Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060739

ARTICLE

DCS-SOCP-SVM: A Novel Integrated Sampling and Classification Algorithm for Imbalanced Datasets

Xuewen Mu* **and Bingcong Zhao**

School of Mathematics and Statistics, Xidian University, Xi'an, 710071, China

*Corresponding Author: Xuewen Mu. Email: xwmu@xidian.edu.cn

Received: 08 November 2024; Accepted: 28 January 2025; Published: 16 April 2025

ABSTRACT: When dealing with imbalanced datasets, the traditional support vector machine (SVM) tends to produce a classification hyperplane that is biased towards the majority class, which exhibits poor robustness. This paper proposes a high-performance classification algorithm specifically designed for imbalanced datasets. The proposed method first uses a biased second-order cone programming support vector machine (B-SOCP-SVM) to identify the support vectors (SVs) and non-support vectors (NSVs) in the imbalanced data. Then, it applies the synthetic minority over-sampling technique (SV-SMOTE) to oversample the support vectors of the minority class and uses the random under-sampling technique (NSV-RUS) multiple times to undersample the non-support vectors of the majority class. Combining the above-obtained minority class data set with multiple majority class datasets can obtain multiple new balanced data sets. Finally, SOCP-SVM is used to classify each data set, and the final result is obtained through the integrated algorithm. Experimental results demonstrate that the proposed method performs excellently on imbalanced datasets.

KEYWORDS: DCS-SOCP-SVM; imbalanced datasets; sampling method; ensemble method; integrated algorithm

1 Introduction

Imbalanced datasets lead to a significant challenge to the effectiveness of machine learning models due to the uneven distribution of classes [1]. This issue is prevalent in fields such as cancer malignancy grading [2], industrial system monitoring [3], and text mining [4]. In real-world scenarios, the decision plane obtained using classifiers tends to be biased towards the majority class. From a practical standpoint, identifying minority class samples is more crucial. Therefore, when classifying imbalanced datasets, a customized algorithm is required to meet practical applications.

To better classify datasets, Vapnik proposed the support vector machine (SVM) in the 1990s [5]. SVM is a supervised learning model used for data classification and regression analysis. Its core idea is to find an optimal decision boundary (hyperplane) that maximizes the margin between different classes. The hard-margin SVM is the basic form of SVM, suitable for linearly separable datasets. At the same time, soft-margin SVM was introduced to allow some data points to violate the margin constraints by introducing slack variables through penalty parameters.

Algorithm-based and data-level-based methods are mainly used when dealing with imbalanced datasets [6]. Sampling algorithms are a data level-based method [1], which is mainly divided into three types: oversampling, undersampling, and synthetic sampling algorithms. The main idea is to use a certain strategy to increase the number of minority classes or decrease the number of majority classes before using the classifier, to balance the two-class dataset. Besides sampling algorithms, ensemble algorithms are also

frequently used in imbalanced datasets. Ensemble learning improves prediction accuracy by combining the decisions of multiple classifiers to output a single class label [7], which proves to outperform non-ensemble classification methods. Another particular solution for SVM is choosing a second-order cone programming support vector machine (SOCP-SVM) which is proposed by Nath and Bhattacharyya to replace the original SVM to enhance the classification [8], whose objective function is a quadratic function with a second-order cone constraints. It can adjust the error rates of the two-class datasets to allocate weights, which leads to better performance in classifying imbalanced datasets.

Many scholars have proposed further optimization strategies to address the shortcomings of SVM when dealing with imbalanced datasets. In 2020, Kim and John proposed a hybrid neural network and cost-sensitive support vector machine (hybrid NN-CSSVM) algorithm. This algorithm combines multiple neural network structures to extract features from different modal data, integrating them with SVM optimized through cost-sensitive learning, resulting in better performance on imbalanced datasets [9]. In 2021, Wei and Huang proposed a new method based on the sample feature oversampling technique and multi-class least squares support vector machine, transforming multi-class imbalanced dataset problems into multiple binary imbalanced dataset problems. This algorithm demonstrated superior performance and robustness [10]. In 2022, Yu and Fu proposed a novel cost-sensitive learning model CSSVM for classifying imbalanced datasets. This model combines the advantages of SVM and asymmetric LINEX loss function, achieving efficient cost-sensitive learning by assigning different instance costs [11]. At the same time, Hasib et al. proposed a new hybrid framework named HUSCSLBoost was proposed to address the class imbalance. This framework integrates three key steps: data cleaning using Tomek-Link to eliminate noise, data balancing via random under-sampling to create balanced subsets, and cost-sensitive learning through CSLBoost, which incorporates cost concepts based on sample hardness [12]. These recent research methods tend to combine different models and algorithms, the method which combines multiple models proposed in this paper outperforms the traditional SVM. In 2023, Shajalal, Md, and Hajek, Petr proposed a deep neural network model has been proposed for predicting product backorder, incorporating advanced data balancing techniques such as SMOTE, weight boosting, and hybrid sampling methods. These methods optimize training data distribution, enabling the model to achieve state-of-the-art performance in standard metrics and profit-based evaluations [13]. Tanveer, Mishra, and Richhariya propose a novel fuzzy-based approach to handle class-imbalanced and noisy datasets. Two models, IF-RELSTSVM and F-RELSTSVM, are introduced, leveraging intuitionistic and hyperplane-based fuzzy membership functions, respectively. These methods dynamically calculate membership values through projection, effectively addressing noise and imbalance [14]. In 2024, Rezvani, Pourpanah, and Lim provide a comprehensive review of methods addressing class imbalance in SVM and its variants. They categorize SVM-based approaches into three groups: re-sampling, algorithmic, and fusion methods. The study highlights the strengths and limitations of each category, showing that fusion methods often achieve the best performance by combining re-sampling and algorithmic improvements, albeit at a higher computational cost [15]. Meanwhile, Fofanah, Abdul Joseph, and Chen, David proposed novel architecture for imbalanced graph data named GATE-GNN, addressing the limitations of traditional methods like resampling and reweighting by leveraging ensemble modules and spatial embeddings [16]. Tested on benchmark graph datasets, GATE-GNN outperforms leading models, achieving superior classification accuracy and reduced training time, highlighting its practical application potential. These recent research methods tend to combine different models and algorithms, the method that combines multiple models proposed in this paper outperforms the classifier that has not been optimized.

Gao and Jian proposed an ensemble sampling learning algorithm based on SVM [17]. This algorithm first uses biased support vector machine (B-SVM) to divide vectors into support vectors (samples near the decision margin) and non-support vectors (samples far from the decision margin), then applies synthetic

minority over-sampling technique (SV-SMOTE) to increase support vectors in the dataset and uses random undersampling technique (NSV-RUS) to eliminate non-support vectors without removing support vectors. Repeatedly using this method produces multiple majority-class samples, which combined with minority-class samples processed by SV-SMOTE, yield multiple balanced datasets. Each dataset is classified using an SVM classifier, and the ensemble algorithm combines different classifiers through voting to form an ensemble classifier (SVMen).

The ensemble algorithm [17] presented in this paper enhances the classification of imbalanced datasets by incorporating second-order cone programming support vector machines (SOCP-SVM) into the ensemble framework. Traditional SVMs struggle with noise, outliers, and skewed decision boundaries in imbalanced data, resulting in poor classification of the minority class. SOCP-SVM addresses these limitations by optimizing the decision boundary and cost function to mitigate the majority class's influence. The algorithm first applies a biased second-order cone programming support vector machine(B-SOCP-SVM) to differentiate between support vectors and non-support vectors. Then use the synthetic minority over-sampling technique (SV-SMOTE) to generate additional minority class samples based on the support vectors. For the majority class, it performs a variant of random under-sampling (NSV-RUS) multiple times to produce balanced datasets. Each dataset is classified using SOCP-SVM, and the final classification is derived through the SOCP-SVM ensemble (SOCP-SVMen). This approach proves to outperform the original SOCP-SVM. The main contribution of this paper is:

(i) We propose a novel hybrid approach that combines a second-order cone programming support vector machine (SOCP-SVM) with synthetic sampling techniques to address imbalanced datasets. This integration improves both memorization and generalization performance, especially in cases with noisy or skewed data.

(ii) By introducing SV-SMOTE, a synthetic minority over-sampling technique that operates exclusively on support vectors. This method ensures that the most crucial minority class samples contribute effectively to the decision boundary, reducing the negative impact of noise and outliers that often affect traditional SVM models. Moreover, a modified random under-sampling algorithm (NSV-RUS) is applied to the majority class, carefully preserving support vectors.

(iii) The combination of SOCP-SVM and these improved sampling methods leads to an ensemble learning framework, enhancing the classifier's ability to generalize across imbalanced datasets. The ensemble model, referred to as SOCP-SVMen, exhibits superior classification performance in comparison to traditional SVM methods, particularly in classifying minority class samples.

(iv) Experimental results confirm the strong performance of the proposed hybrid model, highlighting improvements in both minority-class classification and overall model accuracy.

The structure of this article is as follows: Section 2 describes the basic knowledge required to understand the algorithm, Section 3 provides the detailed implementation steps of the algorithm, Section 4 presents the experimental results of the proposed method compared to the control methods, and Section 5 concludes with the final remarks.

2 Basic Formulation

2.1 Support Vector Machine (SVM)

In the 1990s, Vapnik introduced a Support Vector Machine (SVM) to enhance the performance of classifiers [5]. Consider a dataset $T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m)\}$ where $x_i \in R^n$, $y_i \in \{1, -1\}$, and $i = 1, 2, \cdots, m$ (m denoting the number of data points). The task is to find a hyperplane that acts as a decision boundary to separate samples of different classes. The goal of SVM is to find a hyperplane that maximizes

the margin between the two classes, improving the classifier's generalization capability. This hyperplane can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 \\ &s.t.\ y_i\left(w^T x_i + b\right) \geq 1, i = 1, 2, \cdots, m \end{aligned} \tag{1}$$

where w is the normal vector to the hyperplane, b is the offset term, x_i is the sample point, and y_i is the class label of the sample point (+1 or −1).

Soft margin SVM introduces a slack variable ξ_i to handle the linearly inseparable cases, the optimization problem can be formulated as:

$$\begin{aligned} &\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ &s.t.\ y_i\left(w^T x_i + b\right) \geq 1 - \xi_i,\ \ i = 1, 2, \cdots, m \\ &\quad \xi_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{2}$$

where C is a regularization parameter that balances margin maximization and classification error penalty. The decision boundary of the classifier can be obtained by solving the above optimization problem.

Although SVMs have been widely applied over the past 20 years, their computational complexity and suboptimal performance in handling imbalanced datasets have notable drawbacks. Better classification results will be achieved if the constraint is replaced by a second-order cone [8]. For non-linearly separable datasets, the data can be mapped to a higher-dimensional space using a kernel function to make it linearly separable in that space. Common kernel functions include the linear kernel, polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel.

2.2 Sampling Algorithms

To better address imbalanced datasets, researchers have developed various sampling algorithms, including oversampling, undersampling, and synthetic sampling methods. Oversampling algorithms balance datasets by increasing the number of minority class samples, thereby improving the classifier's ability to recognize minority classes. The Synthetic Minority Over-sampling Technique (SMOTE) is the most representative method [18]. SMOTE generates new samples by randomly selecting k nearest neighbors of minority class samples and performing linear interpolation between these points to achieve a balanced state. SMOTE has been extensively studied for its excellent performance, and its common variants include Borderline-SMOTE [19], which tends to generate more synthetic samples at the boundary to improve classifier performance. AdaSYN [20], which generates samples based on the density of minority class samples, creates more samples in low-density regions to enhance the classifier's learning capability in these areas. These superior performances have led scientists to categorize these methods under a new term: SMOTE-like algorithms. The basic form of SMOTE can be expressed by the following formula:

$$x_{new} = x_i + \lambda\left(x_j - x_i\right) \tag{3}$$

where x_{new} is the newly generated data point, the x_i is the data point we choose in the minority class, x_j is the neighbor of the x_i and λ is a random value from $(0, 1)$.

Undersampling algorithms balance datasets by reducing the number of majority-class samples. Common undersampling methods include Edited Nearest Neighbor (ENN) [21] and Tomek Link [22] algorithms. ENN maintains data balance by removing majority class samples whose labels are inconsistent with their

nearest neighbors. Specifically, for each majority class sample, the algorithm identifies its k nearest neighbors. If the sample's label is inconsistent with the majority of its neighbors' labels, then it is removed. This method effectively eliminates noise and boundary samples, thereby improving classifier performance. Tomek Link removes majority class samples which are pairs of samples from different classes that are each other's nearest neighbors. By identifying all Tomek Link pairs in the dataset and removing the majority of class samples in these pairs, the class boundaries are clarified, enabling the classifier to better distinguish between different classes.

Synthetic sampling algorithms combine oversampling and undersampling methods to better handle imbalanced data. Common synthetic sampling algorithms include SMOTE-ENN [23] and SMOTE-Tomek [24]. SMOTE-ENN first generates new minority class samples using SMOTE, then cleans the dataset using ENN. SMOTE-Tomek first generates new minority class samples using SMOTE, then cleans the dataset using Tomek Link. Synthetic sampling algorithms combine the advantages of both oversampling and undersampling methods, achieving a balanced dataset while eliminating noise. By effectively applying different sampling algorithms, imbalanced datasets can be managed, enhancing the classifier's ability to recognize minority classes and improving overall classification performance. The sampling method adopted in this paper is also a synthetic sampling algorithm.

The aforementioned sampling algorithms are commonly used for imbalanced datasets but also have critical issues. Oversampling can introduce noisy points that interfere with the classifier while undersampling can remove data points useful for the hyperplane. To address these issues, the proposed algorithm differentiates between support vectors (SVs) and non-support vectors (NSVs), applying oversampling only on SVs and undersampling only on NSVs. Specifically, the method employs a variant of SMOTE, called Support Vector-SMOTE (SV-SMOTE), for oversampling support vectors, and a Random Undersampling algorithm (NSV-RUS) for undersampling non-support vectors.

2.3 Ensemble Algorithms

To enhance classifier performance and prediction accuracy, researchers have developed various ensemble algorithms, Bagging, Boosting, and Stacking are the most representative methods. Bagging [25] improves model stability and accuracy by constructing multiple independent classifiers and averaging or voting on their predictions. Specifically, Bagging generates multiple sub-datasets by randomly sampling from the original dataset, training a classifier on each sub-dataset. The final prediction is the average or majority vote of all classifiers' predictions. A classic implementation of Bagging is Random Forest, which integrates multiple decision tree classifiers, significantly improving model generalization and noise resistance.

Ensemble algorithms improve prediction accuracy and robustness by combining the predictions of multiple models, reducing the errors that a single model might make. Combining multiple models also mitigates the impact of noise and bias in the training data, making the overall model more robust and reliable. In this method, multiple datasets are obtained through random undersampling, and each dataset yields a model. The final prediction accuracy is enhanced by combining these models using a voting mechanism.

The work presented in the paper introduces a novel approach to mitigate issues inherent in traditional sampling methods when dealing with imbalanced datasets. By differentiating between SVs and NSVs and then applying specific sampling strategies, the proposed DCS-SOCP-SVM method demonstrates improved classification performance and offers a promising direction for future research in handling imbalanced data.

2.4 Deep Learning in the Imbalanced Dataset

Deep learning has demonstrated remarkable success in various domains, including computer vision, natural language processing, and medical diagnosis. However, its performance in imbalanced datasets presents unique challenges. Unlike traditional machine learning algorithms, deep learning models rely heavily on large volumes of data for training. When faced with imbalanced datasets, these models often become biased toward the majority class, leading to suboptimal performance in minority class prediction. This issue arises primarily because the loss functions commonly used, such as cross-entropy, inherently prioritize overall accuracy rather than focusing on underrepresented classes.

To address these challenges, researchers have proposed various strategies. One prominent approach involves the design of specialized loss functions, such as the focal loss introduced by Lin et al. [26], which dynamically scales the loss for hard-to-classify examples, thereby focusing the learning process on minority classes. Another significant advancement is the use of class-balanced loss [27], which reweights the contributions of each class based on their inverse frequency to mitigate the impact of class imbalance. Additionally, ensemble learning and deep learning have shown promise, with ensemble methods combining multiple models and recent efforts integrating them with deep learning to enhance predictive performance [28]. This article proposes Deep Density Hybrid Sampling (DDHS) to address imbalanced data by learning a low-dimensional latent space, preserving class proximity, and generating diverse synthetic samples. Combined with boosting, DDHS boosting outperforms other ensemble methods in experiments [29].

3 Proposed Method: DCS-SOCP-SVM

In this section, we propose a novel integrated sampling method for imbalanced datasets, which extends the method proposed by Jian et al. [17]. Specifically, we replace the classifier used to differentiate support vectors from B-SVM to B-SOCP-SVM and the general classifier from SVM to SOCP-SVM. Section 3.1 presents the specific implementation of the algorithm, Section 3.2 provides details on the classifiers used, Section 3.3 gives information on the sampling algorithms employed, and Section 3.4 details the ensemble algorithm used in our method.

3.1 Overview of Proposed Method: DCS-SOCP-SVM

The following describes the specific steps of the DCS-SOCP-SVM algorithm:

Step 1: Identify the support vectors (SVs) and non-support vectors (NSVs) of both minority and majority classes using the B-SOCP-SVM method (Eq. (11)). Support Vectors are the data points for which the constraints involving $\boldsymbol{\xi_i}$ are active, whereas Non-Support Vectors lie outside the margin and do not influence the decision boundary.

Step 2: Apply the Synthetic Minority Over-sampling Technique (SMOTE) to the support vectors (SVs) of the minority class. The new minority class dataset consists of new support vectors (SVs) and the original non-support vectors (NSVs) from the minority class.

Step 3: Employ various random undersampling methods to eliminate some of the non-support vectors (NSVs) in the majority class, resulting in multiple new sets of NSVs, while also removing noisy support vectors (SVs) from the majority class. The new majority class datasets consist of the processed support vectors (SVs) and the new sets of NSVs.

Step 4: Combining the new minority class dataset with each of the new majority class datasets to obtain multiple balanced training sets. These balanced training sets are then used to create an ensemble of multiple SOCP-SVM models. For each sample point, the final class is determined by majority voting among the classifiers in the ensemble, referred to as SOCP-SVMen.

In addition to the step-by-step breakdown provided in Algorithm 1, the entire process of the proposed method is represented in Fig. 1. This figure illustrates the key stages and transitions between each step of the algorithm, offering a clearer view of the workflow and interactions among different components.

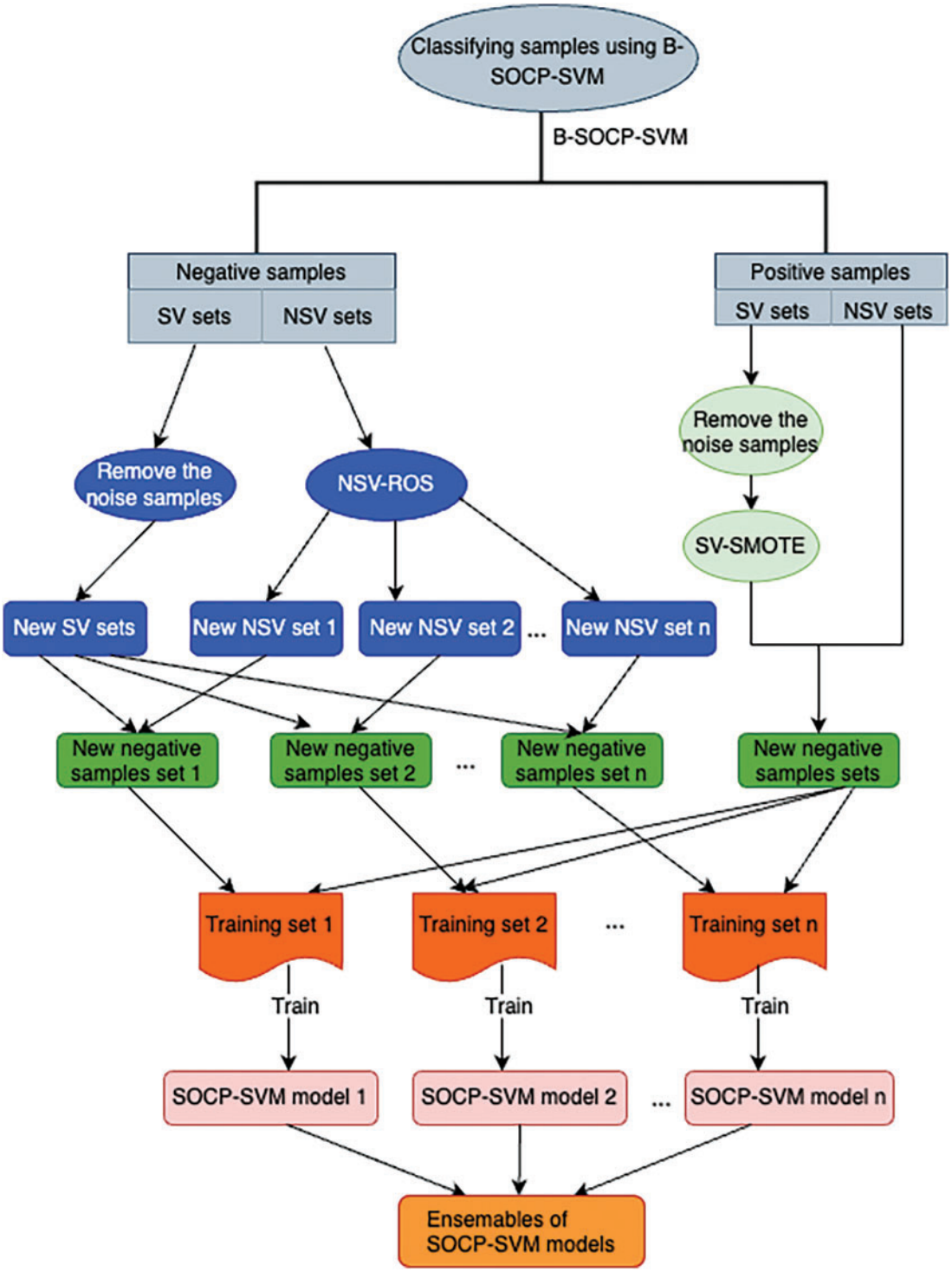

Figure 1: DCS-SOCP-SVM

3.2 Second-Order Cone Programming Support Vector Machine (SOCP-SVM)

The major innovation of this method is to optimize the original method [17] by replacing the classifiers for distinguishing support vectors (B-SVM) and the general classifier (SVM) with second-order cone programming. This results in B-SOCP-SVM and SOCP-SVM.

Suppose X_1 and X_2 are random vectors generating positive and negative class samples, and their means and covariance matrices are $(\mu_i, \sum_i)$, $i = 1, 2$, where $\sum_i \in R^{n\times n}$ is a symmetric positive definite matrix. Consider the following quadratic chance-constrained formulation:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &s.t.\ \ \Pr\left\{w^T X_1 + b \geq 1\right\} \geq \eta_1 \\ &\qquad \Pr\left\{w^T X_2 + b \leq -1\right\} \geq \eta_2 \end{aligned} \tag{4}$$

where $\eta_i \in (0,1)$, $i = 1, 2$ represents the probability that a random sample lies on one side of the hyperplane with at least a probability of η_i. To ensure that the correct classification rate of each class exceeds η_i for the worst-case distribution (μ_i, Σ_i), the constraints can be transformed as follows:

$$\begin{aligned} &\inf_{X_1\sim(\mu_1,\Sigma_1)} \Pr\left\{w^T X_1 + b \geq 1\right\} \geq \eta_1 \\ &\inf_{X_2\sim(\mu_2,\Sigma_2)} \Pr\left\{w^T X_2 + b \leq -1\right\} \geq \eta_2 \end{aligned} \tag{5}$$

where $X \sim (\mu, \sum)$ denotes the family of distributions with a common mean μ and covariance $\sum$. In order to better solve the above constraints, we introduce the multivariate Chebyshev inequality, which can be described as follows:

Theorem 1: *Let X be an n-dimensional random vector. The mean and covariance of X are $\mu \in R^n$ and $\sum \in R^{n\times n}$, respectively. Given $w \in R^n$, $w \neq 0$ and $b \in R$, define:*

$$H(w,b) = \left\{z \,|\, w^T z < b,\ z \in R^n\right\}$$

be half-space. Then:

$$\Pr\left\{z \in R^n\right\} = \frac{s^2}{s^2 + w^T \sum w}$$

The proof of Theorem 1 can be found in the paper [8].

Apply the Theorem 1 with $X = X_i$ and $H = H_i$, the constraints can be transformed by setting:

$$\Pr\left\{X_i \in R^n\right\} = \frac{w^T \sum_i w}{\left(w^T\mu - b\right)^2 + w^T \sum_i w} \tag{6}$$

which can be transformed as follows:

$$w^T\mu - b \leq \sqrt{\frac{1-\eta}{\eta}}\sqrt{w^T \sum_i w} \tag{7}$$

Based on the Theorem 1, the inequalities (5) can be transformed as follows:

$$\begin{aligned} &w^T\mu_1 + b \geq 1 + \kappa_1\sqrt{w^T\sum_1 w} \\ &-\left(w^T\mu_2 + b\right) \geq 1 + \kappa_2\sqrt{w^T\sum_2 w} \end{aligned} \tag{8}$$

where $\kappa_k = \sqrt{\frac{\eta_k}{1-\eta_k}}, k = 1, 2$. Therefore, coupled with the model (4), we obtain the following model:

$$\begin{aligned} &\min_{w,b} \frac{1}{2}\|w\|^2 \\ &s.t.\, w^T\mu_1 + b \geq 1 + \kappa_1\sqrt{w^T\sum_1 w} \\ &-\left(w^T\mu_2 + b\right) \geq 1 + \kappa_2\sqrt{w^T\sum_2 w} \end{aligned} \tag{9}$$

where $\Sigma_i = S_i S_i^T, i = 1, 2$. To better handle non-linear separable problems, slack variable ξ_i is introduced to form a soft-margin second-order cone programming SVM, making it suitable for some non-linear separable scenarios, the formula is described as follows:

$$\begin{aligned} &\min_{w,b,\xi} -r + C\sum_{i=1}^{m}\xi_i \\ &s.t.\, w^T\mu_1 + b \geq r - \xi_i + \kappa_1\sqrt{w^T\sum_1 w} \\ &-\left(w^T\mu_2 + b\right) \geq r - \xi_i + \kappa_2\sqrt{w^T\sum_2 w} \\ &-1 \leq w_j \leq 1, j = 1, 2, \cdots, n \\ &\xi_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{10}$$

The classifier above is the most common in the algorithm. Additionally, to process the original dataset and distinguish support vectors from non-support vectors, we optimize it using second-order cone programming to enhance performance, resulting in B-SOCP-SVM. This approach is essentially a soft-margin SVM combined with cost-sensitive learning methods, resulting in:

$$\begin{aligned} &\min_{w,b,\xi} -r + C_+\sum_{i\in I^+}^{m}\xi_i + C_+\sum_{i\in I^-}^{m}\xi_i \\ &s.t.\, w^T\mu_1 + b \geq r - \xi_i + \kappa_1\sqrt{w^T\sum_1 w} \\ &-\left(w^T\mu_2 + b\right) \geq r - \xi_i + \kappa_2\sqrt{w^T\sum_2 w} \\ &-1 \leq w_j \leq 1, j = 1, 2, \cdots, n \\ &\xi_i \geq 0, i = 1, 2, \cdots, m \end{aligned} \tag{11}$$

where C_+ and C_- are penalty parameters corresponding to each class. Eqs. (10) and (11) are the primary classifiers used in the algorithm.

3.3 The Applied Two Sampling Methods

Two main sampling algorithms are employed in our method: a variant of the SMOTE (Synthetic Minority Over-sampling Technique) algorithm and a variant of the random undersampling technique to create balanced datasets. The SMOTE algorithm generates synthetic samples by linear interpolation between two neighboring samples, thereby increasing the number of minority class samples to balance the dataset.

Support vectors (SVs) have a greater influence on the decision hyperplane compared to non-support vectors (NSVs). In this study, synthetic samples are generated exclusively from the support vectors of the minority class using SMOTE, while keeping the number of non-support vectors constant. This method is referred to as SV-SMOTE. The formula used for generating synthetic samples is as follows:

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{sv} + \lambda \left(\boldsymbol{x}_{knn} - \boldsymbol{x}_{sv} \right) \tag{12}$$

where $\boldsymbol{\lambda}$ is a random value in the range is (0, 1), $\boldsymbol{x_{sv}}$ represents a support vector from the minority class, and $\boldsymbol{x_{knn}}$ is a random sample from the k-nearest neighbors of $\boldsymbol{x_{sv}}$. Additionally, let m denote the number of samples from the majority class among the k-nearest neighbors. If the ratio $\frac{\boldsymbol{k}}{\boldsymbol{m}}$ exceeds or equals 0.8 based on past experience, the sample is considered noise and should be discarded. In our paperwork, $\boldsymbol{k}$ is set to 5. The new dataset for the minority class comprises synthetic support vectors and the original non-support vectors from the minority class. The step of Algorithm 1 can be described as follows:

Algorithm 1: SV-SMOTE algorithm

Input: Minority class support vectors $\boldsymbol{X}_{sv}$, k-nearest neighbors $\boldsymbol{k}$
Output: New synthetic samples $\boldsymbol{X}_{new}$
for each support vector $\boldsymbol{x}_{sv} \in \boldsymbol{X}_{sv}$ **do**
  Find the k-nearest neighbors of $\boldsymbol{x}_{sv}$, denoted as $\boldsymbol{X}_{knn}$
  for each neighbor $\boldsymbol{x}_{knn} \in \boldsymbol{X}_{knn}$ **do**
    Generate a random value $\lambda \in (0,1)$
    Compute the synthetic sample using linear interpolation:
      $$\boldsymbol{x}_{new} = \boldsymbol{x}_{sv} + \lambda \left(\boldsymbol{x}_{knn} - \boldsymbol{x}_{sv} \right)$$
    Let m be the number of majority class samples in $\boldsymbol{X}_{knn}$
    if the ratio $\frac{k}{m} > 0.8$ **then**
      Consider the sample noise and discard it
    else
      Add $\boldsymbol{x}_{new}$ to the set of synthetic samples.
    end if
  end for
end for
Return the set of synthetic samples $\boldsymbol{X}_{new}$

The random undersampling algorithm randomly selects and removes a portion of the samples from the majority class. However, since support vectors play a crucial role in the performance of the SVM, we apply random undersampling without removing support vectors (NSV-RUS) to reduce the number of non-support vectors in the majority class to preserve critical information. The goal of the sampling strategy is to ensure that the number of majority-class samples equals the number of minority-class samples, which can be achieved using the random undersampling method. Specifically, the support vectors from the majority class are retained, and the remaining majority class samples are randomly selected until the number of majority class samples equals the number of minority class samples. The steps of the Algorithm 2 can be described as follows:

Algorithm 2: Random Undersampling Algorithm (NSV-RUS)

1. **Input:** Majority class samples $S_{majority}$, Minority class samples S_{minority}
2. Output: New balanced dataset S_{balanced}
3. Let $SV_{majority}$ be the support vectors from $S_{majority}$
4. Let $NSV_{majority}$ be the non-support vectors from $S_{majority}$
5. Retain all support vectors $SV_{majority}$
6. Randomly select samples from $NSV_{majority}$ until the total number of majority samples equals the number of minority samples:
$$\boldsymbol{n}_{majority} = \boldsymbol{n}_{minority}$$
7. Form the new balanced dataset $S_{balanced} = SV_{majority} \cup S'_{majority} \cup S_{\text{minority}}$, where $S'_{majority}$ is the subset of randomly selected majority class samples
8. **Return** S_{balanced}

3.4 Ensemble SOCP-SVM Classifier (SOCP-SVMen)

To maintain diversity among datasets that have been undersampling, multiple new majority class NSV sets are obtained by multiple NSV-RUS methods. These new NSV sets are combined with the original SV set to form multiple new majority-class datasets. Based on these majority class datasets, a new balanced training set is created by combining the new minority class dataset with each of the new majority class datasets. Each balanced training set generates a SOCP-SVM classifier model, resulting in multiple SOCP-SVM classifiers. The SOCP-SVMen can be created and defined as follows:

$$\boldsymbol{SOCP-SVMen} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{SOCP-SVM_i} \tag{13}$$

For each sample point, the final result is determined by the majority vote among the predictions of the individual classifiers.

4 Numerical Experiments

In this section, we evaluate the performance of the proposed DCS-SOCP-SVM method. Eight datasets are selected in our paperwork: cmc2, Haberman, wilt2, yeast, abalone, ecoli, car, and balance. To demonstrate the advantages of the proposed algorithm, the selected datasets mainly focus on those with high imbalance characteristics. These datasets can be downloaded from the University of California, Irvine (UCI) repository [30]. Different methods are applied to process the original datasets and compare their performance, including the original SOCP-SVM, SMOTE-SOCP-SVM (applying the SMOTE algorithm to the original dataset before using SOCP-SVM for classification), ENN-SOCP-SVM (applying the ENN algorithm to the original dataset before using before SOCP-SVM for classification), DCS-SVM, and SOCP-DCS-SVM. There are many evaluations that can be used to assess the dataset [31]. In this paper, the comparison of these models is based on three evaluation metrics: G-mean [32], F-measure [33], and AUC [34]. These evaluation metrics are detailed in Section 4.1. The main results and a summary of classification performance are provided in Section 4.2.

4.1 Data Information and Evaluation Metrics

The proposed method and other models were applied to eight datasets from the UCI repository for binary classification tasks [30]. Table 1 presents basic information about the selected standard datasets, including the target minority class, the number of variables, the number of samples in the datasets, and

the imbalance ratio of the datasets. The definition of imbalance ratio is the number of positive samples over the number of negative samples. More information about the datasets can be found on the UCI repository website.

Table 1: Benchmark data sets for class learning

Dataset	Target	Variables	Examples	Imbalance ratio
Ecoli	M	30	569	1.7
Haberman	Class2	3	306	2.78
Wilt2	Class1	5	4839	17.54
Yeast	NUC	8	1332	14.9
Abalone	Class7	8	4177	9.68
Balance	B	4	625	11.75
Car	Class1	6	1729	2.33
Cmc2	Class1	9	1473	1.34

Given the emphasis on minority classes, traditional evaluation metrics such as accuracy and error rate are not suitable for imbalanced datasets. The main evaluation relies on three metrics to comprehensively assess the results: G-mean [32], F-measure [33], and AUC [34]. These three metrics better measure the classifier's performance on imbalanced datasets, as defined below.

4.1.1 G-Mean

The definition of G-mean is the product of the true positive rate and the true negative rate, a commonly used solution for evaluating imbalanced datasets [32]. It is defined as follows:

$$G - mean = \sqrt{\frac{TP \times TN}{(TP + FN) + (TN + FP)}} \tag{14}$$

Here, TP is the number of true positives, FN is the number of false negatives, TN is the number of true negatives, and FP is the number of false positives. G-mean effectively calculates the geometric mean of class accuracies for both the majority (negative) and minority (positive) classes. If the classifier is biased towards one class, the value will approach 0; if both classes are correctly classified, the value will approach 1.

4.1.2 F-Measure

The F-measure balances Recall and Precision [33]. It is defined as follows:

$$F - measure = \frac{2 \times Precision + Recall}{Precision + Recall} \tag{15}$$

This formula includes a coefficient β which aims to adjust the emphasis on precision rate or recall rate. Typically, β is set to 1. Recall is defined as the proportion of actual positive instances correctly identified by the model. Precision is defined as the proportion of predicted positive samples that are actually positive. An improvement in either Recall or Precision increases the F-measure value, indicating better classification performance for the positive class.

4.1.3 AUC

AUC (Area Under the Curve) is another evaluation metric in imbalanced dataset as it is insensitive to changes in class distribution particularly for binary classification tasks. It is estimated through various techniques, the most used is the trapezoidal method, which is a geometrical method based on linear interpolation between each point on the ROC curve [34]. The approximation of AUC which is widely used in binary classification is:

$$AUC = \frac{1 + \frac{TP}{TP+FN} - \frac{FP}{FP+TN}}{2} \tag{16}$$

It provides a simple numerical assessment of the overall classifier performance ranging from 0.5 to 1, 0.5 indicates performance equivalent to random guessing, and 1 denotes perfect classification.

4.2 Experimental Results

In this section, we compare the performance of the proposed DCS method with four other methods on the test datasets:

1. Original SOCP-SVM without sampling the test dataset [8].
2. ENN-SOCP-SVM (using the ENN method on the test dataset before using SOCP-SVM).
3. SMOTE-SOCP-SVM (applying the SMOTE algorithm to the test dataset).
4. DCS-SVM using a regular support vector machine as the classifier [17].

The DCS-SOCP-SVM method aims to address overfitting, information loss in undersampling methods, and trivial information increase in oversampling methods, as well as reducing the computational cost brought by support vector machines. In the experiments, SOCP-SVM was used as the classifier. Each method was repeated five times, with the average results used as the final outcome. These experiments were implemented using MATLAB2023a and Python3.11.2 on a personal computer equipped with an AMD4600, 3.70 GHz processor.

The first method directly operates on the original dataset without sampling (NS method). The second method uses the ENN method for sampling; the reason for not using random sampling as a control is that the ENN algorithm generally performs better than random undersampling. In the experiments, random undersampling is used to obtain more classifiers. In this method, k is chosen as 3, which means that for each sample point, if the nearest 3 elements of the same class outnumber those of the different class, it is retained; otherwise, it is removed. The strategy in the third method is to make the number of majority class samples equal to the number of minority class samples as much as possible to maximize the algorithm's performance. Both major sampling algorithms were implemented using Python's imbalanced-learn library [35]. The parameters and implementations of the fourth method are consistent with those proposed in Chapter 3, with the only difference being that B-SVM is used to distinguish support vectors and non-support vectors, and a traditional support vector machine is used as the final classifier instead of a second-order cone programming support vector machine, resulting in an ensemble SVM (SVMen). The fourth method and the proposed algorithm were both implemented using MATLAB.

Given the importance of minority class samples in imbalanced datasets, the G-mean and F-measure of the minority class samples were chosen to evaluate. As shown in Table 2, the proposed DCS-SOCP-SVM method achieved the highest G-mean of 0.85721. The proposed algorithm achieved the highest value on six datasets, while the SMOTE and SOCP-SVM combined algorithms achieved the highest value on four datasets ranking second. The undersampling algorithm and the ordinary DCS algorithm achieved the highest value

on the two datasets, outperforming the traditional SOCP-SVM. Table 3 shows that the proposed DCS-SOCP-SVM method achieved the highest average value of 0.94611. The proposed algorithm outperformed other datasets on all datasets or was paired with them, while the SMOTE and SOCP-SVM combined algorithm achieved the highest value on three datasets, ranking second. The undersampling algorithm and the ordinary DCS algorithm achieved the highest value on two datasets, while the unoptimized SOCP-SVM achieved the highest value on one dataset. Table 4 shows that the proposed DCS-SOCP-SVM algorithm achieved the highest value of 0.84815. The proposed algorithm outperformed the other datasets on five datasets, while the SOCP-SVM optimized with SMOTE achieved the highest value on four datasets. The ENN-optimized SOCP-SVM and the original DCS algorithm followed closely, achieving the highest value on three and two datasets, respectively. The unoptimized classifier performed the worst, with the lowest performance on all datasets. Combining the data from Tables 2–4, it can be observed that using DCS as a sampling method outperforms traditional oversampling algorithms such as SMOTE and undersampling algorithms such as ENN, indicating better performance of the new sampling algorithm compared to traditional sampling algorithms. For algorithms that use DCS, the method using SOCP-SVM as the classifier outperforms the method using SVM as the classifier, indicating that the classifier mentioned in this paper outperforms traditional SVM classifiers. Combining the data and the conclusions from the tables, it can be concluded that the method proposed in this paper is more advantageous than traditional methods.

Table 2: G-mean

	NS-SOCP-SVM	SMOTE-SOCP-SVM	ENN-SOCP-SVM	DCS-SVM	DCS-SOCP-SVM
Abalone	0.82848	0.83980	0.82014	0.82204	0.83980
Balance	0.96666	0.99959	0.97510	0.98272	0.95323
Car	0.74824	0.74501	0.74898	0.74756	0.75881
Cmc2	0.72507	0.73308	0.71169	0.73929	0.74738
Ecoli	0.94868	0.96225	0.96225	0.96225	0.96225
Haberman	0.75593	0.92582	0.82752	0.87833	0.91352
Wilt2	0.69063	0.69861	0.69177	0.69063	0.71039
Yeast	0.96515	0.96077	0.96515	0.96515	0.97236

Table 3: F-measure

	NS-SOCP-SVM	SMOTE-SOCP-SVM	ENN-SOCP-SVM	DCS-SVM	DCS-SOCP-SVM
Abalone	0.89170	0.89296	0.88193	0.90713	0.91005
Balance	0.99187	0.99828	0.99187	0.99187	0.99828
Car	0.88387	0.89102	0.88387	0.89731	0.90352
Cmc2	0.72507	0.82828	0.79828	0.79828	0.83202
Ecoli	0.98462	0.96269	0.98462	0.98462	0.98462
Haberman	0.92894	0.94915	0.94915	0.94915	0.96188
Wilt2	0.97224	0.98532	0.98532	0.98532	0.98532
Yeast	0.98983	0.99320	0.96963	0.97784	0.99320

Table 4: AUC

	NS-SOCP-SVM	SMOTE-SOCP-SVM	ENN-SOCP-SVM	DCS-SVM	DCS-SOCP-SVM
Abalone	0.76942	0.84065	0.82280	0.83710	0.84065
Balance	0.96721	0.96967	0.97541	0.97541	0.98317
Car	0.75301	0.74612	0.76488	0.76032	0.75382
Cmc2	0.72507	0.73337	0.71528	0.72760	0.74210
Ecoli	0.88557	0.88740	0.88740	0.88740	0.88740
Haberman	0.76902	0.92857	0.92857	0.90381	0.91654
Wilt2	0.69208	0.69960	0.69315	0.69315	0.69960
Yeast	0.96575	0.96154	0.96154	0.96575	0.96154

5 Conclusion

Sampling methods are commonly employed to address the issue of skewed sample distribution in imbalanced datasets. However, these methods can lead to the loss of crucial information or the introduction of irrelevant information during classification, ultimately affecting the prediction accuracy of minority class samples in imbalanced datasets. Given the different contributions of support vectors (SVs) and non-support vectors (NSVs) to classification, this paper proposes a new ensemble sampling classification algorithm based on SOCP-SVM (DCS-SOCP-SVM). In this method, SVs and NSVs are identified through the B-SOCP-SVM method. The SV-SMOTE is used to increase the number of minority class samples, and NSV-RUS is employed multiple times to reduce the number of majority class samples, resulting in different training sets. Imbalanced datasets were selected from the UCI repository and experiments were conducted using various existing sampling methods. The results demonstrate that the proposed DCS-SOCP-SVM method outperforms other methods (SOCP-SVM, ENN-SOCP-SVM, SMOTE-SOCP-SVM, and DCS-SVM).

The proposed method offers significant potential for future optimization. For example, the SMOTE algorithm used in this approach could be substituted with more advanced techniques such as ADASYN or Borderline-SMOTE to improve performance. Similarly, the random undersampling component could be enhanced by leveraging algorithms like Edited Nearest Neighbors (ENN), and various datasets could be generated by experimenting with different parameter configurations. Additionally, more sophisticated ensemble learning techniques, such as Boosting or Stacking, could be applied to combine classifiers or integrate diverse models, enabling a deeper exploration of performance improvements. Moreover, this method can be extended to handle multi-class datasets, broadening its applicability and impact.

Acknowledgement: Not applicable.

Funding Statement: This work was supported by the Natural Science Basic Research Program of Shaanxi (Program No. 2024JC-YBMS-026).

Author Contributions: The authors confirm their contribution to the paper as follows: study conception and design: Xuewen Mu; analysis and interpretation of results: Xuewen Mu; draft manuscript preparation: Bingcong Zhao. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are available from the corresponding author, Bingcong Zhao, upon reasonable request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Abbreviation

SVM	Support vector machine
SOCP-SVM	Second-order cone programming support vector machine
SV	Support vector
NSV	Non-support vector
B-SVM	Biased support vector machine
B-SOCP-SVM	Biased second-order cone programming support vector machine
SV-SMOTE	Support vector-SMOTE
NSV-RUS	Random under-sampling removing non-support vectors
SVMen	Support vector machine ensemble classifier
SOCP-SVMen	Second-order cone programming support vector machine ensemble classifier
hybrid NN-CSSVM	Hybrid neural network and cost-sensitive support vector machine

References

1. Krawczyk B. Learning from imbalanced data: open challenges and future directions. Prog Artif Intell. 2016;5(4):221–32. doi:10.1007/s13748-016-0094-0.
2. Krawczyk B, Galar M, Jeleń Ł, Herrera F. Evolutionary undersampling boosting for imbalanced classification of breast cancer malignancy. Appl Soft Comput. 2016;38(8):714–26. doi:10.1016/j.asoc.2015.08.060.
3. Ramentol E, Gondres I, Lajes S, Bello R, Caballero Y, Cornelis C, et al. Fuzzy-rough imbalanced learning for the diagnosis of high voltage circuit breaker maintenance: the SMOTE-FRST-2T algorithm. Eng Appl Artif Intell. 2016;48(17):134–9. doi:10.1016/j.engappai.2015.10.009.
4. Munkhdalai T, Namsrai OE, Ryu KH. Self-training in significance space of support vectors for imbalanced biomedical event data. BMC Bioinform. 2015;16(S7):1–8. doi:10.1186/1471-2105-16-S7-S6.
5. Vapnik VN. An overview of statistical learning theory. IEEE Trans Neural Netw. 1999;10(5):988–99. doi:10.1109/72.788640.
6. Yadav S, Bhole GP. Handling imbalanced dataset classification in machine learning. In: 2020 IEEE Pune Section International Conference; 2020 Dec 16–18; Pune, India. p. 38–43.
7. Dietterich TG. Ensemble methods in machine learning. Mult Classif Syst. 2000;1857:1–15. doi:10.1007/3-540-45014-9.
8. Nath JS, Bhattacharyya C. Maximum margin classifiers with specified false positive and false negative error rates. In: Proceedings of the 2007 SIAM International Conference on Data Mining; 2007 Apr 26–28; Minneapolis, MN, USA. 2007. p. 35–46.
9. Kim KH, Sohn SY. Hybrid neural network with cost-sensitive support vector machine for class-imbalanced multimodal data. Neural Netw. 2020;130(6):176–84. doi:10.1016/j.neunet.2020.06.026.
10. Wei J, Huang H, Yao L, Hu Y, Fan Q, Huang D. New imbalanced bearing fault diagnosis method based on Sample-characteristic Oversampling Technique (SCOTE) and multi-class LS-SVM. Appl Soft Comput. 2021;101(9):107043. doi:10.1016/j.asoc.2020.107043.
11. Fu S, Yu X, Tian Y. Cost sensitive ν-support vector machine with LINEX loss. Inf Process Manag. 2022;59(2):102809. doi:10.1016/j.ipm.2021.102809.
12. Hasib KM, Showrov MIH, Al Mahmud J, Mithu K. Imbalanced data classification using hybrid under-sampling with cost-sensitive learning method. In: Patgiri R, Bandyopadhyay S, Borah MD, Balas VE, editors. Edge analytics. Singapore: Springer; 2022. p. 423–35.
13. Shajalal M, Hajek P, Abedin MZ. Product backorder prediction using deep neural network on imbalanced data. Int J Prod Res. 2023;61(1):302–19. doi:10.1080/00207543.2021.1901153.
14. Fofanah AJ, Chen D, Wen L, Zhang S. Addressing imbalance in graph datasets: introducing gate-GNN with graph ensemble weight attention and transfer learning for enhanced node classification. Expert Syst Appl. 2024;255(5):124602. doi:10.1016/j.eswa.2024.124602.

15. Tanveer M, Mishra R, Richhariya B. Projection based fuzzy least squares twin support vector machine for class imbalance problems. arXiv:2309.15886. 2023.
16. Rezvani S, Pourpanah F, Lim CP. Methods for class-imbalanced learning with support vector machines: a review and an empirical evaluation. Soft Comput. 2024;28(4):11873–94. doi:10.1007/s00500-024-09931-5.
17. Jian C, Gao J, Ao Y. A new sampling method for classifying imbalanced data based on support vector machine ensemble. Neurocomputing. 2016;193:115–22. doi:10.1016/j.neucom.2016.02.006.
18. Fernández A, Garcia S, Herrera F, Chawla NV. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res. 2018;61:863–905. doi:10.1613/jair.1.11192.
19. Han H, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Int Conf Intell Comput. 2005;3644:878–87. doi:10.1007/11538059.
20. He H, Bai Y, Garcia EA, Li S. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE International Joint Conference on Neural Networks; Hong Kong SAR, China. p. 1322–8.
21. Tomek I. Two modifications of CNN. IEEE Trans Syst Man Cybern. 1976;SMC-6(11):769–72. doi:10.1109/TSMC.1976.4309452.
22. Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Trans Syst Man Cybern. 1972;SMC-2(3):408–21. doi:10.1109/TSMC.1972.4309137.
23. Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor Newsl. 2004;6(1):20–9. doi:10.1145/1007730.1007735.
24. Batista GE, Bazzan AL, Monard MC. Balancing training data for automated annotation of keywords: a case study. Wob. 2003;3:10–8.
25. Breiman L. Bagging predictors. Mach Learn. 1996;24(2):123–40. doi:10.1007/BF00058655.
26. Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. In: IEEE Conference on Computer Vision and Pattern Recognition; 2017 Jul 21–26; Honolulu, HI, USA. p. 2980–8.
27. Cui Y, Jia M, Lin TY, Song Y, Belongie S. Class-balanced loss based on effective number of samples. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019 Jun 15–20; Long Beach, CA, USA. p. 9268–77.
28. Mohammed A, Kora R. A comprehensive review on ensemble deep learning: opportunities and challenges. J King Saud Univ Comput Inf Sci. 2023;35(2):757–74. doi:10.1016/j.jksuci.2023.01.014.
29. Liu CL, Chang YH. Learning from imbalanced data with deep density hybrid sampling. IEEE Trans Syst Man Cybern Syst. 2022;52(11):7065–77. doi:10.1109/TSMC.2022.3151394.
30. Asuncion A, Newman D. UCI machine learning repository. Irvine, CA, USA: University of California; 2007.
31. Vujovic Z. Classification model evaluation metrics. Int J Adv Comput Sci Appl. 2021;12(6):599–606. doi:10.14569/issn.2156-5570.
32. Braytee A, Liu W, Kennedy P. A cost-sensitive learning strategy for feature extraction from imbalanced data. In: Neural Information Processing: 23rd International Conference; 2016 Oct 16–21; Kyoto, Japan. p. 78–86.
33. Kubat M, Matwin S. Addressing the curse of imbalanced training sets: one-sided selection. Proc ICML. 1997;97(1):179.
34. Fawcett T. An introduction to ROC analysis. Pattern Recognit Lett. 2006;27(8):861–74. doi:10.1016/j.patrec.2005.10.010.
35. Lemaître G, Nogueira F, Aridas CK. Imbalanced-learn: a Python toolbox to tackle the curse of imbalanced datasets in machine learning. J Mach Learn Res. 2017;18(17):1–5.