On optimization of SVMs kernels and parameters for electricity price forecasting,” in Proc. IEEE Electr. Power Energy Conf. (EPEC), Oct. 2016, pp. 1–6.

[32] G. Li, W. Liu, and H. Yun, “A new data preprocessing method for bus load forecasting,” Power Syst. Technol., vol. 34, no. 2, pp. 149–154, Feb. 2010.