



Research article

CCKEL: Compensation-based correlated k -labelsets for classifying imbalanced multi-label data

Qianpeng Xiao¹, Changbin Shao^{1,2}, Sen Xu³, Xibei Yang¹ and Hualong Yu^{1,*}

¹ School of Computer, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu, China

² Jiangsu Key Laboratory of Media Design and Software Technology, Jiangnan University, Wuxi, Jiangsu, China

³ School of Information Engineering, Yancheng Institute of Technology, Yancheng, Jiangsu, China

* **Correspondence:** Email: yuhualong@just.edu.cn; Tel: +86-15952894360.

Abstract: Imbalanced data distribution and label correlation are two intrinsic characteristics of multi-label data. This occurs because in this type of data, instances associated with certain labels may be sparse, and some labels may be associated with others, posing a challenge for traditional machine learning techniques. To simultaneously adapt imbalanced data distribution and label correlation, this study proposed a novel algorithm called compensation-based correlated k -labelsets (CCKEL). First, for each label, the CCKEL selects the $k-1$ strongest correlated labels in the label space to constitute multiple correlated k -labelsets; this improves its efficiency in comparison with the random k -labelsets (RAkEL) algorithm. Then, the CCKEL transforms each k -labelset into a multiclass issue. Finally, it uses a fast decision output compensation strategy to address class imbalance in the decoded multi-label decision space. We compared the performance of the proposed CCKEL algorithm with that of multiple popular multi-label imbalance learning algorithms on 10 benchmark multi-label datasets, and the results show its effectiveness and superiority.

Keywords: multi-label learning; class imbalance; random k -labelsets; label correlation; decision output compensation

1. Introduction

Multi-label learning, specifically used to classify instances that simultaneously relate to multiple

class labels, is a popular machine learning paradigm. In real life, an object with multiple labels is common. For instance, a photograph may simultaneously picture mountains, snow, clouds, and the sky, a news article may involve several different topics, and a piece of music may express many emotions. In recent years, multi-label learning was turned into a hotspot in the field of machine learning [1]; meanwhile, this technique has also been widely used to retrieve images [2,3], classify text [4,5], predict drug-induced pathology in multi-organ systems [6], study proteins subcellular localization [7,8], recognize protein functions [9], and construct recommendation systems [10].

Many algorithms have been proposed to classify multi-label data; these could be roughly divided into two different groups: problem transformation and algorithm adaption. The former transforms the multi-label learning task into one or more single-label learning tasks [11] or label-ranking (LR) tasks [12]. Algorithm adaption aims to adapt traditional machine learning algorithms to directly handle multi-label data [13,14].

Alternatively, researchers may consider the order of correlations among labels to categorize algorithms, dividing all multi-label learning algorithms into three groups: first-order approaches, which treat each label independently [15], second-order methods, which take advantage of pairwise relations between labels to address multi-label learning issues [12], and high-order algorithms, which utilize high-order relations among labels to classify multi-label data [16,17]. In comparison to first- and second-order approaches, high-order methods can generally yield a better performance, as most real-world applications have high-order relations among their labels [1]. The classifier chains (CC) [16] and random k -labelsets (RA k EL) [17] algorithms are two of the most popular high-order multi-label learning algorithms. The former gradually adds labels into a feature space to train the binary classifier into predicting the next label; the RA k EL randomly divides the label space into multiple k -label subspaces, transforming each to be a multi-class learning task based on the combination of these k -labels. Then, it trains a multi-class classifier on each multi-class task. Specifically, the RA k EL effectively addresses two problems simultaneously: the excessive number of classes being transformed by the label powerset (LP) method, and sparse instances for each class. However, due to the RA k EL always randomly selecting k -label sets, it cannot be guaranteed to focus on the boundaries between labels.

In addition to label correlation, imbalanced data distribution is also an intrinsic characteristic of multi-label data [18]. In general, for any one specific label, there are more instances correlated with it than those that are not. For example, in the Google image repository, the number of images containing the *mountains* label is certainly much lower than the number of images excluding that label. Such an imbalanced class distribution tends to make those traditional multi-label learning models ineffective in recognizing real active labels in a test instance. Therefore, it is necessary to focus on this issue when a multi-label learning algorithm is designed. In single-label learning, there are several techniques dealing with this class imbalance problem, including resampling [19–22], cost-sensitive learning [23–25], decision output compensation [26,27], and ensemble learning [28–33]. However, these single-label class imbalance learning algorithms cannot be directly used on multi-label data due to the existence of complex label correlations. Some recent studies focused on this problem and provided several solutions by borrowing the idea of single-label class imbalance learning techniques, including ML-RUS [34], ML-ROS [34], ML-SMOTE [35], COCOA [36], ECCRU [37], and LDAML-IMB [38]. We note that these emerging multi-label imbalance learning algorithms either fail to properly take advantage of label correlations or require very-high time complexity.

In this study, we propose an improved RA k EL algorithm, called compensation-based correlated k -labelsets (CC k EL), which simultaneously amends the flaw of RA k EL and considers how to address

the class imbalance problem. Specifically, the $CCkEL$ first adopts the Jaccard coefficient [39] as a similarity metric to find the $k-1$ strongest correlated labels with each label existing in a label space to create the corresponding k labelset. This replaces the random strategy used by $RAkEL$; by selecting correlated labels, it forces the learning model to pay more attention to the boundary details among k strongly correlated labels. Then, the $CCkEL$ transforms each k -label learning task into a multi-class learning task, in which each class represents a unique label combination. Next, considering that the transformed multi-class datasets might be very skewed, a rapid decision output compensation strategy is designed to enhance the predicted accuracy of active labels during the final decision. The performance of the proposed $CCkEL$ is compared here with several popular multi-label imbalance learning algorithms on 10 benchmark multi-label datasets, and the results show that the proposed $CCkEL$ has excellent classification performance and low time complexity, indicating its effectiveness and superiority.

The key contributions of this study are as follows:

- 1) A correlated label-selection strategy relying on the Jaccard coefficient is proposed to replace the random label-selection strategy used by the $RAkEL$ algorithm, which makes the learning model focus more on the boundary details among correlated labels;
- 2) To address the class imbalance issue in the decoded multi-label decision space, a fast decision output compensation strategy relying on the imbalance ratio of labels is designed. It can simultaneously improve the adaption of the learning model to imbalanced data distribution and save running time.
- 3) A novel multi-label imbalance learning algorithm named $CCkEL$ is presented. Experimental results show that the $CCkEL$ algorithm acquires the lowest average rank in terms of F-micro, HammingLoss, and SubsetAccuracy metrics among all compared algorithms. Meanwhile, it has a moderate running time consumption, which is a little higher than simple baseline algorithms but clearly lower than several other state-of-the-art imbalance multi-label learning algorithms.

The remainder of this paper is organized as follows. Section 2 describes the class imbalance problem in multi-label data, reviews the related work about multi-label imbalance learning algorithms, and illustrates the limitations of the $RAkEL$ algorithm. Section 3 presents the details of the proposed $CCkEL$ algorithm. In Section 4, the experimental results and analysis are presented in detail. Finally, Section 5 concludes the findings of this study.

2. Problem description and related work

2.1. Problem description

A multi-label dataset can be represented as $D = \{(x_i, y_i) | 1 \leq i \leq N\}$, where N denotes the number of instances, $x_i \in X$ indicates the feature vector of the i th instance, and $y_i \subseteq Y = \{0, 1\}^q$ represents the q -dimensional label vector corresponding to the i th instance. Derived by the multi-label dataset D , a multi-label learning model can be induced aiming to $f: X \mapsto Y$, i.e., constructing a mapping between the feature space X and the label space Y .

As indicated in [18], for any one label l_i , the number of instances correlated (active) with that label always tends to be lower than the number of instances that are not correlated (inactive) with it. Therefore, each binary classification task corresponding to an independent label existing in the label space is always imbalanced. Here, we use $|D_i^+|$ and $|D_i^-|$ to denote the number of active and inactive

instances corresponding to the label l_i , respectively. Obviously, we have $|D_i^+| + |D_i^-| = N$. Then, several quantized metrics [34] to describe the class imbalance level of a multi-label dataset are drawn as follows:

$$\text{Card}(D) = \frac{\sum_{i=1}^q |D_i^+|}{N} \quad (1)$$

$$\text{Dens}(D) = \frac{\text{Card}(D)}{q} \quad (2)$$

$$\text{IR}_i = \frac{|D_i^-|}{|D_i^+|} \quad (3)$$

$$\text{MeanIR} = \frac{\sum_{i=1}^q \text{IR}_i}{q} \quad (4)$$

It is clear that the metrics Card, Dens, and MeanIR reflect the imbalance level of the whole multi-label dataset, while IR_i only denotes the imbalance level of the i th label. The smaller Card and Den and the larger MeanIR are, the higher is the class imbalance level.

Taking the Yeast dataset as an example, we counted the active instances on each label and further calculated the corresponding IR_i and MeanIR (Figure 1). It is noteworthy that, to some extent, there is a class imbalance problem on each class label. Specifically, Label 2 has the lowest IR (1.33), while Label 14 presents the highest IR (70.09); the MeanIR reaches 8.95, indicating that the class imbalance is a non-negligible problem when constructing multi-label learning models.

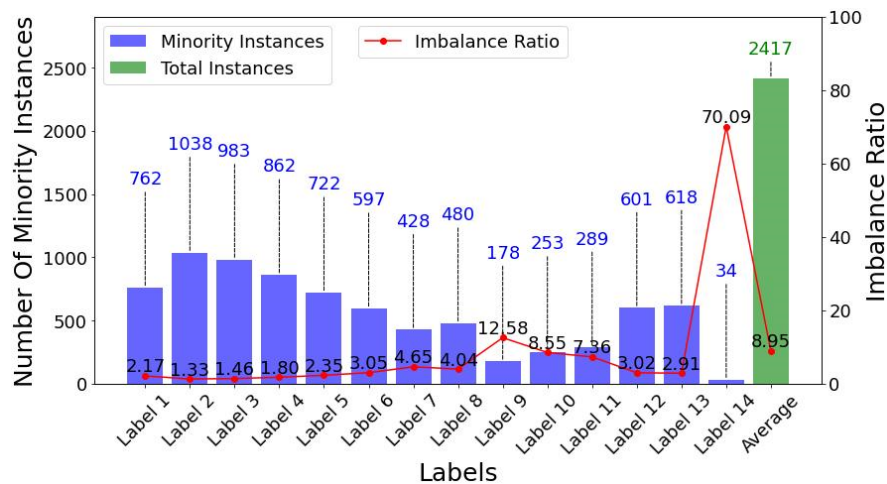


Figure 1. Distribution of minority samples and imbalance ratios in the Yeast dataset.

2.2. A brief review of related work on multi-label imbalance learning algorithms

In recent years, researchers have focused on the class imbalance problem of multi-label data and tried to propose several solutions considering their characteristics. As indicated in the previous section, there are two differences between single- and multi-label imbalance learning tasks. First, in multi-label data, it is difficult to precisely distinguish majority and minority instances, since each instance simultaneously correlates with multiple class labels. Additionally, there are correlations among labels that cannot be easily isolated.

In the context of sampling, Charte et al. [34] proposed the ML-RUS (multi-label random

undersampling) and ML-ROS (multi-label random oversampling) algorithms. In their design, only the class label with a lower IR than the MeanIR is considered to be handled. Based on this principle, we either randomly remove the majority instances or copy the minority instances. To promote the robustness of sampling, Charte et al. [35] further proposed the ML-SMOTE (multi-label SMOTE) algorithm, which generates synthetic minority instances between two adjacent minority instances. These algorithms alleviate the class imbalance problem to some extent but fail to focus on label correlations. In other words, it is difficult to distinguish minority and majority instances, since a so-called majority instance on one label may be the minority instance on another label. When the instance is removed, it would be beneficial for the first label but harmful for the second label. To address this problem, Liu and Tsoumakas [37] integrated the sampling techniques into the framework of the classifier chain [16] and proposed the ECC-RU. This algorithm executes random undersampling on each label independently to deal with the class imbalance problem and adopts a classifier chain to address label correlation problems. In comparison with several other multi-label sampling algorithms, the ECC-RU shows significant superiority [37].

Besides sampling, the multi-label class imbalance problem can also be addressed in other terms. Zhang et al. [36] proposed an ensemble solution named cross-coupling aggregation (COCOA), which uses a pairwise strategy to exploit label correlations and a threshold strategy to deal with class imbalance. Specifically, for each label pair $\langle l_i, l_j \rangle$, it first transforms all instances into one of three classes according to their label values on the label pair: the first class contains all instances belonging to l_i , the second class covers all instances neither belonging to l_i nor l_j , and the third class includes the remainder instances. Then, a calibrating multi-class learning model is trained on the transformed task, and the corresponding threshold needs to be determined by an optimization procedure. Finally, the predicted label set for a test example is obtained by querying the predictive models of all label pairs. The COCOA algorithm is robust but is also extremely time consuming, as it requires training $K \times q$ triple-class models in total, where K denotes a preset number of correlated labels for each label. In [38], Peng et al. proposed an algorithm called LDAML-IMB that adopts the latent Dirichlet allocation (LDA) model to explore label correlations. Specifically, the LDAML-IMB treats each instance as a document and each label as a word in the document. It takes advantage of the LDA to determine the distribution of topics, which reflects the label correlations; then, it integrates the topic information to extend the original feature space and finally decides the predicted set of labels for each instance according to the corresponding topic probabilities. The LDAML-IMB adopts a similar method than COCOA to tackle the class imbalance problem; thus, it is noteworthy that the LDAML-IMB algorithm is relatively time consuming.

2.3. Random k -labelsets and its limitations

The random k -labelsets (RA k EL) algorithm was initially proposed to overcome the problem of generating an excessive number of classes in the label powerset (LP) algorithm, further improving the robustness of the learning model [17]. It firstly randomly extracts multiple k -label subsets from the original label space and then transforms each of them to be a 2^k -class problem according to different combinations of label values. Next, for each transformed multi-class learning task D_i , the RA k EL trains a corresponding learning model h_i . Finally, it adopts the following rule to make a decision for each label l_i of a test instance x .

$$V(x, l_i) = \begin{cases} 1, & \text{if } \varphi(x, l_i) \geq t \\ 0, & \text{if } \varphi(x, l_i) < t \end{cases} \quad (5)$$

where $t \in [0,1]$ denotes the decision threshold, which is, in general, set to be 0.5, and

$$\varphi(x, l_i) = \frac{\sum_{j=1}^S h_j(x, l_i)}{S} \quad (6)$$

where S denotes the number of k -label subsets that contain the label l_i , i.e., $l_i \in D_j$. Figure 2 presents an example to describe the flow path of the RAkEL algorithm.

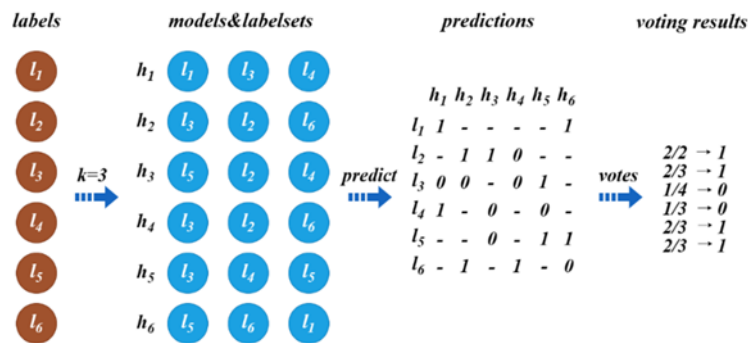


Figure 2. An example to describe the flow path of the RAkEL algorithm, where $t = 0.5$.

The RAkEL can be seen as an improved version of the LP algorithm. However, we note that there are still two limitations in the RAkEL algorithm. Although the RAkEL considers the high-order label correlations, the label combinations are random, which may cause the model to only explore a rough boundary while neglecting some details about the classification boundary. Also, the RAkEL fails to solve the class imbalance problem. In this work, we will try to address both problems.

3. Methods

3.1. Exploring label correlations

As stated in Section 2.3, the RAkEL algorithm randomly explores label correlations. This tends to generate a rough classification boundary based on the discrepancies among k irrelevant labels, instead of an accurate boundary determined by the differences among k closely correlated labels. In this study, we modified the random fashion adopted by RAkEL and proposed to constitute labelsets among k closely correlated labels. Specifically, considering that a multi-label classification task generally involves a binary label space where each element is labeled either 0 or 1, we decided to use the Jaccard coefficient [39] to detect label correlations.

Suppose D^i and D^j store the training instances that are active on label l_i and l_j , respectively; then, the Jaccard coefficient between these two labels can be calculated as follows,

$$J(l_i, l_j) = \frac{|D^i \cap D^j|}{|D^i \cup D^j|} \quad (7)$$

It is obvious that a larger Jaccard coefficient means a closer correlation between two corresponding labels. For example, in an image retrieval task, the images containing *sandbeach* always include *sea*

but are in general irrelevant to *waterfall*. When training a learning model, it is more difficult but also useful to explore the difference between the two labels *sandbeach* and *sea* than distinguish between *sandbeach* and *waterfall*.

For each label l_i in the label space Y , the $k-1$ strongest correlated labels with that label can be found by sorting $q-1$ calculated Jaccard coefficients in descending order. Therefore, it only requires generating q k -labelsets in total.

3.2. An empirical decision output compensation strategy

In this study, we consider adopting a decision output compensation strategy, i.e., a threshold strategy, to address the imbalanced data classification problem. On the transformed multi-class datasets, the instances belonging to some classes might be extremely sparse, causing their distributions to not being precisely described, which undermines some traditional sampling or cost-sensitive learning methods [20].

Although there are several mature decision output compensation algorithms, e.g., SVM-OTHR [26] and ODOC-ELM [27], they are too complex and time consuming to be used in this scenario. In addition, these methods tend to overfit the learning model, as the optimum solution is always sought out based on the training data.

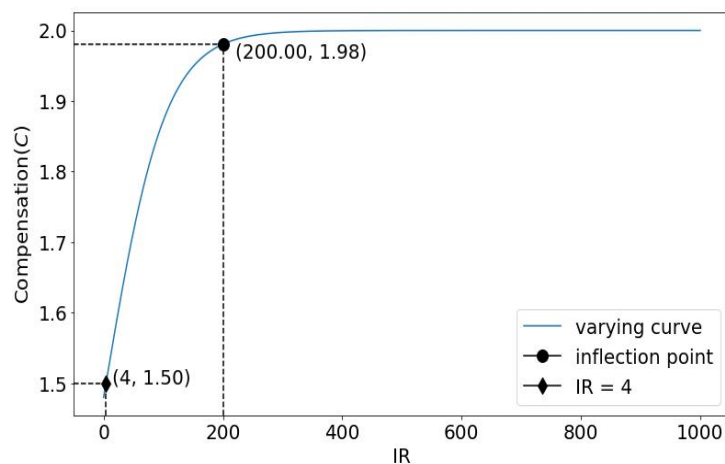


Figure 3. The varying curve of the modified function f .

Based on the considerations above, in this study we developed a fast decision output compensation strategy. Different from the addition compensation strategy adopted by SVM-OTHR and ODOC-ELM, our proposed strategy adopts the multiplication compensation. Specifically, it rewrites the decision output as

$$\varphi(x, l_i) = C \times \varphi(x, l_i) \quad (8)$$

where the modified factor C can be calculated by,

$$C = f(\text{IR}_i) = 1 + \frac{1}{1 + e^{-\lambda_1(\text{IR}_i - \lambda_2)}} \quad (9)$$

where f denotes the modified function, IR_i indicates the class imbalance ratio of label l_i , and λ_1 and λ_2

are two constants; the former controls the varying gradient of the modified function and the latter indicates the inflection position of the varying curve of the modified function. In this study, we empirically designate λ_1 as 0.02 and λ_2 as 4. The setting of λ_2 satisfies the assumption that the performance of a dataset is impacted by the imbalanced data distribution only when the class imbalance ratio is larger than 4 [20]. Similarly, in this study, when and only when the class imbalance ratio of a label is equal to or larger than 4, its decision output is compensated. Specifically, the modified factor C lies in the range (1.5, 2), and its varying curve is drawn in Figure 3.

3.3. Description of the CkEL algorithm

Based on the proposed label correlation exploration strategy and the empirical decision output compensation strategy, the flow path of the CkEL algorithm can be described in detail as follows.

Input:

Training set: D
 Size of labelset: k
 Learning model: h
 Test instance: x

Output:

Predicted label set for x : \hat{y}

Training procedure:

1. $\mathbf{W} = \emptyset$;
2. $\mathbf{C} = \emptyset$;
3. **For** $i=1:q$
4. Calculate IR_i of the label l_i by Eq (3);
5. $\mathbf{W} = \mathbf{W} \cup IR_i$;
6. **If** $IR_i \geq 4$
7. Calculate C_i by Eq (9);
8. $\mathbf{C} = \mathbf{C} \cup C_i$;
9. **End if**
10. **End for**
11. $\mathbf{Q} = [0]_{q \times q}$;
12. **For** $i=1:(q-1)$
13. **For** $j=(i+1):q$
14. Calculate $J(l_i, l_j)$ by Eq (7) and use it to replace the values of the i th line and j th column, and the j th line and i th column in \mathbf{Q} ;
15. **End for**
16. **End for**
17. $\mathbf{P} = [0]_{q \times k}$;
18. **For** $i=1:q$
19. Sort all values in the i th line of \mathbf{Q} in descending order and then find the top $(k-1)$ labels to combine l_i to constitute its k -labelset, then use their mark numbers to replace the i th line in \mathbf{P} ;
20. Transform the k -labelset to be the corresponding multi-class task, and train the learning model h_i ;
21. **End for**

Testing procedure:

22. **For** $i=1:q$
23. Decode $h_i(x)$ by visiting the i th line in \mathbf{P} ;
24. **End for**
25. **For** $i=1:q$
26. Calculate $\varphi(x, l_i)$ by Eq (6);
27. **If** $(\mathbf{W}(i) \geq 4)$
28. Update $\varphi(x, l_i)$ by visiting \mathbf{C} using Eq (8);
29. **End if**
30. Calculate $V(x, l_i)$ by using Eq (5);
31. **End for**

Output \hat{y} , which is the predicted label set for the test instance x .

Specifically, \mathbf{W} and \mathbf{C} are two vectors that store the class imbalance ratio and the modified factor of each label, respectively. \mathbf{Q} is a matrix to store the Jaccard coefficient between any two labels, while \mathbf{P} is the matrix that stores the information about all k -labelsets.

It is noteworthy that several state-of-the-art algorithms, including RAKEL [17], COCOA [36], ECCRU [37], LDAML-IMB [38], and our proposed CCKEL, all focus on high-order label correlations. Here, we wish to analyze their differences, further highlighting the rationality of our proposed algorithm. First, both RAKEL and the proposed CCKEL generate q k -labelsets and deal with them by label powerset strategy. However, each k -labelset in the RAKEL is generated randomly, making it possible to involve some irrelevant labels, further lowering the efficiency of each sub-learning model. In addition, the RAKEL fails to consider the potential class imbalance problems in multi-label data. In contrast, the CCKEL improves the efficiency of each sub-learning model, as it constructs each k -labelset among closely correlated labels, and solves the class imbalance issue through a fast threshold strategy. As for COCOA, it only considers pairwise label relations; when there are multiple label correlations, it cannot capture these relations. Meanwhile, it conducts an optimization procedure to seek the optimal threshold, thus being extremely time consuming. In contrast, the CCKEL can rapidly capture high-order label relations and calculate an appropriate threshold. ECCRU explores high-order label correlations by introducing old labels into the feature space and solves the class imbalance problem by adopting random undersampling (RUS) strategy. In contrast to our proposed CCKEL algorithm, the ECCRU tends to introduce more noisy label correlations, which could destroy feature-label mappings, while losing classification information and decreasing the quality of each sub-learning model, as with the adoption of the RUS technique. In comparison to several other algorithms, the LDAML-IMB seems to be more robust, as it can not only capture accurate high-order label correlations but also solve the class imbalance problem in a rational fashion. However, it is extremely time consuming, as both the LDA approach and optimized threshold strategy require high time complexities. This further explains why our proposed CCKEL algorithm may be more effective and efficient than these state-of-the-art algorithms.

Regarding time complexity, the RAKEL algorithm has time complexity $O(q \cdot F_M(N, d, 2^k))$ for training and $O(q \cdot F'_M(d, 2^k))$ for testing, where N denotes the number of training instances, d represents the dimension of feature space X , $F_M(N, d, 2^k)$ denotes the time complexity of transforming a k -labelset into a multi-class problem and training the corresponding classification model, and $F'_M(d, 2^k)$ denotes the time complexity for predicting an instance by a trained multi-class learning model and then denoting it into the original k -label space [1]. Then, the CCKEL has two more steps than the RAKEL during training. One is calculating the Jaccard coefficients and sorting them with the time complexity $O(q^2N)$. The other one is calculating the modified factor C , which has a time complexity $O(1)$. Therefore, the training time complexity of the CCKEL is $O(q \cdot (qN + F_M(N, d, 2^k)))$, and the testing time complexity of the CCKEL is $O(q \cdot F'_M(d, 2^k))$.

4. Experiments and discussion

4.1. Description of the used datasets

In experiments, we used 10 multi-label datasets collected from two open-source data repositories:

OpenML¹ and Mulan². The details about these datasets are presented in Table 1, where *#Instances*, *#Features*, and *#Labels* represent the number of instances, the number of features, and the number of labels contained in the corresponding dataset, respectively. Card, Dens, and MeanIR are three popular multi-label imbalance level metrics, which have been described in Eqs (1), (2), and (4), while IR (min) and IR (max) denote the minimum and maximum imbalance ratios among all labels within the corresponding dataset.

Table 1. Details about the used datasets.

Dataset	#Instances	#Features	#Labels	Card	Dens	IR (min)	IR (max)	MeanIR
Bibtex	7395	183	159	2.402	0.015	6.097	144.000	87.699
Birds	645	260	19	1.014	0.053	5.262	106.500	32.859
Emotions	593	72	6	1.869	0.074	1.246	3.007	2.320
Enron	1702	1001	53	3.378	0.064	1.009	1701.000	136.867
Flags	194	19	9	3.392	0.485	1.042	11.125	3.819
Image	2000	135	5	1.236	0.247	2.448	3.890	3.116
Medical	978	1449	45	1.245	0.028	2.677	977	328.069
Scene	2407	294	6	1.074	0.179	3.516	5.613	4.662
SlashDot	3782	1079	22	1.134	0.081	5.476	1259.667	125.039
Yeast	2417	103	14	4.233	0.325	1.329	70.088	8.954

4.2. Experimental settings

We compared the proposed CCKEL with the baseline algorithm RAKEL [17] and several state-of-the-art imbalanced multi-label learning algorithms, namely ML-RUS [34], ML-ROS [34], ML-SMOTE [35], COCOA [36], ECCRU [37], and LDAML-IMB [38]. All these algorithms were implemented by Python, and they all ran on a hardware environment of Intel(R) Core(TM) i5-11400H @ 2.70GHz.

Specifically, we used five popular performance metrics to evaluate various algorithms, including Micro F-measure (F-micro), Macro F-measure (F-macro), HammingLoss, SubsetAccuracy, and running time. These metrics can be calculated as follows:

$$F - \text{micro} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

$$F - \text{macro} = \frac{1}{q} \sum_{i=1}^q \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \quad (11)$$

$$\text{HammingLoss} = \frac{1}{N} \sum_{i=1}^N \frac{\sum_{j=1}^q \mathbf{I}(y_{i,j} \neq \hat{y}_{i,j})}{q} \quad (12)$$

$$\text{SubsetAccuracy} = \frac{1}{N} \sum_{i=1}^N \mathbf{I}(y_i = \hat{y}_i) \quad (13)$$

where \mathbf{I} is an indicator function equal to 1 when the condition is satisfied; otherwise, it is equal to 0. As for several other variables, they can be calculated as follows:

¹ <https://www.openml.org>

² <https://mulan.sourceforge.net/>

$$\text{Precision} = \frac{\frac{1}{q} \sum_{i=1}^q \text{TP}_i}{\frac{1}{q} \sum_{i=1}^q \text{TP}_i + \frac{1}{q} \sum_{i=1}^q \text{FP}_i} \quad (14)$$

$$\text{Recall} = \frac{\frac{1}{q} \sum_{i=1}^q \text{TP}_i}{\frac{1}{q} \sum_{i=1}^q \text{TP}_i + \frac{1}{q} \sum_{i=1}^q \text{FN}_i} \quad (15)$$

$$\text{Precision}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FP}_i} \quad (16)$$

$$\text{Recall}_i = \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i} \quad (17)$$

where TP_i , FP_i , and FN_i represent the true positive rate, false positive rate, and false negative rate on the label l_i , respectively.

To guarantee the impartiality of compared experiments, all algorithms adopted C4.5 decision tree [40] as the base level single-label learning model. We selected C4.5 decision tree as it has been widely adopted in experiments of several relevant algorithms [17,40]. In practical applications, it could be replaced by any other classifier according to the users' preferences. Except for the CCKEL, all other algorithms used the default settings, which have been suggested in the corresponding references. The CCKEL empirically set $k = \max(q/3, 3)$. In addition, to truly reflect the performance of each algorithm, we conducted 10 times five-fold cross-validations and then output the average results.

4.3. Results and discussion

The experimental results of the five performance metrics are presented in Tables 2–6. Additionally, to more intuitively compare the quality of various algorithms, a bar graph depicting the average rankings of each algorithm on five different metrics is also provided in Figure 4.

Table 2. F-micro performance of various algorithms, where \square and \bullet indicate the best and the second-best results on each dataset, respectively.

Dataset	Algorithm							
	CCKEL	RAkEL	COCOA	ML-SMOTE	ML-ROS	ML-RUS	ECCRU	LDAML-IMB
Bibtex	0.4101 ± 0.0007 \bullet	0.4073 ± 0.0009	0.3523 ± 0.0023	0.3601 ± 0.0022	0.3624 ± 0.0015	0.3513 ± 0.0005	0.4359 ± 0.0022 \square	0.3645 ± 0.0014
Birds	0.4507 ± 0.0186 \bullet	0.4353 ± 0.0214	0.4142 ± 0.0181	0.3967 ± 0.0208	0.3793 ± 0.0259	0.3587 ± 0.0172	0.4708 ± 0.0122 \square	0.3962 ± 0.0164
Emotions	0.5767 ± 0.0163	0.5633 ± 0.0148	0.6119 ± 0.0181 \bullet	0.5630 ± 0.0163	0.5764 ± 0.0205	0.5716 ± 0.0095	0.6312 ± 0.0098 \square	0.5941 ± 0.0137
Enron	0.5525 ± 0.0044 \square	0.5505 ± 0.0050 \bullet	0.4829 ± 0.0129	0.4749 ± 0.0057	0.4732 ± 0.0088	0.4567 ± 0.0086	0.5123 ± 0.0028	0.4749 ± 0.0071
Flags	0.6806 ± 0.0198 \bullet	0.6526 ± 0.0164	0.6915 ± 0.0129 \square	0.6405 ± 0.0215	0.6558 ± 0.0149	0.6637 ± 0.0219	0.6514 ± 0.0059	0.6596 ± 0.0212
Image	0.4790 ± 0.0157	0.4622 ± 0.0109	0.5120 ± 0.1140 \bullet	0.4500 ± 0.0204	0.4718 ± 0.0135	0.4632 ± 0.0045	0.5421 ± 0.0130 \square	0.4776 ± 0.0067
Medical	0.8187 ± 0.0055 \square	0.8128 ± 0.0031 \bullet	0.7970 ± 0.0065	0.7905 ± 0.0069	0.7795 ± 0.0066	0.7684 ± 0.0149	0.8044 ± 0.0070	0.7952 ± 0.0069
Scene	0.6376 ± 0.0144 \bullet	0.6130 ± 0.0147	0.6092 ± 0.0097	0.5959 ± 0.0145	0.5899 ± 0.0064	0.5673 ± 0.0084	0.6825 ± 0.0063 \square	0.5955 ± 0.0041
SlashDot	0.5467 ± 0.0078 \square	0.5364 ± 0.0011	0.5273 ± 0.0059	0.5207 ± 0.0082	0.5204 ± 0.0047	0.4906 ± 0.0073	0.5424 ± 0.0041 \bullet	0.5258 ± 0.0032
Yeast	0.5717 ± 0.0129	0.5629 ± 0.0047	0.5854 ± 0.0052 \square	0.5333 ± 0.0044	0.5375 ± 0.0047	0.5584 ± 0.0098	0.5844 ± 0.0119 \bullet	0.5453 ± 0.0039
Average ranking	2.1	4.0	3.3	6.4	6.1	6.7	2.2	4.7

Table 3. F-macro performance of various algorithms, where \square and \bullet indicate the best and the second-best results on each dataset, respectively.

Dataset	Algorithm							
	CCkEL	RAkEL	COCOA	ML-SMOTE	ML-ROS	ML-RUS	ECCRU	LDAML-IMB
Bibtex	0.2281 ± 0.0011	0.2246 ± 0.0007	0.2922 ± 0.0011 \bullet	0.2808 ± 0.0033	0.2843 ± 0.0023	0.2667 ± 0.0008	0.3479 ± 0.0029 \square	0.2854 ± 0.0017
Birds	0.2827 ± 0.0264	0.2651 ± 0.0183	0.3110 ± 0.0175 \bullet	0.2782 ± 0.0254	0.2780 ± 0.0180	0.2504 ± 0.0188	0.3453 ± 0.0137 \square	0.2917 ± 0.0217
Emotions	0.5581 ± 0.0135	0.5478 ± 0.0191	0.6056 ± 0.0178	0.5543 ± 0.0213	0.5693 ± 0.2100	0.5577 ± 0.0129	0.6270 ± 0.0106 \square	0.5881 ± 0.0137 \bullet
Enron	0.1891 ± 0.0063	0.1850 ± 0.0046	0.1962 ± 0.0124	0.1888 ± 0.0088	0.1932 ± 0.0071	0.1718 ± 0.0077	0.2067 ± 0.0081 \square	0.1980 ± 0.0049 \bullet
Flags	0.5989 ± 0.0162 \square	0.5093 ± 0.0141	0.5810 ± 0.0250 \bullet	0.5126 ± 0.0244	0.5318 ± 0.0193	0.5424 ± 0.0266	0.5488 ± 0.0184	0.5484 ± 0.0204
Image	0.4799 ± 0.0175	0.4642 ± 0.0130	0.5139 ± 0.0138 \bullet	0.4526 ± 0.0228	0.4739 ± 0.0137	0.4617 ± 0.0054	0.5478 ± 0.0126 \square	0.4805 ± 0.0079
Medical	0.3767 ± 0.0145	0.3708 ± 0.0110	0.3934 ± 0.0132 \square	0.3867 ± 0.0089	0.3873 ± 0.0102	0.3467 ± 0.0283	0.3865 ± 0.0064	0.3918 ± 0.0091 \bullet
Scene	0.6482 ± 0.0140 \bullet	0.6174 ± 0.0159	0.6248 ± 0.0095	0.6057 ± 0.0147	0.6003 ± 0.0057	0.5789 ± 0.0092	0.6950 ± 0.0071 \square	0.6061 ± 0.0042
SlashDot	0.3332 ± 0.0038	0.3162 ± 0.0069	0.3465 ± 0.0193 \bullet	0.3279 ± 0.0073	0.3311 ± 0.0115	0.3129 ± 0.0056	0.3696 ± 0.0068 \square	0.3326 ± 0.0041
Yeast	0.3625 ± 0.0117	0.3559 ± 0.0064	0.4379 ± 0.0067 \square	0.3733 ± 0.0088	0.3870 ± 0.0101	0.3973 ± 0.0097	0.4111 ± 0.0150 \bullet	0.3963 ± 0.0043
Average ranking	4.4	7.0	2.1	5.8	5.0	6.7	1.7	3.3

Table 4. HammingLoss performance of various algorithms, where \square and \bullet indicate the best and the second-best results on each dataset, respectively.

Dataset	Algorithm							
	CCkEL	RAkEL	COCOA	ML-SMOTE	ML-ROS	ML-RUS	ECCRU	LDAML-IMB
Bibtex	0.0120 ± 0.0003 \square	0.0121 ± 0.0004 \bullet	0.0242 ± 0.0001	0.0199 ± 0.0001	0.0200 ± 0.0002	0.0199 ± 0.0001	0.0174 ± 0.0004	0.0199 ± 0.0001
Birds	0.0427 ± 0.0016 \square	0.0434 ± 0.0013 \bullet	0.0771 ± 0.0046	0.0599 ± 0.0017	0.0683 ± 0.0033	0.0707 ± 0.0017	0.0615 ± 0.0018	0.0647 ± 0.0015
Emotions	0.2347 ± 0.0078 \square	0.2401 ± 0.0088 \bullet	0.2927 ± 0.0153	0.2551 ± 0.0104	0.2649 ± 0.0091	0.2728 ± 0.0117	0.2512 ± 0.0074	0.2545 ± 0.0074
Enron	0.0495 ± 0.0003	0.0489 ± 0.0004 \square	0.0781 ± 0.0017	0.0679 ± 0.0007	0.0689 ± 0.0014	0.0723 ± 0.0013	0.0644 ± 0.0005	0.0694 ± 0.0008
Flags	0.2767 ± 0.0148	0.2625 ± 0.0224 \square	0.2829 ± 0.0108	0.2859 ± 0.0224	0.2796 ± 0.0114	0.2768 ± 0.0185	0.2863 ± 0.0028	0.2792 ± 0.0144
Image	0.2184 ± 0.0067 \square	0.2323 ± 0.0135 \bullet	0.3133 ± 0.0062	0.2516 ± 0.0089	0.2679 ± 0.0075	0.2762 ± 0.0044	0.2560 ± 0.0072	0.2631 ± 0.0039
Medical	0.0098 ± 0.0003 \square	0.0099 ± 0.0004 \bullet	0.0116 ± 0.0003	0.0117 ± 0.0004	0.0123 ± 0.0004	0.0129 ± 0.0008	0.0114 ± 0.0004	0.0115 ± 0.0004
Scene	0.1210 ± 0.0078 \square	0.1219 ± 0.0058	0.1741 ± 0.0050	0.1386 ± 0.0042	0.1501 ± 0.0020	0.1599 ± 0.0064	0.1217 ± 0.0027 \bullet	0.1456 ± 0.0200
SlashDot	0.0401 ± 0.0006 \square	0.0410 ± 0.0003 \bullet	0.0502 ± 0.0006	0.0461 ± 0.0015	0.0467 ± 0.0006	0.0525 ± 0.0008	0.0522 ± 0.0008	0.0463 ± 0.0004
Yeast	0.2281 ± 0.0033 \square	0.2312 ± 0.0024 \bullet	0.3090 ± 0.0041	0.2637 ± 0.0031	0.2813 ± 0.0032	0.2788 ± 0.0049	0.2526 ± 0.0065	0.2750 ± 0.0024
Average ranking	1.2	1.9	7.3	4.3	6.0	6.4	4.0	4.6

Table 5. SubsetAccuracy performance of various algorithms, where \square and \bullet indicate the best and the second-best results on each dataset, respectively.

Dataset	Algorithm							
	CCkEL	RAkEL	COCOA	ML-SMOTE	ML-ROS	ML-RUS	ECCRU	LDAML-IMB
Bibtex	0.1805 ± 0.0021 \square	0.1793 ± 0.0013 \bullet	0.0810 ± 0.0008	0.0998 ± 0.0020	0.0997 ± 0.0011	0.0893 ± 0.0022	0.1367 ± 0.0033	0.1010 ± 0.0016
Birds	0.5054 ± 0.0167 \square	0.5020 ± 0.0096 \bullet	0.3519 ± 0.0248	0.4307 ± 0.0270	0.3833 ± 0.0236	0.3688 ± 0.0332	0.4056 ± 0.0164	0.3890 ± 0.0107
Emotions	0.2167 ± 0.0211 \square	0.1954 ± 0.0202	0.1548 ± 0.0255	0.1651 ± 0.0221	0.1619 ± 0.0270	0.1516 ± 0.0218	0.2020 ± 0.0233 \bullet	0.1826 ± 0.0155
Enron	0.1158 ± 0.0058 \bullet	0.1194 ± 0.0068 \square	0.0837 ± 0.0073	0.0883 ± 0.0083	0.0850 ± 0.0049	0.0741 ± 0.0118	0.1008 ± 0.0056	0.0852 ± 0.0029
Flags	0.1438 ± 0.0204 \square	0.1057 ± 0.0206	0.0835 ± 0.0405	0.0665 ± 0.0303	0.0778 ± 0.0202	0.0912 ± 0.0274	0.1099 ± 0.0036 \bullet	0.0938 ± 0.0196
Image	0.3042 ± 0.0162 \square	0.2841 ± 0.0146 \bullet	0.1817 ± 0.0072	0.2307 ± 0.0128	0.2220 ± 0.0170	0.2110 ± 0.0155	0.2795 ± 0.0210	0.2309 ± 0.0056
Medical	0.6880 ± 0.0152 \square	0.6778 ± 0.0152 \bullet	0.6335 ± 0.0178	0.6263 ± 0.0077	0.6152 ± 0.0105	0.5917 ± 0.0341	0.6488 ± 0.0139	0.6315 ± 0.0117
Scene	0.4844 ± 0.0209 \bullet	0.4703 ± 0.0188	0.3349 ± 0.0150	0.3994 ± 0.0119	0.3766 ± 0.0089	0.3444 ± 0.0245	0.4944 ± 0.0112 \square	0.3880 ± 0.0054
SlashDot	0.3816 ± 0.0058 \square	0.3725 ± 0.0039 \bullet	0.3228 ± 0.0116	0.3304 ± 0.0088	0.3246 ± 0.0142	0.2930 ± 0.0137	0.3503 ± 0.0101	0.3303 ± 0.0051
Yeast	0.0969 ± 0.0117 \bullet	0.0881 ± 0.0058	0.0374 ± 0.0055	0.0509 ± 0.0052	0.0426 ± 0.0070	0.0420 ± 0.0031	0.1214 ± 0.0068 \square	0.0475 ± 0.0029
Average ranking	1.3	2.3	7.1	4.8	6.2	7.2	2.5	4.6

Table 6. Running time (seconds) of various algorithms, where \square and \bullet indicate the best and the second-best results on each dataset, respectively.

Dataset	Algorithm							
	CCKEL	RAkEL	COCOA	ML-SMOTE	ML-ROS	ML-RUS	ECCRU	LDAML-IMB
Bibtex	2919.67	2826.61	1592.60	959.81 \bullet	1653.67	529.67 \square	2026.41	3342.11
Birds	9.35	8.70	13.23	8.55	6.85 \bullet	1.75 \square	11.58	21.78
Emotions	1.23	1.08	1.38	2.56	0.84 \bullet	0.33 \square	2.53	5.16
Enron	49.04	46.73	51.01	41.67	37.14 \bullet	11.85 \square	64.47	290.88
Flags	0.31	0.29	0.34	0.40	0.08 \bullet	0.07 \square	0.47	1.34
Image	6.01	5.21 \bullet	8.82	10.88	7.71	2.89 \square	10.65	50.34
Medical	18.63	17.02	16.14	13.02	10.27 \bullet	4.13 \square	22.38	37.62
Scene	18.27	17.80 \bullet	27.39	26.76	23.20	8.53 \square	30.88	52.67
SlashDot	123.75	118.74	162.40	116.56 \bullet	194.98	55.48 \square	119.89	127.87
Yeast	20.96	19.09 \bullet	33.21	26.51	25.80	8.35 \square	42.98	48.08
Average ranking	4.5	3.4	5.4	4.3	3.4	1.0	6.2	7.8

The results in Tables 2–6 and Figure 4 show that, on the first four performance metrics, the proposed CCKEL is superior to the RAkEL baseline algorithm, indicating that it is necessary to constitute strongly correlated k -labelsets and adopt a compensation strategy to deal with imbalanced data distributions. Clearly, we also note that the CCKEL is more time consuming than the RAkEL as it takes $O(q^2N)$ more time to calculate Jaccard similarities and sort the corresponding k -labelsets.

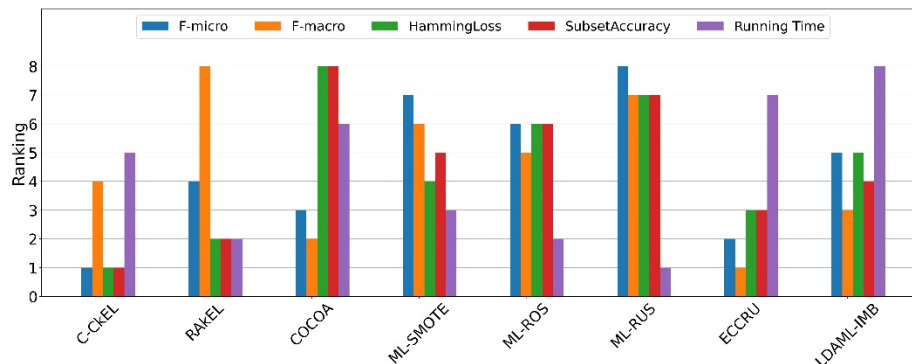


Figure 4. Average rankings of the algorithms compared on five different metrics.

In addition, we observe that several complex algorithms, e.g., COCOA, ECCRU, and LDAML-IMB, clearly perform better than those simple-sampling-based algorithms, especially on two specific imbalance performance metrics: F-micro and F-macro. The reason may be that they all focus on complex label correlations. Of course, we also note that these three algorithms are more time consuming than several others.

Regarding the three simple-sampling-based algorithms, i.e., ML-RUS, ML-ROS, and ML-SMOTE, they performed clearly worse than the others. We believe this to be because in multi-label spaces, it is not easy to distinguish which instances are significant or not. Meaning, adding or removing an instance might alleviate the class imbalance level of a label while increasing that of another label. Therefore, although the corresponding references [34,35] have stated that these sampling-based algorithms are effective, their performance improvement is not significant enough.

It is obvious that the proposed CCKEL algorithm performs best among all comparative algorithms. In particular, the CCKEL has acquired the lowest average ranking in terms of F-micro, HammingLoss,

and SubsetAccuracy metrics, despite being superior in the F-macro metric to the RAKEL baseline algorithm and three sampling-based algorithms. In contrast to three complex ensemble learning algorithms, our proposed CCKEL algorithm is obviously more time saving. The observations above verify the rationality of our algorithm design. We believe that the CCKEL acquires superiority from two aspects: 1) It focuses on how to draw the boundary details by constructing k -labelsets among several closely correlated labels, and 2) it can adaptively amend the negative impact of skewed data distribution for each label by taking advantage of the prior imbalance ratio information of that label.

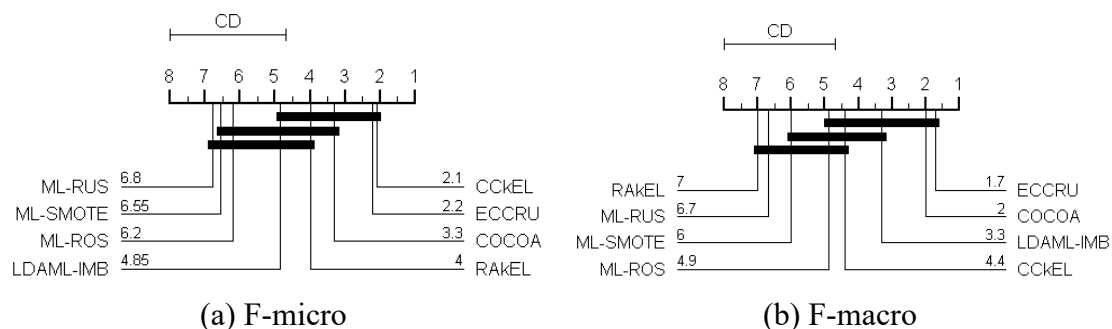
We also note that the proposed CCKEL algorithm presents particular superiority on those datasets with high MeanIR. Specifically, in terms of the F-micro metric, which mostly reflects the quality of a multi-label imbalance learning algorithm, the CCKEL yielded the best results on three highly imbalanced datasets (i.e., Enron, Medical, and SlashDot) and the second-best results on two moderate imbalanced datasets (i.e., Bibtex and Birds). In contrast, on those datasets with a low imbalance level or few labels, the CCKEL cannot show significant superiority to other algorithms. We consider that, on these types of datasets, there is either a lack of label correlations or invalid compensations for decision thresholds.

4.4. Significance analysis

In order to further analyze the relative performance of the algorithms involved in this study, the Nemenyi test was adopted as a post-hoc test for the Friedman test [41,42]. Here, we focus on the relative performance between our proposed CCKEL algorithm and the other algorithms. If the average rank of the CCKEL and that of another algorithm differs by at least one critical difference (CD) unit, then we consider their performance to be significantly different. Specifically, the CD is calculated as follows

$$CD = q_{\alpha} \sqrt{\frac{L(L+1)}{6H}} \quad (18)$$

where q_{α} denotes the significance level, L indicates the number of compared algorithms in the experiments, and H represents the number of datasets used in the experiments. In a CD diagram, if the average ranks of two algorithms are less than a CD unit, then they would be connected by a thick line to indicate that their difference is not significant at a specific significance level. Figure 5 presents the CD diagrams in terms of five different metrics.



Continued on next page

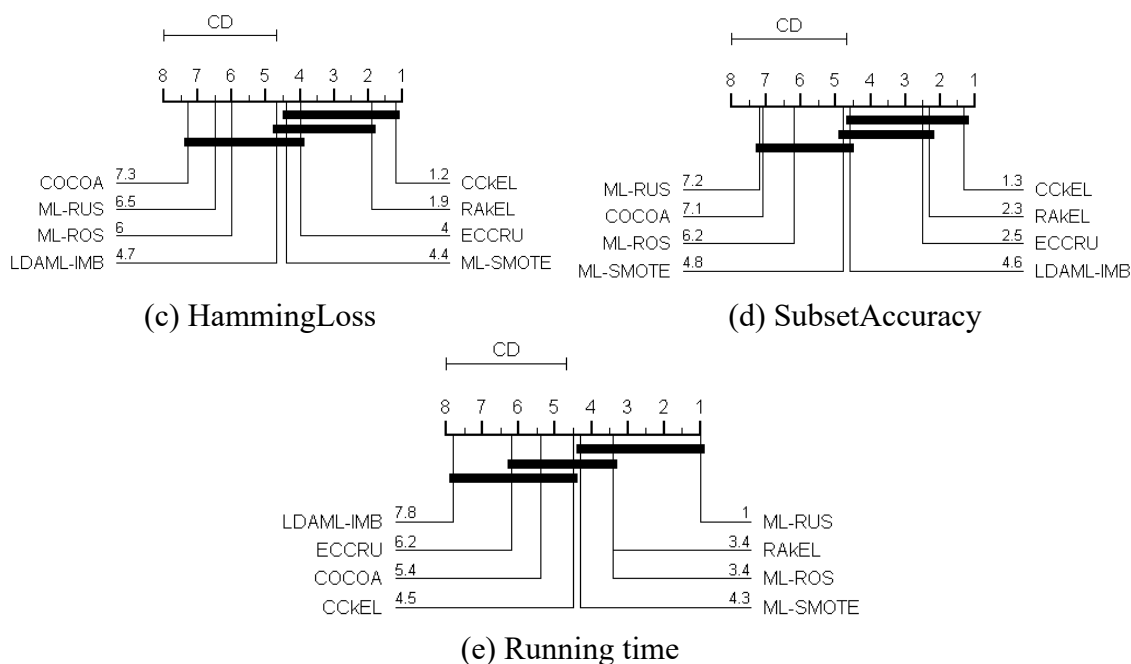


Figure 5. CD diagrams of various comparative algorithms at a standard level of significance $\alpha = 0.05$.

The results in Figure 5 illustrate that the proposed *CCkEL* algorithm performs best on F-micro, HammingLoss, and SubsetAccuracy metrics, although its superiority is not significant in comparison to several other algorithms. Specifically, on the F-micro metric, the *CCkEL* is significantly superior to several sampling-based algorithms (i.e., ML-RUS, ML-ROS, and ML-SMOTE). On the HammingLoss metric, the *CCkEL* performs significantly better than COCOA, ML-ROS, ML-RUS, and LDAML-IMB algorithms. For the SubsetAccuracy metric, the proposed *CCkEL* algorithm not only significantly outperforms three sampling-based algorithms but also presents a significant superiority in comparison to the LDAML-IMB algorithm. Also, it is noteworthy that on the F-macro metric, the *CCkEL* algorithm performs worse than three other complex algorithms (ECCRU, COCOA, and LDAML-IMB), but the differences among them are not significant. On the running time metric, we observe that the proposed *CCkEL* algorithm is significantly more time consuming than the ML-RUS algorithm but more time saving than the LDAML-IMB algorithm. In summary, the statistical results indicate that the proposed *CCkEL* algorithm is an accurate and stable solution for classifying imbalanced multi-label data.

4.5. Investigation about the impact of parameters

Finally, we investigated the impact of several key parameters used in our proposed algorithms. Figure 6 shows the variation of various performances with gradually increasing k on nine multi-label datasets, where three representative k values, $k = 3$, $k = q/3$, and $k = q/2$ are specifically labeled. Although there are some fluctuations, most curves present a common trend, i.e., the performance first rises with the increase of k value and then gradually declines. This means that the *CCkEL* algorithm is not appropriate for either designating a too small value or assigning an oversize k value. We believe that a too-small value of k tends to lose some important correlation information, while an oversized k can not only destroy the concentration of the learning model but also greatly add the number of classes

in each transformed multi-class learning model to make it more complicated and inaccurate. In addition, with the increment of k , the running time of the $CCkEL$ algorithm also tends to be higher. Further, we observed the $CCkEL$ algorithm to be relatively stable in the context of a specific k value. According to the results from Figure 6, we suggest users to designate k as $\max(q/3, 3)$ by a rule-of-thumb setting.

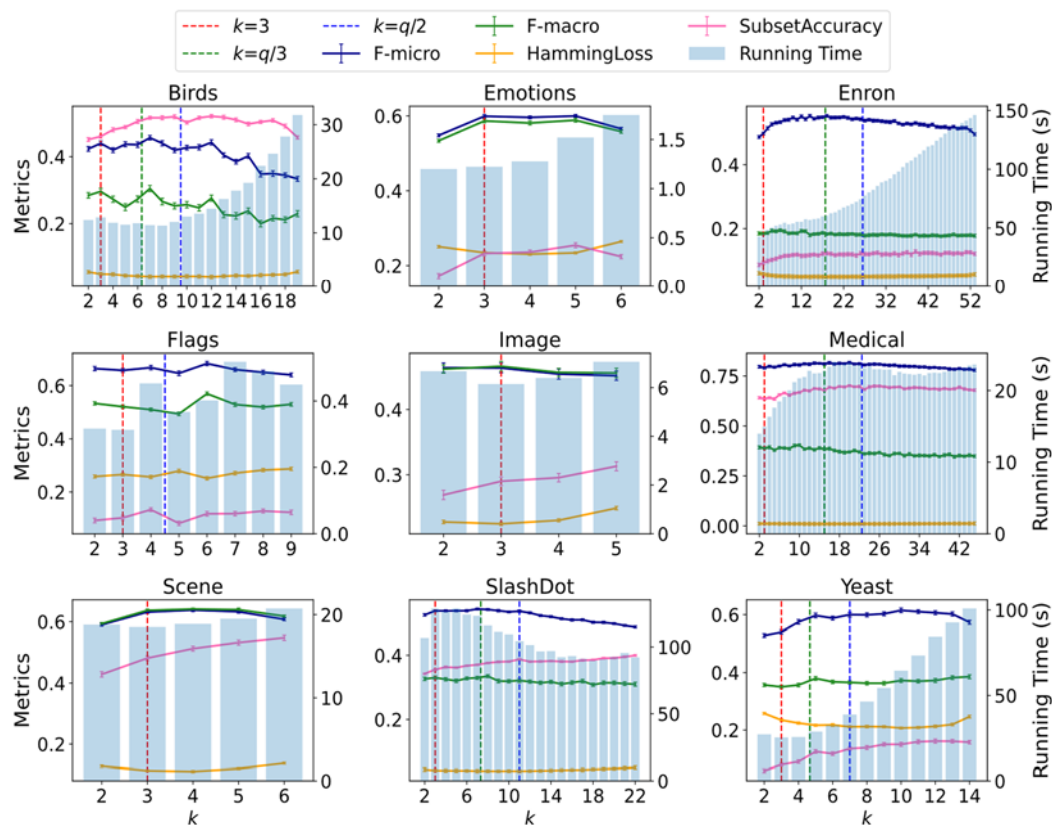


Figure 6. The performance variation of the $CCkEL$ algorithm by changing the parameter k .

Figures 7 and 8 present the performance variance of F-micro and F-macro metrics with a joint change of the parameters λ_1 and λ_2 . Specifically, λ_1 varies from 0.01 to 0.20 with an increment of 0.01, and λ_2 varies from 1 to 20 with an increment of 1. We randomly conducted 10 experiments on each parameter combination and used the average result to indicate its performance. The results in Figures 7 and 8 present a small performance fluctuation, meaning that although these two parameters can influence classification performance, the influence is not large enough. We consider this to be associated with two reasons. The first one is that the modified factor C varies in a moderate range (1, 2), which can more or less amend the skewed decision threshold, making the worst performance to also be acceptable. The second one is that, in multi-label data, the instances on some labels are easy to be classified, while it may be difficult to distinguish the instances on some other labels; thus, the compensation may effectively improve the classification performance on some labels but destroy that of some other labels. All together, the results reflect that it is relatively safe to designate a small value for the gradient parameter λ_1 and a moderate control parameter λ_2 .

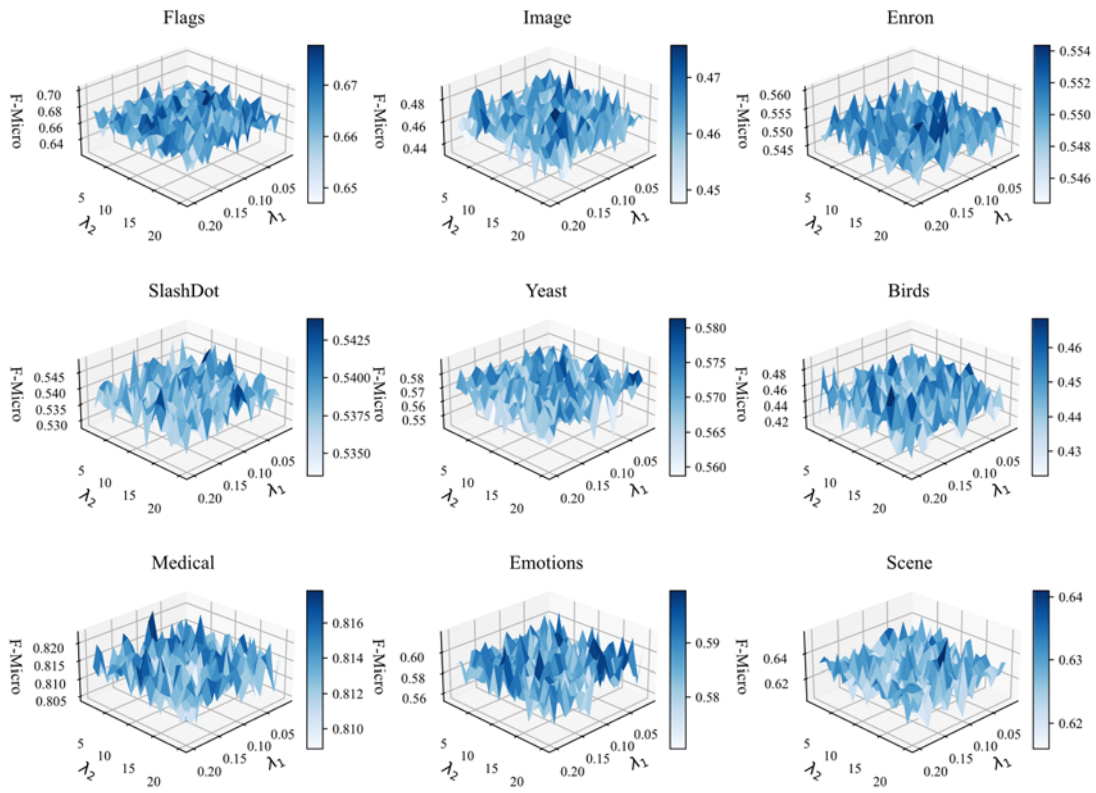


Figure 7. The F-micro performance variation of the CCkEL algorithm with joint change of the parameters λ_1 and λ_2 .

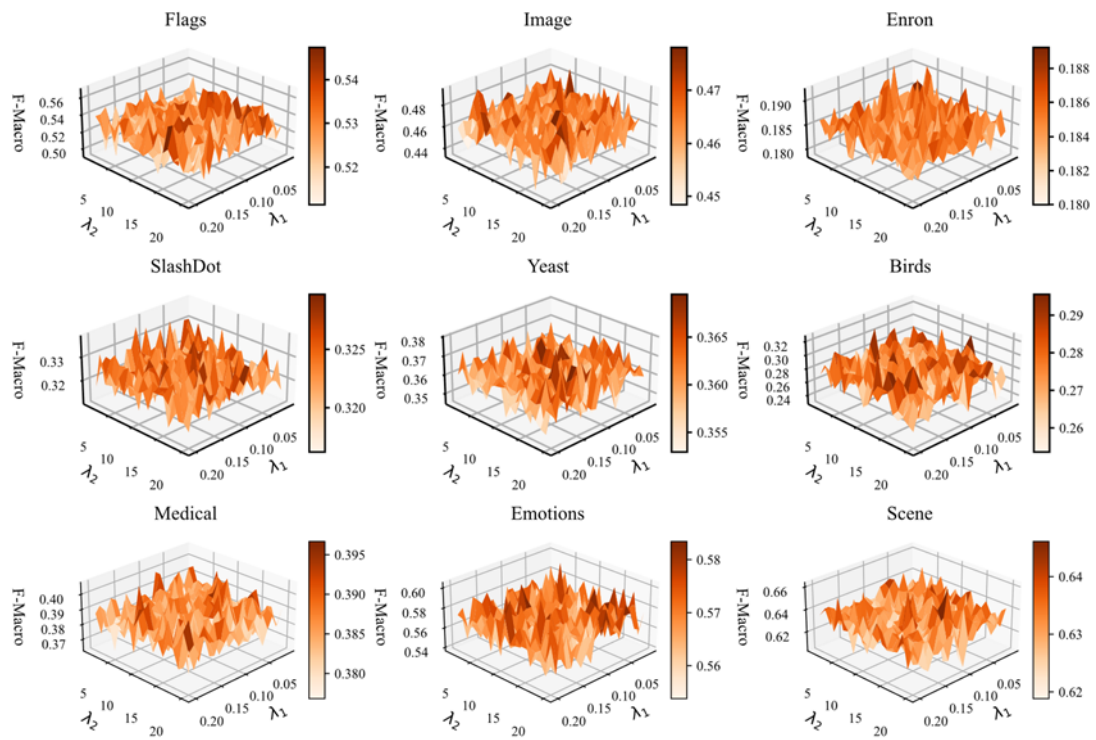


Figure 8. The F-macro performance variation of the CCkEL algorithm with joint change of the parameters λ_1 and λ_2 .

5. Conclusions and future research

In this paper, we proposed a novel multi-label imbalance learning algorithm named *CCkEL*, which simultaneously takes advantage of the strong label correlation information existing in the label space and adapts the skewed data distribution in the decision space. It modifies the original *RAkEL* algorithm by extracting strongly correlated labels to make the learning model focus more on the boundary details. Additionally, a fast compensation-based strategy has been developed to amend the biases caused by skewed data distribution on the learning model trained on each transformed multi-class dataset. The experimental results on ten benchmark multi-label datasets illustrate the superiority of the proposed algorithm in comparison with several state-of-the-art multi-label imbalance learning algorithms. Also, the running time results showed that the *CCkEL* is a relatively efficient algorithm, especially when compared with several complex ensemble-based multi-label imbalance learning models.

There are two significant limitations in the proposed algorithm. First, the *CCkEL* is not flexible enough, as it limits the number of correlated labels for any one specific label once k has been designated. That means that if a label has none or only a few correlated labels, some noisy information could be introduced; on the other hand, if a label correlates with a lot of other labels, some useful information could be missed. Second, the compensation is adaptive but not optimum, as we abandon the optimized threshold strategies [26,27]. Accordingly, its computational complexity improves tremendously.

In future work, we plan to explore more flexible strategies to get rid of the limitation of the parameter k for further improving the adaption of the algorithm. In addition, the threshold strategy is expected to be replaced by more effective and efficient cost-sensitive learning models to the transformed multi-class learning tasks. Also, we wish to extend the scalability of the experimental data in terms of features, instances, and labels, and adopt more real-world applications containing large-scale multi-label data [43–45] to further verify the effectiveness and robustness of the proposed *CCkEL* algorithm.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62176107, 62076111, 62076215, and the Open Project of Jiangsu Key Laboratory of Media Design and Software Technology (Jiangnan University).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. M. L. Zhang, Z. H. Zhou, A review on multi-label learning algorithms, *IEEE Trans. Knowl. Data Eng.*, **26** (2013), 1819–1837. <https://doi.org/10.1109/TKDE.2013.39>
2. Z. Shao, W. Zhou, X. Deng, M. Zhang, Q. Cheng, Multilabel remote sensing image retrieval based on fully convolutional network, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **13** (2020), 318–328. <https://doi.org/10.1109/JSTARS.2019.2961634>
3. Z. Zhang, Q. Zou, Y. Lin, L. Chen, S. Wang, Improved deep hashing with soft pairwise similarity for multi-label image retrieval, *IEEE Trans. Multimedia*, **22** (2019), 540–553. <https://doi.org/10.1109/TMM.2019.2929957>
4. X. Zhang, J. Xu, C. Soh, L. Chen, LA-HCN: label-based attention for hierarchical multi-label text classification neural network, *Expert Syst. Appl.*, **187** (2022), 115922. <https://doi.org/10.1016/j.eswa.2021.115922>
5. Z. Yang, F. Emmert-Streib, Optimal performance of Binary Relevance CNN in targeted multi-label text classification, *Knowledge-Based Syst.*, **284** (2023), 111286. <https://doi.org/10.1016/j.knosys.2023.111286>
6. R. Su, H. Yang, L. Wei, S. Chen, Q. Zou, A multi-label learning model for predicting drug-induced pathology in multi-organ based on toxicogenomics data, *PLoS Comput. Biol.*, **18** (2022), e1010402. <https://doi.org/10.1371/journal.pcbi.1010402>
7. S. Wan, M. K. Mak, S. Y. Kung, mGOASVM: Multi-label protein subcellular localization based on gene ontology and support vector machines, *BMC Bioinf.*, **13** (2012), 1–16. <https://doi.org/10.1186/1471-2105-13-290>
8. K. C. Chou, Advances in predicting subcellular localization of multi-label proteins and its implication for developing multi-target drugs, *Curr. Med. Chem.*, **26** (2019), 4918–4943. <https://doi.org/10.2174/0929867326666190507082559>
9. H. Wang, L. Yan, H. Huang, C. Ding, From protein sequence to protein function via multi-label linear discriminant analysis, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **14** (2016), 503–513. <https://doi.org/10.1109/TCBB.2016.2591529>
10. M. R. G. A. De Oliveira, P. M. Ciarelli, E. Oliveira, Recommendation of programming activities by multi-label classification for a formative assessment of students, *Expert Syst. Appl.*, **40** (2013), 6641–6651. <https://doi.org/10.1016/j.eswa.2013.06.011>
11. M. L. Zhang, Y. K. Li, X. Y. Liu, X. Geng, Binary relevance for multi-label learning: an overview, *Front. Comput. Sci.*, **12** (2018), 191–202. <https://doi.org/10.1007/s11704-017-7031-7>
12. J. Fürnkranz, E. Hüllermeie, E. Loza Mencía, K. Brinker, Multilabel classification via calibrated label ranking, *Mach. Learn.*, **73** (2008), 133–153. <https://doi.org/10.1007/s10994-008-5064-8>
13. M. L. Zhang, Z. H. Zhou, ML-KNN: A lazy learning approach to multi-label learning, *Pattern Recognit.*, **40** (2007), 2038–2048. <https://doi.org/10.1016/j.patcog.2006.12.019>
14. M. L. Zhang, Z. H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, *IEEE Trans. Knowl. Data Eng.*, **18** (2006), 1338–1351. <https://doi.org/10.1109/TKDE.2006.162>
15. M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification, *Pattern Recognit.*, **37** (2004), 1757–1771. <https://doi.org/10.1016/j.patcog.2004.03.009>
16. J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, *Mach. Learn.*, **85** (2011), 333–359. <https://doi.org/10.1007/s10994-011-5256-5>
17. G. Tsoumakas, I. Katakis, I. Vlahavas, Random k -labelsets for multilabel classification, *IEEE Trans. Knowl. Data Eng.*, **23** (2010), 1079–1089. <https://doi.org/10.1109/TKDE.2010.164>

18. A. N. Tarekegn, M. Giacobini, K. Michalak, A review of methods for imbalanced multi-label classification, *Pattern Recognit.*, **118** (2021), 107965. <https://doi.org/10.1016/j.patcog.2021.107965>
19. A. Zhang, H. Yu, S. Zhou, Z. Huan, X. Yang, Instance weighted SMOTE by indirectly exploring the data distribution, *Knowledge-Based Syst.*, **249** (2022), 108919. <https://doi.org/10.1016/j.knosys.2022.108919>
20. A. Zhang, H. Yu, Z. Huan, X. Yang, S. Zheng, S. Gao, SMOTE-RkNN: A hybrid re-sampling method based on SMOTE and reverse k -nearest neighbors, *Inf. Sci.*, **595** (2022), 70–88. <https://doi.org/10.1016/j.ins.2022.02.038>
21. K. E. Bennin, J. Keung, P. Phannachitta, A. Monden, S. Mensah, MAHAKIL: Diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction, *IEEE Trans. Software Eng.*, **44** (2018), 534–550. <https://doi.org/10.1109/TSE.2017.2731766>
22. M. Zhang, T. Li, X. Zheng, Q. Yu, C. Chen, D. D. Zhou, et al., UFFDFR: Undersampling framework with denoising, fuzzy c-means clustering, and representative sample selection for imbalanced data classification, *Inf. Sci.*, **576** (2021), 658–680. <https://doi.org/10.1016/j.ins.2021.07.053>
23. R. Batuwita, V. Palade, FSVM-CIL: fuzzy support vector machines for class imbalance learning, *IEEE Trans. Fuzzy Syst.*, **18** (2010), 558–571. <https://doi.org/10.1109/TFUZZ.2010.2042721>
24. C. L. Castro, A. P. Braga, Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data, *IEEE Trans. Neural Networks Learn. Syst.*, **24** (2013), 888–899. <https://doi.org/10.1109/TNNLS.2013.2246188>
25. Z. H. Zhou, X. Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, *IEEE Trans. Knowl. Data Eng.*, **18** (2005), 63–77. <https://doi.org/10.1109/TKDE.2006.17>
26. 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, *Knowledge-Based Syst.*, **76** (2015), 67–78. <https://doi.org/10.1016/j.knosys.2014.12.007>
27. H. Yu, C. Sun, X. Yang, W. Yang, J. Shen, Y. Qi, ODOC-ELM: Optimal decision outputs compensation-based extreme learning machine for classifying imbalanced data, *Knowledge-Based Syst.*, **92** (2016), 55–70. <https://doi.org/10.1016/j.knosys.2015.10.012>
28. G. Collell, D. Prelec, K. R. Patil, A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data, *Neurocomputing*, **275** (2018), 330–340. <https://doi.org/10.1016/j.neucom.2017.08.035>
29. P. Lim, C. K. Goh, K. C. Tan, Evolutionary cluster-based synthetic oversampling ensemble (ECO-Ensemble) for imbalance learning, *IEEE Trans. Cybern.*, **47** (2017), 2850–2861. <https://doi.org/10.1109/TCYB.2016.2579658>
30. S. E. Roshan, S. Asadi, Improvement of Bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization, *Eng. Appl. Artif. Intell.*, **87** (2020), 103319. <https://doi.org/10.1016/j.engappai.2019.103319>
31. H. Yu, J. Ni, An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data, *IEEE/ACM Trans. Comput. Biol. Bioinf.*, **11** (2014), 657–666. <https://doi.org/10.1109/TCBB.2014.2306838>
32. H. G. Zefrehi, H. Altincay, Imbalance learning using heterogeneous ensembles, *Expert Syst. Appl.*, **142** (2020), 113005. <https://doi.org/10.1016/j.eswa.2019.113005>
33. X. M. An, S. Xu, A selective evolutionary heterogeneous ensemble algorithm for classifying imbalanced data, *Electron. Res. Arch.*, **31** (2023), 2733–2757. <https://doi.org/10.3934/era.2023138>

34. F. Charte, A. J. Rivera, M. J. del Jesus, F. Herrera, Addressing imbalance in multilabel classification: Measures and random resampling algorithms, *Neurocomputing*, **163** (2015), 3–16. <https://doi.org/10.1016/j.neucom.2014.08.091>
35. F. Charte, A. J. Rivera, M. J. del Jesus, F. Herrera, ML-SMOTE: Approaching imbalanced multi-label learning through synthetic instance generation, *Knowledge-Based Syst.*, **89** (2015), 385–397. <https://doi.org/10.1016/j.knosys.2015.07.019>
36. M. Zhang, Y. K. Li, H. Yang, Towards class-imbalance aware multi-label learning, *IEEE Trans. Cybern.*, **52** (2020), 4459–4471. <https://doi.org/10.1109/TCYB.2020.3027509>
37. B. Liu, G. Tsoumakas, Dealing with class imbalance in classifier chains via random undersampling, *Knowledge-Based Syst.*, **192** (2020), 105292. <https://doi.org/10.1016/j.knosys.2019.105292>
38. Y. Peng, E. Huang, G. Chen, C. Wang, J. Xie, A general framework for multi-label learning towards class correlations and class imbalance, *Intell. Data Anal.*, **23** (2019), 371–383. <https://doi.org/10.3233/IDA-183932>
39. J. Rice, R. J. Belland, A simulation study of moss floras using Jaccard's coefficient of similarity, *J. Biogeogr.*, **9** (1982), 411–419. <https://doi.org/10.2307/2844573>
40. J. R. Quinlan, Improved use of continuous attributes in C4.5, *J. Artif. Intell. Res.*, **4** (1996), 77–90. <https://doi.org/10.1613/jair.279>
41. J. Demsar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.*, **7** (2006), 1–30. <https://doi.org/10.1007/s10846-005-9016-2>
42. S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Inf. Sci.*, **180** (2010), 2044–2064. <https://doi.org/10.1016/j.ins.2009.12.010>
43. S. Pandya, T. R. Gadekallu, P. K. Reddy, W. Wang, M. Alazab, InfusedHeart: A novel knowledge-infused learning framework for diagnosis of cardiovascular events, *IEEE Trans. Comput. Social Syst.*, **2022** (2022), 1–10. <http://doi.org/10.1109/TCSS.2022.3151643>
44. L. Zhang, J. Wang, W. Wang, Z. Jin, Y. Su, H. Chen, Smart contract vulnerability detection combined with multi-objective detection, *Comput. Networks*, **217** (2022), 109289. <https://doi.org/10.1016/j.comnet.2022.109289>
45. X. Liu, T. Shi, G. Zhou, M. Liu, Z. Yin, L. Yin, et al., Emotion classification for short texts: an improved multi-label method, *Humanit. Social Sci. Commun.*, **10** (2023), 1–9. <https://doi.org/10.1057/s41599-023-01816-6>



AIMS Press

©2024 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)