Hindawi
Scientific Programming
Volume 2022, Article ID 4872230, 19 pages
https://doi.org/10.1155/2022/4872230

Research Article

An ISVM Algorithm Based on High-Dimensional Distance and Forgetting Characteristics

Wenhao Xie ,[1] Jinfeng Li,[2] Juanni Li,[1] and Xiaoyan Wang[1]

[1] *School of Science, Xi'an Shiyou University, Xi'an 710065, China*
[2] *School of Computer Science and Technology, Xidian University, Xi'an 710126, China*

Correspondence should be addressed to Wenhao Xie; xwhaoxwhao@163.com

Received 21 March 2022; Revised 14 August 2022; Accepted 26 October 2022; Published 11 November 2022

Academic Editor: Jiangbo Qian

Copyright © 2022 Wenhao Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the face of the batch, dynamic access data, or the flow of data that continuous changes over time, the traditional support vector machine algorithm cannot dynamically adjust the previous classification model. To overcome this shortcoming, the incremental support vector machine (ISVM) algorithm is proposed. However, many incremental support vector algorithms still have shortcomings such as low efficiency, memory limitation, and poor generalization. This paper puts forward the new ISVM algorithm, HDFC-ISVM* algorithm, based on the high-dimensional distance and forgetting characteristics. This paper firstly proposes the original HDFC-ISVM algorithm that first learns the distribution characteristics of the samples according to the distance between the samples and the normative hyperplane. Then, it introduces the forgetting factor. In the incremental learning process, the classifier gradually accumulates the spatial distribution knowledge of samples, eliminates the samples that have no contributions to the classifier, and selectively forgets some useless samples according to the forgetting factor, which overcomes the shortcomings such as low efficiency and poor accuracy of some algorithms. But, the original HDFC-ISVM algorithm is sensitive to parameters, and different settings of the parameters have a great impact on the final classification accuracy of the algorithm. Therefore, on the basis of the original algorithm, an improved algorithm HDFC-ISVM* based on the adjustments to the initialization strategy and updating rules of the forgetting factor is proposed. The initialization strategy and updating rules of the forgetting factor are adjusted to adapt datasets with different distributions in this improved algorithm. The rationality of the improved strategy about the forgetting factor is discussed theoretically. At the same time, the proposed algorithm has better classification accuracy, classification efficiency, and better generalization ability than other algorithms, which is verified by experiments.

1. Introduction

Support vector machine (SVM) is a classic machine learning method. It is proposed by Cortes and Vapnik in 1995 according to VC dimension theory of the statistical theory and the structural risk minimization principle (SRM) [1]. SVM can maximally increase the predictive ability of the learning model, and the obtained classification model by learning has high prediction accuracy for the independent testing samples even if it is built on a small training set. Also, SVM is a convex quadratic programming problem, so it can get the extreme value of solution in the global scope to obtain the optimal solution [2–4]. SVM uses a large interval factor to control the learning process. In dealing with the classification of high-dimensional data and the classification of limited samples, it avoids the defects that neural networks and other methods are easy to fall into local maxima or over fitting.

The nonlinear classification model based on SVM can be summarized as the following convex semipositive definite programming problem:

$$\begin{aligned} &\min \frac{1}{2}w^2 + C\sum_{i=1}^{l}\xi_i \\ &\text{s.t.}\, y_i\left(w\cdot\varphi\left(x_i\right)+b\right) - 1 + \xi_i \geq 0 \\ &\xi_i \geq 0, i = 1, \ldots, l. \end{aligned} \tag{1}$$

At this time, the necessary and sufficient condition of the optimal solution can be obtained for this optimization problem is that the corresponding Karush–Kuhn–Tucker (KKT) conditions are held [5].

SVM was originally designed to solve the problem of binary classification of balanced data obtained in batch. With the development of machine learning research, SVM has also expanded from the initial binary classification problem and regression problem to other machine learning topics, such as feature selection, semisupervision, top order learning, ordered regression, outlier detection, and multi-perspective learning [6]. At the same time, the extended algorithms based on SVM are also applied to more complex data classification or regression. For example, for effectively reducing the impact of noise in the dataset, solving the noise sensitivity and instability of resampling, realizing the high-precision classification of imbalanced data, and the efficient classification of dynamically obtained data or stream data, such as the research of its extended algorithms has always been one of the research directions in the field of machine learning. These studies are of great significance for support vector machines and their variants [7–10]. In these new topics, the models evolved from SVM inherit most of the original characteristics, such as interval theory, kernel techniques, and structural risk minimization and also inherit the defects of the original SVM model.

In the face of the batch, dynamic access data or the flow of data that continuously changes over time, the traditional support vector machine algorithm cannot dynamically adjust the previous classification model. To overcome this shortcoming, the incremental support vector machine (ISVM) algorithm is proposed. In recent years, the incremental learning based on SVM, the ISVM algorithm has attracted a lot of researchers' attention [11–14]. In every incremental learning process of the ISVM, how to effectively retain the historical information, selectively discard and forget the useless training data, and save the storage space while maintaining the classification accuracy is the key of the ISVM classification algorithm.

Scholars at home and abroad have carried out a lot of research work on ISVM algorithms based on sample preselection strategy. For example, Xiao et al. proposed a new incremental learning algorithm—α-SVM [15]. Wang analyzed that the samples near the classification boundary were easy to become support vectors, and then selected the nonsupport vectors near the classification boundary into the incremental update and proposed a redundant ISVM learning algorithm [16]. Yao et al. proposed a fast ISVM learning method based on local sensitive hashing in order to improve the classification accuracy of large-scale high-dimensional data. This method firstly used the local sensitive hash to quickly find similar data, and then selected the samples that may become SVs in the increment on the basis of the SVM algorithm, and then used these samples together with the existing SVs as the basis for subsequent training [17]. Tang combined the strict incremental process of the classical ISVM algorithm with the idea of passive-aggressive online learning to effectively solve the problem of how to better select the new SVs in the online process of the classical ISVM algorithm [11]. Zhang et al. introduced RCMDE as the feature extraction method and proposed an improved ISVM fault classifier based on the whale optimization algorithm (WOA) to diagnose and predict bearing faults [18]. The above incremental models are all based on sample preselection strategies. In addition, many scholars and experts have proposed ISVM learning algorithms based on KKT conditions and the Lagrange multiplier methods [19–21].

As can be seen from the abovementioned research, in the incremental learning process, with the addition of new samples, how to select new support vector sets so that useful information will not be discarded while retaining the original training results has become an important content in the construction of the ISVM learning model.

2. Description of the Original HDFC-ISVM Algorithm

Previously, many classical ISVM learning algorithms have been proposed, including Simple_ISVM [22], KKT_ISVM [23, 24], CSV_ISVM [14], GGKKT_ISVM [25], CD_ISVM [26], and other ISVM algorithms mentioned above, these algorithms provide the different selection methods of incremental learning training samples from different perspectives. However, the ability of the classifier to gradually accumulate the spatial distribution knowledge of samples is still not fully developed, so the accuracy and efficiency can be further improved. In order to further learn the distribution characteristics of samples, this new ISVM algorithm called "HDFC-ISVM" algorithm based on the high-dimensional distance and forgetting characteristics is proposed in this paper. It can fully train the ability of the classifier to accumulate the knowledge of the spatial distribution of samples. The flow of this algorithm is shown in Figure 1.

2.1. The Distance from Every Sample to the Optimal Hyperplane in High-Dimensional Euclidean Space.

The training of the SVM classification hyperplane is only related to the support vectors, and the support vectors are the ones that fall on the normative hyperplane $wx + b = \pm 1$.In n-dimensional Euclidean space, let the mapping function be $\varphi(x)$. If the projection point x' of point x to the optimal hyperplane Π: $w \cdot \varphi(x) + b = 0$, then it satisfies the following formula:

$$w \cdot \varphi(x') + b = 0, \tag{2}$$

where $w = (w_1, w_2, \ldots, w_n), b = (b_1, b_2, \ldots, b_n), x = (x_1, x_2, \ldots, x_n)^T, x' = (x_1', x_2', \ldots, x_n')^T$ and SVs are the set of support vectors.

As can be seen from Figure 2, the vector $\overrightarrow{x'x}$ is parallel to the normal vector $\vec{w}$ of the hyperplane Π, which satisfies the following formula:

$$\left\| w \cdot x'x \right\| = \|w\| \cdot \|x'x\| = \|w\| \cdot d, \tag{3}$$

where d is the distance from point x to the hyperplane Π. In addition, the following formula holds:

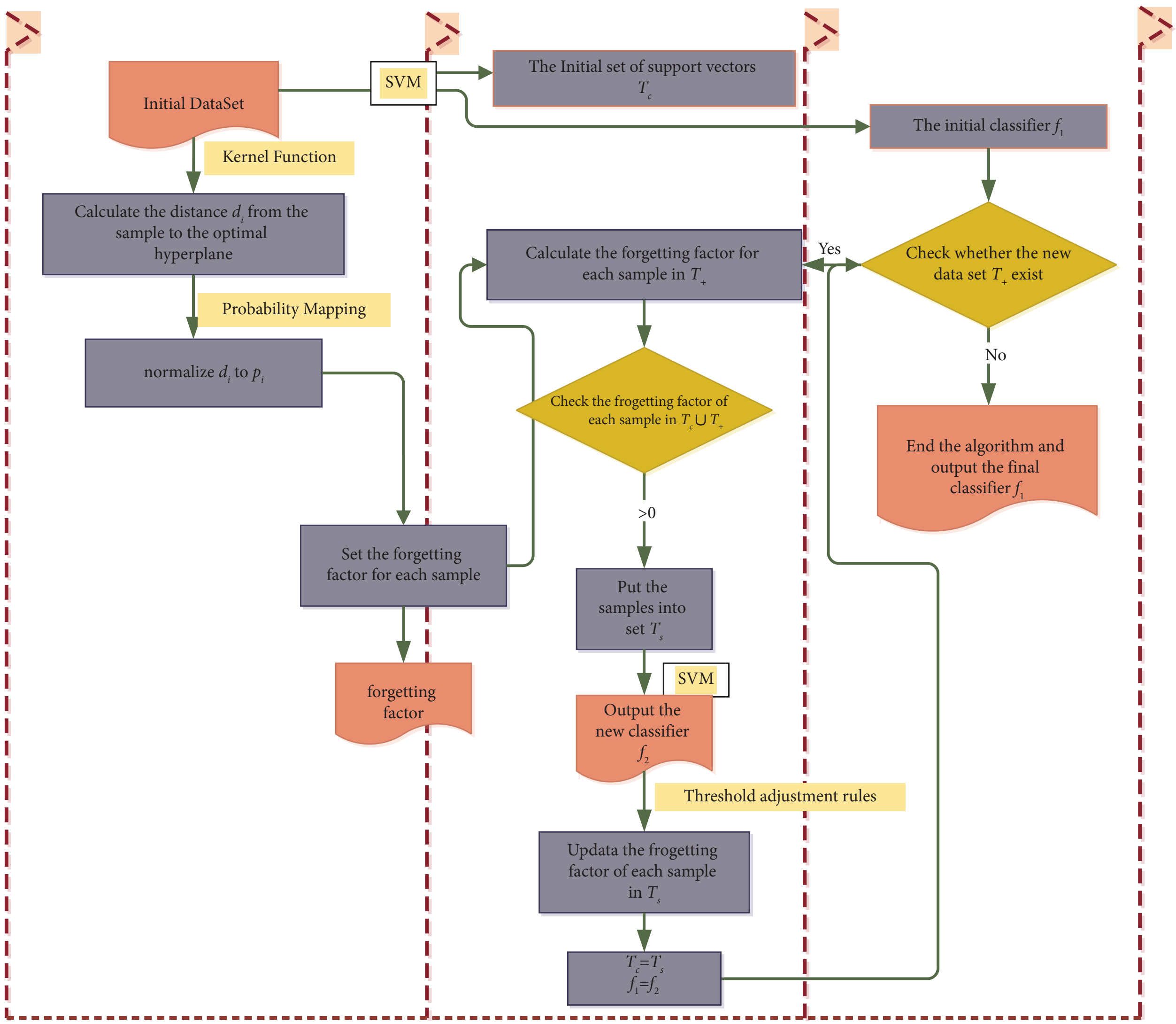


FIGURE 1: The flow of HDFC-ISVM algorithm.

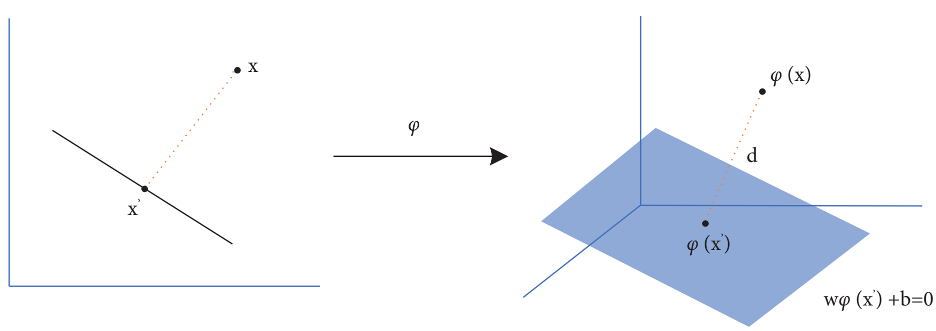


FIGURE 2: The distance of point x to the optimal hyperplane: $w \cdot \varphi(x) + b = 0$.

$$\begin{aligned}
w \cdot x' x &= (w_1, w_2, \ldots, w_n) \cdot (x_1 - x_1', x_2 - x_2', \ldots, x_n - x_n') \\
&= w_1 (x_1 - x_1') + w_2 (x_2 - x_2') + \cdots + w_n (x_n - x_n') \\
&= w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \\
&\quad - (w_1 x_1' + w_2 x_2' + \cdots + w_n x_n') \\
&= w \cdot x - (-b) \\
&= w \cdot x + b.
\end{aligned} \tag{4}$$

Formula (5) is given by formulas (3) and (4):

$$\|w\| \cdot d = |w \cdot \varphi(x) + b|. \tag{5}$$

Therefore, the distance from any point in n-dimensional Euclidean space to the optimal hyperplane is obtained as follows:

$$d = \frac{|w \cdot \varphi(x) + b|}{\|w\|}. \tag{6}$$

2.2. The Distance between Every Sample and the Optimal Hyperplane Is Calculated under the Action of Kernel Function. For nonlinear separability problems, it is necessary to introduce the mapping function $\varphi(x)$ to map the samples to the high-dimensional space, and then realize the linearly separability or approximately linearly separability of the samples. The literature [27] theoretically proves that under the action of kernel function, the higher the dimension of samples is, the higher the probability of linear separability is after they are mapped to a higher dimensional space and a better classification effect can be obtained. For this reason, this paper first calculates the distance between the sample and the hyperplane in the high-dimensional space under the action of the kernel function.

Let the mapping function be $\varphi(x)$ and the kernel function be $K(x_i, x_j)$. At this time, $w = \sum \alpha_i y_i \varphi(x_i)$ and put it into formula (6) to obtain formulas as follows:

$$\begin{aligned}
\|w \cdot x_k\| &= \|\sum \alpha_i y_i \varphi(x_i) \cdot \varphi(x_k)\| \\
\| &= \sum \alpha_i y_i K(x_i, x_k)\|,
\end{aligned} \tag{7}$$

$$\begin{aligned}
\|w\| &= \left|w \cdot w^T\right|^{(1/2)} = \left|\sum \alpha_i y_i \varphi(x_i) \sum \alpha_j y_j \varphi(x_j)\right|^{(1/2)} \\
&= \left|\sum \alpha_i y_i \varphi(x_i) \alpha_j y_j \varphi(x_j)\right|^{(1/2)} \\
&= \left|\sum \alpha_i \alpha_j y_i y_j \varphi(x_i) \varphi(x_j)\right|^{(1/2)} \\
&= \left|\sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)\right|^{(1/2)}.
\end{aligned} \tag{8}$$

Substitute formulas (7) and (8) into formula (6) to obtain the distance between the sample and the hyperplane in the high-dimensional space as the following formula:

$$d = \frac{\left|\sum \alpha_i y_i K(x_i, x_k) + b\right|}{\left|\sum \alpha_i \alpha_j y_i y_j K(x_i, x_j)\right|^{(1/2)}}. \tag{9}$$

2.3. Mapping and Normalization. HDFC-ISVM algorithm firstly trains an optimal classification hyperplane from the initial data, and then obtains the normative hyperplane $wx + b = \pm 1$ where the support vectors are located. In each increment process, formula (9) is used to calculate the distance between the newly added positive samples, negative samples, and the corresponding normative hyperplane $wx + b = \pm 1$, respectively. The distance between the positive sample x_i^+ and the hyperplane $wx + b = +1$ is denoted as d_i^+, and the distance between the negative sample x_i^- and the hyperplane $wx + b = -1$ is denoted as d_i^-, as shown in Figure 3.

Thus, it can be seen, this algorithm uses the distances between the samples in the high-dimensional space and the hyperplane (see formula (9)) to describe the geometric distribution of the samples. In order to better reflect the distribution information of the samples, these distances are then mapped into the corresponding probability value.

Definition 1. Let $D^+$and D^- represent the distance sets of the positive and negative samples from their respective normative hyperplanes that are the set $D^+ = \{d_i^+ \mid i \in X^+\}$ and the set $D^- = \{d_i^- \mid i \in X^-\}$. Let $d_{\max}^+, d_{\min}^+, d_{\max}^-, d_{\min}^-$ be the maximum and minimum values of samples in sets $D^+$and D^-, respectively, then the following definitions can be obtained:

$$\begin{aligned}
p_i^+ &= \frac{d_{\max}^+ - d_i^+}{d_{\max}^+ - d_{\min}^+}, \\
p_j^- &= \frac{d_{\max}^- - d_j^-}{d_{\max}^- - d_{\min}^-}, i \in D^+, j \in D^-.
\end{aligned} \tag{10}$$

In essence, Definition 1 normalizes the distance from each sample to the corresponding normative hyperplane and maps the distance in Euclidean space to the interval (0,1) .

2.4. Forgetting Factor and Initialization

2.4.1. Forgetting Factor. Before introducing HDFC-ISVM algorithm proposed in this paper, we first give Definition 2.

Definition 2 see([15]). The definitions for the following types of samples are given as follows:

① The SV samples that have never been selected for any round of training in the sample set are called the in-class samples, which usually account for a large proportion in the dataset

② The samples that always appear in each round of SV sets are called the boundary samples

③ The samples which jitter appear in the SV set are defined as the quasi-boundary samples

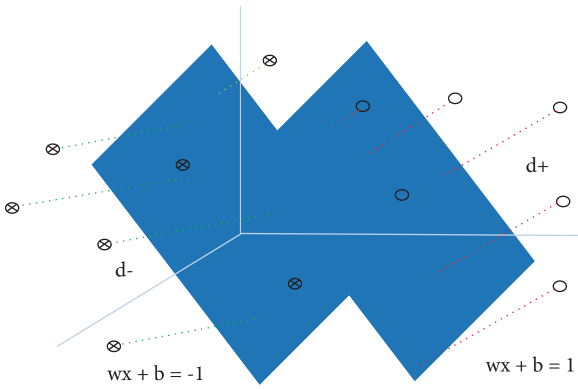


FIGURE 3: The distance of point x to the normative hyperplane: $wx + b = \pm 1$.

It is not difficult to see from Definition 2 that different types of samples have different contributions to the final classifier due to their different geometric distribution characteristics. The boundary samples contain most of the information of classification. The quasi-boundary samples are the supplement and correction of the information carried by boundary samples, while the classification information carried by the inner samples can be covered by the boundary samples and the quasi-boundary samples. Therefore, only by focusing on boundary samples and quasi-boundary samples can the classification effect of incremental learning be better improved.

Based on the above analysis, we can eliminate the inner samples according to certain rules to reduce the storage of historical samples, and then select some quasi-boundary samples to accelerate the convergence speed of the SV set and improve the accuracy of incremental learning. In order to achieve the abovementioned purpose, in the process of incremental learning, HDFC-ISVM algorithm introduces the forgetting factor α to select the dataset.

For the new round of incremental learning dataset X_{add}, firstly the distances are mapped using formula (10), each distance is mapped to a set of p values, and all the p values are ordered from small to large. This algorithm sets the value of 60% quantiles as p_s, the value of 75% quantiles for p_q, and 90% quantiles for p_m. We set the following factors for every sample x_i in the set X_{add} according to its p_i value:

① If $p_i \leq p_s$, then for the data point x_i, its forgetting factor $\alpha_i = 0$ is assigned

② If $p_s < p_i \leq p_q$, then for the data point x_i, its forgetting factor $\alpha_i = 0.1$ is assigned

③ If $p_q < p_i \leq p_m$, then for the data point x_i, its forgetting factor $\alpha_i = 0.15$ is assigned

④ If $p_i > p_m$, then for the data point x_i, its forgetting factor $\alpha_i = 0.3$ is assigned

2.4.2. Threshold Adjustment Rule. Determine whether each sample x_i in the dataset X is in the SV set X_{sv} which is obtained after the new round of SVM training. If it is in this set, the corresponding forgetting factor $\alpha_i = \alpha_i + 0.05$; if the sample point x_i is not in the set X_{sv}, the corresponding forgetting factor $\alpha_i = \alpha_i - 0.1$, and the forgetting factor corresponding to every sample of the set X is updated according to this rule.

2.5. The Steps of HDFC-ISVM Algorithm. Based on the above analysis, a new ISVM algorithm—HDFC-ISVM algorithm is proposed in this paper. It tries its best to retain some samples that may become support vectors, discard the useless samples for classifier training, and improve the classification efficiency of the algorithm based on ensuring the accuracy of the algorithm. The specific process of HDFC-ISVM algorithm is described in Algorithm .

2.6. Experiment and Result Analysis. In this paper, simulated experiments are carried out for the four algorithms using different datasets with different sample numbers, different dimensions, and different distribution characteristics, and the experimental results are analyzed and evaluated.

2.6.1. Experimental Datasets of the Original HDFC-ISVM Algorithm. This experiment selects some datasets in the UCI database. The specific information of the experimental datasets is shown in Table 1.

2.6.2. The Experimental Results of the Original HDFC-ISVM Algorithm. This experiment uses the 4 incremental learning algorithms: simple-ISVM algorithm, KKT-ISVM algorithm, CSV-ISVM algorithm, and HDFC-ISVM algorithm proposed in this paper to make the incremental learning trains and compare their learning effects for the 6 datasets in Table 1. In this experiment, the initial training dataset contains 500 samples, and 500 samples are added each time for incremental learning until all training samples are completed. The comprehensive performance of the classification effects of the above algorithms is evaluated through the following indicators.

$$\text{ACC} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \tag{11}$$

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{12}$$

$$\text{TNR} = \frac{\text{TN}}{\text{TN} + \text{FP}}, \tag{13}$$

$$F_1 - \text{score} = \frac{2\left((\text{TP}/\text{TP} + \text{FP}) \cdot (\text{TP}/\text{TP} + \text{FN})\right)}{(\text{TP}/\text{TP} + \text{FP}) + (\text{TP}/\text{TP} + \text{FN})}, \tag{14}$$

where TP,TN, FP, and FN, respectively, represent true positive cases(that is the number of positive samples predicted to be positive samples by the model), true negative cases (that is the number of negative samples predicted to be negative samples by the model), false positive cases (that is the number of negative samples predicted to be positive samples by the model), and false negative cases (that is the

Premise: suppose $T = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}, x_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i = 1, 2, \ldots, n$, T is the initial sample set, and T_+ is the new sample set;
Objective: it will find the SVM classifier based on $T \bigcup T_+$.
Step 1. The initial dataset T is trained to obtain the initial classifier $f_1 T$ and the initial SV set T_c, and the forgetting factor corresponding to each sample in the set T_c is calculated.
Step 2. Check whether the incremental set T_+ exists. If not, the algorithm ends, and $f_1 T$ is the final classifier; otherwise, it will enter into step 3.
Step 3. For the incremental set T_+, the forgetting factor of every sample of T_+ is calculated according to the classifier $f_1 T$.
Step 4. Set $T_{\text{total}} = T_+ \bigcup T_c$, it will select the samples whose forgetting factor satisfied $\alpha_i > 0$ to construct the set T_s.
Step 5. A new round of SVM training is carried out for the dataset T_s to obtain a new classifier f_2.
Step 6. For the classifier f_2, the above threshold adjustment rule is used to update the forgetting factor of each sample in the dataset T_s, setting $T_c = T_s, f_1 = f_2$ and then turning to step 2.

Algorithm 1: The process of HDFC-ISVM algorithm.

Table 1: The description of the datasets used in the experiment.

Datasets	The number of training samples	The number of testing samples	The samples' dimension
Breast_cancer	2000	770	9
German	3500	1500	20
Heart	4250	2500	13
Image	6500	1050	18
Mushroom	5644	1000	22
Thyroid	2800	1500	5

number of positive samples predicted to be negative samples by the model). ACC is the accuracy rate, which represents the proportion of the number of samples correctly predicted by the model to the total number of samples. TPR is sensitivity, which denotes the proportion of the samples correctly predicted to be positive samples in all positive samples, and TNR is specificity, which denotes the proportion of the samples correctly predicted to be negative samples in all negative samples. $F1$-score takes into account both accuracy and recall of classification models. It can be regarded as a weighted average of model accuracy and recall.

Tables 2–7 respectively show the predicted results of the above four incremental learning algorithms for the 6 datasets listed in Table 1, where "Iteration count" refers to the incremental learning times, "ACC" refers the accuracy rate of the classifiers, and "Time" refers to the training time of this incremental learning. In addition, *TPR* and *TNR* values in Table 8 represent the sensitivity and specificity indexes of the classifiers. Finally, aiming at the training sets and the testing sets, we compare the classification accuracy after each incremental learning by taking their average values.

According to the abovementioned experimental results, we obtained the comparison graphs of the accuracy and cumulative time of the abovementioned four algorithms for all the datasets, as shown in Figures 4 and 5. The comparison graph of TPR and TNR values of all the algorithms for the 6 datasets is shown in Figure 6. Figure 7 shows the average classification accuracy for the training sets and testing sets according to Table 9. In Table 9, "Train_Acc" represents the average precision of training datasets, and "Test_Acc" represents the average precision of testing datasets.

As can be seen from Figure 4, during the incremental learning process, the classification accuracy of Simple-ISVM and KKT-ISVM algorithms fluctuates greatly and the robustness of the classifiers is poor because the two algorithms ignore the set of quasi-boundary vectors that may become SVs. However, the classification accuracy of CSV-ISVM algorithm and HDFC-ISVM algorithm fluctuates less during the incremental process, and shows an overall growth trend. The classification accuracy of HDFC-ISVM algorithm is slightly lower than other algorithms under the initial state, but it can continuously learn the spatial distribution knowledge of samples and adjust the training set through the forgetting factors during the incremental learning process. So that it can obtain slightly higher accuracy than Simple-ISVM and KKT-ISVM algorithms at last.

From Figure 5, it can be seen that the difference of training time of the four algorithms is not big in the initial stages of training, but the total run time of the four algorithms has a huge difference after several incremental learning. The average running time of Simple-ISVM algorithm and KKT-ISVM algorithm is 50% longer than CSV-ISVM algorithm and HDFC-ISVM algorithm. Comparatively, the average cumulative running time of HDFC-ISVM algorithm is less than that of the other algorithms, so this algorithm has a great advantage in training efficiency.

It can be intuitively seen from Figure 6, compared with other algorithms, HDFC-ISVM algorithm has little difference in sensitivity to positive and negative samples for all the datasets, and the maximum accuracy difference is only within 5%, which makes this algorithm obtain better classification accuracy for both positive and negative samples.

Figure 7 shows the average classification accuracy of the classifiers for the training sets and the testing sets after all incremental learning. HDFC-ISVM algorithm has certain advantages in the average classification accuracy for the training set and testing set. In addition, the difference between the average classification accuracy of HDFC-ISVM

Table 2: The comparison of classification performance of the four algorithms for "breast_cancer."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	0.947	0.071	0.966	0.127	0.978	0.086	0.928	0.067
2	0.952	0.229	0.950	0.191	0.974	0.131	0.959	0.053
3	0.945	0.223	0.932	0.229	0.982	0.121	0.973	0.096
4	0.947	0.238	0.952	0.145	0.974	0.100	0.985	0.070

Table 3: The comparison of classification performance of the four algorithms for "German."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	0.998	0.227	1.000	0.213	0.994	0.142	0.93	0.117
2	0.792	0.353	0.864	0.189	0.988	0.147	0.954	0.169
3	0.924	0.383	0.923	0.231	0.976	0.133	0.994	0.214
4	0.98	0.391	0.943	0.146	0.982	0.138	0.997	0.23
5	0.988	0.473	0.906	0.160	0.992	0.149	0.998	0.144
6	0.996	0.466	0.959	0.183	1	0.148	0.997	0.17

Table 4: The comparison of classification performance of the four algorithms for "heart."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	1	0.103	1	0.101	0.998	0.131	0.985	0.093
2	0.984	0.146	0.984	0.128	0.996	0.099	0.98	0.133
3	1	0.151	0.988	0.090	0.994	0.117	1	0.08
4	0.998	0.113	0.982	0.104	0.996	0.131	1	0.081
5	1	0.148	0.99	0.104	0.998	0.082	1	0.08
6	0.998	0.146	0.984	0.115	0.998	0.119	1	0.105
7	1	0.132	0.984	0.122	0.993	0.082	1	0.097
8	1	0.124	0.978	0.133	0.996	0.104	1	0.132

Table 5: The comparison of classification performance of the four algorithms for "image."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	1	0.071	1.000	0.163	0.998	0.088	0.986	0.107
2	0.94	0.134	0.938	0.197	0.998	0.098	0.962	0.104
3	0.976	0.270	0.968	0.166	0.998	0.157	0.977	0.122
4	0.976	0.175	0.950	0.121	0.996	0.150	0.985	0.105
5	0.982	0.217	0.934	0.134	1	0.164	0.989	0.134
6	0.982	0.291	0.948	0.142	0.998	0.116	0.985	0.172
7	0.98	0.197	0.956	0.146	0.996	0.099	0.992	0.130
8	0.998	0.299	0.950	0.167	0.998	0.117	0.989	0.154
9	0.998	0.331	0.948	0.145	1	0.115	0.996	0.171
10	0.99	0.289	0.985	0.181	0.998	0.156	0.992	0.158
11	0.992	0.299	0.979	0.140	0.998	0.152	0.993	0.121
12	0.996	0.259	0.976	0.146	0.998	0.150	0.999	0.147

Table 6: The comparison of classification performance of the four algorithms for "mushroom."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	1	0.1828	1	0.1641	0.999	0.1319	0.998	0.132
2	0.996	0.1797	1	0.1719	0.94	0.1873	0.989	0.187
3	0.9823	0.245	0.994	0.1428	0.97	0.1575	0.994	0.158
4	0.994	0.3161	0.998	0.1838	0.986	0.13	0.991	0.13

Table 6: Continued.

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
5	1	0.5385	0.998	0.1862	0.997	0.1997	1	0.2
6	0.996	0.5215	0.982	0.1805	0.999	0.1751	0.997	0.175
7	0.994	0.6904	0.98	0.1546	0.999	0.197	0.999	0.197
8	0.994	0.6168	0.918	0.1906	0.996	0.1462	0.999	0.146
9	0.998	0.7751	1	0.1299	0.997	0.1971	1	0.197
10	0.9986	0.7634	0.994	0.1934	0.999	0.1681	1	0.168
11	0.9917	1.0033	0.998	0.2045	0.993	0.2079	1	0.208

Table 7: The comparison of classification performance of the four algorithms for "thyroid."

Iteration count	Simple-ISVM		KKT-ISVM		CSV-ISVM		HDFC-ISVM	
	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)	ACC	Time (s)
1	0.996	0.060	1.000	0.053	0.994	0.060	0.999	0.044
2	0.994	0.058	0.996	0.042	0.967	0.079	0.996	0.041
3	0.934	0.086	0.941	0.055	0.934	0.041	0.970	0.065
4	0.960	0.159	0.960	0.149	0.987	0.063	0.975	0.051
5	0.992	0.158	0.992	0.135	1.000	0.038	0.999	0.052

Table 8: The comparison of the TPR value and TNR value of the four algorithms for all the datasets.

Data_set	Algorithm	TPR (%)	TNR (%)
Breast_cancer	Simple-ISVM	96.67	90.51
	KKT-ISVM	97.88	88.80
	CSV-ISVM	99.56	93.53
	HDFC-ISVM	97.54	92.97
German	Simple-ISVM	95.25	92.66
	KKT-ISVM	94.01	91.18
	CSV-ISVM	99.71	96.78
	HDFC-ISVM	98.16	96.91
Heart	Simple-ISVM	99.95	99.49
	KKT-ISVM	99.09	98.10
	CSV-ISVM	99.61	99.47
	HDFC-ISVM	99.65	99.60
Image	Simple-ISVM	98.56	98.32
	KKT-ISVM	96.80	98.77
	CSV-ISVM	98.66	95.66
	HDFC-ISVM	99.78	99.79
Mushroom	Simple-ISVM	99.80	97.45
	KKT-ISVM	97.96	99.90
	CSV-ISVM	98.75	98.56
	HDFC-ISVM	99.74	99.72
Thyroid	Simple-ISVM	98.27	94.62
	KKT-ISVM	98.22	95.00
	CSV-ISVM	98.92	95.23
	HDFC-ISVM	99.85	99.01

algorithm for the training set and the testing set is only 3.59%, which is lower than 5.61% of Simple-ISVM algorithm and 3.89% of CSV-ISVM algorithm. Therefore, HDFC-ISVM algorithm has strong generalization performance.

2.7. Parameter Sensitivity Analysis of the Original HDFC-ISVM Algorithm. The influence of hyperparameters involved in HDFC-ISVM algorithm on experimental results is studied, and a series of hyperparameters involved in this algorithm are tested to fully explore the sensitivity of the algorithm to the introduced hyperparameters.

2.7.1. Sensitivity Analysis of Parameters p_s, p_q, and p_m. For HDFC-ISVM algorithm proposed above, the incremental learning dataset X_{add} is mapped to a set of p values using formula (10), and then three hyperparameters

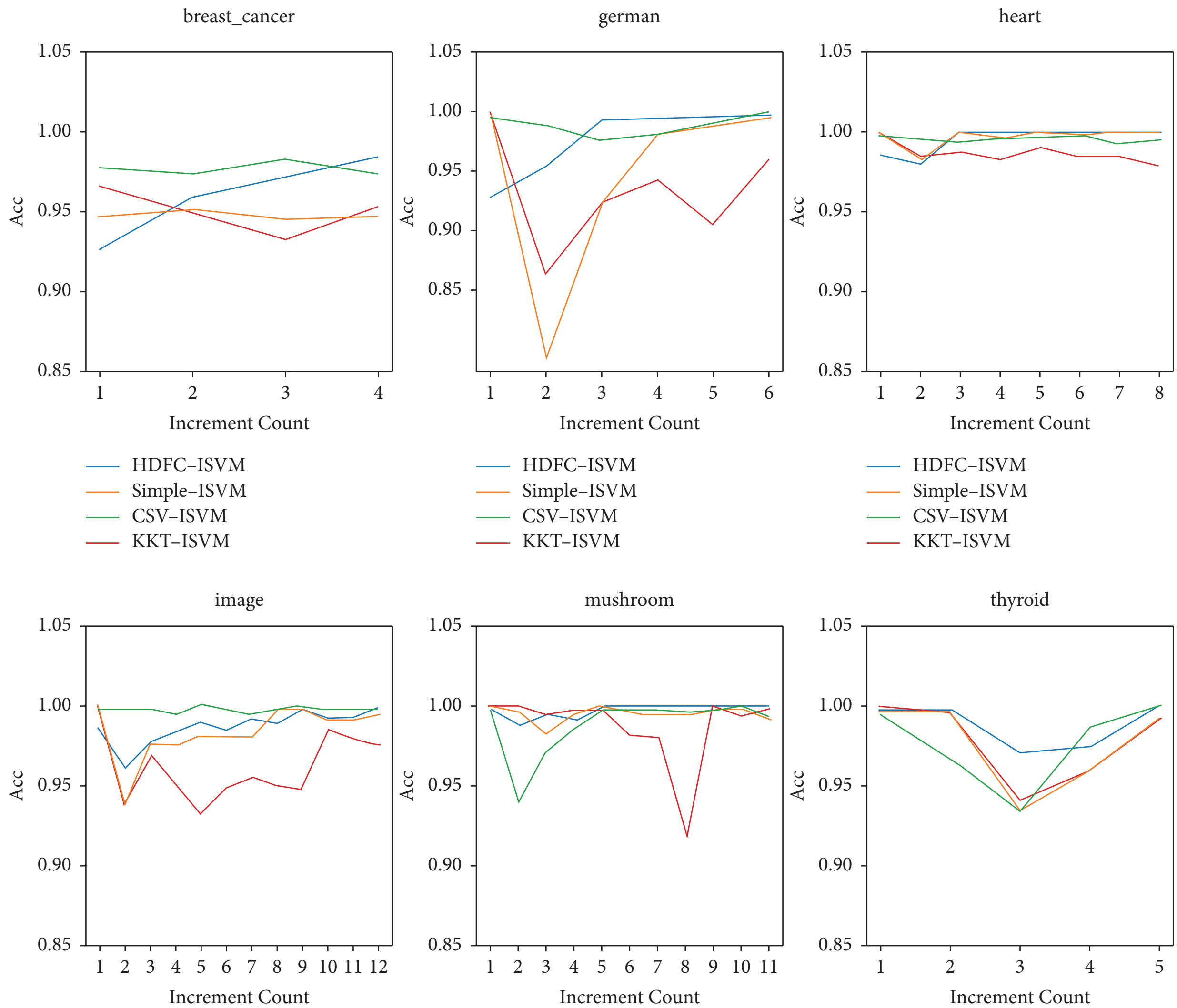


Figure 4: The classification accuracy of the four algorithms for all the datasets during the process of incremental learning.

p_s, p_q, and p_m are introduced as the thresholds for initializing the forgetting factor. In order to explore the sensitivity of this algorithm to the initial threshold of the forgetting factor, four groups of different parameters are set for the hyperparameters p_s, p_q, and p_m, and tests are carried out on the abovementioned 6 datasets. The experimental results are shown in Table 10. The first column in Table 10 is the values of the hyperparameters p_s, p_q, *and* p_m used in the previous experiment.

It can be seen from Table 10 that the values of different groups have a greater impact on the experimental results. Appropriately increasing the values of the hyperparameters can improve the classification accuracy to a certain extent. However, if the values of the hyperparameters are too large, the classification accuracy will decrease. Meanwhile, it can be seen from Table 10 that different hyperparameters have different influences on the classification accuracy for different datasets. Experimental results show that setting different hyperparameters will make the accuracy fluctuate to a certain extent, so HDFC-ISVM algorithm is sensitive to parameters p_s, p_q, and p_m.

2.7.2. Sensitivity Analysis of Parameter α. For HDFC-ISVM algorithm proposed above, the assignment strategy of forgetting factor α has a great impact on the algorithm performance, and different assignment of α will lead to different tendencies in selecting candidate support vectors. Therefore, this paper still adopts the abovementioned 6 datasets and selects different assignment strategies to explore the influence of α on the experimental results. The specific results are shown in Table 11. Here, all the results in Table 11 are the accuracy when p_s, p_q, and p_m is set to 0.6, 0.75, and 0.9 and α is set to different values. Table 11 lists the combinations of 4 different thresholds of parameter α corresponding to the four different situations and the first column shows the values of α used in the previous experiment.

It can be seen from Table 11, the different assignment strategies of α will have great influences on the final results of this algorithm. If the value of α is too small, the forgetting factor cannot play its due effect, resulting in some data samples being forgotten prematurely. When the value of forgetting factor α increases, the testing accuracy will improve for some datasets, while it will decrease for others. At

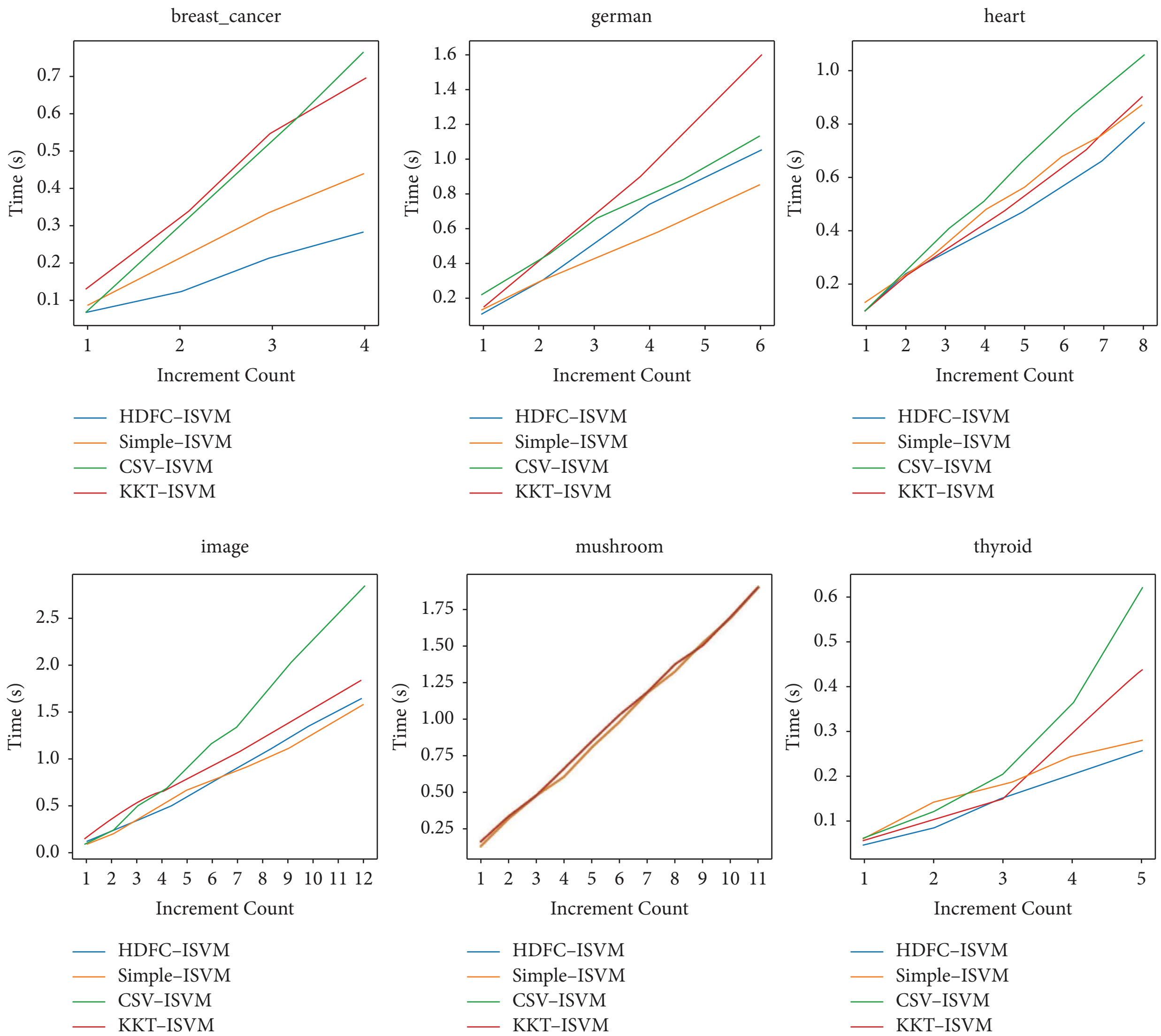


Figure 5: The cumulative time of the four algorithms for all the datasets during the process of the incremental learning.

the same time, increasing the assignment of α leads to more data to be learned incrementally, which increases the training time. Therefore, it can be seen that the algorithm is also sensitive to parameter α.

2.7.3. The Conclusion about the Parameter Sensitivity. The experimental results show that the original HDFC-ISVM algorithm is sensitive to both p_s, p_q, and p_m and α, and different parameters need to be adjusted for different datasets to achieve the best classification effect. When the values of $p_s, p_q, and\ p_m$ are 0.6, 0.75, and 0.9, respectively, the performance of this algorithm is relatively stable for the abovementioned 6 datasets. When the parameters' values are increased or decreased, the accuracy of the classifier will fluctuate for different datasets. Similarly, when the values of α are 0.1, 0.15, and 0.3, the testing accuracy of the classifier for all the 6 datasets is high, while increasing the values of the parameters will lead to rapid decline of testing accuracy for some datasets.

2.8. The Improvement Strategy for the Original HDFC-ISVM Algorithm. From the abovementioned sensitivity experiments, it can be seen that different settings of parameters have a great impact on the final classification accuracy for the original HDFC-ISVM algorithm. Because of too many parameters, the algorithm cannot adapt to the datasets with different distributions. The datasets with different distributions often need different groups of hyperparameters to achieve the ideal classification results. So, it needs to adjust the initialization strategy and update a rule for forgetting factor α to some extent. The following article will do this work and the algorithm after adjusting the forgetting factor initialization rule and updating strategy is called HDFC-ISVM $*$. The new rules are as follows.

2.8.1. Initialization Process. For the new round of incremental learning dataset X_{add}, we first use formula (10) to perform probability calculation to obtain the set of p values of all samples, and then the forgetting factor is initialized by

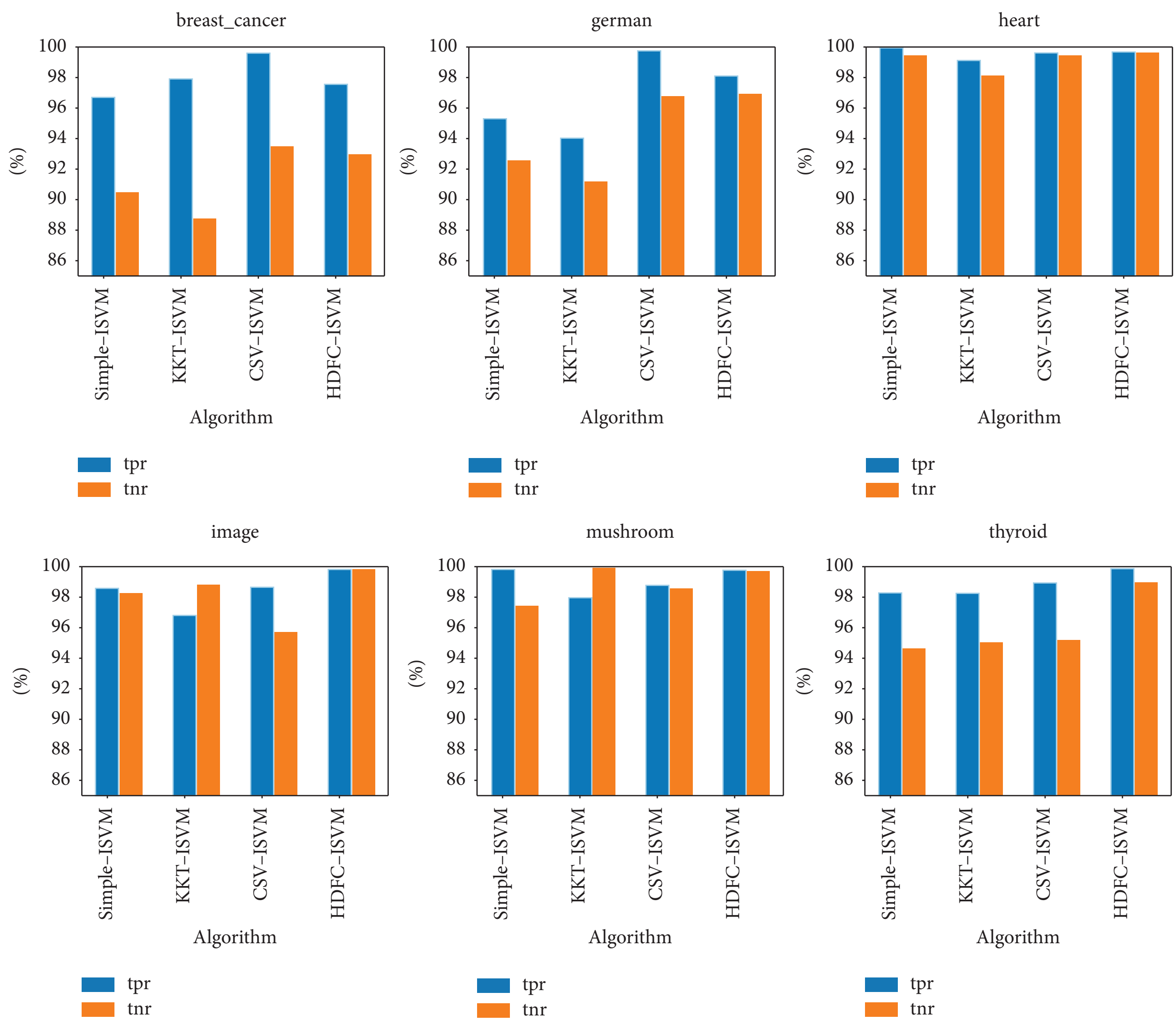


Figure 6: The comparison of TPR and TNR values of all the algorithms for the 6 datasets.

assigning to the samples in dataset X_{add} by the following formula:

$$\alpha_i = \begin{cases} \operatorname{ceil}\left\{(p_i - p_{\min})^2\right\}, & p_i > \theta, \\ 0, & \text{else}, \end{cases} \tag{15}$$

where p_i is calculated by formula (10), $p_{\min}$ represents the minimum in the set of p values for this round, ceil{ } represents the result taken up by one decimal place, and $\theta \in (0, 1)$ represents the regulating parameter.

All the samples in incremental dataset X_{add} are marked with the corresponding forgetting factors by the above-mentioned method, then the dataset of the previous round is combined, and the samples with forgetting factor $\alpha_i > 0$ are screened out. The samples constitute dataset X_{ch}, and a new round of SVM training is conducted for dataset X_{ch}.

2.8.2. Updating Rule for the Forgetting Factor. In order to make the forgetting factor self-adaptive update, reduce the setting of parameters and improve the generalization performance of the model, this paper proposes a new forgetting factor update strategy, that is, before a new round of incremental training, the forgetting factor is updated for the original data as follows:

$$\alpha_{i+1} = \alpha_i + \frac{1}{1+\alpha_i}\text{is.Contains}\left(X_{SV}, x_i\right) + \frac{1}{\theta}\max_{j\in SV}\left\{\min\left(\frac{x_i \cdot x_j^T}{\|x_i\| \cdot \|x_j\|} - 1, 0\right)\right\}, \tag{16}$$

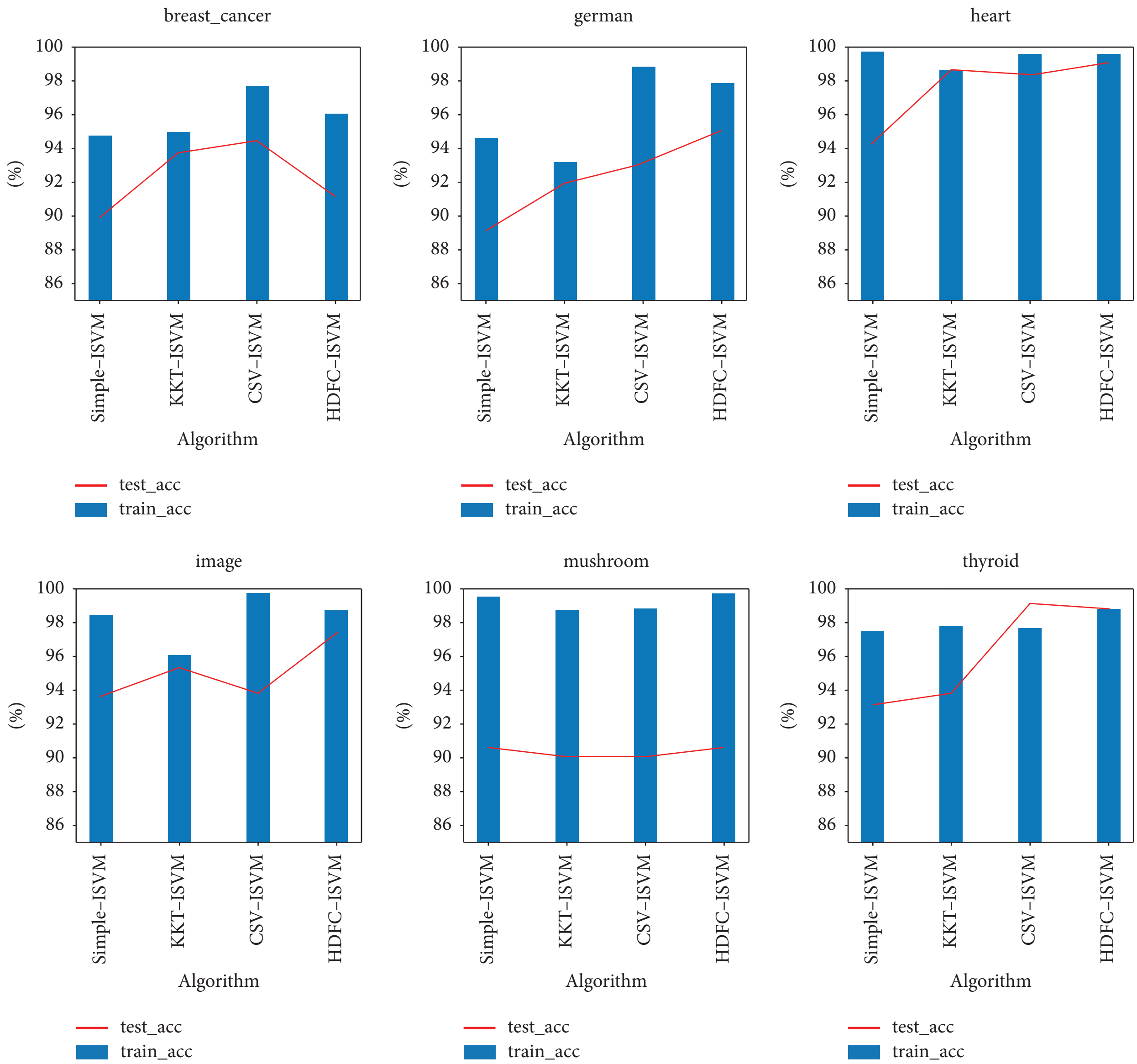


Figure 7: The comparison of average classification accuracy for the training sets and testing sets.

where is $\cdot$ Contains $(\cdot)$ indicates whether the set X_{SV} contains x_i. If it does, return 1, otherwise, return 0. x_j represents the member of set X_{SV}.

The interpretation of formula (16) is as follows. When x_i is the support vector after the last round of training, then function is $\cdot$ Contains $(\cdot)$ returns 1 and $\max_{j\in SV}\{min\ ((x_i \cdot x_j^T/x_i \cdot x_j) - 1, 0)\}$ returns 0, and $(1/1 + \alpha_i)$ acts as a weight to adjust the increment of the forgetting factor. When the forgetting factor is large, the increment will decrease in each round to ensure the sensitivity of the forgetting factor to the candidate support vectors. On the contrary, if x_i is not the support vector after the last round of training, then function is $\cdot$ Contains $(\cdot)$ returns 0. At this point, the forgetting factor is reduced by function $\max_{j\in SV}\left\{min((x_i \cdot x_j^T/x_i \cdot x_j) - 1, 0)\right\}$ because the inner function $min((x_i \cdot x_j^T/x_i \cdot x_j) - 1, 0)$ discriminates the distance between x_i and support vector x_j by comparing the cosine similarity between them. We think that the closer x_i is to the support vector, the more likely it is to be a support vector. Therefore, the distance mapping between x_i and the nearest support vector is obtained through the calculation of function $\max_{j\in SV}\{min\ ((x_i \cdot x_j^T/x_i \cdot x_j) - 1, 0)\}$. When updating α_i, this algorithm adjusts the attenuation size by threshold $(1/\theta)$. The closer it is to the current support vector, the smaller attenuation of α_i is, and the further it is, the greater attenuation of α_i is. In this way, the forgetting factor is initialized and updated, the number of parameters is reduced, the algorithm can adjust the updating rules adaptively by data distribution for different datasets, and the generalization performance of this algorithm is improved.

2.9. Analysis of Experiments and Results of HDFC-ISVM Algorithm*

2.9.1. Experimental Datasets for the Improved Algorithm. In order to better test the algorithm performance after adjustment, the experiment added 6 datasets in the UCI

Table 9: The average accuracy of the training datasets and testing datasets.

Data_set	Algorithm	Train_Acc (%)	Test_Acc (%)
Breast_cancer	Simple-ISVM	94.78	89.91
	KKT-ISVM	95.00	93.70
	CSV-ISVM	97.69	94.50
	HDFC-ISVM	96.12	91.16
German	Simple-ISVM	94.63	89.22
	KKT-ISVM	93.23	91.97
	CSV-ISVM	98.85	93.10
	HDFC-ISVM	97.81	94.99
Heart	Simple-ISVM	99.75	94.32
	KKT-ISVM	98.63	98.61
	CSV-ISVM	99.60	98.34
	HDFC-ISVM	99.57	99.12
Image	Simple-ISVM	98.42	93.70
	KKT-ISVM	96.10	95.34
	CSV-ISVM	99.79	93.83
	HDFC-ISVM	98.72	97.50
Mushroom	Simple-ISVM	99.50	90.61
	KKT-ISVM	98.75	90.15
	CSV-ISVM	98.86	90.18
	HDFC-ISVM	99.71	90.53
Thyroid	Simple-ISVM	97.52	93.22
	KKT-ISVM	97.78	93.84
	CSV-ISVM	97.65	99.16
	HDFC-ISVM	98.79	98.77

Table 10: The accuracy (%) of the original HDFC-ISVM algorithm under different settings of parameters p_s, p_q, and p_m.

Data_set	Different values of parameters p_s, p_q, and p_m			
	(0.6, 0.75, and 0.9)	(0.1, 0.4, and 0.7)	(0.3, 0.6, and 0.9)	(0.4, 0.6, and 0.8)
Breast_cancer	91.16	88.47	92.37	93.63
German	94.99	90.23	84.40	95.14
Heart	99.12	96.51	99.04	100.00
Image	97.50	94.79	98.97	98.81
Mushroom	90.53	86.67	93.74	88.66
Thyroid	98.77	95.15	98.40	98.67

Table 11: The accuracy (%) of the original HDFC-ISVM algorithm under different settings of parameter α.

Data_set	Different values of parameters α			
	(0, 0.1, 0.15, and 0.3)	(0, 0.05, 0.1, and 0.15)	(0, 0.05, 0.1, and 0.3)	(0, 0.15, 0.2, and 0.3)
Breast_cancer	91.16	90.67	92.71	93.19
German	94.99	91.67	91.97	93.93
Heart	99.12	99.04	99.04	99.87
Image	97.50	95.83	95.30	98.18
Mushroom	90.53	90.14	90.16	91.14
Thyroid	98.77	94.26	94.91	97.66

library on the basis of the original 6 datasets, namely 12 experimental datasets, and the latest dataset information is shown in Table 12.

2.9.2. The Experimental Results. Based on the abovementioned experiments, this round of experiment compares the training results of Simple-ISVM [22], KKT-ISVM [23, 24], CSV-ISVM [14], GGKKT-ISVM [25], CD-ISVM [26], HDFC-ISVM, and HDFC-ISVM* (HDFC-ISVM* algorithm is the improved algorithm based on the original HDFC-ISVM algorithm) for the abovementioned 12 datasets in Table 12. In this experiment, for all the algorithms mentioned above, the initial training datasets contain 500 samples. Each time 500 samples are added for incremental learning until all training samples are trained, and the value of θ is 0.3. The ACC index and F_1-score index are introduced simultaneously to evaluate the performance of the classifier

Table 12: The description of the updated experimental datasets.

Datasets	The number of training samples	The number of testing samples	The samples' dimension
Breast_cancer	2000	770	9
German	3500	1500	20
Heart	4250	2500	13
Image	6500	1050	18
Mushroom	5644	1000	22
Thyroid	2800	1500	5
Titanic	10255	750	3
Splice	10875	5000	60
Diabetes	4680	1000	8
Credit	6000	1000	65
Spambase	4600	1500	57
Waveform	3344	1000	40

Table 13: The F_1-score values comparison of all the algorithms for different datasets.

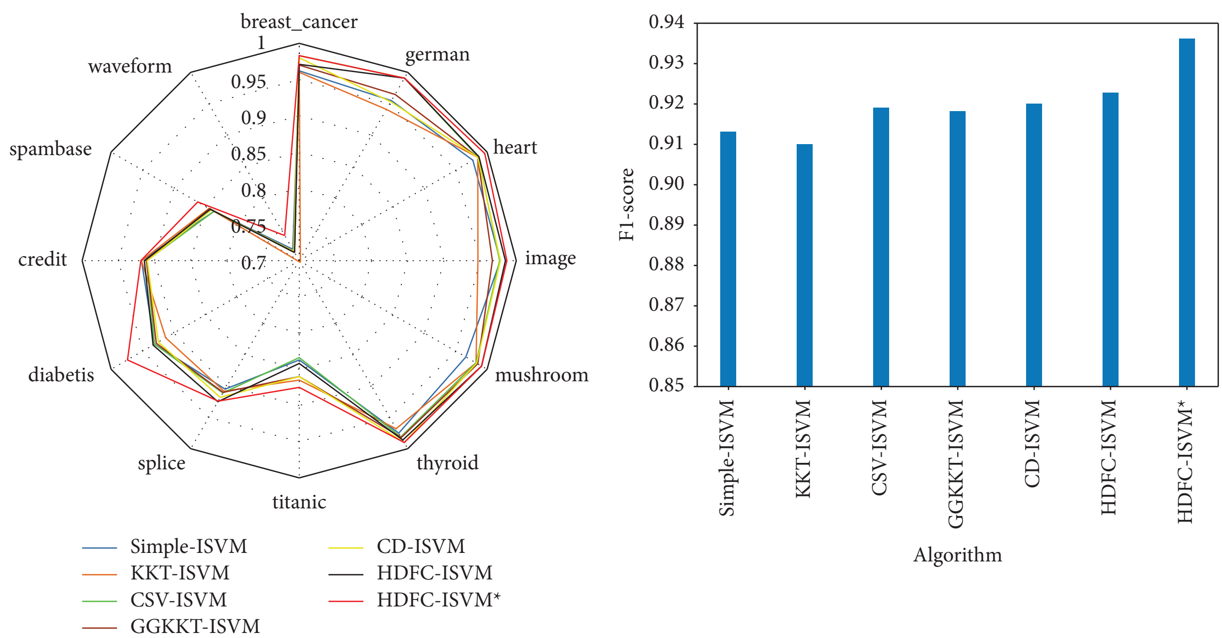
Datasets	F_1-score						
	Simple-ISVM	KKT-ISVM	CSV-ISVM	GGKKT-ISVM	CD-ISVM	HDFC-ISVM	HDFC-ISVM*
Breast_cancer	0.962	0.960	0.983	0.970	0.980	0.971	0.983
German	0.955	0.941	0.991	0.965	0.953	0.991	0.991
Heart	0.977	0.985	0.986	0.985	0.985	0.987	0.996
Image	0.978	0.947	0.977	0.967	0.978	0.985	0.987
Mushroom	0.966	0.982	0.982	0.985	0.983	0.991	0.991
Thyroid	0.975	0.968	0.981	0.982	0.990	0.986	0.990
Titanic	0.837	0.865	0.834	0.860	0.860	0.842	0.875
Splice	0.905	0.908	0.911	0.910	0.918	0.925	0.924
Diabetes	0.931	0.913	0.931	0.928	0.925	0.933	0.974
Credit	0.918	0.918	0.911	0.911	0.910	0.914	0.919
Spambase	0.836	0.844	0.836	0.843	0.840	0.842	0.862
Waveform	0.718	0.697	0.713	0.713	0.715	0.713	0.740
Mean	0.913	0.910	0.919	0.918	0.920	0.923	0.936

(see formulas (11) and (14)). The specific experimental results are as follows.

It can be seen from Table 13 and Figure 8 that the F_1-score values of HDFC-ISVM algorithm before and after the improvement are significantly improved for the abovementioned 12 datasets. The mean value of F_1-score of HDFC-ISVM* for all datasets is 0.936, 1.3 percent points higher than HDFC-ISVM algorithm before the improvement and 2.6 percent points higher than KKT-ISVM algorithm. The F_1-score mean value of HDFC-ISVM* algorithm for different datasets is higher than that of other algorithms, which proves that the improved algorithm has advantages in accuracy and recall rate compared with other algorithms. At the same time, it can be obtained from Tables 14 and 15 that the average training accuracy of the HDFC-ISVM* algorithm is 92.29% for all datasets, and the average testing accuracy 90.80% for all datasets. This algorithm is not only obviously better than other algorithms but also has better effect than the original HDFC-ISVM algorithm. It can be seen from Figure 9, the testing accuracy of the improved HDFC-ISVM* algorithm on almost all datasets is no lower than HDFC-ISVM algorithm, especially for the "mushroom" dataset, the testing accuracy of the improved algorithm is improved by 8.61%, and the testing accuracy for the "breast_cancer" dataset is improved by 5.46% compared to HDFC-ISVM algorithm. The experimental results show that by adjusting the initialization and update strategies of the forgetting factor, the new algorithm can better adjust the data of each training round and adjust the update strategy of the forgetting factor adaptively, so as to train the classifier with a better effect.

2.9.3. Sensitivity Analysis of Parameter θ. In order to further explore the influence of parameter θ on experimental results, the first 6 datasets in the abovementioned experiments are taken to test the accuracy of HDFC-ISVM* algorithm. The experimental results represent the accuracy (%) of the algorithm for different testing sets, with values of θ are 0.1, 0.2, 0.3, and 0.4, respectively. The experimental results are shown in Table 16.

It can be seen from the results in Table 16, different values of θ have a certain degree of influence on the experimental results. When the value of θ increases from 0.1 to 0.4, the classification accuracy of each dataset also fluctuates. In general, when the value of θ is 0.3, the algorithm performance is optimal. Further increasing the value of θ will not increase the classification accuracy of the algorithm, but will affect the algorithm's perception to the overall distribution of the datasets and reduce the classification accuracy of the algorithm because too many samples are deleted.



(a) (b)

Figure 8: The comparison of F_1-score values of the different algorithms for all datasets. (a) The radar graph of F_1-score value and (b) the comparison diagram of F_1-score means.

Table 14: The train_ACC values comparison of all the algorithms for different datasets.

Datasets	Train_ACC (%)						
	Simple-ISVM	KKT-ISVM	CSV-ISVM	GGKKT-ISVM	CD-ISVM	HDFC-ISVM	HDFC-ISVM*
Breast_cancer	94.78	95.0	97.69	96.50	97.85	96.12	98.86
German	94.63	93.23	98.85	95.60	97.70	97.81	99.50
Heart	99.75	98.63	99.60	98.89	98.90	99.57	100
Image	98.42	96.10	99.79	97.56	98.30	98.72	99.47
Mushroom	99.5	98.75	98.86	98.76	98.72	99.71	99.96
Thyroid	97.52	97.78	97.65	97.80	99.40	98.79	99.44
Titanic	76.94	81.98	77.03	81.85	81.83	80.75	82.33
Splice	90.84	92.26	90.19	92.20	92.19	92.26	92.26
Diabetes	94.33	94.73	94.95	94.10	94.20	94.98	97.81
Credit	86.52	85.12	85.60	85.60	85.80	86.01	86.09
Spambase	78.81	79.73	79.05	79.80	79.81	79.72	80.56
Waveform	68.34	65.67	68.97	67.20	68.53	68.97	71.28
Mean	90.03	89.92	90.68	90.49	91.10	91.11	92.29

Table 15: The test_ACC values comparison of all the algorithms for different datasets.

Datasets	Test_ACC (%)						
	Simple-ISVM	KKT-ISVM	CSV-ISVM	GGKKT-ISVM	CD-ISVM	HDFC-ISVM	HDFC-ISVM*
Breast_cancer	89.91	93.70	94.50	93.95	94.20	91.16	96.62
German	89.22	91.97	93.10	92.00	93.20	94.99	99.13
Heart	94.32	98.61	98.34	98.50	98.65	99.12	99.98
Image	93.70	95.34	93.83	95.40	95.56	97.50	99.24
Mushroom	90.61	90.15	90.18	90.30	90.25	90.53	99.14
Thyroid	93.22	93.84	99.16	95.20	99.20	98.77	99.26
Titanic	74.85	80.65	75.45	76.23	76.30	76.38	79.21
Splice	87.29	90.34	90.16	90.18	90.20	90.33	90.31
Diabetes	91.89	91.49	91.81	91.50	91.75	91.90	93.66
Credit	85.17	83.15	83.14	83.20	84.40	85.16	85.17
Spambase	76.9	77.24	76.99	77.35	77.87	77.13	78.53
Waveform	66.76	64.84	66.77	66.78	66.85	66.78	69.40
Mean	86.15	87.61	87.78	87.55	88.20	88.31	90.80

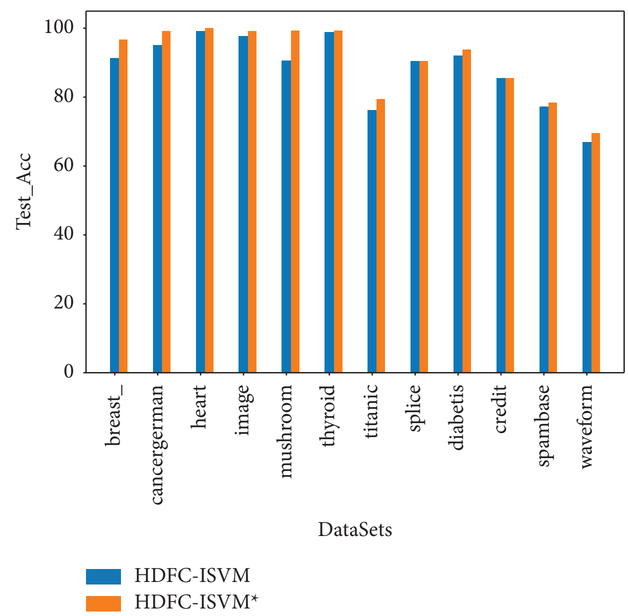


Figure 9: Precision comparison diagram of the testing set before and after HDFC-ISVM algorithm improvement.

Table 16: Classification accuracy of HDFC-ISVM* algorithm with different values of parameter θ.

Datasets	The different values of parameter θ			
	$\theta = 0.1$	$\theta = 0.2$	$\theta = 0.3$	$\theta = 0.4$
Breast_cancer	95.22	96.23	96.62	96.23
German	97.31	99.01	99.13	96.67
Heart	99.45	100.00	99.98	100.00
Image	97.26	99.25	99.24	99.08
Mushroom	96.73	98.99	99.14	98.77
Thyroid	98.51	99.26	99.26	98.66

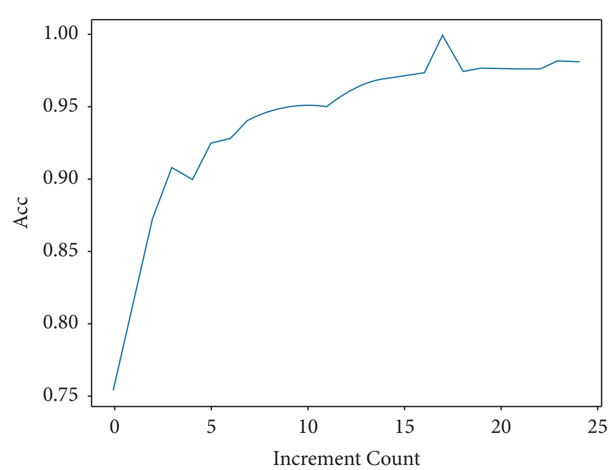


Figure 10: The incremental learning training precision graph of HDFC-ISVM* algorithm.

2.9.4. Application of HDFC-ISVM in Image Detection.* In order to explore the actual effect of HDFC-ISVM* algorithm in image classification, this paper adopts "catsvs-dogs" dataset provided by Kaggle as a training dataset. 5000 images are selected for classification to explore the effect of the proposed algorithm in image classification.

In this experimental training set, 2500 pictures of cats and 2500 pictures of dogs are selected for training. In this experiment, 20% of the training pictures are extracted by the method of 5 fold cross validation, AlexNet convolutional neural network [28] is used to extract image features, and 4096 dimensional features are finally extracted as input data.

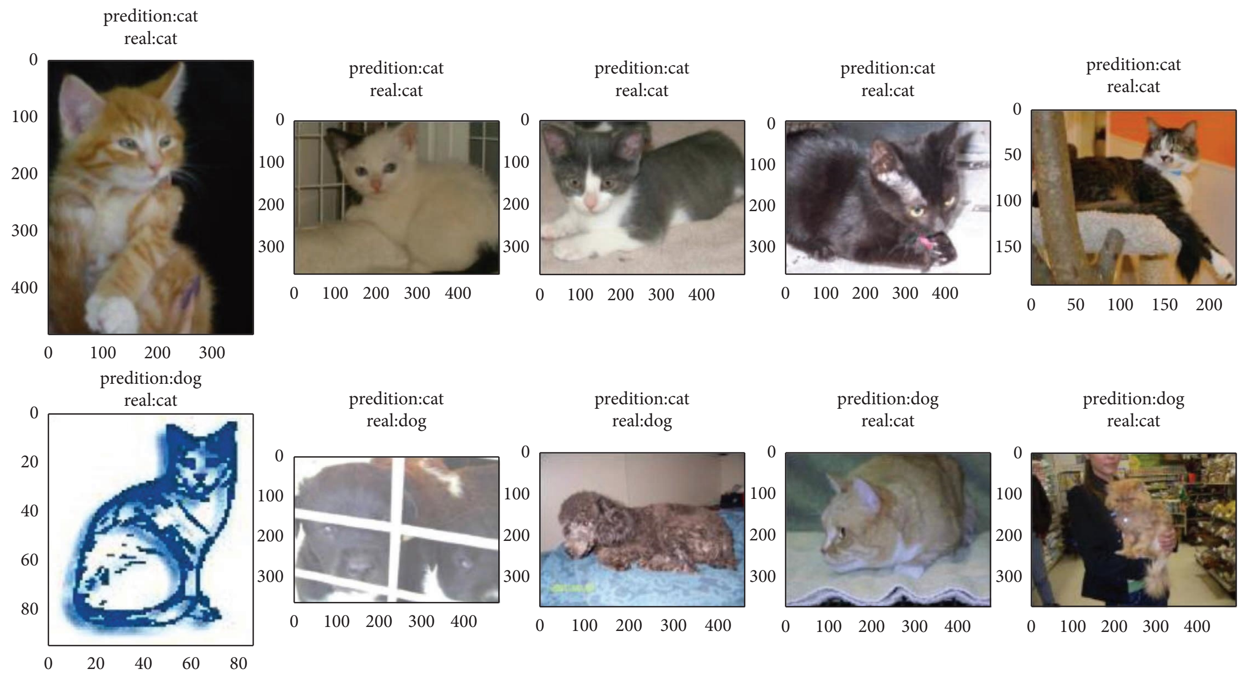


Figure 11: The partial classification results display of HDFC-ISVM* algorithm for dataset "catsvsdogs."

Table 17: The classification comparison results of AlexNet and HDFC-ISVM* for dataset "catsvsdogs."

Results	AlexNet	HDFC-ISVM*
Total time (s)	2593	514
Test_ACC (%)	95.23	97.93

It marks the cat as −1 and the dog as +1. The dataset is divided into 25 incremental learning units, each batch has 200 image data, and the experiment is carried out using HDFC-ISVM* algorithm to obtain data such as running time and test accuracy. The specific experimental results are as follows.

Figure 10 shows the incremental learning training precision of HDFC-ISVM* algorithm for the dataset "catsvsdogs." The first row in Figure 11 shows the partially correctly classified images, and the second row shows the partially incorrectly classified images. It can be seen from the experimental results, HDFC-ISVM* algorithm has achieved a good classification effect. In this experiment, the convolutional neural network algorithm—AlexNet algorithm is compared with HDFC-ISVM* algorithm proposed in this paper. The comparison results are shown in Table 17. It can be seen from Table 17, HDFC-ISVM ∗ algorithm has higher classification accuracy and better classification efficiency for image dataset "catsvsdogs."

3. Conclusion

In this paper, an improved incremental learning algorithm, HDFC-ISVM, is proposed, which achieves a good classification effect. On this basis, aiming at the sensitivity of parameters, the initialization strategy and update rule of the forgetting factor are adjusted to some extent, and an improved algorithm, HDFC-ISVM* algorithm, is proposed at last.

The algorithm has the following innovations:

(1) It uses the distance formula in the high-dimensional space to better express the spatial distribution law of samples;

(2) Forgetting factor screening method is proposed and relevant screening strategies are formulated to retain as much as possible part of the datasets that may become support vectors to improve the classification accuracy. On this basis, the initialization strategy and update rule of the forgetting factor are further adjusted. The experimental results show that HDFC-ISVM* algorithm has a good classification effect on most datasets and has the same sensitivity to positive and negative samples. The experiments verify that HDFC-ISVM* algorithm has higher average classification accuracy for the training sets and testing sets compared with other algorithms.

Finally, the experimental results show that HDFC-ISVM* algorithm has better generalization performance and classification effects than other ISVM algorithms and can be correctly applied to image classification. In the image detection classification experiment, HDFC-ISVM* is compared with the relatively new convolutional neural network algorithm—AlexNet algorithm for image dataset "catsvsdogs." The results proved that HDFC-ISVM* algorithm has higher classification accuracy and classification effect than AlexNet algorithm.

HDFC-ISVM* incremental learning algorithm proposed in this paper has good classification accuracy and classification effect. However, it can be seen from Tables 14 and 15 that for datasets—"waveform," "spambase," and "credit," the classification accuracy of HDFC-ISVM* algorithm is better than other algorithms, but the overall classification accuracy is not very high. These datasets have features such as uneven positive and negative sample sizes to varying degrees and high-dimensional data, which may be the reason why most ISVM algorithms have low classification accuracy for these imbalanced datasets, especially for highly imbalanced datasets because in each round of incremental learning, the distribution of the training samples is very different from the distribution of the overall samples due to the extreme imbalance of these datasets. So, the accuracy of classifier trained by incremental learning algorithm is reduced. Therefore, for the incremental learning of the imbalanced datasets, especially those with large differences in the number of positive and negative samples, further research work can be carried out in the future. For example, we can consider assigning different weights to the forgetting factors of training samples in each round of incremental learning, especially considering the huge difference in the number of positive and negative samples. The optimization of appropriate forgetting factor updating strategy will make the training samples in each round of incremental learning have the same distribution characteristics as the samples in the original total dataset as far as possible, so as to improve the incremental learning effect of such imbalanced datasets. In addition, future research work can continue to explore the application of ISVM algorithm in image classification, outlier detection, and other fields.

Data Availability

The datasets used to support the findings of this study have been deposited in the UCI dataset, and they are available openly (the URL of the UCI dataset is https://archive.ics.uci.edu/ml/index.php). In addition, dataset "catsvsdogs" was provided by Kaggle (the URL of dataset "catsvsdogs" is https://www.kaggle.com/datasets).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Natural Science Foundation of Shaanxi Province (Project No. 2022JM-409) and the Key R&D Program in Shaanxi Province (Project No. 2021GY-084).

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
[2] C. J. C. Burges, "A Tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, Article ID 121167, 1998.
[3] S. Abe, *Support Vector Machines for Pattern Classification*, Springer Press, Beilin, Germany, 2005.
[4] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K Murthy, "Improvements to platt's SMO algorithm for SVM classifier design," *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001.
[5] Q. Wu, *Research on Extended Support Vector Machine Algorithm*, Science Press, Beijing, China, 2015.
[6] F. Zhu, *Research on Some Issues in Support Vector Machines*, Nanjing University of Science & Technology, Nanjing, China, 2019.
[7] B. B. Hazarika, D. Gupta, and P. Borah, "An intuitionistic fuzzy kernel ridge regression classifier for binary classification," *Applied Soft Computing*, vol. 112, no. 4, Article ID 107816, 2021.
[8] D. Gupta and U. Gupta, "On robust asymmetric Lagrangian ν-twin support vector regression using pinball loss function," *Applied Soft Computing*, vol. 102, no. 3, Article ID 107099, 2021.
[9] B. B. Hazarika and D. Gupta, "Density weighted twin support vector machines for binary class imbalance learning," *Neural Processing Letters*, vol. 54, no. 2, pp. 1091–1130, 2022.
[10] A. Soula, K. Tbarki, R. Ksantini, S. B. Said, and Z. Lachiri, "A novel incremental Kernel Nonparametric SVM model (iKN-SVM) for data classification: An application to face detection," *Engineering Applications of Artificial Intelligence*, vol. 89, pp. 103468.1–103468, 2020.
[11] T. L. Tang, *Research on Support Vector Machine Incremental Learning*, Zhejiang University of Technology, Hangzhou, China, 2018.
[12] C. Hou and Z. H. Zhou, "One-pass learning with incremental and decremental features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2776–2792, 2018.
[13] R. MelloA, M. R. Stemmer, and A. L. Koerich, "Incremental and decremental fuzzy bounded twin support vector machine," *Information Sciences*, vol. 526, pp. 20–38, 2020.
[14] R. Chitrakar and C. Huang, "Selection of Candidate Support Vectors in incremental SVM for network intrusion detection," *Computers & Security*, vol. 45, pp. 231–241, 2014.
[15] R. Xiao, J. C. Wang, and Z. X. Sun, "An approach to incremental SVM learning algorithm," *Journal of Nanjing University (Natural Science Edition)*, vol. 38, no. 2, pp. 152–157, 2002.
[16] W. J. Wang, "A Redundant Incremental Learning Algorithm for SVM," in *Proceedings of the International Conference on Machine Learning & Cybernetics*, pp. 734–738, Helsinki, Finland, June 2008.
[17] M. H. Yao, X. M. Lin, and X. B. Wang, "Fast incremental learning algorithm of SVM with locality sensitive hashing," *Computer Science*, vol. B11, pp. 88–91, 2017.
[18] X. Zhang, J. M. Zhao, H. Z. Teng, and G Liu, "A novel faults detection method for rolling bearing based on RCMDE and ISVM," *Journal of Vibroengineering*, vol. 21, no. 8, pp. 2148–2158, 2019.
[19] G. J. Chen, *Research on the Key Issues and Applications of Anomaly Detection Based on Support Vector Machines*, Taiyuan University of Technology, Taiyuan, China, 2016.
[20] Y. F. Li, B. Su, and G. S. Liu, "An incremental learning algorithm for SVM based on combined reserved set," *Journal of Shanghai Jiaotong University*, vol. 50, no. 7, pp. 1054–1059, 2016.
[21] L. Y. Li, *Research on Incremental Learning of Support Vector Machine Based on Robustness*, Wuhan Textile University, Wuhan, China, 2018.

[22] N. A. Syed, H. Liu, and K. K. Sung, "Handling concept drifts in incremental learning with support vector machines," in *Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 317–321, San Diego, CA, USA, August 1999.

[23] Y. L. Yang, J. Che, Y. Y. Li, Y. Zhao, and S. Zhu, "An incremental electric load forecasting model based on support vector regression," *Energy*, vol. 113, pp. 796–808, 2016.

[24] W. Y. Cheng and C. F. Juang, "An incremental support vector machine-trained TS-type fuzzy system for online classification problems," *Fuzzy Sets and Systems*, vol. 163, no. 1, pp. 24–44, 2011.

[25] W. H. Xie, G. Q. Liang, and P. C. Yuan, "Research on the incremental learning SVM algorithm based on the improved generalized KKT condition," *Journal of Physics: Conference Series*, vol. 1237, no. 2, Article ID 022150, 2019.

[26] J. F. Li and W. H. Xie, "Research of incremental learning algorithm for SVM based on class center diameter," *Journal of Physics: Conference Series*, vol. 1894, no. 1, Article ID 012074, 2021.

[27] N. Y. Deng and Y. J. Tian, *A New Method in Data Mining: Support Vector Machine*, Science and Technology Press, Beijing, China, 2004.

[28] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, 2012.