


ITC 2/50 Information Technology and Control Vol. 50 / No. 2 / 2021 pp. 319-331 DOI 10.5755/j01.itc.50.2.27752	Least Squares Support Vector Machine Regression Based on Sparse Samples and Mixture Kernel Learning	
	Received 2020/09/28	Accepted after revision 2021/05/03
	 http://dx.doi.org/10.5755/j01.itc.50.2.27752	

HOW TO CITE: Ma, W., Liu, H. (2021). Least Squares Support Vector Machine Regression Based on Sparse Samples and Mixture Kernel Learning. *Information Technology and Control*, 50(2), 319-331. <https://doi.org/10.5755/j01.itc.50.2.27752>

Least Squares Support Vector Machine Regression Based on Sparse Samples and Mixture Kernel Learning

Wenlu Ma, Han Liu

School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China;
e-mails: liuhan@xaut.edu.cn, mwlntybbrq@163.com

Corresponding author: liuhan@xaut.edu.cn

Least squares support vector machine (LSSVM) is a machine learning algorithm based on statistical theory. Its advantages include robustness and calculation simplicity, and it has good performance in the data processing of small samples. The LSSVM model lacks sparsity and is unable to handle large-scale data problem, this article proposes an LSSVM method based on mixture kernel learning and sparse samples. This algorithm reduces the initial training set to a sub-dataset using a sparse selection strategy. It converts the single kernel function in the LSSVM model into a mixed kernel function and optimizes its parameters. The reduced sub-dataset is used for training LSSVM. Finally, a group of datasets in the UCI Machine Learning Repository were used to verify the effectiveness of the proposed algorithm, which is applied to real-world power load data to achieve better fitting and improve the prediction accuracy.

KEYWORDS: LSSVM, mean-shift, mixture kernel, IABC.

1. Introduction

Support Vector Machine (SVM) [25] is one of the most important algorithms in the field of machine learning. SVM detection method has been widely employed on account of the advantages of small sample learning, good generalization ability and high accuracy. At present, it is under the background of large samples in the era of big data. Due to its super large amount of calculation in large samples, the attention of SVM has declined, but it is still a commonly used machine learning algorithm [9, 18, 26]. The applications of the SVM have been significantly increased in the last years in multiple sectors as a successful machine learning approach in modeling the relationship between the input and the output in regression problems [8, 30, 31].

The main advantages of the SVM algorithm are: (1) It is very effective to solve the classification problem and regression problem of high dimensional features, and it still has a good effect when the feature dimension is greater than the number of samples. (2) Only a part of the support vectors is used to make hyperplane decisions without relying on all data. (3) A large number of kernel functions can be very flexible to solve various nonlinear classification regression problems. (4) When the sample size is not massive data, the classification accuracy is high and the generalization ability is strong.

The main disadvantages of the SVM algorithm are: (1) SVM is not suitable for use when the sample size is very large and the kernel function mapping dimension is very high. (2) There is no universal standard for the choice of kernel function for nonlinear problems, and it is difficult to choose a suitable kernel function.

Least squares support vector machine (LSSVM) [24] is an improved form of SVM, the difference being that SVM is a quadratic programming problem with linear inequality constraints. The calculation process is complex and requires a large computational space. LSSVM is a loss function that uses the sum of error squares as a training set, which is equivalent to converting the quadratic planning problem into a linear equation solution, which makes the problem much easier to solve. Although LSSVM inherits the advantages of SVM, it is with this conversion step that the final decision function is correlated with all the

samples, so that LSSVM loses its understanding of the sparseness of the feature. When processing large-scale data, with the increase of data sample size and the diversity of structure, computer memory can easily overflow, which affects the prediction accuracy and generalisation ability of the algorithm. As a result of this situation, LSSVM appears to be unable to cope with large sample problems.

To solve the problem of the sparsity of solutions, many researchers have put forward new and improved algorithms, which mainly solve the sparseness problem from the standpoint of the training sample set. Suykens et al. proposed a pruning algorithm based on the size of the support value after LSSVM model training [23]. This algorithm deletes the sample points corresponding to the smaller support value and retains the sample points with the larger support value to decide on the model. The disadvantage of this method is that the model training is performed twice, the solution process is complicated and time-consuming. Subsequently, an LSSVM algorithm with a fixed size sample set and a corresponding improved algorithm [3, 4], and the method for combining LSSVM with other machine learning algorithms, have appeared [10, 13]. The core concept of these algorithms is to compress large datasets into smaller sub-datasets, and then train them in the LSSVM model [15, 16]. Since the reduced sub-sample set carries almost all the important information of the original sample, it can be used as a training sample for the LSSVM model [11, 12, 14, 20]. [11] and [12] solved the problem of fault diagnosis, [20] and [14] proposed a deep structure of LSSVM to solve classification problems. To increase the accuracy, a variety of deep network models based on SVM have been proposed in [6, 7, 19, 21, 29] and successfully applied to various classification and regression prediction scenarios. A support vector machine classification algorithm based on depth kernel theory was proposed that can be applied to large-scale data sets [21, 29]. A deep learning model based on support vector machines and a probability output network has also been proposed [7, 19]. To have a good representation of the data distribution, one can take an algorithm with subset selection, or a random subset as a simpler scheme in [22].

However, these algorithms cannot guarantee sufficiently large reduction datasets, run time is long, and the prediction accuracy is not high. The predic-

tion accuracy of LSSVM is also affected by the kernel function and parameters, which makes the selection of kernel function a key consideration. So far, there is no definite theory or method to support how to determine the kernel function and parameters. Improper parameter selection can lead to the problem of overfitting or underfitting the regression model.

To solve the above problems, it is necessary to further research on LSSVM. An LSSVM regression algorithm based on sparse samples and hybrid kernel learning is proposed in this paper. For large-scale data sets, an effective sparse selection strategy is adopted to reduce the large-scale data set to a smaller subset, and the optimization algorithm is used to optimize the mixture kernel function to solve the LSSVM sparsity problem.

The remainder of this article is structured as follows. Section 2 and Section 3 provides a brief review on LSSVM and sparse subset selection strategy, respectively. The proposed method based on (Improved Artificial Bee Colony-Mixture Kernel LSSVM) IABC-MixKLSSVM with sparsity IABC-MixKLSSVM (SIABC-MixKLSSVM) is presented in Section 4. In Section 5, the experimental results of the related algorithms are given, and the results are analysed and summarized.

2. Description of LSSVM

Given a training data set $\{(x_k, y_k)\}_{k=1}^N$, $x_k \in R$ and $y_k \in R$ denote the input and output of LSSVM, respectively, and m is the dimension of the input. Then the LSSVM model can be described as solving constrained optimization problems:

$$\begin{aligned} \min J(w, e) &= \frac{1}{2} w^T w + \gamma \sum_{k=1}^N e_k^2 \\ \text{s.t. } y_k &= w^T \phi(x_k) + b + e_k, \quad k = 1, \dots, N. \end{aligned} \tag{1}$$

In this, w and b are the weights and bias to be adjusted, respectively. e_k denotes the error for the k^{th} sample, γ is a positive regularisation parameter and $\phi(\cdot)$ denotes a nonlinear function mapping x_k into a high-dimensional feature space. Construct the Lagrangian function of the optimization problem by formula (1):

$$\begin{aligned} J(w, b, e, \alpha) &= J(w, e) \\ &- \sum_{k=1}^N \alpha_k \{w^T \phi(x_k) + b + e_k - y_k\} \end{aligned} \tag{2}$$

where $\alpha_k \in R$ is the Lagrange multiplier corresponding to the k^{th} sample, and the corresponding sample points are called Support Vectors (SVs). According to the KKT conditions, the equivalent equations are obtained by eliminating vector w and e :

$$\begin{bmatrix} 0 & 1^T \\ 1 & \Omega + \gamma^{-1}I \end{bmatrix} \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{3}$$

where $\Omega = K(x_k, x_j) = \phi(x_k)^T \phi(x_j)$, $I = (1, \dots, 1)^T$, $y = \{y_1, \dots, y_N\}$ and $\alpha = \{\alpha_1, \dots, \alpha_N\}^T$ is the kernel function matrix.

The final LSSVM model can be written as

$$y(x) = \sum_{k=1}^N \alpha_k K(x, x_k) + b. \tag{4}$$

Kernel types and parameters affect the prediction accuracy of the LSSVM algorithm training model, and the selection of kernel functions plays an important role in processing learning tasks. In the LSSVM model, a kernel function is used to map the input data to a high-dimensional space. Because each kernel function has its characteristics and has different effects on the performance of the LSSVM. The two types of kernel functions—Gaussian and polynomial—were combined to create a Mixture Kernel (MixK) functions in the LSSVM model. MixK does not need to change the original mapping space to ensure the effectiveness of its functions [17]

$$\begin{aligned} K_{\text{mix}}(x, x_k) &= \delta K_{\text{poly}}(x, x_k) \\ &+ (1 - \delta) K_{\text{Gauss}}(x, x_k). \end{aligned} \tag{5}$$

According to the Mercer condition, if K_1 and K_2 are kernel on $X \times X$, $X \subseteq R^n$, $\delta \in [0, 1]$, then

$$K(x, x_k) = K_1(x, x_k) + K_2(x, x_k). \tag{6}$$

K is still a kernel function. Therefore, $K_{\text{mix}}(x, x_k)$ satisfies the kernel function property of Mercer's condition.

Therefore, the prediction output of the LSSVM model is:

$$y(x) = \text{sgn} \left(\sum_{k=1}^N \alpha_k K_{\text{mix}}(x, x_k) + b \right). \quad (7)$$

3. Sparse Subset Selection

Clustering is a machine learning technique that groups some data points. As one of the most well-known clustering algorithms, K-Means has the advantages of fast running speed and wide application, but the biggest shortcoming is that it needs to preset the number of clusters and initial points. When the training sample set is large, the K-Means algorithm needs to sort each iteration when calculating the median vector, and the clustering effect is affected. In 1975, Fukunaga et al. proposed a mean-shift algorithm, which is a sub-parameter method based on density gradient rise [5]. It is widely used in target tracking, data clustering, classification [27] and other scenarios. The basic idea is to randomly select an initialisation centre point, calculate the average value of the distance vector from all points to the centre point within a certain range of the centre point, and then calculate the average value to obtain an offset mean. Then move the centre point to the offset mean position, and through this repeated movement, the centre point can be gradually approached to the best position. This idea is similar to the gradient drop method, which can reach the local or global optimal solution of the gradient by constantly moving towards the gradient descent [1, 2]. The geometric explanation is as follows: if the sample point x_i obeys the distribution of a probability density function $f(x)$, because the gradient of the non-zero probability density function points to the direction where the probability density increases the most, the sample points in the S_h region fall more along the direction of the probability density gradient, the mean-shift vector $M_h(x)$ should point to the direction of the probability density gradient [28]. In other words, the mean shift algorithm is essentially a gradient-based optimisation algorithm.

Given a point x_i in d dimensional space, then the basic form of a mean shift vector is defined as:

$$M_h = \frac{1}{k} \sum_{x_i \in S_h} (x_i - x), \quad (8)$$

where k indicates that k points fall into the S_k area. S_k is a high-dimensional sphere with radius h , a set of y points satisfying the following relationship:

$$S_h(x) = \{y : (y - x)^T (t - x) \leq h^2\}. \quad (9)$$

With the increase of distance, the smaller the effect is. Thus, there are the following improvements:

$$M_h(x) = \frac{\sum_{i=1}^n G\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i)(x_i - x)}{\sum_{i=1}^n G\left(\left\|\frac{x_i - x}{h}\right\|^2\right) w(x_i)} \quad (10)$$

$$m_h(x) = \frac{\sum_{i=1}^n G(\cdot) w(x_i) x_i}{G(\cdot) w(x_i)}. \quad (11)$$

Combined formula (10) and (11), the Mean-shift's modified mean shift formula is:

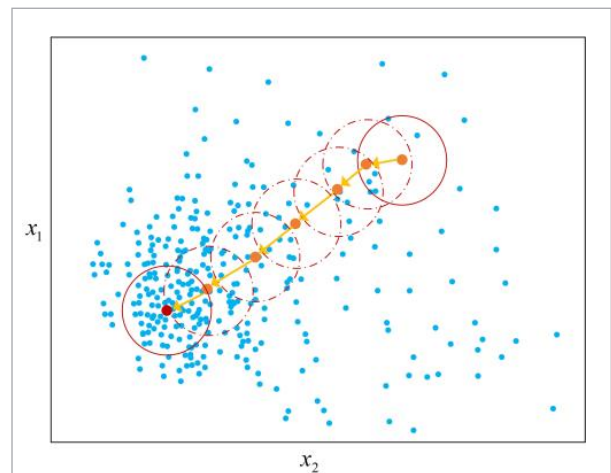
$$M_h = m_h(x) + x, \quad (12)$$

where $G(\cdot) = G\left(\left\|\frac{x_i - x}{h}\right\|^2\right)$, $w(x_i)$ denote weight.

The Mean-shift clustering movement process is shown in Figure 1.

Figure 1

The mean-shift clustering movement process



Because LSSVM lacks sparsity and is not suitable for large-scale datasets, this article uses the mean-shift clustering method to obtain sparse subsets, which is beneficial to LSSVM model training and prediction. The process is shown in Algorithm 1:

Algorithm 1. Mean-shift algorithm process

Step 1 Select an initial center c randomly in a given data set, h is the radius of S_h , the threshold of the $\|shift\|$ is σ .

Step 2 Calculate c to each element in the set M , and add these vectors to get the vector $shift$.

Step 3 Update the center point $c = c + shift$, the moving distance is $\|shift\|$.

Step 4 Repeat Steps 2,3,4 until the shift converges to a σ , obtain c at this time.

Step 5 Calculate the distance d between c and center C of the last iteration,

If $d < h/2$
merge
else

generate a new cluster point.

Step 6 Repeat 1, 2, 3, 4, 5 until all points are marked as accessed.

4. SIABC-MixKLSSVM

To solve the problem that the LSSVM model is not suitable for large-scale datasets, this article uses the sparse strategy to reduce the dataset, which decreases the computational cost and complexity. The parameters of the LSSVM model with a mixture kernel are optimized by IABC to improve the accuracy of regression prediction. Figure 2 shows the schematic diagram of SIABC-MixKLSSVM, and Algorithm 2 provides the detailed process of SIABC-MixKLSSVM.

Algorithm 2. SIABC-MixKLSSVM

Step 1 Preprocess the original data.

Step 2 Initialize the whole parameters of the proposed method

Step 3 Selection subset by Algorithm 1, Obtain the final training datasets.

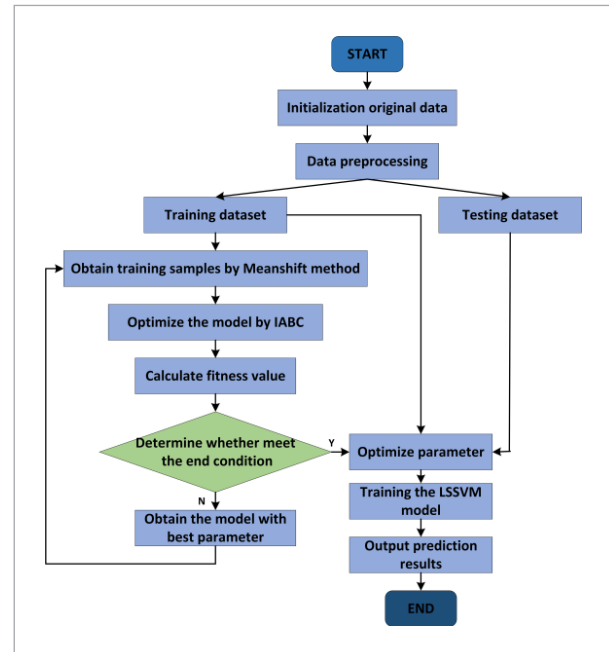
Step 4 Utilize IABC to select appropriate kernel functions and optimize the corresponding parameters on the final datasets.

Step 5 Obtain the model with the best parameters.

Step 6 Test the model.

Figure 2

Flow chart of SIABC-MixKLSSVM algorithm



5. Experiments

To test the performance of the proposed algorithm, we used the dataset in UCI Machine Learning Repository to test various algorithms and analyse the results. All experiments are conducted on an Intel Core i7-3770 CPU @3.20GHz processor with 4GB RAM in a MATLAB 2018a environment. To avoid randomness in the experimental results, each data and model must be run 10 times. All given input datasets are normalized to zero mean and unit variance. The kernel function and optimization parameter setting and value range used in the algorithm are shown in Table 2.

Table 2

Parameters setting of experimental datasets

Parameter	SN	δ	σ_{Gauss}	d_{poly}	γ
Value	30	[0,1]	[1,10]	[1,10]	[1,100]

In addition, when the mean shift method is used to sparse samples, the spherical radius h of S_h need to be set manually, and its value varies with the size of the data set, $\sigma=10^{-5} * h$.

5.1. Test Data and Evaluation Indicators

The data for this experiment comes from 14 datasets in the UCI repository: Energy Efficiency (Heating), Energy Efficiency (Cooling), Concrete Compressive Strength, Airfoil Self-Noise, Red Wine Quality, White Wine Quality, Bias correction of numerical prediction model temperature forecast (Bias Minimum temperature), Bias correction of numerical prediction model temperature forecast (Bias Maximum temperature), Electrical Grid Stability Simulated (EGSSD), Condition Based Maintenance of Naval Propulsion Plants (CBMNPP Compressor), Condition based Maintenance of Naval Propulsion Plants (CBMNPP Turbine), Bike sharing, Superconductivity, Gas Turbine CO and NOx Emission (Gas). Details are shown in Table 1.

To quantitatively evaluate the effectiveness of the proposed algorithm, five quantitative criteria are provided—root mean square error (RMSE), mean absolute error (MAE), standard error (STDE), Mean Absolute Percentage Error (MAPE) and TIME. The values in bold are the best results in the comparisons.

5.2. Experimental Results and Analysis

To verify the effectiveness of the mean-shift clustering algorithm proposed in this article, and to sparse the dataset. We randomly generate 450 points in a two-dimensional space to form a visual dataset, as shown in Figure 3. The blue circles represent all the points in the dataset, and the pink asterisks represent the updated points after each iteration. The pink asterisks form the trajectory of the mean-shift method in finding the extreme points. According to the description of Algorithm 1, Figure 3 shows the process of finding cluster points by the mean shift method. Randomly select a center point in the data set, take the spherical area with a radius h as the initial set M , and move along the direction of increasing initial point density. This process is repeated until the moving distance $shift < \sigma$ ($\sigma = 10^{-3} * h$, $h = 0.75$), and then stop and record the center at this time. The distance d between the center at the time and the center of the previous iteration are calculated. If $d < d/2$, the two centers will be merged, otherwise, the center point at this time is a new cluster point. Repeat the

Table 1

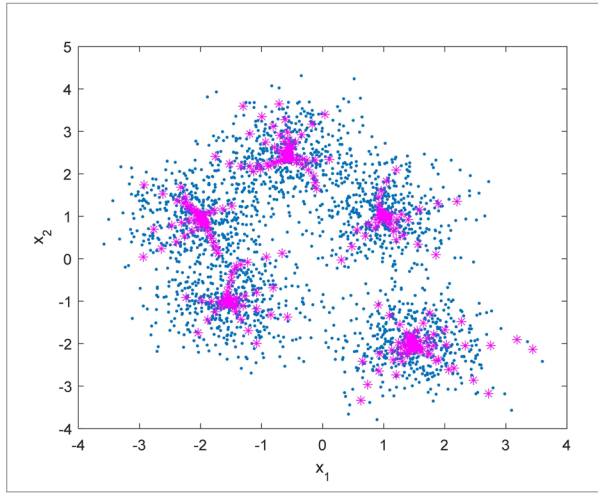
Data used in experiments

Datasets	Data types	Number of Attributes	Number of Instances
Energy Efficiency(Heating)	Multivariate	8	768
Energy Efficiency(Cooling)	Multivariate	8	768
Concrete Compressive Strength	Multivariate	9	1030
Airfoil Self-Noise	Multivariate	6	1503
Red Wine Quality	Multivariate	12	1599
White Wine Quality	Multivariate	12	4898
Bias Minimum temperature	Multivariate	25	7750
Bias Maximun temperature	Multivariate	25	7750
EGSSD	Multivariate	14	10000
CBMNPP Compressor	Multivariate	16	11934
CBMNPP Tutbine	Multivariate	16	11934
Bike sharing	Multivariate	16	17389
Superconductivtiy	Multivariate	81	21263
Gas	Multivariate	11	36733

above process and finally obtain five clusters as shown in Figure 3. Figure 4 shows the final clustering result obtained by the mean-shift method. Five colors represent five clusters, and the mean-shift method is continuously updated and iterated, and finally five cluster centers are obtained, with black asterisk marked.

Figure 3

Motion trajectory of clustering by the mean-shift method

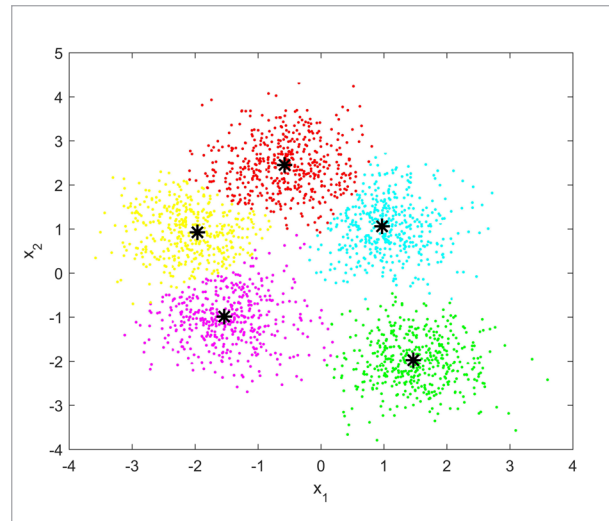


A Combined Cycle Power Plant (CCPP) is composed of gas turbines, steam turbines and heat recovery steam generators. In a CCPP, the electricity is generated by gas and steam turbines, which are combined in one cycle, and is transferred from one turbine to another. The dataset contains 9568 data points collected from a CCPP over six years (2006-2011) when the power plant was set to work with a full load. Features consist of hourly average ambient variables: Temperature (T), Ambient Pressure (AP), Relative Humidity (RH) and Exhaust Vacuum (EV) to predict the net hourly electrical energy output (EP) of the plant.

Figure 5 shows the comparison between the predicted value and actual value of SIABC-MixKLSSVM and several other methods in the CCPP dataset. It includes four comparison algorithms: Sparse LSSVM (S-LSSVM), Sparse Mixture Kernel LSSVM (S-Mix-KLSSVM), Sparse IABC LSSVM (SIABC-LSSVM) and IABC-LSSVM. Among them, the curve fitting in (a) and (b) is poor, the curve fitting in (c) is better than that in (a) and (b), and the curve fitting in (d) is better than (c). The fit of the fitted curve in (e) is the best of all methods. This shows that the SIABC-Mix-

Figure 4

Clustering results of the mean-shift method



KLSSVM method can achieve effective predictability and high predictive ability for large-scale datasets.

Table 3 shows the comparison results of different algorithms based on the LSSVM model and three popular algorithms in the training set and the testing set. In the training set, the MAE and STDE of SIABC-MixKLSSVM are smaller, and the RMSE and MAPE of S-LSSVM are smaller. The overall effect of the latter algorithm is better. In the test set, in addition to MAPE, the evaluation index of SIABC-Mix-KLSSVM is smaller and the performance is better. In summary, whether on a training set or a prediction set, the prediction effect of SIABC-Mix-KLSSVM is the best of all the algorithms. The performance of the algorithm based on the LSSVM model is better than the other three algorithms, especially compared with the algorithm with a deep structure, the result of the SIABC-MixKLSSVM algorithm is better than the BP and ELM algorithm, which shows the effectiveness and feasibility of the algorithm proposed in this paper.

Table 4 shows the test results of five evaluation indicators in five algorithms for data sets of different sizes in the UCI database. In table 4, bold indicates that the smaller the value, the better the prediction effect (that is, the optimal effect). It can be seen that compared with the five given algorithms, the values of the five evaluation indices in the S-LSSVM are higher, indicating that the performance is the worst in all datasets. By comparing S-LSSVM and SIABC-MixKLSSVM,

Figure 5

The prediction results of five methods on the CCPP data set

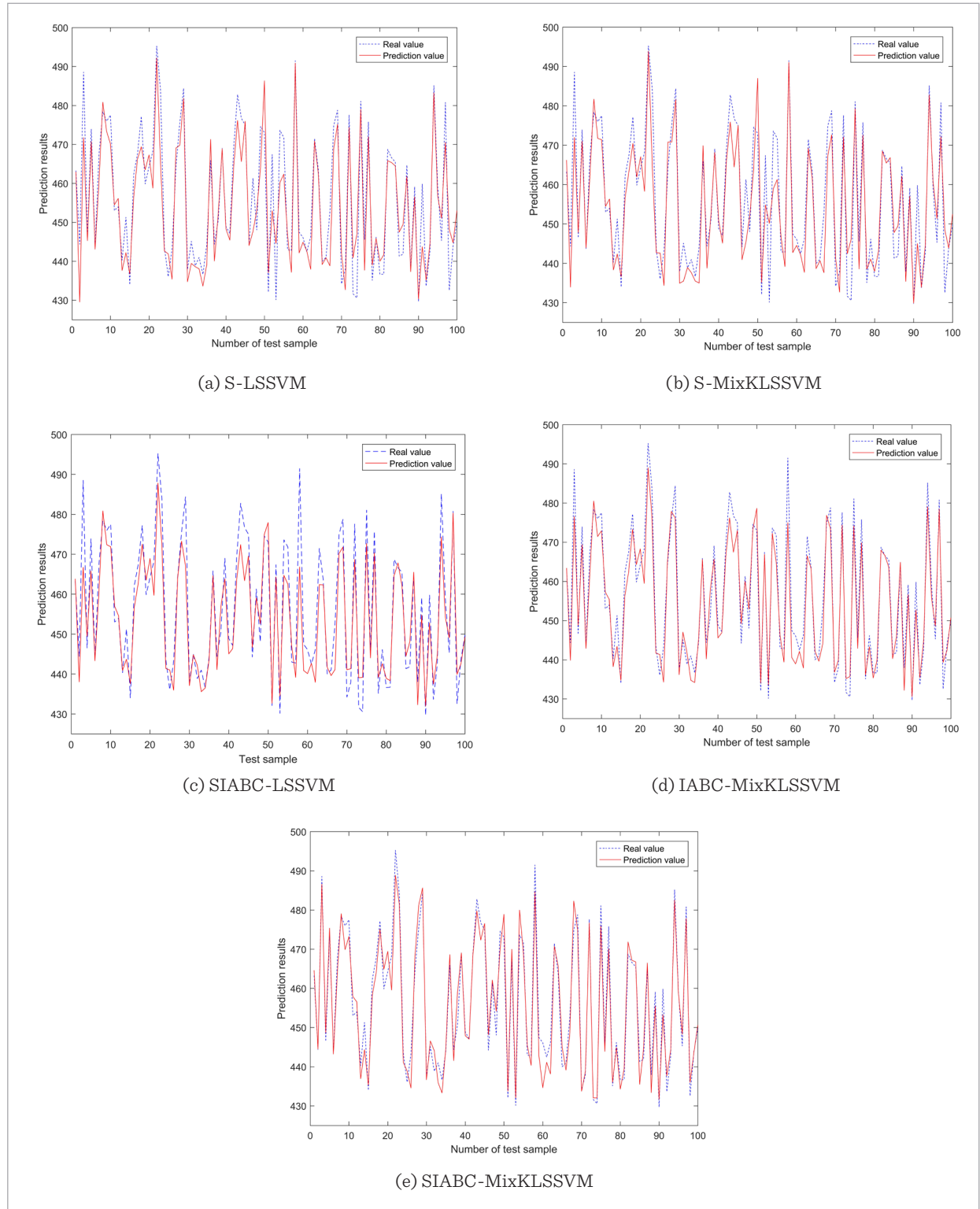


Table 3

Performance comparisons of other algorithms on CCPP data set

Datasets	Indices	S-LSSVM	RVM	BP	ELM	SIABC-MixKLSSVM
Training set	RMSE	7.70E-01	5.22E+00	7.92E+00	3.32E+01	1.60E+00
	MAE	1.13E-01	4.07E+00	1.18E+01	5.62E+01	2.63E-01
	STDE	2.06E+00	3.27E+00	2.62E+00	8.47E+00	4.71E+00
	MAPE	2.00E-03	9.00E-03	2.60E-02	1.23E-01	6.00E-04
Testing set	RMSE	6.14E+00	7.22E+00	1.71E+01	1.69E+01	4.50E+00
	MAE	4.60E+00	5.50E+00	1.47E+01	1.47E+01	3.59E+00
	STDE	4.08E+00	4.67E+00	8.86E+00	8.37E+00	2.71E+00
	MAPE	1.01E-02	1.21E-01	3.21E-01	3.21E-02	7.90E-02
	TIME	3.96E-02	1.40E+00	2.19E+00	2.80E-02	7.91E-02

Table 4

Performance comparisons of other algorithms on UCI data set

Datasets	Indices	S-LSSVM	RVM	BP	ELM	SIABC-MixKLSSVM
Energy Efficiency (Heating)	RMSE	4.75E+00	5.37E+00	8.91E+00	8.55E+00	4.66E+00
	MAE	2.52E+00	4.99E+00	7.88E+00	7.44E+00	3.47E+00
	STDE	4.05E+00	2.00E+00	4.17E+00	4.23E-00	3.11E+00
	MAPE	2.50E-01	3.91E-01	6.68E-01	6.16E-01	2.50E-01
	TIME	4.05E-02	3.00E-03	4.17E+00	3.25E-02	4.75E-02
Energy Efficiency (Cooling)	RMSE	6.65E+00	3.63E+00	9.36E+00	5.62E+00	4.98E+00
	MAE	4.96E+00	3.16E+00	8.05E+00	4.61E+00	3.75E+00
	STDE	4.44E+00	1.80E+00	4.81E+00	3.23E+00	3.30E+00
	MAPE	3.33E-01	1.75E-01	4.61E-01	2.40E-01	2.53E-01
	TIME	3.96E-02	2.90E-03	1.81E+00	4.36E-02	4.46E-02
Concrete Compressive Strength	RMSE	1.08E+01	1.40E+01	1.20E+01	1.27E+01	7.61E+00
	MAE	8.10E+00	1.11E+01	8.90E+00	9.81E+00	5.79E+00
	STDE	7.29E+00	8.52E+00	8.12+00	8.16E+00	4.96E+00
	MAPE	2.34E-01	3.22E-01	3.53E-01	3.07E-01	1.50E-01
	TIME	1.41E-01	5.60E-03	1.82E+00	2.83E-02	7.20E-02
Airfoil Self-Noise	RMSE	4.17E+00	6.65E+00	7.34E+00	6.94E+00	2.75E+00
	MAE	2.92E+00	5.30E+00	6.40E+00	5.21E+00	1.97E+00
	STDE	3.00E+00	4.03E+00	3.60E+00	4.60E+00	1.93E+00
	MAPE	2.47E-02	4.53E-02	5.22E-02	4.49E-02	1.66E-02
	TIME	1.95E-01	3.50E-03	2.31E+00	3.91E-02	2.19E-01
Red Wine Quality	RMSE	7.54E-01	6.48E-01	6.68E-01	7.43E-01	6.43E-01
	MAE	6.60E-01	5.26E-01	5.69E-01	6.48E-01	5.17E-01
	STDE	3.67E-01	3.91E-01	3.52E-01	3.66E-01	3.82E-01
	MAPE	1.26E-01	9.72E-02	1.09E-01	1.28E-01	9.66E-02
	TIME	1.26E-01	8.20E-03	1.91E+00	2.81E-02	1.45E-01

Table 4 (continued)

Datasets	Indices	S-LSSVM	RVM	BP	ELM	SIABC-MixKLSSVM
White Wine Quality	RMSE	9.02E-01	7.66E-01	1.05E+00	8.20E-01	8.11E-01
	MAE	6.99 E-01	6.27E-01	8.57E-01	5.93E-01	6.17E-01
	STDE	5.70E-01	4.40E-01	6.18E-01	5.69E-01	5.26E-01
	MAPE	1.24E-01	1.07E-02	1.36E-01	1.02E-02	1.06E-01
	TIME	1.32E+00	2.98E-02	1.94E+00	2.84E-02	1.39E+00
Bias Minimum temperature	RMSE	2.01E+00	1.94E+00	2.04E+00	1.90E+00	2.01E+00
	MAE	1.60E+00	1.54E+00	1.65E+00	1.48E+00	1.69E+00
	STDE	1.22E+00	1.19E+00	1.20E+00	1.18E+00	1.09E+00
	MAPE	6.57E-02	7.46E-02	7.35E-02	6.25E-02	6.95E-02
	TIME	8.69E-02	4.96E-01	2.09E+00	1.41E-02	9.29E-02
Bias Maximum temperature	RMSE	2.73E+00	2.20E+00	2.82E+00	2.60E+00	2.53E+00
	MAE	2.27E+00	1.80E+00	2.37E+00	2.12E+00	2.16E+00
	STDE	1.51E+00	1.28E+00	1.53E+00	1.51E+00	1.31E+00
	MAPE	7.52E-02	6.38E-02	7.79E-02	7.20E-02	7.35E-02
	TIME	2.97E-02	1.70E-02	2.15E+00	2.93E-02	1.68E-02
EGSSD	RMSE	3.30E-03	1.06E-02	1.39E-03	6.06E-02	4.70E-3
	MAE	2.80E-03	8.20E-03	1.54E-03	7.80E-02	3.40E-03
	STDE	3.30E-03	6.70E-03	6.30E-03	2.13E-02	1.80E-03
	MAPE	7.52E-01	3.15E-00	7.34E-00	1.10E-01	5.57E-02
	TIME	5.66E-00	5.93E+01	3.34E-00	1.79E-01	6.15E-00
CBMNPP Compressor	RMSE	2.24E-02	2.48E-02	2.14E-01	2.57E-02	2.19E-02
	MAE	2.14E-02	2.43E-02	2.03E-01	2.54E-02	2.13E-02
	STDE	6.90E-03	4.90E-03	6.95E-02	3.70E-03	4.99E-03
	MAPE	2.23E-02	2.54E-02	2.12E-01	2.66E-02	2.24E-01
	TIME	3.26E-02	7.93E-01	2.04E+00	3.76E-02	3.67E-02
CBMNPP Turbine	RMSE	7.65E-03	7.30E-03	9.96E-02	7.44E-03	7.20E-03
	MAE	6.54E-03	6.30E-03	7.67E-02	6.44E-03	6.10E-03
	STDE	3.90E-03	3.70E-03	6.36E-02	3.73E-02	3.99E-03
	MAPE	6.60E-03	6.40E-03	7.77E-02	6.52E-03	6.20E-03
	TIME	1.77E-01	7.16E-01	2.17E+00	3.93E-02	3.85E-02
Bike-sharing	RMSE	1.64E-00	7.51E+01	1.59E+01	3.04E+02	1.48E-00
	MAE	3.38E-01	5.26E+01	1.61E+01	3.80E+02	1.41E-01
	STDE	9.33E-00	5.36E+01	7.74E-00	1.18E+02	2.76E-00
	MAPE	1.69E-03	6.79E-01	1.15E+00	1.80E+01	1.40E-03
	TIME	4.24E-02	1.65E-00	1.20E+01	3.25E-02	2.43E-02
Super	RMSE	1.48E-00	1.74E+01	1.28E-00	5.87E+01	7.56E-01
	MAE	3.38E-01	1.28E+01	1.18E-00	5.70E+01	1.87E-01
	STDE	2.76E-00	1.17E+01	6.72E-01	2.90E+01	1.36E-00
	MAPE	3.56E-01	2.59E+01	8.90E-03	3.08E+01	5.09E-01
	TIME	2.43E-00	1.15E+01	1.20E+01	8.44E-00	1.48E-00
Gas	RMSE	1.98E-01	1.14E+01	1.30E-00	1.73E+01	1.46E-01
	MAE	1.33E-02	8.58E-00	1.00E-00	1.51E+01	1.18E-02
	STDE	1.36E-00	7.56E-00	8.32E-01	9.80E-00	7.98E-01
	MAPE	1.00E-04	6.67E-02	7.40E-03	1.17E-01	1.00E-04
	TIME	6.45E-02	3.28E+01	2.59E-00	1.76E-01	6.23E-02

it is not difficult to find that sample sparseness is performed in both algorithms. Although it is a bit longer in terms of TIME, SIABC-MixKLSSVM predicts better results. This shows that optimizing the kernel function of the sparse LSSVM model helps to improve the prediction results. The RVM algorithm performs better in the Energy Efficiency (Cooling), and Bias Maximum temperature dataset. In other data sets, the evaluation indicators of SIABC-MixKLSSVM are smaller than RVM, and the performance is better than RVM. This shows that when the amount of data is very small, RVM has the advantage of being more sparse than SVM. As the amount of data increases, the accuracy of RVM drops significantly, which is unsuitable.

Compared with BP and ELM algorithms, except that ELM performs better in White Wine Quality and Bias Minimum temperature datasets, SIABC-MixKLSSVM has the best evaluation indicators in other data sets. Especially when the amount of data is more than ten thousand levels, the sparse strategy is used to reduce the data set, which greatly shortens the running time of the algorithm, optimizes the mixed kernel function parameters of the LSSVM model, and improves the prediction accuracy.

It can be seen from the above analysis that the method proposed in this paper is effective and feasible. This shows that the algorithm in this paper can not only sparse samples, but also improve the prediction ability of the algorithm by optimizing the parameters of the mixed kernel function, and it has better competitiveness. If the feature dimension is much larger than the number of samples, the prediction accuracy of this algorithm is not high.

References

1. Cheng, Y. Mean Shift, Mode Seeking, and Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1995, 17(8), 790-799. <https://doi.org/10.1109/34.400568>
2. Comaniciu, D., Meer, P. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 2002, 24(5), 603-619. <https://doi.org/10.1109/34.1000236>
3. De Brabanter, K., De Brabanter, J., Suykens, J. A. K., De Moor, B. Optimized Fixed-Size Kernel Models for Large Data Sets. *Computational Statistics & Data Analysis*, 2010, 54(6), 1484-1504. <https://doi.org/10.1016/j.csda.2010.01.024>
4. Espinoza, M., Suykens, J. A. K., De Moor, B. Fixed-Size Least Squares Support Vector Machines: A Large Scale Application in Electrical Load Forecasting. *Computational Management Science*, 2006, 3(2), 113-129. <https://doi.org/10.1007/s10287-005-0003-7>
5. Fukunaga, K., Hostetler, L. The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. *IEEE Transactions on Information Theory*, 1975, 21(1), 32-40. <https://doi.org/10.1109/TIT.1975.1055330>
6. Kim, S., Chol, Y., Lee, M. Deep Learning with Support Vector Data Description. *Neurocomputing*, 2015, 165, 111-117. <https://doi.org/10.1016/j.neucom.2014.09.086>
7. Kim, S., Yu, Z., Kil, R. M., Lee, M. Deep Learning of Support Vector Machines with Class Probability Output Networks. *Neural Networks*, 2015, 64, 19-28. <https://doi.org/10.1016/j.neunet.2014.09.007>

6. Conclusions

LSSVM is an improved version of the SVM algorithm, but it lacks the sparsity of SVM, its single kernel function leads to low generalisation ability and accuracy in the case of a large dataset. In response to this situation, the authors used a sparsity strategy to reduce the initial samples in a subset, to improve the sparsity of the kernel function, and to solve the poor sparsity problem of LSSVM in the case of a large dataset. The single kernel function in the LSSVM model was changed to a mixed kernel function and the IABC algorithm was used to optimize the parameters, which improves the prediction accuracy. In the standard UCI dataset, the experimental results and analysis show that the sparse selection strategy can effectively solve the problem that the LSSVM model is not suitable for large-scale datasets and the SIABC-MixKLSSVM proposed in this article is effective. At the same time, the algorithm was applied to the real-world power load data. The SIABC-MixKLSSVM also achieves a better fitting effect, which shows that the algorithm has higher forecasting accuracy. The future work is due to the idea of deep learning, combining the traditional SVM algorithm and deep structure to form a multi-layer LSSVM model to solve specific problems such as time series forecasting and bearing fault diagnosis.

Acknowledgement

This study was supported by the National Natural Science Foundation of China (No.61973248).

8. Kouziokas, G. N. A New W-SVM Kernel Combining PSO-Neural Network Transformed Vector and Bayesian Optimized SVM in GDP Forecasting. *Engineering Applications of Artificial Intelligence*, 2020, 92, 103650. <https://doi.org/10.1016/j.engappai.2020.103650>
9. Kouziokas, G. N. SVM Kernel Based on Particle Swarm Optimized Vector and Bayesian Optimized SVM in Atmospheric Particulate Matter Forecasting. *Applied Soft Computing*, 2020, 93, 106410. <https://doi.org/10.1016/j.asoc.2020.106410>
10. Li, B. Y., Wang, Q. W., Hu, J. L. A Fast SVM Training Method for Very Large Datasets. *Proceedings of International Joint Conference on Neural Networks*, Atlanta, Georgia, USA, June 14-19, 2009, 1784-1789. <https://doi.org/10.1109/IJCNN.2009.5178618>
11. Li, K., Zhang, R., Li, F., Su, L., Wang, H., Cheng, P. A New Rotation Machinery Fault Diagnosis Method Based on Deep Structure and Sparse Least Squares Support Vector Machine. *IEEE Access*, 2019, 7, 26571-26580. <https://doi.org/10.1109/ACCESS.2019.2901363>
12. Li, X., Yang, Y., Pan, H. Y., Cheng, J., Cheng, J. A Novel Deep Stacking Least Squares Support Vector Machine for Rolling Bearing Fault Diagnosis. *Computers in Industry*, 2019, 110, 36-47. <https://doi.org/10.1016/j.com-pind.2019.05.005>
13. Liu, B., Xiang, H. An Approximate Linear Solver in Least Square Support Vector Machine Using Randomized Singular Value Decomposition. *WuHan University Journal of Natural Sciences*, 2015, 20(4), 283-290. <https://doi.org/10.1007/s11859-015-1094-9>
14. Ma, W. L., Liu, H. Classification Method Based on the Deep Structure and Least Squares Support Vector Machine. *Electronics Letters*, 2020, 56(11), 538-541. <https://doi.org/10.1049/el.2019.3776>
15. Mall, R., Suykens, J. A. K. Sparse Reductions for Fixed-Size Least Squares Support Vector Machines on Large Scale Data. *Advances in Knowledge Discovery and Data Mining*, Springer, Berlin-Heidelberg, 2013, 161-173. https://doi.org/10.1007/978-3-642-37453-1_14
16. Mall, R., Suykens, J. A. K. Very Sparse LSSVM Reductions for Large-Scale Data. *IEEE Transactions on Neural Network and Learning Systems*, 2015, 26(5), 1086-1097. <https://doi.org/10.1109/TNNLS.2014.2333879>
17. Mehmet, G., Ethem, A. Regularizing Multiple Kernels Learning Using Response Surface Methodology. *Pattern Recognition*, 2011, 44(1), 159-171. <https://doi.org/10.1016/j.patcog.2010.07.008>
18. Panda, A. K., Rapur, J. S., Tiwari, R. Prediction of Flow Blockages and Impending Cavitation in Centrifugal Pumps Using Support Vector Machine (SVM) Algorithms Based on Vibration Measurements. *Measurement*, 2018, 130, 44-56. <https://doi.org/10.1016/j.measurement.2018.07.092>
19. Park, W. J., Kil, R. M. Pattern Classification with Class Probability Output Network. *IEEE Transactions on Neural Networks*, 2009, 20(10), 1659-1673. <https://doi.org/10.1109/TNN.2009.2029103>
20. Qi, Z., Wang, B., Tian, Y. J., Zhang, P. When Ensemble Learning Meets Deep Learning: A New Deep Support Vector Machine for Classification. *Knowledge-Based Systems*, 2016, 107(Sep.1), 54-60. <https://doi.org/10.1016/j.knosys.2016.05.055>
21. Rebai, I., BenAyed, Y., Mahdi, W. Deep Kernel-SVM Network. *International Joint Conference on Neural Networks*, (IJCNN), Vancouver, BC, Canada, July 24-29, 2016, 1995-1960. <https://doi.org/10.1109/IJCNN.2016.7727439>
22. Suykens, J. A. K. Deep Restricted Kernel Machines Using Conjugate Feature Duality. *Neural Computation*, 2017, 29(8), 2123-2163. https://doi.org/10.1162/neco_a_00984
23. Suykens, J. A. K., Lukas, L., Vandewalle, J. Sparse Approximation Using Least Squares Support Vector Machine Classifiers. *2000 IEEE International Symposium on Circuits and Systems*, (ISCAS), Geneva, Switzerland, May 28-31, 2000, 757-760. <https://doi.org/10.1109/IS-CAS.2000.856439>
24. Suykens, J. A. K., Vandewalle, J. Least Squares Support Vector Machine Classifiers. *Neural Processing Letters*, 1999, 9(3), 293-300. <https://doi.org/10.1023/A:1018628609742>
25. Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer, New York, 1995. <https://doi.org/10.1007/978-1-4757-2440-0>
26. Wang, C., Han, F., Zhang, Y., Lu, J. An SAE-Based Resampling SVM Ensemble Learning Paradigm for Pipeline Leakage Detection. *Neurocomputing*, 2020, 403, 237-246. <https://doi.org/10.1016/j.neucom.2020.04.105>
27. Wang, X., Liu, H., Ma, W. L. Sparse Least Squares Support Vector Machines Based on Meanshift Clustering Method. *IFAC-PapersOnLine*, 2018, 51(18), 292-296. <https://doi.org/10.1016/j.ifacol.2018.09.315>
28. Yamauchi, H., Lee, S., Lee, Y., Ohtake, Y., Belyaev, E., Seidel, H.-P. Feature Sensitive Mesh Segmentation

- with Mean Shift. International Conference on Shape Modeling and Applications 2005, (SMI' 05), Piscataway, MA, USA June 13-17, 2005, 236-243. <https://doi.org/10.1109/SMI.2005.21>
29. Zareapoor, M., Shamsolmoali, P., Jain, D. K., Haoxiang, W., Jie, Y. Kernelized Support Vector Machine with Deep Learning: An Efficient Approach for Extreme Multiclass Dataset. *Pattern Recognition Letters*, 2018, 115, 4-13. <https://doi.org/10.1016/j.patrec.2017.09.018>
<https://doi.org/10.1016/j.patrec.2017.09.018>
30. Zeng, N., Qiu, H., Wang, Z., Liu, W., Zhang, H., Li, Y. A New Switching-Delayed-PSO-Based Optimized SVM Algorithm for Diagnosis of Alzheimer's Disease. *Neurocomputing*, 2018, 320(DEC.3), 195-202. <https://doi.org/10.1016/j.neucom.2018.09.001>
31. Zhang, X. Y., Li, C. S., Wang, X., Wu, H. A Novel Fault Diagnosis Procedure Based on Improved Symplectic Geometry Mode Decomposition and Optimized SVM. *Measurement*, 2021, 173, 108644. <https://doi.org/10.1016/j.measurement.2020.108644>



This article is an Open Access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 (CC BY 4.0) License (<http://creativecommons.org/licenses/by/4.0/>).