International Journal of Innovative Computing, Information and Control
Volume **22**, Number **1**, February **2026**

ICIC International ©2026 ISSN 1349-4198
pp. 175–192

COVARIANCE-BASED FEATURE WEIGHTED SUPPORT VECTOR MACHINE

Cong Yu[1], Hanming Liu[1,*], Zhaofa Liu[2], Pengcheng Hu[1], Qifeng Liu[1] and Taigui Shi[1]

[1]School of Mathematics and Computer Science
Gannan Normal University
Shida South Road, Rongjiang New District, Ganzhou 341000, P. R. China
jing@csysgz.com; { 2645522860; 3113703411 }@qq.com; m15851736589@163.com
*Corresponding author: liuhanming@gnnu.edu.cn

[2]Department of Nature Science and Computer
Ganzhou Teachers College
No. 35, Gannan Road, Rongjiang New District, Ganzhou 341000, P. R. China
963647895@qq.com

Received April 2025; revised August 2025

Abstract. *Feature weighting is an important technique to improve the performance of support vector machine (SVM), especially in high-dimensional and noisy data scenarios. However, existing feature weighted SVM methods – most of which rely on information-theoretic criteria – often suffer from some gaps such as high computational cost and lack of adaptability. Even they tend to perform poorly when handling noisy or redundant features. To address these limitations, this study proposes a covariance-based feature weighted SVM (CFWSVM), which can effectively evaluate the importance of each feature and improve the performance of SVM. To evaluate its effectiveness, CFWSVM was tested on 20 simulation and 15 real-world datasets, and compared with a traditional SVM and three feature weighted SVM variants (IGFWSVM, IGRFWSVM and MIFWSVM), as well as two recently published algorithms, 1D Convolutional Neural Network and Cost-Sensitive Random Forest. Experimental results show that CFWSVM achieved the highest accuracy and F1 score on most datasets, while maintaining strong AUC and runtime performance. Finally, CFWSVM was applied to gene expression classification tasks, where it also demonstrated outstanding performance. These findings indicate that CFWSVM can obtain a decision boundary with good linearity. The well adaptability, high classification accuracy and computational efficiency of CFWSVM make it a good candidate when facing complex, high-dimensional and noisy data.*

Keywords: Covariance, Feature weighted, Support vector machine, High-dimensional data, Noisy data

1. **Introduction.** A support vector machine (SVM) is a classical algorithm in the field of artificial intelligence (AI), which has undergone decades of continuous development and research since its introduction by Vapnik [1]. SVM is widely used across many fields due to its high interpretability and excellent performance in small-sample learning. However, the advent of AI has ushered in an era of high-dimensional data across scientific fields [2], including natural language processing [3], computer vision [4], machine learning [5] and bioinformatics [6]. These domains often encounter complex data characterized by numerous features but limited samples, presenting challenges for traditional SVM modeling. In the scenario of high-dimensional small sample data, a traditional SVM usually relies on dimensionality reduction or feature selection to reduce data complexity, or use sample

DOI: 10.24507/ijicic.22.01.175

weighted support vector machine (WSVM) [7,8] to enhance samples by assigning different weights to samples of each class. However, dimensionality reduction and feature selection methods suffer from drawbacks such as information loss, requiring manual intervention and lacking interpretability. Both traditional SVM and WSVM models assume implicitly that the contributions from all features towards classification are the same. In real world, however, different features may vary in their contribution when constructing the classification hyperplane, potentially compromising classification performance. To address the issue, one of the promising approaches is feature weighted. The earliest feature weighted support vector machine (FWSVM) model was proposed by Wang et al. [9]. The core framework of the model is to assign a weight to each feature to reflect its contribution to the classification. Initially, this model used information gain to weigh the features. Subsequently, more feature weighted algorithms were proposed, such as mutual information [10] and information gain ratio [11].

However, existing feature weighted methods for SVM, such as the information gain feature weighted support vector machine (IGFWSVM), mutual information feature weighted support vector machine (MIFWSVM) and information gain ratio feature weighted support vector machine (IGRFWSVM), are almost all based on information theory. IGFWSVM determines the feature weights by measuring the improvement in purity brought by the features. MIFWSVM quantifies the feature importance by the mutual information between the features and the class labels. And IGRFWSVM addresses the bias of IGFWSVM towards multi-valued features. Although these information-theoretic weighted methods are effective, they have several limitations. Firstly, information-theoretic methods may lack intuitiveness and involve high computational complexity. For example, concepts like information entropy [12] and mutual information [13], while mathematically sound, could be less intuitive for most users. These methods can indicate the degree of contribution to the classification but do not explicitly show whether a feature is positively or negatively correlated with the class labels. Secondly, these methods often require substantial computational resources, particularly when dealing with a large number of features. The highly computational complexity can lead to longer training times, which is especially for large-scale datasets. In contrast, statistical methods for feature weighting are usually easier to understand and may offer more straightforward interpretability. They can provide clearer insights into the positive or negative correlations of features with the class labels and typically demand fewer computational resources, making them more attractive for large-scale data.

To address the aforementioned challenges, this study introduces a novel feature weighted support vector machine by employing the covariance between the features and labels, termed CFWSVM. Two main steps are involved in CFWSVM model. One is to quantify the relationships between features and labels by a covariance matrix. The other is to weigh the features via the quantified relationships. This approach effectively captures the associations between the features and labels, thereby enhancing the accuracy of the feature contributions to the classification. The experimental results demonstrate that CFWSVM model exhibits significantly superior generalization performance compared to the traditional SVM model, particularly in datasets with more redundant features. This is approved by the validation experiments across 15 UCI and ASU datasets. Notably, in high-dimensional datasets, CFWSVM achieves an average accuracy improvement of up to 70.96% compared to the existing FWSVM algorithm. Furthermore, CFWSVM reduces the runtime by up to 65.22% compared to its predecessor, FWSVM algorithm.

The rest of this paper is organized as follows. Section 2 introduces the foundational FWSVM model, and Section 3 presents the proposed CFWSVM model. We report experimental results of the comparisons with existing methods on both simulated and real-world datasets in Section 4. Section 5 demonstrates the application of CFWSVM to gene expression datasets. Finally, in Section 6 the paper was concluded and the directions for future research were outlined.

2. **Feature Weighted Support Vector Machine.** SVM is a supervised learning model, and used for classification and regression. In classification, SVM maps the training data to a higher-dimensional space and finds the hyperplane that can separate the data from different classes as distinctly as possible. However, it overlooks the feature importance to the classification. To more comprehensively reflect the nature of the data, especially given that different features have varying importance in the real world, considering feature weights and extending SVM model to FWSVM becomes essential [9].

For a given training dataset $\boldsymbol{T} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_i, y_i), \ldots, (\boldsymbol{x}_m, y_m)\}$, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ij}, \ldots, x_{in}]^T$ (n is the number of features) and y_i are the ith sample and its label respectively, x_{ij} is the feature value of the jth feature in the ith sample. FWSVM multiplies the input matrix by the feature weighted diagonal matrix $\boldsymbol{P}$ of size $n \times n$, denoted as $\phi(\boldsymbol{X}, \boldsymbol{P})$. Then the optimal separating hyperplane can be obtained by solving the following optimization problem.

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{m} \xi_i \right) \\ &\text{s.t.} \begin{cases} y_i(\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i, \boldsymbol{P}) + b) \geq 1 - \xi_i \\ \xi_i \geq 0, \ (i = 1, 2, \ldots, m) \end{cases} \end{aligned} \tag{1}$$

where ξ_i is the error, and C is the penalty coefficient. This equation includes two purposes: a) finding the minimum 2-norm value of $\|\boldsymbol{w}\|^2$ to ensure the minimum margin, and b) minimizing $\sum_{i=1}^{m} \xi_i$ to reduce the classification error.

The Lagrangian of Equation (1) can be constructed via introducing the non-negative Lagrange multipliers α_i of sample $\boldsymbol{x}_i$ and the corresponding constraints μ_i of slack variables:

$$L_p(\boldsymbol{w}, \beta, \xi, \alpha, \mu) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i, \boldsymbol{P}) + b) - 1 + \xi_i \right] - \sum_{i=1}^{N} \mu_i \xi_i \tag{2}$$

In this way, we can transform the optimization problem into its dual problem, and then calculate the Karush-Kuhn-Tucker (KKT) conditions of the original problem.

$$\frac{\partial L_p(\boldsymbol{w}, b, \xi, \alpha, \mu)}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{i=1}^{N} \alpha_i y_i \phi(\boldsymbol{x}_i, \boldsymbol{P}) = 0 \tag{3}$$

$$\frac{\partial L_p(\boldsymbol{w}, b, \xi, \alpha, \mu)}{\partial b} = -\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{4}$$

$$\frac{\partial L_p(\boldsymbol{w}, b, \xi, \alpha, \mu)}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \tag{5}$$

$$\alpha_i \geq 0, \ \mu_i \geq 0 \tag{6}$$

$$\alpha_i \left[y_i(\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i, \boldsymbol{P}) + b) - 1 + \xi_i \right] = 0 \tag{7}$$

$$\mu_i \xi_i = 0 \tag{8}$$

$$y_i(\boldsymbol{w} \cdot \phi(\boldsymbol{x}_i, \boldsymbol{P}) + b) - 1 + \xi_i \geq 0, \ \xi_i \geq 0, \ i = 1, 2, \ldots, N \tag{9}$$

Thus, the original optimization problem (1) can be transformed into its dual problem

$$\begin{aligned}&\max_{\alpha}\left(\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\phi(\boldsymbol{x}_i,\boldsymbol{P})\phi(\boldsymbol{x}_j,\boldsymbol{P})\right)\\&\text{s.t.}\begin{cases}\sum_{i=1}^{m}\alpha_i y_i=0\\0\le\alpha_i\le C,\ (i=1,\ldots,m)\end{cases}\end{aligned}\tag{10}$$

The decision function of the algorithm is

$$f(\boldsymbol{x})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i\phi(\boldsymbol{x}_i,\boldsymbol{P})\cdot\phi(\boldsymbol{x},\boldsymbol{P})+b\right)\tag{11}$$

We introduce a feature weighting kernel function

$$K_P(\boldsymbol{x}_i,\boldsymbol{x}_j)=\phi(\boldsymbol{x}_i,\boldsymbol{P})\cdot\phi(\boldsymbol{x}_j,\boldsymbol{P})=K\left(\boldsymbol{x}_i^T\boldsymbol{P},\boldsymbol{x}_j^T\boldsymbol{P}\right)\tag{12}$$

Then, the non-linear classification decision function can be denoted as

$$f(\boldsymbol{x})=\text{sign}\left(\sum_{i=1}^{N}\alpha_i y_i K\left(\boldsymbol{x}_i^T\boldsymbol{P},\boldsymbol{x}^T\boldsymbol{P}\right)+b\right)\tag{13}$$

In this study, the most suitable Gaussian kernel function

$$\begin{aligned}K_P(\boldsymbol{x}_i,\boldsymbol{x}_j)&=\exp\left(-\gamma\frac{\|\boldsymbol{x}_i^T\boldsymbol{P}-\boldsymbol{x}_j^T\boldsymbol{P}\|^2}{2\sigma^2}\right)\\&=\exp\left(-\gamma\frac{(\boldsymbol{x}_i-\boldsymbol{x}_j)\boldsymbol{P}\boldsymbol{P}^T(\boldsymbol{x}_i-\boldsymbol{x}_j)^T}{2\sigma^2}\right),\quad(\gamma>0)\end{aligned}\tag{14}$$

is used.

3. **Covariance-Based Feature Weighted Support Vector Machine.** The matrix $\boldsymbol{P}$ influences significantly the performance and computational efficiency for a feature weighted method. Existing methods for acquiring $\boldsymbol{P}$ are primarily based on information theory, encompassing techniques such as information gain, mutual information and information gain ratio. These methods aim to assess the importance of features in classification by quantifying the information between the features and labels.

The information gain method draws on the information gain calculation idea of decision tree ID3. For the training dataset $\boldsymbol{T}$, it first calculates its information entropy $Ent(\boldsymbol{T})$, and then calculates the information gain of the feature set $\boldsymbol{F}=\{\boldsymbol{f}_1,\boldsymbol{f}_2,\ldots,\boldsymbol{f}_i,\ldots,\boldsymbol{f}_n\}$ to $\boldsymbol{T}$ to obtain the diagonal elements of $\boldsymbol{P}$:

$$p_{ii}=Gain(\boldsymbol{f}_i)=Ent(\boldsymbol{T})-\sum_{j\in Values\_f_i}\left(p\left(\boldsymbol{f}_i^{(j)}\right)*Ent\left(\boldsymbol{f}_i^{(j)}\right)\right),\quad i=1,\ldots,n\tag{15}$$

where $Values\_f_i$ represents all possible values of the feature $\boldsymbol{f}_i$, $p\left(\boldsymbol{f}_i^{(j)}\right)$ denotes the proportion of samples with the jth value of $\boldsymbol{f}_i$, and $Ent\left(\boldsymbol{f}_i^{(j)}\right)$ indicates the entropy associated with the jth value of $\boldsymbol{f}_i$.

The mutual information method [14] calculates the mutual information $Info(\boldsymbol{y};\boldsymbol{f}_i)$ between each feature $\boldsymbol{f}_i$ and the class label $\boldsymbol{y}$, which is used as the diagonal element of $\boldsymbol{P}$:

$$p_{ii}=Info(\boldsymbol{y};\boldsymbol{f}_i)=Ent(\boldsymbol{y})-Ent(\boldsymbol{y}|\boldsymbol{f}_i),\quad i=1,\ldots,n\tag{16}$$

where $Ent(\boldsymbol{y})$ is the information entropy of $\boldsymbol{y}$, and $Ent(\boldsymbol{y}|\boldsymbol{f}_i)$ is the conditional entropy of $\boldsymbol{y}$ given $\boldsymbol{f}_i$.

The information gain ratio feature weighted method [11], proposed by Huang et al. in 2023, is based on the principles of the C4.5 decision tree algorithm. The method calculates firstly the information gain $Gain(\boldsymbol{f}_i)$ of a feature $\boldsymbol{f}_i$, and then divides the gain by the intrinsic value [15] $Intrinsic\_Value(\boldsymbol{f}_i)$ of $\boldsymbol{f}_i$ to reduce the influence of the multi-valued features:

$$p_{ii} = IGR(\boldsymbol{f}_i) = Gain(\boldsymbol{f}_i)/Intrinsic\_Value(\boldsymbol{f}_i), \quad i = 1, \ldots, n \tag{17}$$

For a dataset with m samples, n features and k classes, according to Equations (15) to (17), the time complexities of the information gain and the information gain ratio method are $O(mnk)$, while that of the mutual information method is $O(m^2n)$.

A covariance is frequently employed in feature engineering [16] and data preprocessing [17], which aids in identifying feature subsets highly correlated with the target variable. By constructing the covariance matrix among features, it may discern inter-feature correlations better. Thus, it is possible to select important features and reduce dimensionality by the covariance among the features. In this study, we employed covariance analysis to assess the importance of features on classification, to avoid computing information entropy and then reduce computational complexity. Furthermore, the approach keeps away from the bias towards the features having large numbers of values.

For the training dataset $\boldsymbol{T}$, we define $\boldsymbol{f}_i = [x_{1i}, x_{2i}, \ldots, x_{mi}]$ as the values of all samples for the ith feature in the feature set $\boldsymbol{F}$. The covariance between $\boldsymbol{f}_i$ and the class label $\boldsymbol{y}$ is defined as follows:

$$\mathrm{cov}(\boldsymbol{f}_i, \boldsymbol{y}) = \frac{1}{m-1} \sum_{j=1}^{m} \left(x_{ji} - \overline{\boldsymbol{f}_i}\right)(y_j - \overline{\boldsymbol{y}}) \tag{18}$$

where $\overline{\boldsymbol{f}_i}$ is the mean of $\boldsymbol{f}_i$, and $\overline{\boldsymbol{y}}$ is the mean of $\boldsymbol{y}$. We utilize the covariance $\mathrm{cov}(\boldsymbol{f}_i, \boldsymbol{y})$ as the weighted factor to effectively capture the contribution of feature $\boldsymbol{f}_i$ to sample classification and serve as the measure of feature importance. A larger absolute value of the covariance indicates a stronger influence of the corresponding feature. Consequently, the feature weighted matrix $\boldsymbol{P}$ is constructed as a diagonal $n \times n$ matrix, as shown in Equation (19).

$$\boldsymbol{P} = \begin{bmatrix} \mathrm{cov}(\boldsymbol{f}_1, \boldsymbol{y}) & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \mathrm{cov}(\boldsymbol{f}_2, \boldsymbol{y}) & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \mathrm{cov}(\boldsymbol{f}_i, \boldsymbol{y}) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 0 & \cdots & \mathrm{cov}(\boldsymbol{f}_n, \boldsymbol{y}) \end{bmatrix} \tag{19}$$

It is not difficult to calculate that the time complexity of the covariance feature weighted matrix $\boldsymbol{P}$ is $O(mn)$ which is smaller than that of the existing information theory-based methods.

CFWSVM can explicitly capture the correlation between the features and class labels through the covariance-based weighting matrix $\boldsymbol{P}$ having lower runtime cost, reducing effectively the computational complexity and the inherent distributional bias in information-theoretic approaches. The pseudocode of CFWSVM is presented in Algorithm 1, which directly corresponds to the theoretical derivation discussed above.

4. **Experimental Results and Discussions.** To verify the effectiveness and efficiency of our method, two types of datasets of simulated and real datasets were employed to evaluate its performance by compared to four/six benchmark methods (see Sections 4.1 and 4.3). We firstly compared the methods in the simulation datasets, and then in the real

Algorithm 1: Covariance-based Feature Weighted Support Vector Machine

Input: Dataset $\boldsymbol{D} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$
Output: Optimal parameters, accuracies, support vectors, evaluation metrics
1: Normalize the features using z-score
2: Compute covariance between each feature and the labels
3: Construct feature weight vector $\boldsymbol{w}$
4: Apply element-wise multiplication: $\boldsymbol{X} \odot \boldsymbol{w}$
5: Perform grid search for (C, γ) using 10-fold CV to optimize the parameters of SVM
6: Train SVM with the optimized parameters
7: Evaluate accuracy and extract performance metrics

Table 1. Summary of ASU Dataset

Dataset	Sample	Feature	Label	Remark
Madelon	2600	500	2	NIPS 2003 Feature Selection Competition
Yale	165	1024	15	Face Image Data
ORL	400	1024	40	Face Image Data
COIL20	1440	1024	20	Face Image Data
Colon	62	2000	2	Biological Data
warpPIE10P	210	2420	10	Face Image Data
Lung	203	3312	5	Biological Data
Prostate	102	5966	2	Biological Data
SMK_CAN_187	87	19993	2	Biological Data

Table 2. Summary of UCI Dataset

Dataset	Sample	Feature	Label	Remark
Iris	150	4	3	Iris Classification
Crayo	90	6	2	Cryotherapy
Car	1728	6	4	Car Evaluation
Amphibians	189	15	2	Amphibian Classification
Wdbc	569	30	2	Breast Cancer Data
Parkinson	756	754	2	Parkinson's Detection

datasets. The real datasets include eight high-dimensional and one non-high-dimensional datasets from the ASU Feature Selection Database [18] (Table 1), as well as six non-high-dimensional datasets from the UCI Machine Learning Repository [19] (Table 2). This strategic selection aimed to examine our method's performance in both high- and non-high-dimensional scenarios. The fifteen datasets we employed cover a broad field, and are involved in several data types and classification tasks. See Section 4.2 for the generation of the simulation datasets.

4.1. **Experimental setting.** Four methods, namely SVM, IGFWSVM, MIFWSVM and IGRFWSVM, were employed here as benchmark methods in our experiments, where the last three methods are commonly used feature weighted methods. All methods were implemented by the Python package of Scikit-learn [20]. The experiments were conducted on Windows 11 Home Edition running on an Intel(R) Core (TM) i5-8300H CPU at 2.30 GHz. The datasets were partitioned into training and test sets with an $8 : 2$ ratio. For the reliability of the experimental findings, several different sets of SVM parameters of (C, γ) were set for the comparison by the optimal combination of (C, γ). That is, the regularization parameter C is $\{0.01, 0.1, 1, 10, 100, 1000, 10000, 100000\}$ and the kernel coefficient

γ is across $\{0.01, 0.1, 1, 10, 100\}$, allowing for a meticulous exploration of the parameter space. The metrics are the classification accuracy and algorithm runtime.

4.2. **Simulation experimental results and discussions.** We conducted a simulation experiment to firstly verify the effectiveness and robustness of our method. The ways of generating simulation data are inspired by [21], which allows for changing relation between features and labels to simulate real-world data under various conditions. Two sets of 20 simulation datasets were generated, one without redundant features and the other with. The procedure of generation for each pair of datasets is as follows.

For the dataset without redundant features, the number of classes is five, each comprising 100 instances. The degree of overlap among the classes was set to 1. Without loss of generality, the first three classes are labeled as 0, while the last two are assigned label 1. The number of features is 1000, where the first 10 are influential features and the others are noise features. The 10 influential features are independent each other. The data distribution of class c is normal, with the mean and covariance defined by Equations (20) and (21), respectively. Besides, two redundant features are added to the dataset as a simulation dataset with redundant features. The approach is to randomly select three features (named $\boldsymbol{f}_1^* \sim \boldsymbol{f}_3^*$ here) from the 10 influential features, and then to generate two redundant features by using equations $a\boldsymbol{f}_1^* + b\boldsymbol{f}_2^* + c$ ($a, b, c \in [0, 1)$ are random numbers with uniform distribution) and $e^{\boldsymbol{f}_3^*}$, respectively. Finally, after sampling, the training instances of 400 will be produced for each simulation dataset. A stratified sampling method was used here to reduce the sample bias. Thus, the numbers of samples labeled 0 and 1 are 240 and 160, respectively. The rest instances of 100 will be employed as test samples.

$$a_c = (\underbrace{ac, \ldots, ac}_{10}, \underbrace{0, \ldots, 0}_{990}) \tag{20}$$

$$\sum = \begin{pmatrix} \sum^*_{10\times10} & 0_{10\times990} \\ 0_{990\times10} & 1_{990\times990} \end{pmatrix} \tag{21}$$

where a is the degree of overlap, $\sum^*$ represents the covariance matrix with the diagonal elements of 1 and the off-diagonal elements of ρ. The ρ is the correlation coefficient among the first 10 features, and is set to 0 in this study.

Table 3 shows the results on the 20 simulation datasets, where 10 independent runs were conducted for each set.

Table 3. Simulation results*

Algorithm	Average accuracy	Average runtime (s)
SVM	94.5%/94.3%(3.4%/3.4%)	15.2/16.0(21.5%/21.2%)
IGFWSVM	95.4%/95.6%(2.5%/2.0%)	123.5/121.4(90.3%/89.6%)
MIFWSVM	97.2%/97.2%(0.5%/0.4%)	12.3/13.0(2.8%/3.2%)
IGRFWSVM	96.8%/96.7%(1.0%/0.9%)	17.4/17.9(31.4%/29.9%)
CFWSVM	**97.8%/97.6%**(–/–)	**12.0/12.6**(–/–)

*The percentages improvement of CFWSVM to the benchmarks are shown in the parentheses. The values on the left of "/" represent the results without redundant features, while the values on the right are the results with redundant features.

The results in Table 3 show that the average accuracy of each method with or without redundant features is almost the same, as well as the average runtime is also almost the same. This reflects the good performance of SVM in resisting redundant features. Furthermore, the average accuracy of SVM is as high as 94.5%/94.3%, meaning that SVM can efficiently remove pure noise features. However, all feature weighted methods have higher accuracy than the traditional SVM, indicating the effectiveness of such methods. Among them, CFWSVM has the best performance, achieving the highest accuracy and shortest runtime, with an average improvement of 3.4% and 21.4% compared to the traditional SVM, and an average improvement of 1.2% and 41.2% compared to the three weighted benchmarks. This highlights the effectiveness and robustness of the covariance weighted method of CFWSVM.

4.3. **Real data experimental results and discussions.** To further verify the effectiveness and robustness of our method, a real experiment was conducted on real datasets from ASU and UCI. Tables 4 and 5 delineate the average accuracy and runtime on 10 experimental repeats. These metrics offer a thorough assessment of the algorithm's performance regarding classification precision and efficiency. Further, two recently published

TABLE 4. Accuracy

	Dataset	SVM	IGFWSVM	MIFWSVM	IGRFWSVM	1D-CNN	CSRF	CFWSVM
High dimensionality	Yale	16.8%	5.6%	67.1%	66.8%	70.3%	63.8%	**71.5%**
	ORL	40.7%	0.0%	85.9%	91.1%	90.1%	84.4%	**95.4%**
	COIL20	91.1%	51.6%	94.9%	97.4%	97.0%	98.6%	**99.7%**
	Colon	59.2%	80.0%	76.9%	76.9%	76.9%	73.9%	**85.4%**
	warpPIE10P	91.9%	62.8%	98.1%	97.7%	**99.1%**	93.3%	98.6%
	Lung	89.5%	88.5%	88.5%	89.0%	**92.9%**	89.0%	90.5%
	Prostate	44.8%	69.1%	**91.9%**	88.6%	58.1%	91.4%	91.4%
	SMK_CAN_187	46.3%	47.4%	70.3%	66.6%	61.1%	60.8%	**71.3%**
Non-high dimensionality	Iris	93.7%	94.3%	95.0%	94.0%	93.3%	94.7%	**97.0%**
	Crayo	83.9%	89.4%	87.2%	84.4%	**92.2%**	83.9%	91.1%
	Car	**98.4%**	96.2%	95.7%	96.3%	96.4%	97.2%	98.0%
	Amphibians	72.4%	71.1%	72.4%	70.8%	63.9%	69.0%	**75.5%**
	Wdbc	98.0%	97.5%	97.1%	96.6%	97.0%	94.9%	**98.3%**
	Madelon	51.8%	66.1%	63.5%	61.1%	57.1%	66.1%	**72.6%**
	Parkinson	74.8%	88.8%	91.6%	88.6%	84.4%	83.8%	**94.1%**
	Average	70.2%	67.2%	85.1%	84.4%	82.0%	83.0%	**88.7%**
	Improvement*	26.3%	31.9%	4.3%	5.1%	8.0%	6.8%	–

*It indicates the percentage increase in average for CFWSVM to the benchmarks.

TABLE 5. Runtime (s)

	Dataset	SVM	IGFWSVM	MIFWSVM	IGRFWSVM	1D-CNN	CSRF	CFWSVM
High dimensionality	Yale	5.7	6.4	41.0	11.9	28.0	40.1	**4.2**
	ORL	30.4	28.1	151.6	26.2	62.4	241.3	**23.9**
	COIL20	283.5	299.5	294.9	258.1	212.6	395.5	**215.3**
	Colon	3.6	1.7	10.7	2.9	24.6	9.9	**1.3**
	warpPIE10P	18.5	18.7	36.5	31.2	81.1	83.8	**14.0**
	Lung	19.0	57.5	39.4	44.7	110.7	84.5	**11.7**
	Prostate	**5.8**	10.9	15.3	14.4	108.9	42.6	7.5
	SMK_CAN_187	71.1	94.0	102.2	95.7	676.6	557.7	**61.7**
Non-high dimensionality	Iris	1.0	**0.9**	1.0	2.8	2.4	5.1	1.1
	Crayo	**0.9**	1.0	1.1	2.2	6.4	9.3	**0.9**
	Car	65.9	24.1	38.4	26.8	**3.5**	8.2	36.0
	Amphibians	**1.8**	2.7	7.1	2.8	3.2	10.1	2.9
	Wdbc	5.5	5.0	3.7	**3.3**	5.3	13.4	4.5
	Madelon	469.6	491.9	844.6	881.5	**183.6**	1279.1	475.5
	Parkinson	**5.8**	10.9	15.3	14.4	100.3	123.0	7.5
	Average	65.9	70.2	106.8	94.6	107.3	193.6	**57.9**
	Improvement*	12.1%	17.6%	45.8%	38.8%	46.0%	70.1%	–

*It refers to the average reduction percentage of CFWSVM to the benchmarks.

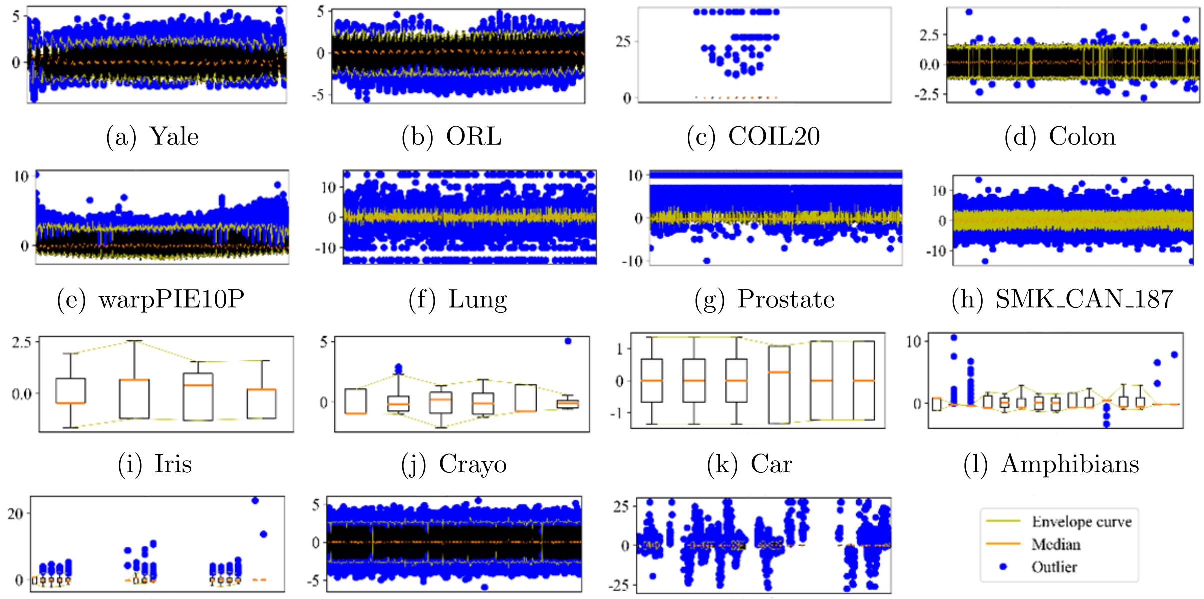
methods, 1D-CNN [22] in deep learning and Cost-Sensitive Random Forest (CSRF) [23] in machine learning, were employed as comparative models in this experimental evaluation.

Table 4 indicates that IGFWSVM achieves the lowest average accuracy, especially on Yale, ORL, COIL20, warpPIE10P and Lung are the lowest as well as Iris and Car are almost the lowest. On the seven datasets, except for Iris being slightly better than SVM, IGFWSVM's accuracies on the other six datasets are consistently lower than that of SVM. However, on the rest eight datasets, IGFWSVM has higher or almost the same accuracy as SVM. Note that those seven datasets are multi-classification problems, while the other eight are binary (Table 1). This implies that IGFWSVM may be more suitable for binary classifications. Furthermore, compared to SVM, a significant decline in accuracy is observed for IGFWSVM on face image datasets, particularly on ORL, where it shows an accuracy of 0. This is likely due to face images being characterized by simple grayscale with a small number of values. The information gain may favor the features with a large number of values [15], potentially leading IGFWSVM to insufficient gain and misleading weights on the face images. Conversely, the variant, IGRFWSVM, shows a marked improvement in accuracy compared to SVM, especially for face images, and less bias towards binary classifications. This is probably due to the information gain ratio's mitigation in the inherent preference of information gain. Additionally, MIFWSVM demonstrates comparable accuracy to IGRFWSVM, with a slight advantage, and surpasses SVM on most datasets. It means that MIFWSVM has a well improvement on SVM.

Notably, Table 4 demonstrates that the highest average accuracy is achieved by our method. Compared to SVM and IGFWSVM, CFWSVM shows a significant improvement. And, more improvements of CFWSVM can be seen on both the MIFWSVM and IGRFWSVM. In the 15 datasets, superior accuracy is consistently exhibited by CFWSVM compared to the benchmarks, except for Prostate and Car where CFWSVM is marginally inferior to the first one. Specifically, only on Car dataset, the accuracy of CFWSVM is slightly lower than that of SVM. The findings highlight the effectiveness of covariance-based feature weighted in enhancing the accuracy.

It is interesting that SVM obtains the highest accuracy on Car dataset. The reason may be that the distributions of the values of Car's features are almost the same (Figure 1(k)). The few differences in features may lead the weights to being a type of additional interference, potentially degrading the performance of a weighted method. As a comparison, Figure 1(i) shows that the Iris dataset has a larger difference in feature distributions, resulting in SVM producing lower accuracy than the four weighted methods. The outliers (or noise) are another factor being the influence on the performance of a weighted method. For example, in seven datasets, including Colon, Crayo, Prostate, SMK_CAN_187, Madelon and Parkinson (Figures 1(d), 1(j), 1(g), 1(h), 1(n), and 1(o)), all four weighted methods achieved higher accuracy than SVM. Even for the other six noisy datasets, except for only two in the Lung dataset, all other datasets have three weighted methods with higher accuracy than SVM. This indicates the robustness of weighted methods to noise.

By and large, our method has significantly better noise resistance than the three weighted benchmarks as well as SVM. On the 13 datasets with noisy features (excluding Figures 1(j) and 1(k)), the average accuracy of CFWSVM is 14.3% higher than that of the three benchmarks. The reason is that the average operation of the product of the covariance's deviations (Equation (18)) to some extent smooths the noise, thus reducing the impact of noise on weighting. However, covariance's smoothing may decrease the sensitivity to feature variances. For instance, as mentioned earlier, CFWSVM does not obtain the highest accuracy on Car. Moreover, on Prostate dataset, CFWSVM's accuracy is not the highest, which is 0.5 percentage points lower than MIFWSVM. This difference can be attributed to the severe skewed distributions of Prostate's features, where each feature deviates by



(a) Yale (b) ORL (c) COIL20 (d) Colon
(e) warpPIE10P (f) Lung (g) Prostate (h) SMK_CAN_187
(i) Iris (j) Crayo (k) Car (l) Amphibians
(m) Wdbc (n) Madelon (o) Parkinson

The horizontal and vertical axes represent the features and normalized feature values, respectively. The yellow curves mark the maximum and minimum envelope curves of the feature values. The "blank" features in Figures 1(c), 1(m) and 1(o) are resulted from their value ranges being almost zero, causing their box plots to degrade as points.

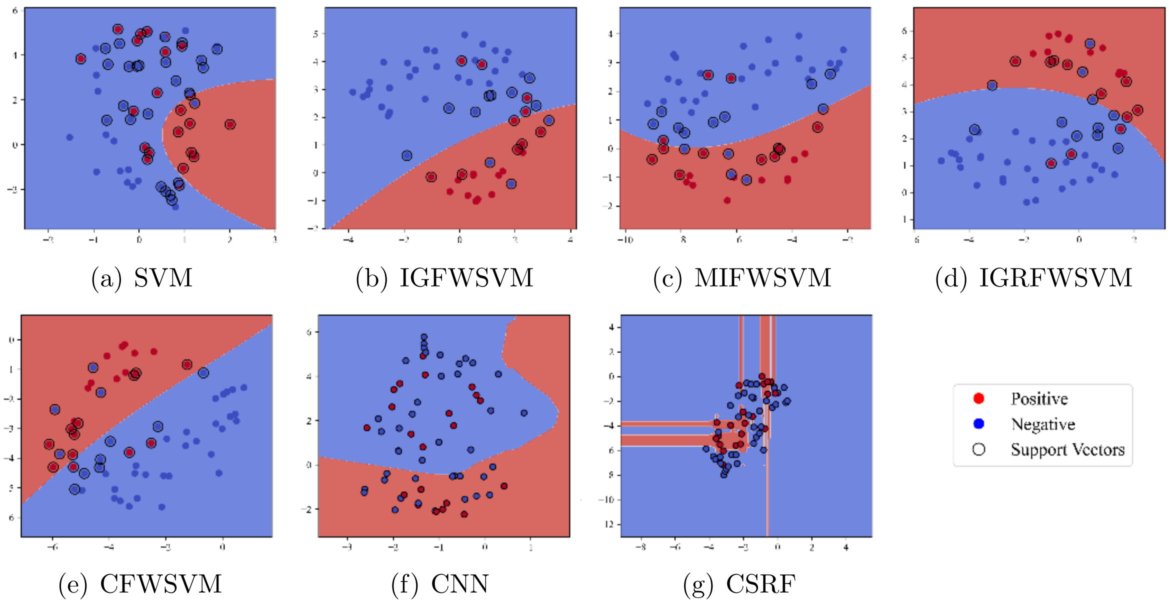
FIGURE 1. Distributions of feature values normalized by Z-score (box plot)

+10 from the median (Figure 1(g)). The skews make the positive deviations "drown" the negative ones when calculating the covariance for CFWSVM, thus reducing CFWSVM's ability to discern feature differences. Unlike CFWSVM, MIFWSVM is less affected by such deviations because it counts a feature value as 1, even if the value deviates significantly from the median. This approach provides a stable way for MIFWSVM to identify feature differences when facing severely skewed features.

For the two recently released algorithms, 1D-CNN and CSRF, only 1D-CNN achieved slightly higher accuracy than CFWSVM in three datasets and ranked the first. And, their average accuracies are all lower than the weighted methods except for IGFWSVM. Especially, they are on average 6.2 percentage points lower than our method. These reflect that although the deep learning and novel machine learning methods have higher algorithm complexity (as shown in Table 5, they have the highest runtime), they cannot guarantee better classification accuracy than feature weighted methods.

The comparative experiments above suggest that CFWSVM obtains the highest average accuracy and demonstrates good robustness. However, runtime is another important consideration for assessing method performance, especially for large-scale datasets or real-time applications. Table 5 shows that compared to SVM in terms of runtime, IGFWSVM, MIFWSVM, IGRFWSVM, 1D-CNN and CSRF improved 33.3%, 13.3%, 33.3%, 26.7% and 6.7% of the datasets, respectively, while CFWSVM improved up to 60%. In terms of average runtime, only CFWSVM shows an improvement of 12.1% compared to SVM, while the other weighted methods all deteriorates, especially MIFWSVM, which decreased by as much as 62.2%. Compared to the weighted methods, the two new algorithms decreased more, with CSRF even decreasing by nearly 200%. The superior runtime of CFWSVM is due to its ability to accurately determine feature weights, which speeds up the algorithm convergence and then compensates for the additional time used for weighting. In contrast, the other weighted methods fail to produce optimal weights, resulting in negligible improvements in convergence speed. This issue is especially severe for MIFWSVM, where the high computational complexity of mutual information leads to the most significant reduction in speed. The reason for the longer runtime of the two new methods is their higher algorithm complexities.

The experimental results on several datasets discussed earlier, show that the weighted methods demonstrated significantly differences in performance when processing data characterized by noise and uneven distribution of the feature values. These disparities are manifested not only in accuracy, but also in the training durations and computational complexities of the models. Such discussion is instrumental for us to understand the efficiency and utility of the methods in real-world scenarios. To clear the decision-making processes of the models, the decision boundary (Figure 2) was examined for each method by employing Colon dataset as a representative.



(a) SVM (b) IGFWSVM (c) MIFWSVM (d) IGRFWSVM

(e) CFWSVM (f) CNN (g) CSRF

FIGURE 2. Decision boundary on Colon

The classification performance of a model is closely related to the position and contour of its decision boundary. More complex models are usually more susceptible to overfitting when compared to their more parsimonious counterparts [24]. A linear decision boundary provides more fault tolerance to training samples, and thus improving the model's generalization. As depicted in Figure 2, the closest linear decision boundary is CFWSVM (Figure 2(e)). It is approved by Table 4, which shows CFWSVM demonstrating the highest accuracy Colon dataset. Following CFWSVM, IGFWSVM exhibits a commendable linearity (Figure 2(b)). Its accuracy on the dataset ranks the second in Table 4. Similarly, SVM has the lowest accuracy in Table 4, and shows the highest curvature of the decision boundary (Figure 2(a)). The two new methods, however, gave the most complex boundaries, which suggest that they may be trapped into an over-fitting easily.

The experimental results demonstrate that, overall, CFWSVM surpasses the benchmarks in accuracy, runtime and decision boundary linearity. The experimental results and the box plots of the feature values indicate that the distribution of feature values impacts method performance significantly. For the datasets with few feature differences, even value distributions, low noise and low dimensionality, a traditional SVM model is preferable for its simplicity and efficiency. In these cases, SVM can effectively capture the essential characteristics of the data without an additional weighting strategy. Conversely, for the high feature dimensionality, which upsurges the probability of redundant features, or the high noise level, traditional SVM model may meet a large challenge. In such scenarios, a weighted method, particularly the covariance-based weighted method, demonstrate its advantages. By assigning different weights to features accurately, CFWSVM may more effectively reduce the influence of noisy features and emphasize crucial ones for classification.

The decision boundaries highlight a significant advantage of covariance-based weighting is that, it could capture the relations between features and labels better, and thus delineate the intrinsic data structure more effectively. This is especially critical for the methods tasked with finding the optimal decision boundary in high-dimensional spaces, such as SVM. A feature weighted method may focus well on the contribution to the model for each feature. This makes the model emphasize further the most influential features, consequently enhancing the model performance. As shown in Figure 2, a covariance-based feature weighted method enables SVM to more accurately identify the difference in classes, reducing the misclassifications and thus exhibiting a more linear decision boundary. This may be an important reason why CFWSVM achieves high classification accuracy and low runtime overall, especially on high-dimensional datasets.

4.4. **Evaluation based on more metrics.** To further evaluate the performance of CFWSVM, we employed precision, recall, F1 score and AUC as the additional metrics. The average values of these metrics are shown in Table 6.

Table 6. Evaluation results of more metrics

Metrics	SVM	IGFWSVM	IGRFWSVM	MIFWSVM	1D-CNN	CSRF	CFWSVM
Precision	76.4%	71.5%	86.8%	85.5%	80.2%	74.9%	**89.9%**
Recall	74.9%	69.8%	84.9%	83.9%	79.4%	73.4%	**89.1%**
F1 score	73.5%	69.0%	84.7%	83.7%	78.6%	73.0%	**88.8%**
AUC	0.905	0.878	0.920	0.906	0.922	0.913	**0.959**
Improvement*	15.4%	22.0%	4.4%	5.8%	10.1%	16.3%	–

*It indicates the percentage increase in average for CFWSVM to the benchmarks.

Table 6 shows that CFWSVM ranks first in all the four metrics and is significantly better than the second, which suggests it has well classification performance further. Its highest F1 score and AUC value are particularly commendable. The highest F1 score indicates that CFWSVM can not only correctly identify more positive instances (high recall) but also ensure that most of the identified instances are indeed positive (high precision). The highest AUC of CFWSVM reflects its excellent discriminative ability in classification, revealing that it can achieve high true positive rates while reducing false positive rates.

4.5. **Discussions on the weighted methods.** To better understand the empirical performance observed in Sections 4.2 and 4.3, we provide a theoretical discussion on the advantages and limitations of the proposed covariance-based feature weighted method, in comparison with traditional information-theoretic approaches such as IGFWSVM, MIFWSVM and IGRFWSVM used as benchmarks in this study.

1) It has higher computational efficiency. The time complexity of the covariance-based method is well reduced compared to the traditional methods. On the other hand, this technique employs only the simple four arithmetic operations, while the traditional methods involve logarithmic operations after the probabilities calculated by counting the values

for each feature. Furthermore, in the case of mutual information, the joint probability distributions can escalate computational cost further.

2) It has no bias towards the features having large numbers of values. The traditional methods need count the values for each feature. This is almost inevitably influenced by the features having large numbers of values.

3) It has better sensitivity. Information theory-based methods calculate the information entropy by the counts of future values, which might not accurately respond to the changes in feature values on class labels. Conversely, covariance-based methods measure the co-variability in feature values and labels, offering a more precise depiction of a feature's influence on the classification. Usually, when the value of a feature with higher importance changes, the change in its label value will also be more pronounced. As depicted in Equation (6), the product of the collaborative changes between feature values and labels enlarges the importance of features. This makes covariance a more responsive measure for assessing the contribution of features to the classification.

4) It has higher robustness. For a training dataset with noise, as we all know, its class labels are unaffected by the noise. Information theory-based methods use frequency counts like a histogram, and can smooth the noise well when the noise is small. Covariance-based methods account for the co-variation between the features and labels; thus, it will lose the enlarging nature mentioned early since the labels do not co-vary with the noise. That is, our method could reduce the influence of noise. However, when the noise is big enough to cause a change in the histogram, an information theory-based method might produce a skewed model. Of course, in that case, a covariance method may be affected by the noise, too. In fact, the noise not only causes $\boldsymbol{f}_i$ to change, but also synchronizes $\overline{\boldsymbol{f}_i}$, so that the difference between them can still be limited to a small range. That is, the covariance weighted methods can obtain a better classification model than the information theory-based methods in the case of big noise. It is worth pointing out that although theoretically, white noise does not affect $\overline{\boldsymbol{f}_i}$ when the dataset has enough samples, the real sample size is far from this, causing $\overline{\boldsymbol{f}_i}$ to change synchronously with $\boldsymbol{f}_i$.

5) It might require more memory. The information theory-based methods primarily require the memory of histograms, while covariance-based methods entail the retention of the two entire vectors and their means, which lead the latter to more memory cost than that of the former, especially for large vectors.

5. Real-World Applications.

5.1. Datasets.

To evaluate the performance of the proposed method, we applied it in some real-world datasets obtained from the Gene Expression Omnibus (GEO) [25] database hosted by NCBI (National Center for Biotechnology Information). The identifications of the datasets are GSE33459 [26], GDS2001 [27], GDS3001 [28] and GDS4999 [29], respectively. These datasets covered a variety of biological and clinical contexts. GSE33459 contains gene expression profiles related to Citrus Greening Disease (HLB). GDS2001 is a gene expression profile of skeletal muscles of utrophin/dystrophin double knockout (DKO) mutants and dystrophin-deficient mdx mutants. GDS3001 is another expression profile of c-Jun N-terminal kinase (JNK) 1 depleted livers of diet-induced obese (DIO) animals. And, GDS4999 is the expression profile of the lung from B6129SF2/J animals depleted for very low-density lipoprotein receptor (VLDLR) and subjected to intranasal challenge with house dust mite (HDM).

Those datasets represent several biological scenarios, from disease diagnostics to functional gene studies, providing a robust basis for verifying our method. By leveraging the datasets, it may demonstrate the generalizability and effectiveness of the method in real-world problems.

5.2. **Data preprocessing.** Since the limited sample size of the datasets, for the samples with missing values, the missing values were filled with zeros instead of removing the samples. Similarly, the datasets were not split by a fixed ratio for training and testing. Instead, an appropriate number of samples from each class were randomly chosen as the testing set while the others as the training set. The partitioning results are presented in Table 7.

Table 7. GEO datasets split

ID	Training samples	Testing samples	Feature	Labels
GDS2001	12	6	12488	6
GDS3001	9	5	22690	3
GDS4999	12	4	35557	4
GSE33459	6	4	30395	2

5.3. **Results and discussions.** The CFWSVM and the benchmarks were applied to the classification tasks on the real-world datasets. Their accuracy and training time are shown in Table 8.

Table 8. Real-world application results (Accuracy/Training time)

Dataset	SVM	IGFWSVM	MIFWSVM	IGRFWSVM	1D-CNN	CSRF	CFWSVM
GDS2001	33.3%/ 10.1ms	33.3%/ 10.9ms	33.3%/ 6.3ms	33.3%/ 9.6ms	66.7%/ 4.9s	33.3%/ 0.4s	**83.3%/ 2.0ms**
GDS3001	60.0%/ 153.9ms	60%/ 15.0ms	60%/ 9.0ms	60%/ 13.0ms	60.0%/ 10.3s	40.0%/ 0.3s	**80%/ 2.0ms**
GDS4999	75%/ 19.9ms	75%/ 18.0ms	75%/ 14.0ms	75%/ 17.9ms	75%/ 19.1s	50%/ 0.5s	**100%/ 2.0ms**
GSE33459	50%/ 11.0ms	50%/ 18.7ms	50%/ 7.0ms	50%/ 12.5ms	50%/ 8.1s	50%/ 0.3s	**75%/ 1.6ms**

Table 8 illustrates that CFWSVM outperforms significantly the benchmarks in terms of both accuracy and computational efficiency. Compared to the weighted methods, its accuracy increased by an average of 30 percentage points, while the training time decreased by an average of 19.8ms. For the two new algorithms, the two numbers are 31.5% and 5485.6ms, respectively. The notable enhancement of the accuracy suggests that CFWSVM can adeptly discern the important features in noisy. Furthermore, the training time of CFWSVM across all datasets remains within 2.0ms, which outpaces significantly the benchmarks and demonstrates extremely high computational efficiency, reflecting the significant optimization of CFWSVM on running time.

By and large, CFWSVM not only improves accuracy significantly but also reduces computational cost markedly. Its comprehensive dominance in classification accuracy and running time demonstrate its effectiveness.

6. **Conclusions.** In this study, a covariance-based feature weighted approach was introduced to SVM and, a new method, CFWSVM, was proposed based on the approach to enhance SVM performance. CFWSVM can effectively and efficiently quantify the correlation between features and labels by employing covariances as the weighting matrix, which

makes the model more focused on the features having pivotal influence on the classification, thus leading to a marked enhancement in SVM's performance. In the simulation experiment, CFWSVM achieved the highest accuracy and shortest runtime, with an average improvement of 3.4% and 21.4% compared to a traditional SVM, and an average improvement of 1.2% and 41.2% compared to the three weighted benchmarks. Besides, the experimental results on real datasets demonstrate that CFWSVM exhibits good performance in high-dimensional, non-high-dimensional and noisy data. Its average accuracy and runtime have improved by 26.3% and 12.1%, respectively compared to a traditional SVM, and by an average of 13.8% and 34.1%, respectively compared to the three weighted benchmarks. Furthermore, in the real-world applications, CFWSVM is significantly superior to the benchmark methods in both accuracy and algorithm efficiency. Comparing the results of simulation and real data experiments as well as the applications, it can be seen that CFWSVM improves significantly on the real data with noisy features compared to the simulation data without noisy features. These results reflect that the covariance-based feature weighted method can better capture the correlation between features and labels (especially for noisy features), thereby improving the accuracy and robustness of the model, highlighting the value of this feature weighted in enhancing SVM performance.

The no-free-lunch theorem [30] for problem-solving algorithms posits that no single method is optimal for all scenarios, each approach has its ideal conditions and merits. Compared to the existing methods, CFWSVM has better adaptability. Its high classification accuracy and computational efficiency make it an excellent candidate when facing complex, high-dimensional and noisy data. In real-world applications, CFWSVM is a recommended and worthwhile alternative when data characteristics are complex and difficult to be preprocessed.

However, CFWSVM focuses only on the correlation between features and labels, and as the existing weighted methods, it does not consider the redundancy of features. This has been validated in the simulation experiment, where all methods show a decrease in performance on the datasets with redundant features. This limitation presents an opportunity for future research efforts.

Acknowledgement. This study was supported by the 14th Five Year Plan for Education Science in Jiangxi Province (21YB169) and the National Natural Science Foundation of China (Grant No: 31660321) awarded to Hanming Liu.

REFERENCES

[1] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer Science & Business Media, 1999.

[2] S. Karami, F. Saberi-Movahed, P. Tiwari et al., Unsupervised feature selection based on variance-covariance subspace distance, *Neural Networks*, 2023.

[3] A. Rogers, M. Gardner and I. Augenstein, QA dataset explosion: A taxonomy of NLP resources for question answering and reading comprehension, *ACM Computing Surveys*, vol.55, no.10, pp.1-45, 2023.

[4] C.-Y. Wang, A. Bochkovskiy and H.-Y. M. Liao, YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.7464-7475, 2023.

[5] N. Kanvinde, A. Gupta, R. Joshi et al., Binary classification for high dimensional data using supervised non-parametric ensemble method, *Intelligent Computing and Networking: Proc. of IC-ICN 2022*, pp.15-23, 2023.

[6] R. Bailey, A. Sarkar, A. Singh et al., Optimal supervised reduction of high dimensional transcription data, *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol.20, no.5, pp.3093-3105, 2023.

[7] X. Zhang, X.-L. Xiao and G.-Y. Xu, Fuzzy support vector machine based on affinities among samples, *Journal of Software*, no.5, pp.951-958, 2006.

[8] X.-W. Fan, S.-X. Du and T.-J. Wu, Weighted support vector machine based classification algorithm for uneven class size problems, *Journal of Image and Graphics*, no.9, pp.70-75, 2003.
[9] T.-H. Wang, S.-F. Tian and H.-K. Huang, Feature weighted support vector machine, *Journal of Electronics & Information Technology*, vol.31, no.3, pp.514-518, 2009.
[10] Y. Chen and Y. Hao, A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction, *Expert Systems with Applications*, vol.80, pp.340-355, 2017.
[11] C. Huang, J. Zhou, J. Chen et al., A feature weighted support vector machine and artificial neural network algorithm for academic course performance prediction, *Neural Computing and Applications*, vol.35, no.16, pp.11517-11529, 2023.
[12] W. Duch, T. Wieczorek, J. Biesiada et al., Comparison of feature ranking methods based on information entropy, *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541)*, vol.2, pp.1415-1419, 2004.
[13] L. Paninski, Estimation of entropy and mutual information, *Neural Computation*, vol.15, no.6, pp.1191-1253, 2003.
[14] H.-J. Xing, M.-H. Ha, B.-G. Hu et al., Linear feature-weighted support vector machine, *Fuzzy Information and Engineering*, vol.1, no.3, pp.289-305, 2009.
[15] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., 1992.
[16] J. Hu, H. Chen, A. A. Heidari et al., Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection, *Knowledge-Based Systems*, vol.213, 106684, 2021.
[17] G. Liu, W. Jia, M. Wang et al., Predicting cervical hyperextension injury: A covariance guided sine cosine support vector machine, *IEEE Access*, vol.8, pp.46895-46908, 2020.
[18] J. Li, K. Cheng, S. Wang et al., Feature selection: A data perspective, *ACM Computing Surveys*, vol.50, no.6, pp.1-45, 2017.
[19] A. Asuncion and D. Newman, *UCI Machine Learning Repository*, https://archive.ics.uci.edu/, Accessed in October 8, 2023.
[20] F. Pedregosa, G. Varoquaux, A. Gramfort et al., Scikit-learn: Machine learning in Python, *The Journal of Machine Learning Research*, vol.12, pp.2825-2830, 2011.
[21] K. Dai, *Research on the Feature Selection Techniques Based on SVM for Network Data*, Ph.D. Thesis, PLA Information Engineering University, 2013.
[22] A. A. Ahmed, W. Ali, T. A. Abdullah et al., Classifying cardiac arrhythmia from ECG signal using 1D CNN deep learning model, *Mathematics*, vol.11, no.3, 562, 2023.
[23] X. Zheng and L. Huang, Unbalanced big data classification based on improved random forest algorithm, *International Journal of Innovative Computing, Information and Control*, vol.20, no.2, pp.575-590, 2024.
[24] J. Zahálka and F. Železný, An experimental test of Occam's razor in classification, *Machine Learning*, vol.82, pp.475-481, 2011.
[25] T. Barrett, S. E. Wilhite, P. Ledoux et al., NCBI GEO: Archive for functional genomics data sets – Update, *Nucleic Acids Research*, vol.41, no.D1, pp.D991-D995, 2012.
[26] U. Albrecht and K. D. Bowman, Gene expression in Citrus sinensis (L.) Osbeck following infection with the bacterial pathogen Candidatus Liberibacter asiaticus causing Huanglongbing in Florida, *Plant Science*, vol.175, no.3, pp.291-306, 2008.
[27] P. E. Baker, J. A. Kearney, B. Gong et al., Analysis of gene expression differences between utrophin/dystrophin-deficient vs mdx skeletal muscles reveals a specific upregulation of slow muscle genes in limb muscles, *Neurogenetics*, vol.7, pp.81-91, 2006.
[28] R. Yang, D. M. Wilcox, D. L. Haasch et al., Liver-specific knockdown of JNK1 up-regulates proliferator-activated receptor γ coactivator 1β and increases plasma triglyceride despite reduced glucose and insulin levels in diet-induced obese mice, *Journal of Biological Chemistry*, vol.282, no.31, pp.22765-22774, 2007.
[29] K. Fredriksson, A. Mishra, J. K. Lam, E. M. Mushaben et al., The very low density lipoprotein receptor attenuates house dust mite-induced airway inflammation by suppressing dendritic cell-mediated adaptive immune responses, *Journal of Immunology*, vol.192, no.10, pp.4497-4509, 2014.
[30] T. F. Sterkenburg and P. D. Grünwald, The no-free-lunch theorems of supervised learning, *Synthese*, vol.199, no.3, pp.9979-10015, 2021.

Author Biography

Cong Yu received the B.Eng. degree in Software Engineering and the M.Eng. degree in Control Science and Engineering from Gannan Normal University, China, in 2022 and 2025, respectively. He is currently a teaching assistant at Changsha Preschool Education College, China. His research interests include machine learning and swarm intelligence algorithms.

Hanming Liu received his Ph.D. degree in Biomedical Engineering from the University of Electronic Science and Technology of China, in 2015. He is currently a professor at the School of Mathematics and Computer Science, Gannan Normal University, China. His research interests include bioinformatics, machine learning, and swarm intelligence algorithms.

Zhaofa Liu received the M.Eng. degree in Electronic Information from Gannan Normal University, China, in 2021. He is currently a lecturer at Ganzhou Teachers College, China. His research interests include bioinformatics and swarm intelligence algorithms.

Pengcheng Hu received the M.Eng. degree in Electronic Information from Gannan Normal University, China, in 2025. He is currently a teaching assistant at School of Artificial Intelligence, Dongguan City University, China. His research interests include machine learning and deep learning.

Qifeng Liu received the Bachelor's degree in Data Science and Big Data Technology from Gannan Normal University, China, in 2023. He is currently an M.Eng. degree candidate in Electronic Information at Gannan Normal University, China. His research interests include bioinformatics, machine learning, and swarm intelligence algorithms.

Taigui Shi received the Bachelor's degree in Computer Science and Technology from Yuchun University, China, in 2023. He is currently an M.Eng. degree candidate in Electronic Information at Gannan Normal University, China. His research interests include bioinformatics, machine learning, and swarm intelligence algorithms.