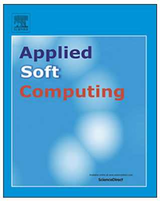
Applied Soft Computing Journal 75 (2019) 702–711

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc



Integration of feature vector selection and support vector machine for classification of imbalanced data

Jie Liu [a], Enrico Zio [b,c,d,*]

[a] School of Reliability and System Engineering in Beihang University, 37 Xueyuan Road, Haidian, Beijing, China
[b] Energy Department, Politecnico di Milano, Milano, Italy
[c] MINES ParisTech / PSL Université Paris, Centre de Recherche sur les Risques et les Crises (CRC), France
[d] Eminent Scholar, Department of Nuclear Engineering, Kyung Hee University, Republic of Korea

HIGHLIGHTS

- In this paper, classification of imbalanced data is considered.
- A feature vector selection method with respect to maximal separability is proposed.
- The decision threshold is optimized considering a specific accuracy metric.
- Experiments on 26 public datasets are considered.
- Comparisons with various kernel methods show the effectiveness of the proposed method.

ARTICLE INFO

Article history:
Received 10 October 2017
Received in revised form 6 September 2018
Accepted 26 November 2018
Available online 6 December 2018

Keywords:
Classification
Feature Vector Selection
Imbalanced data
Support Vector Machine
Separability

ABSTRACT

Support Vector Machine (SVM) has been widely developed for tackling classification problems. Imbalanced data exist in many practical classification problems where the minority class is usually the one of interest. Undersampling is a popular solution for such problems. However, it has the risk of losing useful information in the original data. At the same time, tuning the hyperparameters in SVM is also challenging. By analyzing the geometrical meaning of kernel methods, an approach is proposed in this paper that combines a modified Feature Vector Selection (FVS) method with maximal between-class separability and an easy-tuning version of SVM, i.e. Feature Vector Regression (FVR) proposed in our previous work. In this paper, the modified FVS method selects a small number of data points that can represent linearly all the dataset in the Reproducing Kernel Hilbert Space (RKHS) and the selected data points give also a maximal separability of the imbalanced data in RKHS. The FVR model is also solved analytically, as in least-squared SVM. The decision threshold for classification is optimized to maximize the predefined accuracy metric. Twenty-six imbalanced datasets are considered and comparisons are carried out with several SVM-based methods for imbalanced data. Statistical test shows the effectiveness of the proposed method.

1. Introduction

Classification of imbalanced data has drawn much attention in recent decades [1–4]. Imbalanced data contains one or several classes whose sizes are significantly smaller than the other classes [5]. It differs from the classification of balanced data in a way that the data in one class (named majority class) can be much greater than that in another class (named minority class) [6]. This occurs in many practical problems and the practitioners usually are more interested in the minority class, e.g. banking fraud detection [7], medical diagnosis [8], bioinformatics [9], etc. The methods proposed for tackling imbalanced data classification problems can be divided into three categories: data-level methods, algorithm-level methods and fusion methods [10–12].

In this paper, we focus on the binary classification of imbalanced data with a single Support Vector Machine (SVM) model. Outliers and noise are not specifically considered in the model. An original hybrid approach combining an undersampling method (i.e. FVS) and a modified least squared SVM is reported in this work.

Feature Vector Selection method (FVS) proposed by Baudat and Anouar [13] selects the Feature Vectors (FVs) in RKHS such that the other data points in RKHS can be expressed as a linear combination of the selected FVs. Geometrically, selected FVs form an oblique coordinate system in RKHS and their linear combination can represent any other data in the RKHS. Embedded in the theory of kernel methods, FVS has been used for pattern recognition [14], classification [15], unsupervised feature extraction [16], etc. These works and our previous work in [17] all show that FVS can

[*] Corresponding author at: Energy Department, Politecnico di Milano, Milano, Italy.
E-mail address: enrico.zio@polimi.it (E. Zio).

https://doi.org/10.1016/j.asoc.2018.11.045
1568-4946/© 2018 Elsevier B.V. All rights reserved.

significantly reduce the training data size and yet guarantee the performance of SVM, as the information in the data are (nearly) all stored in the FVs. For imbalance data, FVS can be modified for undersampling the data (both in majority and minority classes) in a way of forward selection without losing much of the information in the original data. To the authors' knowledge, this is the first time that FVS is modified for imbalanced data.

The separability of the data in RKHS with undersampling is very important. Using FVS, each data point in RKHS can be projected onto the oblique coordinate system formed by the selected FVs. For the same dataset, different coordinate systems can be constructed with FVS. Fisher ratio [18] is used in this paper to measure the separability of the original dataset in different coordinate systems formed by the FVs. The FVs of the coordinate system with the largest separability on the original dataset are selected to train a modified version of a least squared SVM model based on FVS, i.e. Feature Vector Regression (FVR) [17].

In FVR, the estimate function is a kernel expansion on the selected FVs and the objective of the optimization is to minimize the prediction error on the whole training dataset, which reduces the chance of overfitting on the FVs [17]. Similar to least-squared SVM, equal constraints are used in the optimization, which keep all the FVs effective in the estimate function. In comparison with classic SVM, there is only one hyperparameter in FVR, which can be determined analytically. This model is adapted for classification, in this paper. The estimate function of FVR is calculated analytically, similar to that in least-squared SVM. The training of the model is, thus, much simplified.

In order to improve the accuracy of the proposed approach, an optimization process, which is similar to Yu et al. [19], is integrated to find the best decision threshold value. The difference is that the objective is to maximize the sum of two accuracy measures, i.e. F-measure and G-mean, instead of maximizing only G-mean as in [19]. This is done because F-measure and G-mean are not always coherent in the experiment results of the previously published papers and the accuracy on the minority class is relatively more important.

Experiments are carried out on 26 imbalanced datasets from Keel dataset Repository [20] to verify the effectiveness of the proposed approach. Comparisons with the several benchmark methods show the superiority of the proposed approach. F-measure, G-mean, Area under the receiver operating characteristics curve are the most popular performance metrics for imbalanced data classification model performance [21]. Several other metrics have also been proposed to improve the flexibility and avoid the disadvantages of the previous ones, such as index of balanced accuracy, class-weighted accuracy, etc. [22,23]. However, there has been no metric accepted by all researchers. In this wok, two of the most widely used performance metrics (i.e. F-measure and G-mean) are used in the experiments for comparing different classification models.

The remaining of the paper is structured as below. Section 2 reviews the SVM methods and accuracy measures for imbalanced data classification. Section 3 presents the model construction for classification of imbalanced data. Experimental results and comparisons are shown in Section 4. Conclusions and remarks are given in Section 5.

2. Review on the SVM methods for imbalanced data and the performance metrics

This section reviews the relevant state-of-art research on imbalanced data classification with SVM and classification model performance metrics. The motivations of our work is presented based on the analysis of the current results.

2.1. SVM methods for imbalanced data

SVM [24] is a powerful and simple data-driven method, which has been widely developed and applied for classification of imbalanced data. SVM handles nonlinear classification problems by mapping the data into a high dimensional feature space (the RKHS), where the classification becomes linear. The classification hyperplane in SVM is only dependent on a few data points (the so called support vectors) and, thus, its robustness is stronger than many other machine learning methods.

Many approaches have been proposed for tackling the classification problem of imbalanced data. According to Galar et al. [25], these approaches can be divided into four categories: algorithm-level approaches, data-level approaches, cost-sensitive approaches and classifier ensembles. One known limitation is that approaches for classification which aim at minimizing the classification error on the whole dataset can degrade the classification accuracy on the minority class (of interest), especially when the classes are highly imbalanced.

In this paper, SVM is considered. A direct way of using SVM on imbalanced data is by combining classic SVM with data-level methods. The imbalanced data is, firstly, balanced, i.e. by make the data sizes of the minority and majority classes comparable through a proper resampling of the training dataset. Possible methods include randomly undersampling the majority class [26], randomly oversampling the minority class [27], synthetic minority oversampling technique (SMOTE) [28], combination of oversampling and undersampling [29], etc. The balanced data is, then, fed to SVM for a traditional training process. Also, the undersampling may cause a loss of information in the majority class and in SVM as pointed out in Jian et al. [3]. Then, the classification hyperplane can be disturbed if informative data points are deleted from the majority class [30]. On the other hand, oversampling increases the dataset size and is not scalable to very large datasets [31]. These methods require also proper setting of undersampling and oversampling rates.

Many algorithm-level methods have also been proposed to make SVM more suitable for imbalanced data. Brefeld and Scheffer [32] modify the primal optimization problem of SVM by maximizing the area under the ROC curve (AUC). Experiments show that the obtained AUC values are higher than in other methods Imam et al. [33] use the Golden search method to find the value of z that shifts the hyperplane to maximize the geometric mean (G-mean) of the accuracy of positive and negative samples in the training dataset. The method is called z-SVM. Lin and Chen [34] propose the method SVM-THR, which adjusts directly the estimation function by adding an optimized constant. The method is proved to outperform other correction techniques experimentally. Cost-Sensitive SVM [35] gives, in the objective function of SVM, relatively larger cost to errors on minority class samples than that for the majority class ones. The same strategy is also developed for Least-Square SVM, i.e. WLSSVM in Mahani et al. [36]. In the recent work, i.e. SVM-OTHR of Yu et al. [19], the decision threshold adjustment strategy is optimized to improve the accuracy of classification for imbalanced data, following the work of Lin and Chen [34]. In [19], comparisons are carried out with several popular and new benchmark methods on SVM, showing the satisfactory performance of SVM-OTHR.

There are also other works on SVM for imbalanced data which however are out of the scope of this paper, e.g. fuzzy total margin support vector machine [37], which is effective especially for imbalanced data with outliers and noise, classifiers ensemble [38,39], twin SVM [4] etc.

Table 1
Confusion matrix for classification.

	Predicted positive class	Predictive negative class
Actual positive class	True Positive (TP)	False Negative (FN)
Actual negative class	False Positive (FP)	True negative (TN)

2.2. Accuracy measures for imbalanced data classification

Different metrics are possible for characterizing the class prediction accuracy, e.g. F-measure, G-mean, AUC, etc.

F-measure and G-mean are two of the most popular ones which are functions of the confusion matrix in Table 1.

F-measure and G-mean are calculated separately as

$$F\text{-}measure = \frac{2 * Precision * Recall}{Precision + Recall}$$

and

$$G\text{-}mean = \sqrt{TPR * TNR},$$

where $Precision = TP/(TP + FP)$, $Recall = TPR = TP/(TP + FN)$ and $TNR = TN/(TN + FP)$.

In [33] and [19], G-mean, which balances the prediction accuracy of the minority class and majority class, is selected as prediction accuracy metric for determining the best decision threshold. However, in practice, the prediction accuracy for the minority class (i.e. positive class) is more important. Thus, in this paper, the sum of F-measure and G-mean is chosen as the accuracy metric for optimizing the decision threshold.

3. The proposed approach for classification of imbalanced data

In this section, the proposed approach combining an improved FVS method and SVM is presented for tackling imbalanced data classification problem.

The proposed approach is divided into two parts: undersampling and model construction. During undersampling, some of the original data points are selected as Feature Vectors (FVs), to maximize the separability of the original dataset, and at the same time the selected FVs can represent the other data points in RKHS as a linear combination among them. By so doing, the information contained in the original data is not lost in the undersampling process.

During the model construction process, as in [17], the kernel expansion is reformed only on the selected FVs and the objective is to minimize the classification error on the whole data. An optimization process is integrated to find the best classification threshold, by maximizing the sum of F-measure and G-mean.

3.1. Undersampling with FVS in RKHS

FVS can select the representative training data points in RKHS and they occupy normally a reduced set of the whole training dataset. Then, a method using the reduced set is integrated with reduced computational burden.

Kernel methods have become very popular in recent decades. This type of methods tackles the nonlinear classification/regression problem by mapping the original data into a high dimensional feature space (i.e. RKHS), where the relation becomes linear. FVS selects part of the training data in RKHS as FVs and all the other data in RKHS can be expressed as a linear combination of the FVs [13]. FVS follows a forward process, which selects the vector that is furthest from the space formed by the currently selected FVs as the next FV. The selection process stops when all the data in RKHS can be expressed as a linear combination of the selected FVs, i.e. all the data in RKHS are contained in the feature space formed by the selected FVs.

The metric for characterizing the distance of a data point $\boldsymbol{x}$ from the current feature space $\boldsymbol{S}$ of the selected FVs is the Local Fitness (LF)

$$LF\left(\boldsymbol{x}\right) = \left|1 - \frac{K_{\boldsymbol{S},x}^{t} K_{\boldsymbol{S},\boldsymbol{S}}^{-1} K_{\boldsymbol{S},x}}{k(\boldsymbol{x},\boldsymbol{x})}\right|, \tag{1}$$

where $K_{\boldsymbol{s},\boldsymbol{s}}$ is the kernel matrix of $\boldsymbol{S}$ and $K_{\boldsymbol{S},x} = \{k\left(\boldsymbol{x}_i, \boldsymbol{x}\right)\}, i = 1, 2, \ldots, M$ and M is the number of selected FVs. If $LF\left(\boldsymbol{x}\right) = 0$, the data point $\boldsymbol{x}$ can be expressed as a linear combination of the currently selected FVs in $\boldsymbol{S}$; otherwise, $\boldsymbol{x}$ is a vector outside the feature space $\boldsymbol{S}$.

A faster version of FVS in [13] is proposed in [17], which extracts the same FVs but in a much shorter time. The pseudo-code is shown in Fig. 1. Note that τ is a small positive value that plays a similar role of ϵ of the ϵ-insensitive loss function in SVM and N_m is the threshold of the maximal number of selected FVs. The FVS process terminates when the maximal local fitness is smaller than τ or the number of selected FVs reaches N_m.

The difference between the original FVS proposed in [13] and the faster version proposed in [17] is that in [17], after selecting one new FV, the dataset $\mathbf{T}$ containing the next new FV is reduced by $\mathbf{T} = \mathbf{T} \backslash \mathbf{E}$ with $\mathbf{E} = \{(\boldsymbol{x}_i, y_i) : LF(\boldsymbol{x}_i) \leq \tau \text{ and } (\boldsymbol{x}_k, y_k) \in \mathbf{T}\}$, because the data points in $\mathbf{E}$ cannot have a LF bigger than τ; while in the original FVS of Baudat and Anouar [13], the next new FV is always selected from the whole training dataset, i.e. the LF of all the data points needs to be calculated each time for selecting a new FV, which is time-consuming.

Geometrically, the selected FVs form an oblique coordinate system in RKHS, which encloses all the data, and the coordinates $\boldsymbol{\beta}$ of one data point $\boldsymbol{x}$ in this coordinate system are calculated as

$$\boldsymbol{\beta} = K_{\boldsymbol{S},\boldsymbol{x}}^{t} K_{\boldsymbol{S},\boldsymbol{S}}^{-1} D_{\boldsymbol{S},\boldsymbol{S}}, \tag{2}$$

where $D_{\boldsymbol{S},\boldsymbol{S}}$ is a diagonal matrix and $D_{\boldsymbol{i},\boldsymbol{i}} = \sqrt{k(\boldsymbol{x}_{FV_i}, \boldsymbol{x}_{FV_i})}$ the norm of $\boldsymbol{x}_{FV_i}$, i.e. the ith FV in $\boldsymbol{S}$.

As FVS reduces dramatically the data size, it may be used for undersampling the imbalanced data, without losing the informative data.

Steps 1–5 for selecting the first FV in Algorithm 1 is computationally burdensome with large datasets. In imbalanced data for classification, the size of the minority class is much smaller than that of the majority class, and, theoretically, at least one FV should be selected from the minority class to retain its information contained therein. Thus, instead of using Steps 1–5 in Algorithm 1 for choosing the first FV, one data point in the minority class is selected by turns as the first FV and Steps 6–11 in Algorithm 1 are repeated to select the following FVs from the whole training dataset. This process is repeated N^+(i.e. the size of minority class) times and each data point in the minority class has been selected once as the first FV. Thus, for different data points in the minority class, different sets of FVs can be selected and the set of FVs with the maximal between-class separability of the original dataset is finally kept.

A simple example is shown in Fig. 1 to illustrate the core idea of FVS. In Fig. 1, $\boldsymbol{\varphi}_1$, $\boldsymbol{\varphi}_2$, $\boldsymbol{\varphi}_3$ and $\boldsymbol{\varphi}_5$ are in the same bi-dimensional feature space. Any pair of two feature vectors (e.g. $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$) that are not colinear in this space can form an oblique coordinate system. All the other vectors in this space can be expressed as a linear combination of these two feature vectors, and, thus, there is no need to select more feature vectors than $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$. For the data points outside this feature space, they cannot be represented by $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$. Thus, $\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2$ need to be combined with a new feature vector (e.g. $\boldsymbol{\varphi}_4$) to form a new coordinate system in the three-dimensional feature space, to express linearly all the data points in this space.

Algorithm 1 FVS in [17]

```
Initialization: Training dataset: T = {$(\boldsymbol{x}_i, y_i)$}, for $i = 1, 2, \dots, N$; Feature space: S = [ ]; Threshold of LF:
$\tau$ (a small positive value); Maximal number of selected FVs: $N_m$
Begin:
1:  for $i = 1$ to N
2:      $\boldsymbol{S} = \{\boldsymbol{x}_i\}$
3:      $LF\_all(i) \leftarrow \sum_{j=1}^{N} LF(\boldsymbol{x}_j)$
4:  end for
5:  $\boldsymbol{S} = \{\boldsymbol{x}_i : LF_{all\,(i)} = min(LF\_all(i))\}$
6:  $LF(\boldsymbol{x}_k) = min(\{LF(\boldsymbol{x}_j)\}), j = 1, 2, \dots, N$
7:  while $LF(\boldsymbol{x}_k) > \tau$ and size($\boldsymbol{S}$) < $N_m$
8:      $\boldsymbol{S} \leftarrow \boldsymbol{S} + \boldsymbol{x}_k$
9:      T = T\E, with E={$(\boldsymbol{x}_i, y_i)$: $LF(\boldsymbol{x}_i) \le \tau$ and $(\boldsymbol{x}_k, y_k) \in$ T}
10:     calculate $LF(\boldsymbol{x}_k)$ with respect to $\boldsymbol{S}$
11: end while
end
```

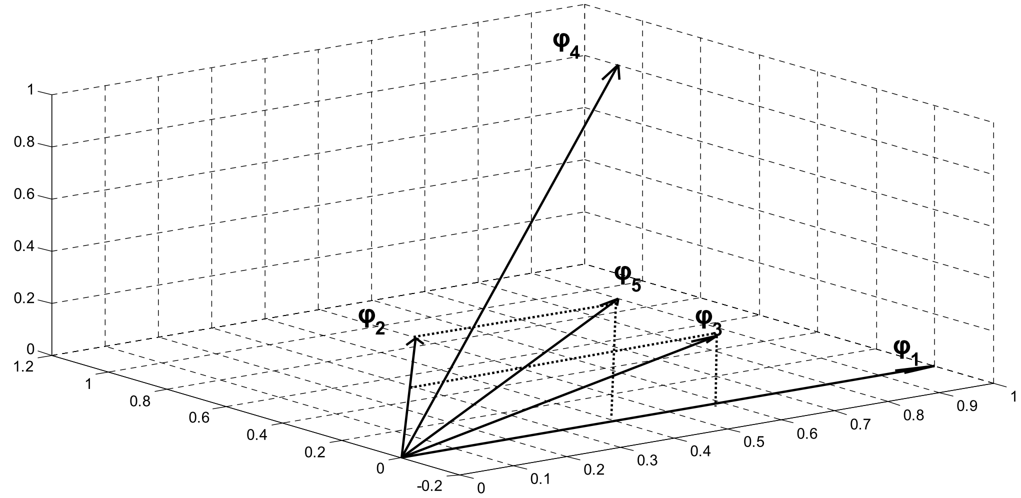


Fig. 1. Illustration of FVS method.

The pseudo-code of FVS modified for maximal between-class separability of binary imbalanced data is shown in Algorithm 2 below.

Algorithm 2 FVS for binary classification of imbalanced data

```
Initialization: Training dataset: T = {$(\boldsymbol{x}_i, y_i)$}, for $i =$
$1, 2, \dots, N^+, N^+{+}1, \dots, N^+ + N^-$
Begin
1:  for $i = 1$ to $N^+$
2:      $\boldsymbol{S}_i = \{\boldsymbol{x}_i\}$
3:      $\boldsymbol{S}_i$ ←steps 6-11 in Algorithm 1
4:      $\lambda_i \leftarrow$ Equation (4)
5:  end for
6:  $\boldsymbol{S} \leftarrow \boldsymbol{S}_j$: max($\{\lambda_i\}$)
end
```

A larger separability can improve the classification accuracy. One problem that emerges in the Step 4 of the previous Algorithm 2 is how to measure the between-class separability of the imbalanced data in RKHS and maximize the separability in the oblique coordinate system formed by the selected FVs. As the data are mapped into the RKHS and their new coordinates $\boldsymbol{\beta}$ can be calculated as Eq. (2), the between-class separability is that of their new coordinates. The new coordinates $\boldsymbol{\beta}$ are used for the construction of the model established in Section 3.2 and, thus, the FVS process is also a feature extraction process. Similar to Mao [40], the Fisher ratio class separability measure is used in this paper to select the best set of FVs.

Suppose the sizes of the minority and majority classes are separately N^+ and N^-, and M FVs are selected: the matrix B of size $(N^+ + N^-) \times M$ is defined as $B_{i,1:M} = \boldsymbol{\beta}_i, i = 1, \dots, N^+$, with $\boldsymbol{\beta}_i$ the coordinates of the ith data point of minority class and $B_{i,1:M} = \boldsymbol{\beta}_i$, $i = N^+ + 1, \dots, N^+ + N^-$, with $\boldsymbol{\beta}_i$ the coordinates of the $(i - N^+)$th data point of majority class. The Fisher ratio class separability λ_j of the data on the direction of the jth FV is calculated as

$$\lambda_j = \frac{(m_{j,+} - m_{j,-})^2}{\delta_{j,+}^2 + \delta_{j,-}^2}, \tag{3}$$

where $m_{j,+}$ and $\delta_{j,+}^2$ are the mean and variance of the jth coordinate of all the data points in the minority class, i.e. $B_{1:N^+,j}$, and $m_{j,-}$ and $\delta_{j,-}^2$ are the mean and variance of the jth coordinate of all the data points in the majority class, i.e. $B_{N^+ + 1:N^+ + N^-,j}$. The separability of the data in the oblique coordinate system formed by the selected FVs is calculated as the mean of the separability in the directions of each FV,

$$\lambda = \frac{1}{M} \sum_{j=1}^{M} \lambda_j. \tag{4}$$

Fig. 2 shows graphically the meaning of separability calculated with Eq. (3). Considering a binary dataset with two features, i.e. x_1 and x_2, the Figure shows that the x_1 feature values can better

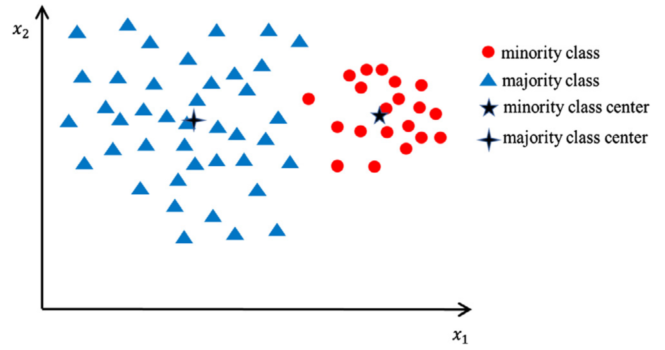


Fig. 2. Illustration of separability-based feature selection.

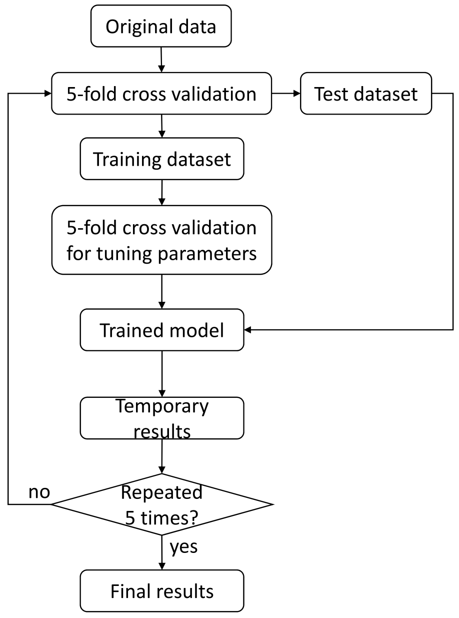


Fig. 3. The flow of the experiment process.

separate the two classes, as the centers of these two classes have very different values for this feature. In the contrary, the other feature x_2 cannot differentiate the two classes, since the values of the two classes for feature x_2 are very close. Thus, in this example, one only needs to selection feature x_1.

3.2. Classification model based on the selected FVs

The classification model here used was originally proposed in [17] for regression (FVR). To make it suitable for the classification problem, an optimization is added to find the best decision threshold value with respect to a specific accuracy metric.

In the nonparametric and semi-parametric representer theorems in [41], it is shown that in kernel algorithms for structural risk minimization in RKHS, the estimate function can be written as a kernel expansion on the training data points, as shown in Eq. (5), with $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ being the kernel function α_i and b being the parameters tuned during the training process.

$$f\left(\boldsymbol{x}\right) = \sum_{i=1}^{N} \alpha_i k(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{5}$$

The kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ represents the inner product of the mapping of $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ RKHS, i.e. $k\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) = \left\langle \varphi\left(\boldsymbol{x}_i\right), \varphi(\boldsymbol{x}_j)\right\rangle$, with $\varphi\left(\boldsymbol{x}\right)$ being the mapping of $\boldsymbol{x}$ in RKHS. The FVS introduced in Section 3.1 shows that every input vector in the training dataset can be expressed as a linear combination of the selected FVs in RKHS, as shown in Eq. (6):

$$\varphi\left(\boldsymbol{x}\right) = \sum_{i=1}^{M} \frac{\beta_i(\boldsymbol{x})}{\sqrt{k(\boldsymbol{x}_{FV_i}, \boldsymbol{x}_{FV_i})}} \varphi\left(\boldsymbol{x}_{FV_i}\right), \tag{6}$$

with $\beta_i(\boldsymbol{x})$ being the jth coordinate of the data point $\boldsymbol{x_i}$ in the oblique coordinate system formed by the FVs, and $\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}$ being the norm of FV $\boldsymbol{x}_{FV_j}$.

Eq. (5) can be rewritten as,

$$\begin{aligned} f\left(\boldsymbol{x}\right) &= \sum_{i=1}^{N} \alpha_i \left\langle \boldsymbol{\varphi}(\boldsymbol{x}_i), \boldsymbol{\varphi}(\boldsymbol{x})\right\rangle + b \\ &= \sum_{i=1}^{N} \alpha_i \left\langle \boldsymbol{\varphi}(\boldsymbol{x}_i), \sum_{j=1}^{M} \frac{\beta_j\left(\boldsymbol{x}\right)}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} \boldsymbol{\varphi}\left(\boldsymbol{x}_{FV_j}\right)\right\rangle + b \\ &= \sum_{j=1}^{M} \frac{\beta_j\left(\boldsymbol{x}\right)}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} \left(f\left(\boldsymbol{x}_{FV_j}\right) - b\right) + b \end{aligned} \tag{7}$$

with $f\left(\boldsymbol{x}_{FV_j}\right), j = 1, 2, \ldots, M$ being the predicted values of the FVs selected from the training dataset $\boldsymbol{T}$. The predicted value of a data point can be expressed as a linear combination of the predicted values of the selected FVs.

The primal optimization problem of FVR is formulated as

$$\begin{aligned} &minimize_{\hat{y}_j, b} \quad W = \frac{1}{N} \sum_{i=1}^{N} \left(g\left(\boldsymbol{x_i}\right) - y_i\right)^2 \\ &subject\ to \quad g\left(\boldsymbol{x_i}\right) = \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} (\hat{y}_j - b) + b, \\ &\qquad\qquad i = 1, 2, \ldots, N, \end{aligned} \tag{8}$$

with M representing the number of FVs selected from the training dataset $\boldsymbol{T} = \left(\boldsymbol{x}_i, y_i\right), i = 1, 2, \ldots, N$ with $y_i \in \{-1, +1\}$, $\hat{y}_j, j = 1, 2, \ldots, M$ being the predicted value of the selected FVs, i.e. $f\left(\boldsymbol{x}_{FV_j}\right)$ in Eq. (7). In Eq. (8), the estimate function $g\left(\boldsymbol{x}\right)$ is a kernel expansion on the selected FVs and the objective is to minimize the estimation error on the whole training dataset, in order to avoid overfitting on the selected FVs. As equal constraints are used in Eq. (8), the output of $g\left(\boldsymbol{x}\right)$ is numeric. For a data point $\boldsymbol{x}$, its predicted value is $g\left(\boldsymbol{x}\right) = \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x})}{\sqrt{k(\boldsymbol{x}_j, \boldsymbol{x}_j)}} (\hat{y}_j - b) + b$. With a given decision threshold γ, the label of the data point $\boldsymbol{x}$ is $+1$ if $g\left(\boldsymbol{x}\right) \geq \gamma$ and -1, otherwise. The consideration of the least-squared error on the whole training dataset makes the model computationally more efficient.

Similar to least-squared MSE, the unknowns, i.e. the predicted values of the FVs and the constant b in the primal optimization problem can be calculated analytically by solving the dual problem of Eq. (8).

The dual problem is

$$\begin{aligned} W = \frac{1}{N} \sum_{i=1}^{N} \Bigg(&\sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} \hat{y}_j \\ &+ b(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}}) - y_i \Bigg)^2. \end{aligned} \tag{9}$$

Setting the partial derivatives of W with respect to $\hat{y}_j$ and b to zero yields,

$$\frac{\partial W}{\partial \hat{y}_{j_0}} = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{\beta_{j_0}(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_{j_0}}, \boldsymbol{x}_{FV_{j_0}})}} * \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} * \hat{y}_j$$

Table 2
Characteristics of the Keel datasets used in this paper.

Dataset	Number of instances	Number of attributes	Imbalance Ratio (IR)	Average # of selected FVs
glass1	214	9	1.82	50
haberman	306	3	2.78	25
new-thyroid1	215	5	5.14	40
yeast3	1484	8	8.1	50
ecoli3	336	7	8.6	50
ecoli-0-6-7_vs_5	220	6	10	50
yeast-1_vs_7	459	7	14.3	50
ecoli4	336	7	15.8	50
abalone-9_vs_18	731	8	16.4	50
shuttle-6_vs_2–3	230	9	22	50
yeast4	1484	8	28.1	50
yeast5	1484	8	32.73	50
poker-8-9_vs_5	2075	10	82	40
poker-8_vs_6	1477	10	85.88	50
abalone19	4174	8	129.44	50
ecoli-0-1_vs_2-3–5	244	7	9.17	50
ecoli-0-1_vs_5	240	6	11	50
ecoli-0-3-4_vs_5	200	7	9	50
ecoli-0-6-7_vs_3–5	222	7	9.09	50
flare-F-5-fold	1066	11	23.79	50
led7digit-0-2-4-5-6-7-8-9_vs_1	443	7	10.97	50
pima	768	8	1.87	50
vehicle1	846	18	2.9	50
winequality-red-4	1599	11	29.17	50
yeast-0-2-5-7-9_vs_3-6–8	1004	8	9.14	50
yeast-0-3-5-9_vs_7–8	506	8	9.12	50

Table 3
Classification results of the proposed method and benchmark methods with respect to F-measure.

	Proposed approach	SSVM	SVM-RUS	SVM-ROS	SVM-SMOTE	CS-SVM	z-SVM	SVM-THR	SVM-OTHR	SVM-ROSE
glass1	.608±.003	.662±.032	.637±.028	.654±.018	.641±.024	.623±.022	.658±.030	.669±.018	**.672±.018**	.593±.005
haberman	**.460±.004**	.269±.034	.424±.023	.405±.023	.405±.033	.423±.018	.451±.032	.364±.038	.443±.030	.421±.082
new-thyroid1	**.948±.003**	.919±.023	.881±.040	.930±.042	.910±.025	.912±.022	.923±.038	.935±.023	.935±.027	.901±.068
yeast3	**.769±.002**	.736±.016	.633±.015	.674±.013	.702±.010	.668±.010	.709±.011	.718±.015	.711±.009	.487±.042
ecoli3	.618±.022	.613±.051	.522±.025	.572±.032	.598±.053	.564±.038	.613±.027	.598±.020	**.635±.023**	.628±.060
ecoli-0-6-7_vs_5	**.821±.004**	.721±.036	.495±.031	.741±.050	.648±.082	.723±.073	.740±.080	.723±.049	.759±.071	.666±.150
yeast-1_vs_7	**.419±.023**	.417±.073	.189±.022	.251±.056	.231±.036	.291±.041	.302±.047	.299±.043	.271±.033	.251±.104
ecoli4	**.831±.011**	.726±.126	.594±.057	.693±.079	.750±.066	.705±.123	.692±.107	.640±.075	.706±.072	.629±.170
abalone-9_vs_18	**.639±.006**	.404±.065	.251±.036	.344±.047	.343±.026	.349±.036	.348±.040	.390±.050	.417±.055	.327±.042
shuttle-6_vs_2–3	.853±.021	.781±.159	.558±.079	.711±.152	.877±.060	.820±.168	.844±.147	.770±.107	.818±.154	**.933±.149**
yeast4	**.420±.002**	.330±.024	.196±.013	.346±.029	.316±.025	.352±.032	.356±.022	.377±.024	.364±.018	.189±.031
yeast5	.683±.012	.646±.034	.466±.044	**.706±.025**	.691±.042	.695±.039	.683±.031	.634±.029	.682±.036	.517±.093
poker-8-9_vs_5	**.174±.005**	.016±.030	.035±.004	.045±.035	.092±.053	.079±.025	.082±.045	.093±.024	.080±.039	.052±.017
poker-8_vs_6	**.719±.037**	.343±.102	.251±.033	.374±.064	.646±.082	.452±.106	.424±.115	.353±.015	.382±.070	.441±.203
abalone19	**.093±.000**	.000±.000	.031±.005	.051±.014	.039±.012	.046±.021	.044±.026	.050±.011	.048±.007	.031±.005
ecoli-0-1_vs_2-3–5	**.765±0.011**	.669±.071	.534±.051	.641±.082	.681±.047	.619±.052	.643±.060	.614±.051	.665±.062	.555±.101
ecoli-0-1_vs_5	**.854±0.014**	.807±.060	.677±.031	.796±.059	.752±.039	.811±.074	.806±.063	.667±.039	.826±.037	.549±.084
ecoli-0-3-4_vs_5	.810 ± 0.053	.828±.055	.640±.077	**.829±.061**	.779±.058	.822±.074	.815±.075	.687±.050	.825±.066	.717±.182
ecoli-0-6-7_vs_3–5	.689 ± 0.163	.676±.078	.550±.035	.680±.054	.620±.088	.708±.077	.693±.074	.677±.058	**.711±.072**	.489±.180
flare-F-5-fold	**.418±0.023**	.112±.031	.199±.017	.176±.031	.195±.031	.163±.029	.195±.026	.222±.041	.245±.033	.392±.075
led7digit-0-2-4-5-6-7-8-9_vs_1	**.773±0.014**	.648±.025	.630±.037	.639±.042	.711±.029	.658±.032	.664±.041	.690±.043	.696±.094	.545±.117
pima	**.655±0.000**	.569±.015	.583±.021	.558±.024	.563±.013	.543±.015	.567±.022	.580±.015	.598±.017	.576±.021
vehicle1	.648 ± 0.000	.642±.021	.645±.015	.642±.014	.645±.017	.646±.018	.663±.018	**.667±.017**	.651±.021	.529±.042
winequality-red-4	.196 ± 0.001	.098±.036	.107±.008	.164±.040	.139±.033	.159±.033	.150±.031	.156±.023	.169±.034	**.197±.052**
yeast-0-2-5-7-9_vs_3-6–8	**.811±0.003**	.798±.014	.595±.027	.690±.015	.709±.021	.687±.013	.685±.037	.761±.018	.709±.018	.742±.116
yeast-0-3-5-9_vs_7–8	**.390±0.000**	.295±.042	.261±.025	.290±.037	.279±.043	.317±.040	.322±.038	.314±.025	.318±.030	.371±.102
mean rank	2.058	6.038	8.712	6.115	6.019	5.673	4.827	5.212	3.423	6.923

Table 4
Classification results of the proposed method and benchmark methods with respect to G-mean.

	Proposed approach	SSVM	SVM-RUS	SVM-ROS	SVM-SMOTE	CS-SVM	z-SVM	SVM-THR	SVM-OTHR	SVM-ROSE
glass1	.687±.002	.736±.022	.714±.024	.732±.017	.716±.022	.706±.021	.733±.020	.710±.013	**.738±.014**	.663±.007
haberman	**.617±.003**	.403±.052	.533±.020	.566±.020	.568±.029	.525±.021	.541±.028	.517±.034	.606±.027	.582±.070
new-thyroid1	**.989±.000**	.961±.019	.958±.017	.961±.033	.957±.020	.955±.017	.960±.024	.981±.011	.963±.019	.977±.016
yeast3	.885±.001	.833±.013	.841±.012	.864±.012	.866±.008	.870±.007	.889±.013	.885±.009	**.892±.008**	.841±.029
ecoli3	.795±.016	.769±.048	.840±.022	.796±.027	.780±.074	.793±.043	.829±.055	.826±.021	.851±.022	**.894±.047**
ecoli-0-6-7_vs_5	**.886±.003**	.807±.049	.830±.039	.830±.054	.761±.105	.808±.071	.855±.047	.868±.027	.857±.053	.864±.071
yeast-1_vs_7	.614±.004	.553±.105	.631±.047	.527±.091	.539±.087	.624±.074	.627±.053	.616±.063	.613±.062	**.673±.125**
ecoli4	.914±.005	.793±.132	**.916±.031**	.807±.082	.857±.049	.790±.135	.843±.063	.848±.078	.867±.073	.892±.079
abalone-9_vs_18	**.852±.007**	.557±.066	.732±.038	.592±.056	.593±.026	.597±.042	.666±.056	.723±.060	.724±.038	.769±.070
shuttle-6_vs_2–3	**.991±.000**	.787±.161	.751±.112	.720±.149	.887±.064	.831±.170	.807±.107	.819±.111	.825±.153	.941±.131
yeast4	.746±.006	.510±.031	**.776±.013**	.697±.034	.683±.043	.678±.031	.727±.042	.727±.029	.760±.026	.761±.037
yeast5	.906±.003	.863±.020	**.943±.017**	.897±.020	.881±.029	.877±.024	.875±.010	.907±.029	.899±.024	.938±.035
poker-8-9_vs_5	.533±.017	.034±.059	.219±.058	.392±.079	**.595±.134**	.511±.055	.494±.090	.497±.108	.502±.080	.519±.093
poker-8_vs_6	.782±.019	.396±.117	.425±.098	.512±.066	.669±.081	.489±.198	.557±.078	.615±.097	.631±.087	**.949±.084**
abalone19	.641±.004	0.00 ± 0.00	**.644±.063**	.369±.081	.346±.103	.253±.071	.328±.096	.277±.038	.349±.072	.642±.062
ecoli-0-1_vs_2-3–5	**.871±.012**	.781±.063	.791±.055	.762±.073	.821±.025	.747±.066	.760±.053	.757±.038	.797±.055	.770±.131
ecoli-0-1_vs_5	**.913±.006**	.853±.067	.868±.030	.849±.062	.866±.050	.864±.071	.857±.041	.859±.025	.894±.021	.795±.106
ecoli-0-3-4_vs_5	.902±.018	.878±.054	.860±.069	.863±.066	.866±.068	.851±.076	**.917±.088**	.853±.015	.869±.066	.885±.124
ecoli-0-6-7_vs_3–5	**.926±.004**	.768±.077	.847±.025	.762±.067	.762±.099	.786±.086	.774±.053	.876±.045	.788±.071	.637±.378
flare-F-5-fold	.763±.019	.229±.051	.769±.024	.501±.084	.464±.046	.473±.062	.479±.025	.501±.088	.640±.059	**.861±.093**
led7digit-0-2-4-5-6-7-8-9_vs_1	**.898±.007**	.847±.021	.831±.022	.816±.055	.844±.017	.840±.026	.832±.033	.806±.017	.851±.107	.835±.049
pima	**.730±.000**	.661±.012	.670±.019	.652±.019	.658±.010	.639±.013	.663±.017	.670±.011	.678±.014	.667±.017
vehicle1	.752±.000	.749±.017	.782±.011	.747±.010	.750±.013	.751±.016	.748±.013	**.797±.014**	.787±.017	.664±.051
winequality-red-4	**.633**±.004	.213±.081	.452±.032	.382±.070	.394±.068	.368±.063	.560±.068	.504±.043	.540±.070	.542±.029
yeast-0-2-5-7-9_vs_3-6–8	.865±.002	.866±.009	.840±.012	.846±.011	.849±.013	.845±.013	.865±.016	**.889±.011**	.884±.007	.824±.094
yeast-0-3-5-9_vs_7–8	**.676±.002**	.486±.044	.570±.035	.556±.048	.543±.050	.592±.048	.542±.065	.616±.022	.637±.034	.532±.091
mean rank	2.615	7.904	4.962	7.231	6.173	7.269	5.731	5.115	3.385	4.615

$$+ \ b * \sum_{i=1}^{N} \frac{\beta_{j_0}(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_{j_0}}, \boldsymbol{x}_{FV_{j_0}})}} * \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}}\right)$$

$$- \sum_{i=1}^{N} \frac{\beta_{j_0}(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_{j_0}}, \boldsymbol{x}_{FV_{j_0}})}} * y_i = 0 \qquad (10)$$

$$\frac{\partial W}{\partial b} = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}} * (1 - \sum_{l=1}^{M} \frac{\beta_l(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_l}, \boldsymbol{x}_{FV_l})}}) * \hat{y}_j$$

$$+ \ b * \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}}\right)^2$$

$$- \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x}_{FV_j}, \boldsymbol{x}_{FV_j})}}\right) * y_i = 0. \qquad (11)$$

The unknowns are calculated with the following equation,

$$\begin{bmatrix} \boldsymbol{\Omega} & \mathbf{H} \\ \boldsymbol{\Gamma}^T & c \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{y}} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{P} \\ l \end{bmatrix}, \qquad (12)$$

where $\boldsymbol{\Omega}$ is a $M \times M$ matrix with $\boldsymbol{\Omega}_{mn} = \sum_{i=1}^{N} \frac{\beta_m(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_m}, \boldsymbol{x_m})}} * \frac{\beta_n(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_n}, \boldsymbol{x_n})}}$, $\mathbf{H}$ is a $M \times 1$ vector with $\mathbf{H}_m = \sum_{i=1}^{N} \frac{\beta_m(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_m}, \boldsymbol{x_m})}} * \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_j}, \boldsymbol{x_j})}}\right)$, $\boldsymbol{\Gamma}$ is a $M \times 1$ vector with $\boldsymbol{\Gamma}_m = \sum_{i=1}^{N} \frac{\beta_m(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_m}, \boldsymbol{x_m})}} * (1 - \sum_{l=1}^{M} \frac{\beta_l(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_l}, \boldsymbol{x_l})}})$, c is a constant and $c = \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_j}, \boldsymbol{x_j})}}\right)^2$; $\hat{\boldsymbol{y}} = (\hat{y}_j), j = 1, 2, \ldots, M$ and b are the unknowns in Eq. (8), $\mathbf{P}$ is a $M \times 1$ vector with $\mathbf{P}_m = \sum_{i=1}^{N} \frac{\beta_m(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_m}, \boldsymbol{x_m})}} * y_i$, $l = \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{M} \frac{\beta_j(\boldsymbol{x_i})}{\sqrt{k(\boldsymbol{x_j}, \boldsymbol{x_j})}}\right) * y_i$.

Similarly to z-SVM and SVM-OTHR, the decision threshold is the optimized value that gives the maximal prediction performance, i.e. maximal value of the sum of F-measure and G-mean.

4. Case studies

In this Section, 26 Keel datasets [20] with different Imbalance Ratios (IR) are selected to test the effectiveness of the proposed approach. In comparison with various popular benchmark methods, the accuracy of the proposed approach with respect to F-measure and G-mean is justified by statistical tests.

Table 2 summarizes the characteristics of different datasets.

Several SVM-based methods are considered as benchmarks, including standard SVM without any class imbalance correction algorithms (SSVM), SVM with random undersampling (SVM-RUS), SVM with random oversampling (SVM-ROS), SVM with SMOTE (SVM-SMOTE), cost-sensitive SVM (CS-SVM), z-SVM, SVM with decision threshold adjustment (SVM-THR), optimized SVM decision threshold adjustment (SVM-OTHR) and SVM with random over sampling examples (SVM-ROSE). Fivefold cross validation is used for tuning the hyperparameters of the benchmark methods. Since the exact same experimental procedure of Yu et al. [19] is adopted in this paper, the results of the benchmark methods are directly taken from that paper. The experiment process is an inner-and-outer 5-fold cross validation and the outer five-fold cross validation is repeated ten times for obtaining trustable results. The experiment process in shown in Fig. 3.

Statistical tests, e.g. t-test, Wilcoxon signed rank test, Friedman test, are efficient and trustable ways for performance comparison of several machine learning approaches on a number of datasets [42–47].

In this paper, Friedman test [48] is adopted to test if the ranks of the results from different approaches are significantly different from the mean rank of all the approaches under the null hypothesis. If the null hypothesis is rejected, Bonferroni–Dunn test can tell if one method is significantly better than another [49]. For k algorithms and n rank results, the Friedman statistic in Eq. (13) ($R_j, j = 1, \ldots, k$ is the mean rank of algorithm j) follows a χ_F^2 distribution with $k-1$ degrees of freedom. If the statistic F_F is smaller than the critical value $F(k-1, (k-1)(n-1))$ for $\alpha = 0.05$ with F-distribution, the null hypothesis is not rejected and no difference exists among the performance of different approaches. Otherwise, the null hypothesis is rejected and, then, if the difference of the mean rank of two approaches is larger than Critical Difference (CD), it is significant or, otherwise, no significant difference exists between the two methods.

$$\chi_F^2 = \frac{12n}{k(k+1)}[\sum_j R_j^2 - \frac{k(k+1)^2}{4}] \tag{13}$$

$$F_F = \frac{(n-1)\chi_F^2}{n(k-1) - \chi_F^2} \tag{14}$$

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6n}} \tag{15}$$

As the Friedman test and Bonferroni–Dunn test consider only the ranks of the different methods and not the difference in accuracy of the different methods for each dataset, the Wilcoxon signed rank test, which considers both, [50] is also considered in this work for the pairwise comparison of the benchmark methods.

4.1. Experimental setting of the proposed approach

The proposed approach includes FVS and FVR, introduced in the previous Sections 3.1 and 3.2, respectively. Some parameters need to be tuned, including the threshold of LF and the parameters related to the kernel function. In the experiments, most of these parameters are found analytically, without using any complicated or time-consuming tuning method as in [51].

(i) Threshold of LF: $\tau = \min(10^{-3}, 1/N)$, with N the size of training dataset. Only the data points with a LF larger than τ are considered as candidates of being the next new FV. The smaller the value of τ, the more training data points are selected as FVs, and vice versa. This setting can avoid selecting insufficient FVs for small datasets causing a loss of useful information and too many FVs for big datasets causing overfitting and computationally burdensome.

(ii) Radial Basis Function (RBF) kernel function, i.e. $k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp(-\frac{(\boldsymbol{x}_1-\boldsymbol{x}_2)^2}{2\sigma^2})$ is used in the proposed approach for FVS and FVR. The parameter σ is calculated with Eq. (16) below, as proposed in [52], and the parameter μ is set to 0.1.

$$\sigma^2 = \mu * \max\{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2, \ i, j = 1, \ldots, N\} \tag{16}$$

From the above, one can observe that the proposed approach can be easily tuned and that the optimization in Eq. (8) related to the training is solved analytically, as in Eq. (12). On the contrary, classic SVM methods need complicated tuning of the parameters and the optimization is performed by solving a quadratic programming problem.

The possible maximal numbers of selected FVs, i.e. N_m are [20 25 30 35 40 45 50] and the best number is tuned during the training process using five-fold cross validation.

Table 5
Statistical test of the experiment results with Friedman and Bonferroni–Dunn tests.

	χ_F^2	F_F	$F_{\alpha=0.05}(9, 225)$	Null hypothesis	$CD_{\alpha=0.1}$
F-measure	82.110	14.291	1.922	rejected	2.625
G-mean	74.956	11.782	1.922	rejected	2.625

4.2. Experiment results and statistical test

For the Keel dataset [20] reported in Table 1, less than 50 data points are selected as FVs in the training process. Random five-fold cross validations are repeated for ten times to reduce the randomness of the classification results. The classification results (characterized by F-measure and G-mean) are reported in the form of mean±standard deviation of the results for the ten times fivefold cross validation.

Table 3 shows the classification results of the proposed approach and the benchmark results. One can observe that the proposed method outperforms the benchmark methods for 19 and 13 out of the 26 imbalanced datasets, with respect separately to F-measure and G-mean. In this sense, the proposed approach gives better results than the benchmark methods. Note that most the results given by the approach proposed in this paper obtain a lower standard deviation in comparison with the benchmark methods. This proves that the proposed approach is stable and generalizes well among different case studies.

The statistical tests introduced in the previous section is adopted to tell if there is any (significant) difference between the performances of the proposed method and the benchmark methods.

The mean ranks of different methods with respect to F-measure and G-mean are shown in the last line of Tables 3 and 4 respectively. The proposed method obtains the best mean ranks for both F-measure and G-mean. Table 5 shows the statistical test results separately for F-measure and G-mean with Friedman and Bonferroni–Dunn tests. Differences exist among the results given by all the methods, as the statistic F_F values (i.e. 14.291 and 11.782 with respect to F-measure and G-mean separately) are much bigger than the critical value $F(9, 225)$ for a $\alpha = 0.05$ (i.e. 1.922), i.e. the null hypothesis is rejected.

Considering both F-measure and G-mean, the proposed method gives always better results than SSVM, SVM-ROS, SVM-SMOTE, CS-SVM and z-SVM, because the differences of the mean ranks of the proposed method with these benchmark methods are bigger than the critical difference value $CD_{\alpha=0.1}$ (i.e. 2.625). Considering solely F-measure, the proposed method gives also higher values than SVM-RUS, SVM-THR and SVM-ROSE. This shows that the sum of F-measure and G-mean is a proper optimization objective for training classification model with imbalanced data. This objective can give a higher F-mean value with a comparable G-measure value.

As mentioned earlier, Friedman and Bonferroni–Dunn tests do not consider the difference in the performance of the methods on the same dataset. Then, the Wilcoxon signed rank test, which considers both the mean ranks and the differences of performance of the methods are also used for the comparison. The significance level is set to be 5%. The results are shown in Table 6. The null hypothesis for the test is that the proposed method gives same results as the corresponding benchmark method against the alternative hypothesis that the proposed method gives better results. In Table 6, the h value 1 indicates that the null hypothesis is rejected and the value equal to −1 indicates that the null hypothesis is not rejected. The scalar p-value between 0 and 1 is the probability of observing a test statistic as or more extreme than the observed value under the null hypothesis. The results in Table 6 show that by considering the difference in accuracy between the proposed

Table 6
Statistical test of the experiment results with Wilcoxon signed rank test.

		SSVM	SVM-RUS	SVM-ROS	SVM-SMOTE	CS-SVM	z-SVM	SVM-THR	SVM-OTHR	SVM-ROSE
F-measure	h value	1	1	1	1	1	1	1	1	1
	p-value	4.72e−05	2.83e−08	1.22e−05	4.44e−06	1.08e−05	7.14e−05	9.19e−06	2.49e−04	1.33e−6
G-mean	h value	1	1	1	1	1	1	1	1	1
	p-value	4.12e−05	1.58e−03	5.65e−06	4.66e−05	2.64e−05	1.52e−04	4.11e−04	1.38e−03	3.44e−2

method and the benchmark methods for each dataset, the proposed method gives always better results than the benchmark methods, with respect to both F-measure and G-mean.

The FVs selected from the undersampling method in Section 3.1 are also fed to a standard SVM. However, this does not work as the *TP* rate is nearly always 0 for the minority class, except for the first two datasets with smaller IR.

The computation cost for training is not theoretically compared in this paper because the process for tuning the hyperparameters of the benchmark methods occupies a big part of the training time. Different methods can be used for tuning the hyperparameters of the benchmark methods and the computation time varies a lot depending on the methods and their settings. Supercomputing makes the computation burden less crucial for the practitioners.

The hyperparameters tuning process for the proposed approach is much simpler than the benchmark methods. The two parameters, i.e. the threshold for LF and the parameters in the kernel function to tune in the proposed approach during the training process are calculated analytically. In comparison with other methods that need sophisticated processes for tuning parameters, the proposed method is not time-consuming. But the time for FVS increases the computational burden of the model training process. For each test data point, $\boldsymbol{\beta}$ needs to be calculated with Eq. (2) in order to calculate the predicted value.

5. Conclusions and perspectives

SVM has been widely applied for tackling classification problems, including also situations with imbalanced data. However, undersampling methods suffer from the loss of useful information and SVM training is dependent on the tuning of its hyperparameters.

By extrapolating the geometrical meaning of kernel methods in RKHS, an approach is proposed in this paper, combining a FVS method and an easy-tuning version of LSSVM, i.e. FVR. All the data in RKHS can be expressed as a linear combination of the selected FVs and the separability between the classes is maximized during the FVS process. The FVR model can be solved analytically and the only hyperparameter can also be given analytically. The sum of F-measure and G-mean is taken as the accuracy metric and the classification threshold is optimized to maximize such a metric.

The experiments performed on 26 datasets of literature show that statistically the proposed method gives better results than the benchmark methods with respect to both F-measure and G-mean. Also, the proposed method gives a low standard deviation in comparison with the benchmark methods, showing the good generalization property of the proposed method.

One possible direction for future work is on the further development of FVR to a cost-sensitive version to improve the results, as the selected FVs are still imbalanced. Also, other performance scores for classification can be considered as objectives in Eq. (8). Another direction is for feature extraction using the coordinate system formed by the FVs. The new coordinates of the data in the coordinate system formed by the FVs can be used directly to train the classification model. As the separability is considered during the FVS process, the results of classification can be expected to be even more satisfactory.

Acknowledgment

This work was supported by Science Challenge Project (No. TZ2018007).

References

[1] G. Ditzler, R. Polikar, An ensemble based incremental learning framework for concept drift and class imbalance, in: Neural Networks (IJCNN), The 2010 International Joint Conference on, IEEE, 2010, pp. 1–8, July.

[2] Y. Yang, S.C. Chen, Ensemble Learning from Imbalanced Data Set for Video Event Detection, in: Information Reuse and Integration (IRI), 2015 IEEE International Conference on, IEEE, 2015, pp. 82–89, August.

[3] C. Jian, J. Gao, Y. Ao, A new sampling method for classifying imbalanced data based on support vector machine ensemble, Neurocomputing (2016).

[4] Y. Xu, Z. Yang, Y. Zhang, X. Pan, L. Wang, A maximum margin and minimum volume hyper-spheres machine with pinball loss for imbalanced data classification, Knowl.-Based Syst. 95 (2016) 75–85.

[5] Y. Sun, A.K. Wong, M.S. Kamel, Classification of imbalanced data: A review, Int. J. Pattern Recognit. Artif. Intell. 23 (04) (2009) 687–719.

[6] G. Menardi, N. Torelli, Training and assessing classification rules with imbalanced data, Data Min. Knowl. Discov. 28 (1) (2014) 92–122.

[7] W. Wei, J. Li, L. Cao, Y. Ou, J. Chen, Effective detection of sophisticated online banking fraud on extremely imbalanced data, World Wide Web 16 (4) (2013) 449–475.

[8] M. Khalilia, S. Chakraborty, M. Popescu, Predicting disease risks from highly imbalanced data using random forest, BMC Med. Inform. Decis. Mak. 11 (2011) 1.

[9] N. García-Pedrajas, J. Pérez-Rodríguez, M. García-Pedrajas, D. Ortiz-Boyer, C. Fyfe, Class imbalance methods for translation initiation site recognition in DNA sequences, Knowl.-Based Syst. 25 (2012) 22–34.

[10] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. (9) (2008) 1263–1284.

[11] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics, Inform. Sci. 250 (2013) 113–141.

[12] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, G. Bing, Learning from class-imbalanced data: review of methods and applications, Expert Syst. Appl. 73 (2017) 220–239.

[13] G. Baudat, F. Anouar, Feature vector selection and projection using kernels, Neurocomputing 55 (1) (2003) 21–38.

[14] M.M. Adankon, M. Cheriet, Model selection for the LS-SVM. Application to handwriting recognition, Pattern Recognit. 42 (12) (2009) 3264–3270.

[15] H. Zareipour, A. Janjani, H. Leung, A. Motamedi, A. Schellenberg, Classification of future electricity market prices, IEEE Trans. Power Syst. 26 (1) (2011) 165–173.

[16] A.R. Teixeira, A.M. Tomé, E.W. Lang, Unsupervised feature extraction via kernel subspace techniques, Neurocomputing 74 (5) (2011) 820–830.

[17] J. Liu, E. Zio, Feature vector regression with efficient hyperparameters tuning and geometric interpretation, Neurocomputing 218 (2016) 411–422.

[18] J.S. Han, S.W. Lee, Z. Bien, Feature subset selection using separability index matrix, Inform. Sci. 223 (2013) 102–118.

[19] H. Yu, C. Mu, C. Sun, W. Yang, X. Yang, X. Zuo, Support vector machine-based optimized decision threshold adjustment strategy for classifying imbalanced data, Knowl.-Based Syst. 76 (2015) 67–78.

[20] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. Del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (3) (2009) 307–318.

[21] N.V. Chawla, Data mining for imbalanced datasets: An overview, in: Data mining and knowledge discovery handbook, Springer, Boston, MA, 2009, pp. 875–886.

[22] T.W. Liao, Classification of weld flaws with imbalanced class data, Expert Syst. Appl. 35 (3) (2008) 1041–1052.

[23] P. Branco, L. Torgo, R. Ribeiro, A survey of predictive modelling under imbalanced distributions, 2015, arXiv preprint. arXiv:150501658.

[24] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (1995) 273–297.

[25] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst. Man Cybern. Part C 42 (2012) 463–484.

[26] N. Japkowicz, The class imbalance problem: Significance and strategies, in: Proc. of the Int'l Conf. on Artificial Intelligence, June, 2000.

[27] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explor. Newsl. 6 (1) (2004) 20–29.

[28] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. (2002) 321–357.

[29] O. Loyola-González, M. García-Borroto, M.A. Medina-Pérez, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, G.De. Ita, An empirical study of oversampling and undersampling methods for lcmine an emerging pattern based classifier, in: Pattern Recognition, Springer Berlin Heidelberg, 2013, pp. 264–273.

[30] R. Akbani, S. Kwek, N. Japkowicz, Applying support vector machines to imbalanced datasets, in: Machine learning: ECML, Springer Berlin Heidelberg, 2004, pp. 39–50.

[31] N. Garcia-Pedrajas, J. Perez-Rodriguez, A. de Haro-Garcia, OligoIS: Scalable instance selection for class-imbalanced data sets, IEEE Trans. Cybern. 43 (2013) 332–346.

[32] U. Brefeld, T. Scheffer, AUC maximizing support vector learning, in: Proceedings of the ICML 2005 workshop on ROC Analysis in Machine Learning, August, 2005.

[33] T. Imam, K.M. Ting, J. Kamruzzaman, z-SVM: An SVM for improved classification of imbalanced data, in: AI 2006: Advances in Artificial Intelligence, Springer Berlin Heidelberg, 2006, pp. 264–273.

[34] W.J. Lin, J.J. Chen, Class-imbalanced classifiers for high-dimensional data, Brief. Bioinform. 14 (1) (2013) 13–26.

[35] Y. Zhang, Z.H. Zhou, Cost-sensitive face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (10) (2010) 1758–1769.

[36] A.S. Mahani, S. Shojaee, E. Salajegheh, M. Khatibinia, Hybridizing two-stage meta-heuristic optimization model with weighted least squares support vector machine for optimal shape of double-arch dams, Appl. Soft Comput. 27 (2015) 205–218.

[37] Y.H. Liu, Y.T. Chen, Face recognition using total margin-based adaptive fuzzy support vector machines, IEEE Trans. Neural Netw. 18 (1) (2007) 178–192.

[38] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, L.I. Kuncheva, Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl.-Based Syst. 85 (2015) 96–111.

[39] L. Yijing, G. Haixiang, L. Xiao, L. Yanan, L. Jinling, Adapted ensemble classification algorithm based on multiple classifier system and feature selection for classifying multi-class imbalanced data, Knowl.-Based Syst. 94 (2016) 88–104.

[40] K.Z. Mao, RBF neural network center selection based on fisher ratio class separability measure, IEEE Trans. Neural Netw. 13 (5) (2002) 1211–1217.

[41] B. Schölkopf, R. Herbrich, A.J. Smola, A generalized representer theorem, in: International Conference on Computational Learning Theory, Springer Berlin Heidelberg, 2001, pp. 416–426, July.

[42] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation, Knowl.-Based Syst. 89 (2015) 385–397.

[43] J. Jacques, J. Taillard, D. Delerue, C. Dhaenens, L. Jourdan, Conception of a dominance-based multi-objective local search in the context of classification rule mining in large and imbalanced data sets, Appl. Soft Comput. 34 (2015) 705–720.

[44] K. Napierała, J. Stefanowski, Addressing imbalanced data with argument based rule learning, Expert Syst. Appl. 42 (24) (2015) 9468–9481.

[45] Y. Zhu, Z. Wang, D. Gao, Gravitational fixed radius nearest neighbor for imbalanced problem, Knowl.-Based Syst. 90 (2015) 224–238.

[46] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets, Inform. Sci. 354 (2016) 178–196.

[47] O. Loyola-González, J.F. Martínez-Trinidad, J.A. Carrasco-Ochoa, M. García-Borroto, Study of the impact of resampling methods for contrast pattern based classifiers in imbalanced databases, Neurocomputing 175 (2016) 935–947.

[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[49] N.K. Sreeja, A. Sankar, Pattern matching based classification using ant colony optimization based feature selection, Appl. Soft Comput. 31 (2015) 91–102.

[50] B. Rosner, R.J. Glynn, M.L.T. Lee, The Wilcoxon signed rank test for paired comparisons of clustered data, Biometrics 62 (1) (2006) 185–192.

[51] W. Zhao, T. Tao, E. Zio, W. Wang, A novel hybrid method of parameters tuning in support vector regression for reliability prediction: Particle swarm optimization combined with analytical selection, IEEE Trans. Reliab. 65 (3) (2016) 1393–1405.

[52] V. Cherkassky, Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, Neural Netw. 17 (1) (2004) 113–126.