IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 30, 2022, accepted February 7, 2022, date of publication February 16, 2022, date of current version February 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151672

A New L_1 Multi-Kernel Learning Support Vector Regression Ensemble Algorithm With AdaBoost

XIAOJIN XIE[1], KANGYANG LUO[2], AND GUOQIANG WANG[1], (Member, IEEE)
[1]School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, Shanghai 201620, China
[2]School of Data Science and Engineering, East China Normal University, Shanghai 200062, China

Corresponding author: Guoqiang Wang (guoq_wang@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 11971302 and Grant 12171307, and in part by the National Statistical Science Research Project of China under Grant 2020LY067.

ABSTRACT This paper proposes a new multi-kernel learning ensemble algorithm, called Ada-L_1MKL-WSVR, which can be regarded as an extension of multi-kernel learning (MKL) and weighted support vector regression (WSVR). The first novelty is to add the L_1 norm of the weights of the combined kernel function to the objective function of WSVR, which is used to adaptively select the optimal base models and their parameters. In addition, an accelerated method based on fast iterative shrinkage thresholding algorithm (FISTA) is developed to solve the weights of the combined kernel function. The second novelty is to propose an integrated learning framework based on AdaBoost, named Ada-L_1MKL-WSVR. In this framework, we integrate FISTA into AdaBoost. At each iteration, we optimize the weights of the combined kernel function and update the weights of the training samples at the same time. Then an ensemble regression function of a set of regression functions is output. Finally, two groups of the experiments are designed to verify the performance of our algorithm. On the first group of the experiments including eight datasets from UCI machine learning repository, the MAEs and RMSEs of Ada-L_1MKL-WSVR are reduced by 11.14% and 9.08% on average, respectively. Furthermore, on the second group of the experiments including the COVID-19 epidemic datasets from eight countries, the MAEs and RMSEs of Ada-L_1MKL-WSVR are reduced by 31.19% and 29.98% on average, respectively.

INDEX TERMS Support vector regression, multi-kernel learning, AdaBoost, ensemble algorithm, regression prediction.

I. INTRODUCTION

Support vector machine (SVM) [1], [2] is an algorithm based on supervised learning mode, which can be used for data classification, model recognition and regression analysis. It has a strong mathematical foundation and theoretical support. SVMs can effectively solve the problems of small samples, nonlinearity, overfitting and local minima, and have been successfully applied in various fields, including text classification [3], image classification [4], bioinformatics [5] and medical diagnosis [6]. Support vector regression (SVR) is an important application of SVMs, which introduces an ε-insensitive loss function in SVM to adapt to the regression problem [7]. In order to achieve nonlinear regression, SVR uses a kernel function to map the sample set to the feature space. SVR has many advantages in solving small sample, nonlinear and high dimensional pattern recognition, and has been widely applied to practical problems, including traffic velocity prediction [8], conductivity prediction [9], spatial prediction of landslide susceptibility [10], and stock price forecasting [11]. However, for the samples containing heterogeneous information, uneven distribution and irregularity, the traditional SVR using single-kernel mapping is not necessarily suitable for sample processing. Therefore, a lot of work has been applied to multi-kernel learning (MKL) [12], which is a more flexible kernel-based learning method. Using MKL instead of the traditional single-kernel learning can greatly improve the interpretability and generalization performance of the model [13].

MKL is the process of obtaining the weights of the combined kernel function. There are many effective learning methods for solving this problem. For example, Rakotomamonjy *et al.* [14] proposed a valid MKL method to select the kernel functions, in which the kernel functions are

The associate editor coordinating the review of this manuscript and approving it for publication was Jingen Ni.

set to be a linear combination of multiple basic kernel functions. Cao *et al.* [15] proposed multi-kernel feature selection based on the $L_{2,1}$ norm, called $L_{2,1}$MKFS, and an proximal optimization algorithm is designed for efficient learning the model. To solve highly complex issues of convex quadratic programming in SVR, a novel two-phase MKL-SVR based on linear programming (MK-LP-SVR) was proposed by Zhang *et al.* [16], and used for feature sparsification and forecasting.

Moreover, some studies have tried to assign different weights to training samples in SVM or SVR to solve the problem of heteroscedasticity in training samples. Ada-SVR-R, proposed by Gao *et al.* [17], used a so-called classification-type loss to increase and decrease the weights of misclassified samples and correctly classified samples, respectively. Tao *et al.* [18] developed a modification of AdaBoost, which was a self-adaptive cost technique for SVM. Elatter *et al.* [19] combined locally weighted regression (LWR) and SVR (LWSVR) to build a load forecasting model, in which a weighted distance algorithm based on Mahalanobis distance was proposed to optimize the bandwidth of the weighting function. Xu *et al.* [20] proposed a weighted twin SVR, that brought different penalties to the samples according to their different locations. In addition, some algorithms were designed so that the weights of each training sample were added as scaling factors of the slack variable in the objective function of SVR [21]–[23]. However, the above algorithms were only improved in one aspect, and few scholars have considered both the adaptive selection of the kernel function and the updation of the weights of the training samples in the framework of SVR.

Inspired by the existing literature, we propose a new multi-kernel ensemble algorithm based on the L_1 norm and weighted support vector regression (WSVR) with AdaBoost, namely, Ada-L_1MKL-WSVR. First, to adaptively choose the optimal combined kernel function, L_1MKL-SVR is proposed. Moreover, we design an accelerated method to solve the weights of the combined kernel function with the L_1 norm. Then, we introduce FISTA into AdaBoost to correct the weights of the training samples, and a new multi-kernel ensemble algorithm is proposed. In this method, the optimization of the weights of the combined kernel function and the updation of the weights of the training samples are both considered. Finally, the subregressor of each iteration is integrated into a strong robust regressor. There are extensive experiments have been performed to validate the performances of Ada-L_1MKL-WSVR. The numerical results are provided to demonstrate the competitiveness of the algorithm proposed in this paper

The remainder of this article is arranged as follows. In Section 2, we review some pertinent basic results to WSVR and Ada-SVR-R. The details of L_1MKL-SVR and Ada-L_1MKL-WSVR are presented in Section 3. Section 4 discusses our simulations and empirical studies, including dataset descriptions, parameter settings, and a comparative analysis of five different algorithms. Finally, some conclusions are drawn in Section 5.

II. RELATED WORK

A. WEIGHTED SUPPORT VECTOR REGRESSION

In SVR, there is a basic assumption that the samples come from the same distribution, that is, the random error items should have the same variance, independent or uncorrelated. However, it is often not satisfactory if we use the standard SVR to establish the model when there is heteroscedasticity in a regression problem. To solve this problem, Sun *et al.* [28] proposed the so-called WSVR, which introduced the appropriate weights to adjust the role of the training samples in SVR. In what follows, we briefly introduces the basic idea of WSVR. More details can be found in [28].

Let $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_N, y_N)\}$ be the training samples, where $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)$ with $\mathbf{x}_i \in R^m, i = 1, 2, \ldots, N$ and $y = (y_1, y_2, \cdots, y_N)$ with $y_i \in R, i = 1, 2, \ldots, N$ are the input of the training samples and the target values, respectively. The purpose of WSVR is to find a regression function $f(\mathbf{x})$ to precisely estimate y when given an input $\mathbf{x}$. To make $f(\mathbf{x})$ available, the standard WSVR can be transformed into the following convex optimization problem:

$$\min_{\mathbf{w},b,\xi,\xi^*} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\lambda_i\left(\xi_i + \xi_i^*\right)$$
$$\text{s.t.} \begin{cases} \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle + b - y_i \le \varepsilon + \xi_i, & i = 1, \ldots, N \\ y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i)\rangle - b \le \varepsilon + \xi_i^*, & i = 1, \ldots, N \\ \xi_i, \xi_i^* \ge 0, & i = 1, \ldots, N \end{cases} \quad (1)$$

where C is the penalty coefficient, $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_N)$ are the weights of the training samples, $\xi = (\xi_1, \xi_2, \cdots, \xi_N)$ and $\xi^* = (\xi_1^*, \xi_2^*, \cdots, \xi_N^*)$ are the slack variables, b is the intercept term, ε is the fitting error, and $\phi(\cdot)$ is the map function, which maps the training samples space to a Hilbert space $\Re$. It should be pointed out that WSVR reduces to the standard SVR if $\lambda_i = 1$ with $i = 1, \cdots, N$. Particularly, the weight λ_i is set as the reciprocal of the variance of the error term δ_i^2, i.e., $\lambda_i = \frac{1}{\delta_i^2}, i = 1, 2, \ldots, N$ in [28].

The Lagrange dual optimization problem associated with the problem (1) is given by

$$\max -\frac{1}{2}\sum_{i,j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\left\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j)\right\rangle + \sum_{i.j=1}^{N}(\alpha_i - \alpha_i^*)y_i - \varepsilon\sum_{i,j=1}^{N}(\alpha_i + \alpha_i^*)$$
$$\text{s.t.} \begin{cases} \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \\ 0 \le \alpha_i, \alpha_i^* \le \lambda_i C, \quad i = 1, \ldots, N \end{cases} \quad (2)$$

where α^* and α are the Lagrange multipliers.

It is well-known that the regression function can be expressed in the following way

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\alpha^* - \alpha) K(\mathbf{x}, \mathbf{x}_i) + b \tag{3}$$

where $K(\mathbf{x}, \mathbf{x}_i) = \langle \phi(\mathbf{x}), \phi(\mathbf{x}_i) \rangle$ represents the kernel function.

B. ADA-SVR-R

In practical research, the variance of the error term $\sigma_i (i = 1, \ldots, N)$ in WSVR is usually unknown and needs to be determined according to the actual situation. To overcome this difficulty, Gao *et al.* [17] proposed an integrated algorithm based on AdaBoost, namely Ada-SVR-R, which can directly appiled to the regression problem by introducing the classification-type loss. At each iteration of WSVR, SVR receives the training samples and produces a regression function by training. Then the weights of the training samples are updated by calculating regression errors based on the classification-type loss. This process is repeated until $error_t$ is larger than 0.5. Finally, the final regression function $F(\mathbf{x})$, i.e.,

$$F(\mathbf{x}) = \frac{\sum_{t=1}^{T} \alpha_t f_t(\mathbf{x})}{\sum_{t=1}^{T} \alpha_t} \tag{4}$$

is obtained.

The so-called Ada-SVR-R [17] is given below.

Algorithm 1 Ada-SVR-R

Input:

Training samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
Setting the parameters of SVR
Threshold: $\varepsilon > 0$

Output:

Final ensemble regression function: $F(\mathbf{x})$

1: Initialize the weights of the training samples: $w_i^t = 1/N, i = 1, 2, \ldots, N$
2: **for** $t = 1, 2, \ldots,$T **do**
3: Set the distribution of the weights of the training samples as: $\lambda_i^t = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t}, i = 1, 2, \ldots, N$
4: Call SVR, providing it with the distribution λ_i^t, and obtain a regression function $f_t(\mathbf{x})$
5: Calculate the weighted classification-type loss of $f_t(\mathbf{x})$: $error_t = \sum_{i=1}^{n} \lambda_i^t [|y_i - f_t(\mathbf{x}_i)| > \varepsilon]$
6: **if** $error_t > \frac{1}{2}$ **then**
7: Set $T = t - 1$, **break**.
8: **end if**
9: Set base learner's weight: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-error_t}{error_t}\right)$
10: Update the weights of the training samples: $w_i^{t+1} = w_i^t \times \begin{cases} \exp(-\alpha_t), & \text{if } |y_i - f_t(\mathbf{x}_i)| < \varepsilon \\ \exp(+\alpha_t), & \text{if } |y_i - f_t(\mathbf{x}_i)| \geq \varepsilon \end{cases}$
11: **end for**
12: **return** $F(\mathbf{x})$

III. METHODOLOGY

In this section, we fisrt introduce the key idea of L_1MKL-SVR, which is the basis of our algorithm, then we provide the details of L_1MKL-WSVR and Ada-L_1MKL-WSVR, respectively.

A. L_1 MULTI-KERNEL LEARNING SUPPORT VECTOR REGRESSION

MKL [14], [29] is one of the most important research topic in kernel machine learning. MKL selects two or more kernel functions as the optimal kernel function from the set of basic kernel functions, and assigns the weight to each kernel function. The combined kernel function constructed by MKL takes into account the characteristics of each constituent kernel function, which improves the accuracy of the model to a certain extent. Unlike a single kernel function, such as SVR, MKL assumes that the input of the training samples $\mathbf{x}_i (i = 1, \ldots, N)$ can be mapped to S different Hilbert spaces $\Re, \mathbf{x}_i \to \phi_s(\mathbf{x}_i)(s = 1, \ldots, S)$, with S mapping functions, and the purpose of MKL is to learn the optimal combined kernel function, which is used to instead of a single kernel function to obtain better prediction effects.

It is well-known that the weights of the combined kernel function obtained by using the L_1 norm is sparse, and it can reduce redundancy and increase the operation efficiency of model. Therefore, we introduces the L_1 norm of the weights of the combined kernel function into the objective function of SVR, namely L_1MKL-SVR, which can be expressed as the following optimization problem, i.e.,

$$\min_{\mathbf{w}, D, b, \xi, \xi^*} \frac{1}{2}\left(\sum_{s=1}^{S} \|\mathbf{w}_s\|\right)^2 + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) + \gamma \|D\|_1$$

$$\text{s.t.} \begin{cases} \sum_{s=1}^{S} \langle \mathbf{w}_s, \sqrt{d_s}\phi_s(\mathbf{x}_i) \rangle + b - y_i \leq \varepsilon + \xi_i, \ i = 1, \ldots, N \\ y_i - \sum_{s=1}^{S} \langle \mathbf{w}_s, \sqrt{d_s}\phi_s(\mathbf{x}_i) \rangle - b \leq \varepsilon + \xi_i^*, \ i = 1, \ldots, N \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \ldots, N \\ d_s \geq 0, \quad s = 1, \ldots, S \end{cases} \tag{5}$$

where γ is the regularization parameter and $D = (d_1, \ldots, d_S)$ are the weights of the combined kernel function with $d_s (s = 1, \ldots, S)$ being the weight of the kernel function $K_s(\mathbf{x}, \mathbf{x}_i) = \langle \phi_s(\mathbf{x}), \phi_s(\mathbf{x}_i) \rangle$. The optimization problem (5) is nonconvex due to the products of d_s and $\mathbf{w}_s$, and it can be resolved by applying the variable transformation $\mathbf{w}_s' = \sqrt{d_s}\mathbf{w}_s$ as in [14], [30], [31]. This yields the following optimization problem, i.e.,

$$\min_{\mathbf{w}', D, b, \xi, \xi^*} \frac{1}{2} \sum_{s=1}^{S} \frac{\|\mathbf{w}_s'\|^2}{d_s} + C \sum_{i=1}^{N} (\xi_i + \xi_i^*) + \gamma \|D\|_1$$

$$\text{s.t.}\begin{cases}\sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle + b - y_i \le \varepsilon + \xi_i, \ i = 1, \ldots, N \\ y_i - \sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle - b \le \varepsilon + \xi_i^*, \ i = 1, \ldots, N \\ \xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, N \\ d_s \ge 0, \quad s = 1, \ldots, S \end{cases} \tag{6}$$

Similar to the WSVR case, the regression function can be defined by solving the Lagrange dual optimization problem of the optimization problem (6).

B. L_1 MULTI-KERNEL LEARNING WEIGHTED SUPPORT VECTOR REGRESSION WITH AdaBoost

To learn the weights of the training samples and the combined kernel function simultaneously, we introduce the weights λ of the training samples into L_1MKL-SVR, namely L_1MKL-WSVR, which can be expressed as the following optimization problem, i.e.,

$$\min_{\mathbf{w}', D, b, \xi, \xi^*} \frac{1}{2}\sum_{s=1}^{S}\frac{\|\mathbf{w}'_s\|^2}{d_s} + C\sum_{i=1}^{N}\lambda_i\left(\xi_i + \xi_i^*\right) + \gamma\|D\|_1$$
$$\text{s.t.}\begin{cases}\sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle + b - y_i \le \varepsilon + \xi_i, \ i = 1, \ldots, N \\ y_i - \sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle - b \le \varepsilon + \xi_i^*, \ i = 1, \ldots, N \\ \xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, N \\ d_s \ge 0, \quad s = 1, \ldots, S \end{cases} \tag{7}$$

One can easily verify that if $\lambda_i = 1$ $(i = 1, 2, \ldots, N)$, L_1MKL-WSVR degenerates to L_1MKL-SVR. Furthermore, the optimization problem (7) can be regarded as the composite objective optimization problem, i.e.,

$$\min_{D \ge 0} Z(D) = M(D) + \gamma\|D\|_1 \tag{8}$$

where

$$M(D) = \frac{1}{2}\sum_{s=1}^{S}\frac{\|\mathbf{w}'_s\|^2}{\tilde{d}_s} + C\sum_{i=1}^{N}\lambda_i\left(\tilde{\xi}_i + \tilde{\xi}_i^*\right) \tag{9}$$

and $(\tilde{\mathbf{w}'}, \tilde{b}, \tilde{\xi}, \tilde{\xi^*})$ is an optimal solution of the following optimization problem, i.e.,

$$\min_{\mathbf{w}', b, \xi, \xi^*} \frac{1}{2}\sum_{s=1}^{S}\frac{\|\mathbf{w}'_s\|^2}{d_s} + C\sum_{i=1}^{N}\lambda_i\left(\xi_i + \xi_i^*\right)$$
$$\text{s.t.}\begin{cases}\sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle + b - y_i \le \varepsilon + \xi_i, \\ y_i - \sum_{s=1}^{S}\langle \mathbf{w}'_s, \phi_s(\mathbf{x}_i)\rangle - b \le \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, N \\ d_s \ge 0, \quad s = 1, \ldots, S \end{cases} \tag{10}$$

In addition, for the given weights D, $M(D)$ can be directly obtained by solving the classical WSVR in (1), which is given by

$$M(D) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)(\hat{\alpha}_j^* - \hat{\alpha}_j)\sum_{s=1}^{S} d_s K_s(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{N}\left[(\varepsilon - y_i)\hat{\alpha}_i^* + (\varepsilon + y_i)\hat{\alpha}_i\right] \tag{11}$$

It should be noted that the optimization problem (8) has the composite structure, where $M(D)$ is convex and differentiable, while $\|D\|_1$ is a nondifferentiable convex function on the feasible domain. To solve the composite objective optimization problem, the common idea is to use the concept of the proximal gradient proposed by Nesterov [25]–[27]. The quadratic function is used to approximate the objective function, and the proximal gradient method is used to solve the new optimization problem. In this work, FISTA is designed to optimize D [24]. By using the quadratic approximation, we can obtain the proximal operator of the objective function $Z(D)$ at point D, i.e.,

$$Q_L(D, D^{(t-1)}) = M(D^{(t-1)}) + \langle D - D^{(t-1)}, \nabla M(D^{(t-1)})\rangle + \frac{L^{(t-1)}}{2}\|D - D^{(t-1)}\|^2 + \gamma\|D\|_1 \tag{12}$$

where

$$\nabla M(D^{(t-1)}) = \left[\nabla M\left(d_1^{(t-1)}\right), \ldots, \nabla M\left(d_S^{(t-1)}\right)\right] \tag{13}$$

with

$$\nabla M\left(d_s^{(t-1)}\right) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\hat{\alpha}_i^* - \hat{\alpha}_i)(\hat{\alpha}_j^* - \hat{\alpha}_j)K_s(\mathbf{x}_i, \mathbf{x}_j) \tag{14}$$

After ignoring the constant term, we can obtain the unique minimum of (12), i.e.,

$$\begin{aligned} D^{(t)} &= P_L(D^{(t-1)}) \\ &= \arg\min_{D}\left\{\frac{1}{2}\left\|D - U^{t-1}\right\|^2 + \frac{\gamma}{L^{(t-1)}}\|D\|_1\right\} \end{aligned} \tag{15}$$

with

$$\begin{aligned} U^{t-1} &= D^{(t-1)} - \frac{1}{L^{(t-1)}}\nabla M(D^{(t-1)}) \\ &= \left[u_1^{(t-1)}, \ldots, u_s^{(t-1)}, \ldots, u_S^{(t-1)}\right] \end{aligned} \tag{16}$$

where D is the iteratively updated by FISTA, $P_L(D^{(t-1)})$ represents a proximal operator, and $L^{(t-1)}$ is the step size of the internal gradient used to control the convergence rate, which is in the form of a linear search. Meanwhile, to speed up the convergence of the system (15), the proximal operator $P_L(H^{(t+1)})$ is used as the beginning of the current iteration, in which $H^{(t+1)}$ is a linear combination of two previous iterations $D^{(t)}$ and $D^{(t-1)}$, i.e.,

$$H^{(t+1)} = D^{(t)} + \frac{k^{(t)} - 1}{k^{(t+1)}}(D^{(t)} - D^{(t-1)}) \tag{17}$$

where $k^{(t)}$ is an auxiliary sequence, whose iteration formula is given by

$$k^{(t+1)} = \frac{1+\sqrt{1+4(k^{(t)})^2}}{2} \tag{18}$$

Due to the separability of the L_1 norm, i.e., $D = (d_1, \ldots, d_S)$, we can update each weight d_s by solving the following one-dimensional problem, that is

$$d_s^{(t)} = \left(u_s^{(t-1)} - \frac{\gamma}{L^{(t-1)}}\right)_+ \operatorname{sgn}\left(u_s^{(t-1)}\right), \quad s = 1, \ldots, S \tag{19}$$

In addition, a projection operator $\mathbf{P}$ is introduced to assure that each weight $d_s^{(t)}$ is nonnegative ($d_s \geq 0$), i.e.,

$$\mathbf{P}(d_s^{(t)}) = \max\left(0, d_s^{(t)}\right) \tag{20}$$

The updation of the weights D of the combined kernel function by FISTA is given in Algorithm 2. It follows from the optimization problem (8) is a convex problem that its global optimum solution can be obtained. Moreover, Algorithm 2 minimizes the substitution function in each iteration to ensure that the original objective function iteratively decreases, and finally the global optimization of the convergence domain problem is achieved. The theoretical proof [24] that the convergence rate of such an algorithm is guaranteed to be $O(1/t^2)$. By optimizing D, we can learn the relative importance between the different kernel functions and perform parameter estimation at the same time. The final regression function obtained from

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\hat{\alpha}^* - \hat{\alpha}) \sum_{s=1}^{S} \hat{d}_s K_s(\mathbf{x}, \mathbf{x}_i) + b. \tag{21}$$

To simultaneously update the weights of the training samples, we embed Algorithm 2 as a hyper parameter optimization method into Algorithm 1. This yields the so-called Ada-L_1MKL-WSVR, which is a new boosting algorithm for regression. Furthermore, it is also an integrated algorithm composed of several regression functions, which is followed in Algorithm 3.

As Algorithm 3 shows, Ada-L_1MKL-SVR mainly performs two tasks at each iteration, including the adaptive selection of the optimal combined kernel function and the updation of the weights of the training samples. Firstly, MKL-SVR trains a set of the training samples to obtain the corresponding regression function. Secondly, the regression errors are calculated based on the classification-type loss. Thirdly, the weights of each training subset are recalculated according to the regression errors. Next, the weight distribution is used to resample the regression samples to form a new training subset. After that, according to the new training subset, the weights of the combined kernel function D are calculated by using Algorithm 2. Finally, the regression functions obtained from each iteration are combined as the final regression function.

There are two contributions from the boosting iteration in Algorithm 3. The first contribution is to skillfully add the L_1 norm of the weights D to the objective function of WSVR, and an accelerated method based on FISTA is used to optimize the weights D. The second contribution is to embed FISTA into AdaBoost, that is, during each iteration, the weights D and the weights λ are optimized and updated, respectively.

The algorithm finally obtains a regression function, which can be regarded as a separating planes ensemble in the weighted average composed of N optimal separated planes with α_t as the confidence of the t-th optimal separation plane. The final regression function is the results of the weighted votes of the multiple regression functions with a prediction accuracy of more than 50%. Without ignoring the normal samples, the algorithm strengthens the training of the abnormal samples to ensure the robustness, that is, the detection of the abnormal samples.

Algorithm 2 Optimize D Based on FISTA

Input:

Training samples: $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $L^{(0)} = l(l \geq 1)$, $\eta = 2$, $k^{(1)} = 1$, $\lambda = (\lambda_1, \ldots, \lambda_N)$, C, ε, γ and tol

Output:

The weights of the combined kernel function: $\hat{D}$

Initialize $D^{(0)} = (d_1^{(0)}, \ldots, d_S^{(0)}) = (1/S, \ldots, 1/S)$
$H^{(1)} = D^{(0)}$
for $t = 1$ to $\ldots$ **do**
  Calculate $M(H^{(t)})$ by using WSVR in (1) and $\nabla M(D^{(t-1)})$ according to (14)
  Find the smallest nonnegative integers i_t such that $Z\left(p_{\bar{L}^{(t)}}(H^{(t)})\right) \leq Q_{\bar{L}^{(t)}}\left(P_{\bar{L}^{(t)}}(H^{(t)}), H^{(t)}\right)$, where $\bar{L}^{(t)} = \eta^{i_t} L^{(t-1)}$
  Set $L^{(t)} = \eta^{i_t} L^{(t-1)}$
  $D^{(t)} = \mathbf{P}\left(P_{L^{(t)}}(H^{(t)})\right)$
  $k^{(t+1)} = \frac{1+\sqrt{1+4(k^{(t)})^2}}{2}$
  $H^{(t+1)} = D^{(t)} + \frac{k^{(t)}-1}{k^{(t+1)}}(D^{(t)} - D^{(t-1)})$
  if $\max(|D^{(t)} - D^{(t-1)}|) < tol$ **then**
    $\hat{D} = D^{(t)}$, **break**
  end if
end for
return $\hat{D}$

IV. EXPERIMENTAL RESULTS AND ANALYSES

Without causing ambiguity in the context, the prediction model based on Algorithm 3 is still written as Ada-L_1MKL-SVR in this section. In order to test the performance of the proposed Ada-L_1MKL-WSVR, We design two groups of the experiments, and compare with four regression models (SVR [7], EGWO-SVR [33], MKL-SVR [8], Ada-SVR-R [17]). The first group of the experiments consists of eight datasets from UCI machine learning repository [32], and the COVID-19 epidemic dataset from eight countries are used in the second group of the experiments.

Algorithm 3 Ada-L_1MKL-WSVR

Input:

Training samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $L^{(0)} = l(l \geq 1)$, $\eta = 2$, $k^{(1)} = 1$, C, ε, γ and tol

Output:

Final ensemble regression function: $F(\mathbf{x})$

Initialize $D^{(0)} = (d_1^{(0)}, \ldots, d_S^{(0)}) = (1/S, \ldots, 1/S)$

Initialize the weights of the training samples: $w_i^t = 1/N, i = 1, 2, \ldots, N$

$H^{(1)} = D^{(0)}$

for $t = 1$ to $\ldots$ **do**

  Set the distribution of the weights of the training samples as: $\lambda_i^t = \frac{w_i^t}{\sum_{i=1}^{N} w_i^t}, i = 1, 2, \ldots, N$

  Call MKL-SVR, provide it with the distribution λ_i^t, and obtain a regression function $f_t(\mathbf{x})$

  Calculate the weighted classification-type loss of $f_t(\mathbf{x})$: $error_t = \sum_{i=1}^{n} \lambda_i^t [|y_i - f_t(\mathbf{x}_i)| > \varepsilon]$

  if $error_t > \frac{1}{2}$ **then**

    $\alpha_t = 0$

  else

    $\alpha_t = \frac{1}{2} \ln\left(\frac{1-error_t}{error_t}\right)$

  end if

  Update the weights of the training samples: $w_i^{t+1} = w_i^t \times \begin{cases} \exp(-\alpha_t), & \text{if } |y_i - f_t(\mathbf{x}_i)| < \varepsilon \\ \exp(+\alpha_t), & \text{if } |y_i - f_t(\mathbf{x}_i)| \geq \varepsilon \end{cases}$

  Calculate $M(H^{(t)})$ by using WSVR in (1) and $\nabla M(D^{(t-1)})$ according to (14)

  Find the smallest nonnegative integers i_t such that $Z\left(p_{\bar{L}^{(t)}}(H^{(t)})\right) \leq Q_{\bar{L}^{(t)}}\left(P_{\bar{L}^{(t)}}(H^{(t)}), H^{(t)}\right)$, where $\bar{L}^{(t)} = \eta^{i_t} L^{(t-1)}$

  Set $L^{(t)} = \eta^{i_t} L^{(t-1)}$

  $D^{(t)} = \mathbf{P}\left(P_{L^{(t)}}(H^{(t)})\right)$

  $k^{(t+1)} = \frac{1+\sqrt{1+4(k^{(t)})^2}}{2}$

  $H^{(t+1)} = D^{(t)} + \frac{k^{(t)}-1}{k^{(t+1)}}(D^{(t)} - D^{(t-1)})$

  if $\max(|D^{(t)} - D^{(t-1)}|) < tol$ **then**

    $\hat{D} = D^{(t)}$, **break**

  end if

end for

return $F(\mathbf{x}) = \frac{\sum_{t=1}^{T} \alpha_t f_t(\mathbf{x})}{\sum_{t=1}^{T} \alpha_t}$

A. PERFORMANCE CRITERIA

The criteria of mean absolute error (MAE) and root mean square error (RMSE) [34] are employed to validate the effectiveness of the models in this paper.

The representations of MAE and RMSE are defined by

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |F(\mathbf{x_i}) - y(\mathbf{x_i})| \tag{22}$$

TABLE 1. Intervals of the parameters.

Parameters	Interval
C	[0.01,0.1,1,10,100,1000]
σ	[0.001,0.005,0.01,0.05,0.1,0.5,1,5,10,50,100]
d	[1,2,3]
γ	[0.01,0.1,1,5,10,50,100]
tol	0.01
ε	0.1
l	$\max\left(\frac{\max_{1<j<m} \|\mathbf{x}_{ij}\|_1}{5}, 1\right)$ $(i = 1, \ldots, N)$

TABLE 2. UCI dataset statistics.

Dataset	Number of samples	Number of features
Boston-housing	506	13
Pyrim	74	27
Triazines	187	60
Concrete data	1030	8
Auto-mpg	398	7
Forestfires	517	10
Qsar-fish-toxicity	908	6
Wine Quality	4898	11

and

$$RMSE = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (F(\mathbf{x_i}) - y(\mathbf{x_i}))^2} \tag{23}$$

respectively. Here N is the total number of the samples, and $F(\mathbf{x_i})$ and $y(i)$ denote the predicted and real values of the t-th sample, respectively.

It is well-known that MAE is the mean value used to measure the absolute error between the predicted and real values, and RMSE represents the square root of the predicted error, which can measure the dispersion of the predicted error. In each group of the experiments, the smaller the MAE and RMSE, the better the performance of the model is.

B. DATA PREPROCESSING AND PARAMETERS SETTINGS

It is well-known that SVRs produce better models when the data are normalized, all data should be normalized or standardized before the prediction. In this paper, we preprocess the raw data in each group of the experiments by using min-max normalization, i.e.,

$$x_{ij}^* = \frac{x_{ij} - \min_i(x_{ij})}{\max_i(x_{ij}) - \min_i(x_{ij})} \tag{24}$$

where x_{ij} represents the j-th value of the i-th attribute, $\max_i(x_{ij})$ and $\min_i(x_{ij})$ represent the maximum and minimum values of the i-th attribute, respectively.

As is known to all, the reasonable selection of the kernel function and its parameters can improve the prediction ability. The commonly used kernel functions include the Gaussian kernel function and polynomial kernel function [35], i.e.,

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2}\right) \tag{25}$$

TABLE 3. Experimental result on Experiment I.

Dataset	Types	MAE	RMSE	Time(s)
Boston-houing	SVR	2.659±0.615	4.162±0.759	**0.197**
	EGWO-SVR	2.454±0.570	4.088±0.471	29.199
	MKL-SVR	2.584±0.344	4.165±0.475	6.967
	Ada-SVR-R	2.589±0.213	4.078±0.232	0.3619
	L_1MKL-SVR	**2.315±0.140**	3.560±0.238	12.597
	Ada-L_1MKL-SVR	2.326±0.132	**3.543±0.203**	12.251
Pyrim	SVR	0.072±0.010	0.091±0.018	**0.014**
	EGWO-SVR	0.072±0.009	0.093±0.023	0.710
	MKL-SVR	0.071±0.009	0.092±0.022	0.209
	Ada-SVR-R	0.071±0.009	0.088±0.017	0.272
	L_1MKL-SVR	0.070±0.009	0.090±0.019	0.234
	Ada-L_1MKL-SVR	**0.059±0.005**	**0.079±0.019**	0.308
Triazines	SVR	0.104±0.062	0.143±0.010	**0.036**
	EGWO-SVR	0.103±0.008	0.142±0.012	3.060
	MKL-SVR	0.114±0.011	0.153±0.015	0.954
	Ada-SVR-R	0.102±0.054	0.139±0.039	0.152
	L_1MKL-SVR	0.099±0.014	0.134±0.012	1.851
	Ada-L_1MKL-SVR	**0.097±0.012**	**0.130±0.010**	1.993
Concrete Data	SVR	4.390±0.267	6.271±0.456	**0.652**
	EGWO-SVR	4.271±0.244	6.231±0.414	130.839
	MKL-SVR	4.245±0.399	6.225±0.356	20.125
	Ada-SVR-R	4.380±0.270	6.258±0.460	1.165
	L_1MKL-SVR	4.249±0.347	6.439±0.582	83.734
	Ada-L_1MKL-SVR	**4.228±0.278**	**6.204±0.366**	87.996
Auto-mag	SVR	2.664±0.615	3.555±0.759	**0.123**
	EGWO-SVR	2.628±0.570	3.470±0.475	16.023
	MKL-SVR	2.497±0.344	3.362±0.475	3.860
	Ada-SVR-R	2.172±0.213	2.933±0.232	0.204
	L_1MKL-SVR	2.038+0.203	2.851±0.238	11.374
	Ada-L_1MKL-SVR	**1.979±0.141**	**2.770+0.132**	16.054
Forestfires	SVR	0.876±0.033	1.149±0.059	**0.244**
	EGWO-SVR	0.847±0.027	1.083±0.058	20.164
	MKL-SVR	0.857±0.035	1.129±0.126	4.427
	Ada-SVR-R	0.869±0.035	1.096±0.055	0.371
	L_1MKL-SVR	**0.835±0.028**	1.067±0.098	3.198
	Ada-L_1MKL-SVR	0.840±0.026	**1.056±0.076**	2.998
Qsar-fish-toxicity	SVR	0.671±0.057	0.944±0.032	**1.155**
	EGWO-SVR	0.668±0.016	0.934±0.024	63.483
	MKL-SVR	**0.655±0.018**	0.938±0.040	20.675
	Ada-SVR-R	0.656±0.043	**0.913±0.026**	1.683
	L_1MKL-SVR	0.637±0.053	0.903±0.091	43.991
	Ada-L_1MKL-SVR	0.632±0.053	0.891±0.086	52.523
Wine Quality	SVR	0.663±0.019	0.860±0.022	**5.045**
	EGWO-SVR	0.679±0.050	0.893±0.070	149.012
	MKL-SVR	0.645±0.006	0.838±0.014	23.857
	Ada-SVR-R	0.606±0.013	0.782±0.011	8.445
	L_1MKL-SVR	0.581±0.008	**0.754±0.011**	86.614
	Ada-L_1MKL-SVR	**0.580±0.008**	0.753±0.017	23.857

and

$$K(\mathbf{x}, \mathbf{x}_i) = (\langle \mathbf{x}, \mathbf{x}_i \rangle + 1)^d \tag{26}$$

where σ represents the width of the Gaussian kernel function, which controls the complexity of the distribution of the feature subspace, and d represents the order of the polynomial kernel function.

In this paper, the Gaussian kernel function and polynomial kernel function are selected to combine the multi-kernel functions. The multi-kernel function is composed of 13 different basic kernel functions including 10 Gaussian kernel functions and 3 polynomial kernel functions with different parameters. In addition, we use the grid search approach to adjust hyper parameters, and the values of all hyper parameters settings are shown in Table 1.

C. EXPERIMENT I

In this subsection, we test the accuracy of L_1MKL-SVR and Ada-L_1MKL-WSVR based on the first group of the experiments. Each dataset is divided into the training set (60%), the validation set (20%) and the testing set (20%) by using train_test_split() function in Python 3.7.2. The training

TABLE 4. COVID-19 dataset statistics.

Country	Number of observed days	First report	Last report	Cumulative confirmed cases	Cumulative deaths
USA	111	28/01/2020	17/05/2020	1507773	90113
Canada	111	28/01/2020	17/05/2020	77257	5801
Germany	111	28/01/2020	17/05/2020	176369	7958
Italy	109	31/01/2020	17/05/2020	224760	31763
France	87	21/02/2020	17/05/2020	179630	27532
Spain	107	01/02/2020	17/05/2020	276505	27563
South Korea	111	31/01/2020	17/05/2020	11050	262
Iran	89	19/02/2020	17/05/2020	120198	6988

TABLE 5. Experimental result on Experiment II.

Country	Types	MAE	RMSE
USA	NCDTRM	32902.72	48042.41
	INCDTRM	12068.99	16261.69
	Ada-L_1MKL-WSVR	**9273.16**	**11289.08**
Canada	NCDTRM	1160.59	1320.66
	INCDTRM	**1007.98**	**1300.06**
	Ada-L_1MKL-WSVR	1395.578	1915.339
Germany	NCDTRM	2717.74	3374.36
	INCDTRM	1595.88	1782.99
	Ada-L_1MKL-WSVR	**1157.13**	**1336.11**
Italy	NCDTRM	3763.30	4104.03
	INCDTRM	3509.84	3856.71
	Ada-L_1MKL-WSVR	**1598.04**	**1911.01**
France	NCDTRM	9239.81	11827.25
	INCDTRM	15694.11	19165.06
	Ada-L_1MKL-WSVR	**3349.91**	**3949.64**
Spain	NCDTRM	10815.32	11061.34
	INCDTRM	5910.42	6619.12
	Ada-L_1MKL-WSVR	**3639.88**	**4186.15**
South Korea	NCDTRM	48.07	61.45
	INCDTRM	**46.15**	57.95
	Ada-L_1MKL-WSVR	47.01	**55.63**
Iran	NCDTRM	5104.99	6613.54
	INCDTRM	5534.86	7046.04
	Ada-L_1MKL-WSVR	**4823.99**	**6233.85**

set is used to train the models, the validation set is used to adjust hyper parameters, and the testing set is used to detect the generalization ability of models. All the experiments are repeated 10 times to demonstrate the robustness of the model (The parameter ''random_state'' in train_test_split() function is set to integers from 0 to 9 in Python 3.7.2).

The descriptive information of these datasets are presented in Table 2, and the experimental results of Experiment I are showed in Table 3.

As the results demonstrated in Table 3, Ada-L_1MKL-WSVR achieves the best prediction on most datasets against the rest of the models. In particular, the performance of Ada-L_1MKL-WSVR is superior to other models on Pyrim data, where the MAE and RMSE are 0.059 ± 0.005 and 0.079 ± 0.019, which are reduced by 20.38% and 11.28% on average. Taking the Triazines data as an example, Ada-L_1MKL-WSVR has the best regression effect with the MAE and RMSE of 0.097 ± 0.012 and 0.130 ± 0.020, respectively, down 0.49% and 3.64% over the second-best model, i.e., L_1MKL-SVR, while the regression effect of SVR is the worst. In the Boston-housing and Forestfires data, the MAE of Ada-L_1MKL-WSVR is slightly larger than that of L_1MKL-SVR, while the variance of the MAE is less than that of L_1MKL-SVR, and their performances are better than those of SVR and Ada-SVR-R. In the Wine Quality data, one can easily verify that Ada-L_1MKL-WSVR and L_1MKL-SVR obtain the best MAE and RMSE, respectively, and there is little difference between them. Both of them are significantly better than that of other models.

In addition, by analyzing the experimental results of SVR, EGWO-SVR and Ada-SVR-R, we can see that the integrated SVRs is superior to SVR. This is due to the fact that Ada-SVR-R trains many times by changing the weighted distribution of the training samples, so as to achieve the effect of multi-kernel learning and increase the integral performance. In general, the two proposed models have smaller MAEs and RMSEs than those of other models, which shows that L_1MKL-SVR can adaptively select the optimal combined kernel function and its parameter. Due to the advantages of AdaBoost, Ada-L_1MKL-WSVR can effectively adjust the weights of the training samples and the integrate multiple weak regressions. In the face of the abnormal dataset, it can obtain more robust regression performance than that of L_1MKL-SVR.

In terms of time complexity, the most efficiency and the least efficiency are SVR and EGWO-SVR. Compared with SVR, EGWO-SVR has higher prediction accuracy, but it needs to constantly update iteration, so the time complexity is high. In addition, our model has higher time complexity than the majority of the comparative models. The time complexity is negatively correlated with the regularization parameter γ. The smaller the regularization parameter, the higher the time complexity is. On the contrary, the larger the regularization parameter, the lower the time complexity is. This is a shortcoming of our model.

D. EXPERIMENT II

In this subsection, we use the COVID-19 epidemic dataset of eight countries to further verify the performance of our model. Table 4 lists the cumulative confirmed cases and deaths in these eight countries, as well as the first and last reporting periods [39].

In this subsection, we use Ada-L_1MKL-WSVR instead of SVR. Furthermore, we take the data before April 17, 2020 as the training samples to predict and analyze the existing cases of the eight countries from April 28, 2020 to May 17, 2020. The NCDTRM [38] and INCDTGM [39] are used as the comparative models in this experiment.

As shown in Table 5, in addition to Canada, the proposed model has the different degrees of the improvement compared with NCDTRM and INCDTRM. Especially in Spain, Italy and France, the prediction results of our model is significantly improved compared with the best-second model, where the MAE and RMSE are 3639.88, 1598.04, 3349.91 and 4186.15, 1191.01, 3949.64, respectively. The MAEs and RMSEs of our model in eight countries are reduced by 31.19% and 29.98% on average, respectively. This shows the effectiveness of our model in introducing multi-kernel learning and ensemble algorithm. On the whole, Ada-L_1MKL-WSVR can effectively improve the regression accuracies in prediction of the COVID-19 epidemic than the rest of the models.

V. CONCLUSION

In this paper, a new multi-kernel learning ensemble algorithm, i.e., Ada-L_1MKL-WSVR, is presented based on the L_1 norm and WSVR with AdaBoost. The L_1 norm of the weights of the combined kernel function is added to the objective function of WSVR, which can effectively select the optimal combined kernel function and its related parameters. Furthermore, we embed FISTA into AdaBoost, rather than a simple combination or the single model. In each iteration, the algorithm simultaneously optimizes and updates the weights D of the combined kernel function and the weights λ of the training samples. Finally, the multiple weakness regressors are integrated into a robust regressor. The numerical experiments are desired to compare the effectiveness and reliability of the algorithm in this paper. However, our algorithm has the higher time complexity than that of some other existing algorithms. In addition, the hyper parameters of our algorithm need to be preseted, and forecasting efficiency change with the hyper parameters.

For future works, it is intended (*i*) to choose the appropriate initial hyper parameters for the better prediction results, and (*ii*) to look in some advanced optimization algorithm to improve the computational efficiency of Ada-L_1MKL-WSVR.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous referees for their useful comments and suggestions, which helped to improve the presentation of this article.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York, NY, USA: Springer, 2000.

[2] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Mach. Learn.*, vol. 46, nos. 1–3, pp. 131–159, 2002.

[3] V. Kumar and B. Subba, "A TfidfVectorizer and SVM based sentiment analysis framework for text data corpus," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.

[4] A. Fred Agarap, "An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification," 2017, *arXiv:1712.03541*.

[5] X. Lin, C. Li, Y. Zhang, B. Su, M. Fan, and H. Wei, "Selecting feature subsets based on SVM-RFE and the overlapping ratio with applications in bioinformatics," *Molecules*, vol. 23, no. 1, p. 52, 2017.

[6] A. U. Haq, J. P. Li, M. H. Memon, J. Khan, A. Malik, T. Ahmad, A. Ali, S. Nazir, I. Ahad, and M. Shahid, "Feature selection based on L1-norm support vector machine and effective recognition system for Parkinson's disease using voice recordings," *IEEE Access*, vol. 7, pp. 37718–37734, 2019.

[7] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[8] J. Xiao, C. Wei, and Y. Liu, "Speed estimation of traffic flow using multiple kernel support vector regression," *Phys. A, Stat. Mech. Appl.*, vol. 509, pp. 989–997, 2018.

[9] A. Karimipour, S. A. Bagherzadeh, A. Taghipour, A. Abdollahi, and M. R. Safaei, "A novel nonlinear regression model of SVR as a substitute for ANN to predict conductivity of MWCNT-CuO/water hybrid nanofluid based on empirical data," *Phys. A, Stat. Mech. Appl.*, vol. 521, pp. 89–97, 2019.

[10] M. Panahi, A. Gayen, H. R. Pourghasemi, F. Rezaie, and S. Lee, "Spatial prediction of landslide susceptibility using hybrid support vector regression (SVR) and the adaptive neuro-fuzzy inference system (ANFIS) with various Metaheuristic algorithms," *Sci. Total Environ.*, vol. 741, Nov. 2020, Art. no. 139937.

[11] Y. Guo, S. Han, C. Shen, Y. Li, X. Yin, and Y. Bai, "An adaptive SVR for high-frequency stock price forecasting," *IEEE Access*, vol. 6, pp. 11397–11404, 2018.

[12] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, 2004.

[13] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, pp. 41–48.

[14] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *J. Mach. Learn. Res.*, vol. 9, pp. 2491–2521, 2008.

[15] P. Cao, X. Liu, J. Zhang, D. Zhao, M. Huang, and O. Zaiane, "$\ell_{2,1}$ norm regularized multi-kernel based joint nonlinear feature selection and over-sampling for imbalanced data classification," *Neurocomputing*, vol. 234, pp. 38–57, 2017.

[16] Z. Zhang, G. Gao, Y. Tian, and J. Yue, "Two-phase multi-kernel LP-SVR for feature sparsification and forecasting," *Neurocomputing*, vol. 214, pp. 594–606, 2016.

[17] L. Gao, P. Kou, F. Gao, and X. Guan, "AdaBoost regression algorithm based on classification-type loss," in *Proc. 8th World Congr. Intell. Control Automat.*, 2010, pp. 682–687.

[18] X. Tao, Q. Li, W. Guo, C. Ren, C. Li, R. Liu, and J. Zou, "Self-adaptive cost weights-based support vector machine cost-sensitive ensemble for imbalanced data classification," *Inf. Sci.*, vol. 487, pp. 31–56, 2019.

[19] E. E. Elattar, J. Goulermas, and Q. H. Wu, "Electric load forecasting based on locally weighted support vector regression," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 40, no. 4, pp. 438–447, 2010.

[20] Y. Xu and L. Wang, "A weighted twin support vector regression," *Knowl.-Based Syst.*, vol. 33, pp. 92–101, 2012.

[21] Y. Xu, X. Lv, and W. Xi, "A weighted multi-output support vector regression and its application," *J. Comput. Inf. Syst.*, vol. 8, no. 9, pp. 3807–3814, 2012.

[22] M. Tang and H. Zhang, "An effective method for weighted support vector regression based on sample simplification," in *Proc. Int. Colloq. Comput., Commun., Control, Manage. (ISECS)*, Aug. 2009, pp. 33–37.

[23] D. Prayogo, M.-Y. Cheng, Y.-W. Wu, and D.-H. Tran, "Combining machine learning models via adaptive ensemble weighting for prediction of shear capacity of reinforced-concrete deep beams," *Eng. Comput.*, vol. 36, no. 3, pp. 1135–1153, 2019.

[24] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[25] Y. Nesterov, "A method of solving a convex programming problem with convergence rate O(1/K^2)," *Sov. Math. Dokl.*, vol. 27, no. 2, pp. 372–376, 1983.

[26] Y. Nesterov, *Introductory Lectures Convex Optimization: A Basic Course*. New York, NY, USA: Springer, 2004.
[27] Y. Nesterov, "Gradient methods for minimizing composite functions," *Math. Program.*, vol. 140, no. 1, pp. 125–161, 2013.
[28] D. Sun, J. Wu, and Z. Hou, "Weighting support vector regression algorithm," (in Chinese), *Comput. Sci.*, vol. 30, no. 11, pp. 38–39, 2003.
[29] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *J. Mach. Learn. Res.*, vol. 12, pp. 2211–2268, 2011.
[30] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proc. 24th Int. Conf. Mach. Learn. (ICML)*, 2007, pp. 1191–1198.
[31] M. Kloft, U. Brefeld, K. Müller, A. Zien, and S. Sonnenburg, "Efficient and accurate Lp-norm multiple kernel learning," in *Proc. Conf. Neural Inf. Process. Syst.*, 2009, pp. 997–1005.
[32] P. Murphy and D. Aha. (2021). UCI Repository of Machine Learning Databases [Machine-Readable Data Repository]. University of California, Irvine, Department of Information and Computer Science. [Online]. Available: https://archive.ics.uci.edu/ml/datasets
[33] M. Liu, K. Luo, J. Zhang, and S. Chen, "A stock selection algorithm hybridizing grey wolf optimizer and support vector regression," *Expert Syst. Appl.*, vol. 179, Oct. 2021, Art. no. 115078.
[34] L. Xu, B. Luo, Y. Tang, and X. Ma, "An efficient multiple kernel learning in reproducing kernel Hilbert spaces (RKHS)," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 13, no. 2, 2015, Art. no. 1550008.
[35] M. Hussain, S. K. Wajid, A. Elzaart, and M. Berbar, "A comparison of SVM kernel functions for breast cancer detection," in *Proc. 8th Int. Conf. Comput. Graph., Imag. Visualizat.*, 2011, pp. 145–150.
[36] N. E. Huang and F. Qiao, "A data driven time-dependent transmission rate for tracking an epidemic: A case study of 2019-nCoV," *Sci. Bull.*, vol. 65, no. 6, pp. 425–427, 2020.
[37] Y. Hu, K. Jin, L. Yang, X. Wang, Y. Zhang, Y. Dai, and Z. Yang, "A dynamic growth rate model and its application in global COVID-19 epidemic analysis," (in Chinese), *Acta Math. Appl. Sin.*, vol. 43, no. 2, pp. 452–467, 2020.
[38] X. Xie, K. Luo, Z. Yin, and G. Wang, "Nonlinear combinational dynamic transmission rate model and its application in global COVID-19 epidemic prediction and analysis," *Mathematics*, vol. 9, no. 18, p. 2307, 2021.
[39] X. Xie, K. Luo, Y. Zhang, J. Jin, H. Lin, Z. Yin, and G. Wang, "Nonlinear combinational dynamic transmission rate model and COVID-19 epidemic analysis and prediction in China," (in Chinese), *Oper. Res. Trans.*, vol. 25, no. 1, pp. 17–30, 2021.

XIAOJIN XIE received the B.S. degree in quality control engineering from Shanghai Dianji University, in 2018. He is currently pursuing the M.Sc. degree in business statistics with the School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, China. His research interests include machine learning and data mining.

KANGYANG LUO received the B.S. degree in mathematics and applied mathematics from Shijiazhuang Tiedao University, in 2017, and the M.S. degree in business statistics from the School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, in 2020. He is currently pursuing the Ph.D. degree with the School of Data Science and Engineering, East China Normal University, China. His research interests include federated learning, high-dimensional data analysis, machine learning, and data mining.

GUOQIANG WANG (Member, IEEE) received the B.S. degree in mathematics and applied mathematics from Shandong Normal University, in 2002, and the M.S. and Ph.D. degrees in operations research from Shanghai University, China, in 2005 and 2009, respectively. From 2012 to 2013, he was a Visiting Research Associate with the Department of Mathematics and Statistics, Curtin University, Australia. He is currently a Full Professor with the School of Mathematics, Physics and Statistics, Shanghai University of Engineering Science, China. His research interests include high-dimensional statistical inference, statistical optimization, machine learning and data mining, optimization theory, methods, and applications.