Parallel Support Vector Machines for Multi-Label Classification in Imbalanced Databases

Yanjie Wang[1*], Lei Song[2]
[1]Institute of Information Engineering, Zhengzhou College of Finance and Economics, Zhengzhou 450000, China
[2]Department of Information Engineering, Zhengzhou Railway Technician College, Zhengzhou 450041, China
E-mail: wyj99yongyou2@163.com, sl9188jsj@126.com
[*]Corresponding author

Keywords: parallel support vector machines, imbalance, sample databases, multi-labeling, categorical mining

Received: February 20, 2025

We propose a multi-label classification mining method using parallel support vector machines for imbalanced sample databases. The samples within the unbalanced sample database are partitioned into the majority sub-cluster and the minority sub-cluster by means of the hierarchical clustering algorithm, thereby achieving the oversampling of the unbalanced sample database. Using hierarchical clustering algorithm to divide into majority and minority sub clusters, complete oversampling of imbalanced sample database. Clustering itself does not directly generate new samples, but it divides the data into sub clusters, allowing oversampling to be more targeted in the sub clusters of minority classes, which can avoid noise or overfitting problems caused by blind oversampling. The role of clustering algorithms is to provide structured data partitioning basis for oversampling. Improve the accuracy of minority class classification in imbalanced sample databases through parallel computing, and use MapReduce to solve SVM dual problems in parallel to optimize hyperplanes for multi label classification. By using the Map function to divide the training sample set into small sample sets and train support vector machines, these support vector machines are then integrated in the Reduce stage to train a new support vector machine as the final decision function, in order to efficiently handle multi label classification problems. The experimental results show that the studied method consistently maintains a high accuracy of 0.95 or higher on the G-means index, far exceeding the comparison methods; In terms of acceleration ratio, when the sample size increased from 1000 to 10000, the acceleration ratio of our method steadily improved from 1.0 to 2.5, while the two comparison methods only reached 1.5 and 2.0 respectively, and there were significant fluctuations.

Povzetek: Za hitro, porazdeljeno in uravnoteženo večoznačno klasifikacijo velikih in neuravnoteženih podatkov z izboljšano natančnostjo manjšinskih razredov ter učinkovito uporabo virov v rudarjenju podatkov je razvit PSVM-MLC, paralelni sistem podpornih vektorjev na osnovi MapReduce. Metoda uporablja hierarhično grozdenje za ciljno nadvzorčeno nadvzorčenje manjšinskih razredov in s tem prepreči šum.

1 Introduction

Machine learning algorithms rely on observational data samples to discover patterns, and employ these patterns to predict future data or data that cannot be directly observed [1]. This has become a crucial technology for resolving numerous practical issues. Support Vector Machine (SVM) is a data mining algorithm. Data mining [2] is the process of using algorithms to search for hidden information from large amounts of data, which may be unknown, interesting, and useful for specific applications. In the classification issue, SVM looks for a hyperplane to maximize the separation between distinct categories, thereby attaining precise classification of new samples [3]. This approach excels in managing high-dimensional data, nonlinear challenges, and small sample datasets, and is extensively utilized in data mining. SVM maps the input space to a higher dimensional feature space by constructing a kernel function, and finds the optimal hyperplane in this feature space to achieve classification. Due to the fact that SVM only considers a small number of support vectors when constructing models, it has a certain robustness to data sparsity and noise. The multi label classification problem refers to the situation where a sample can belong to multiple categories simultaneously [4]. In image recognition, an image may contain multiple objects; In text classification, an article may belong to multiple topics simultaneously. This type of problem poses higher requirements for classification algorithms, which not only need to consider accurate classification of individual labels [5], but also need to deal with the correlation between labels and the imbalance of samples. Sample imbalance is a common problem, where the number of samples in certain categories far exceeds that of other categories, resulting in the model leaning towards majority class samples during training and insufficient learning of minority class samples, which affects the

overall classification performance. For the multi label classification problem [6], this imbalance is even more complex because each sample may belong to multiple imbalanced categories simultaneously. To address these challenges, researchers have proposed various solutions such as oversampling, undersampling, ensemble learning, etc. Exploring parallel support vector machine algorithms and utilizing parallel computing techniques to improve training speed and classification performance has become an important research direction.

In recent years, many scholars have studied multi label classification mining in unbalanced sample databases. For example, Moral-Garcia et al. used Credal C4.5 to rank calibration labels in multi label classification [7]. Credal C4.5 uses imprecise probability to deal with noise in data, which is particularly important in multi label classification. This approach establishes a binary classifier for each pair of labels and employs the calibration function of Credal C4.5 to mitigate the issue of category imbalance to some extent, thereby enhancing the recognition accuracy of minority categories. Consider the correlation between each pair of tags to build tag ranking, which is helpful to more accurately predict multiple tags of an instance. However, the performance of Credal C4.5 is affected by its internal parameters. When dealing with imprecise probability, the setting of upper bound and lower bound functions has a significant impact on the final classification results. Udandarao et al. use the attention based multitask cyclic network to classify multi label physical text [8], and use the deep learning model to automatically extract features from the original text data without manually constructing features, reducing manual intervention and costs. The introduction of attention mechanism enables the model to dynamically focus on key information in the text, further improving the accuracy of feature extraction. Multi task learning allows the model to learn multiple related tasks at the same time. By sharing the presentation layer, different tasks can promote each other and improve the overall performance. In physical text classification, if there is association or sharing of some features between different tags, multi task learning can effectively use these commonalities to improve the classification effect. The attention mechanism can assign varying weights to different segments of the text, enabling the model to focus more on key information pertinent to labels during classification. Nonetheless, in multi-label classification, there exists interference among different labels. Especially when there are multiple keywords related to different tags in the text, the model will cause classification errors due to improper allocation of attention mechanism. Qaraei and Babbar studied the classifier negative sampling method for extreme multi label classification [9]. The negative sampling technique only selects part of the negative samples for training, which significantly reduces the computational complexity and improves the training efficiency. Negative sampling helps the model better learn to distinguish between the boundaries of positive and negative samples. It forces the model to pay more attention to those samples that are clearly marked as negative in the training process, which helps the model to more accurately judge which labels are not applicable to the current instance when predicting. However, negative sampling technology is prone to lead to sample selection bias. In extreme multi label classification, the distribution of labels is often very unbalanced. If the selection of negative samples is not random or representative enough, the model will learn biased feature representation, affecting its generalization ability on new data. Bogatinovski et al. studied the multi label classification method with dataset attributes [10]. When processing multi label datasets, they can better identify and allocate multiple related labels to each instance. Considering the diversity and complexity of dataset attributes, it can learn the potential patterns in the data and show good generalization ability on new data. However, the performance of multi label classification methods largely depends on the quality of data sets and the accuracy of labels. If there are noise or label errors in the dataset, the accuracy of the classification results will be directly affected. Stefanovic et al. proposed a multi label text data class based on self-organizing mapping and latent semantic analysis [11]. Text data is preprocessed using multiple types of filters to remove redundant and irrelevant information. Latent semantic analysis is used for dimensionality reduction processing, mapping high-dimensional text vectors to a low dimensional latent semantic space by constructing a semantic space, while preserving core semantic features. Cosine similarity is applied to optimize multi label classification by quantifying vector directional similarity to identify the label categories that need to be adjusted. The self-organizing mapping neural network discovers data topology structure through competitive learning mechanism, achieves text similarity clustering, and provides decision-making basis for new text category allocation. However, although the linear transformation based on singular value decomposition in latent semantic analysis can capture explicit semantic features, it cannot effectively handle complex language phenomena such as synonym ambiguity and context dependence, resulting in the loss of fine-grained semantic information.

The summary of the existing research mentioned above is shown in Table 1.

Table 1: Summary of existing research

Methods	Data set	Index	Defect
Traditional C4.5 CLR [7]	Unbalanced sample database	Classification accuracy	Neglecting label correlation, G-means<0.85 under imbalanced data
Multi task recurrent network based on attention [8]	CBSE Physics Textbook (Grades 6-12)	Classification accuracy	High computational complexity and fluctuating

			acceleration ratio (1.5-2.0)
Extreme multi label classification method [9]	Unclear	Training efficiency	Sample selection bias affects generalization ability
Dataset attribute method [10]	40 MLC datasets+50 meta features	Multi label classification	Hyperparameter optimization consumes a large number of resources, and the improvement effect is not proportional to the resource consumption
Self organizing mapping and latent semantic analysis [11]	Public website	Correct allocation rate	When latent semantic analysis reduces the data dimension to 40, it obtains 82% correct allocation

To address the issues with the above methods in label classification, this paper explores a multi label classification mining technique for imbalanced sample databases based on parallel support vector machines. The parallelization architecture of parallel support vector machines utilizes the MapReduce framework to block and process large-scale data, significantly improving computational efficiency. By dividing data into sub clusters through hierarchical clustering, it is possible to accurately identify the distribution characteristics of minority class samples, provide structured basis for oversampling, and avoid model bias caused by blind sampling. Not only does it overcome the classification bias problem of traditional SVM in handling imbalanced data, but it also achieves efficient processing of massive data through distributed computing, providing a solution that balances speed and accuracy for multi label classification tasks. Compared to state-of-the-art attention based multi task recurrent networks, this method significantly improves classification performance on imbalanced datasets through structured oversampling and parallelization, providing a better solution for massive data mining.

2 Multi-label classification mining methods for unbalanced sample databases

For imbalanced sample databases, a hierarchical clustering algorithm is used to divide majority and minority class samples into sub clusters. By calculating the sub cluster misclassification rate, the oversampling weight is determined, and sub clusters with higher misclassification rates are given greater weight for priority processing. Based on the roulette wheel mechanism, select seed samples and combine them with neighboring samples to synthesize new data, ensuring the randomness of the synthesized samples and the authenticity of the data distribution. This process balances inter class differences through dynamic weight allocation, while avoiding model bias caused by oversampling, ultimately improving the representativeness of minority class samples and optimizing the overall data distribution.

Implementing parallel SVM algorithm based on MapReduce framework, the Map stage divides the data into subsets and solves local Lagrange multipliers in a distributed manner to extract support vectors. In the Reduce stage, the global support vectors are aggregated and retrained to generate the final classifier. Mapping data to high-dimensional space through kernel functions, constructing a maximum interval hyperplane, and optimizing the model's generalization ability based on the principle of minimizing structural risk. Parallelization significantly improves computational efficiency, effectively solves the problem of imbalanced data classification bias, and enhances the accuracy of minority class recognition.

2.1 Oversampling treatment

To acquire more effective sample information, sampling is conducted on the samples within the unbalanced sample database. When oversampling the imbalanced sample database, the imbalance of data both between and within classes is thoroughly considered. A hierarchical clustering algorithm is employed to partition the majority class samples in the imbalanced dataset into multiple majority class subclusters. Subsequently, the minority class samples are divided into different minority class subclusters based on the majority class samples.

The notions of misclassification rate and oversampling weight are brought in for oversampling the samples within the unbalanced sample database. The misclassification rate is employed to signify the proportion of the quantity of samples misclassified by the support vector machine classifier for a subcluster to the overall number of samples in the entire subcluster [12], represented as $E(C\min_t)$, and then the following holds:

$$E(C\min_t) = k_t / m_t \tag{1}$$

Among them, k_t denotes the number of misclassified samples in the minority class subcluster $C\min_t$, m_t denotes the total number of samples in the minority class subcluster $C\min_t$.

The oversampling weight is the product of the weight of the misclassification rate of the subcluster, the difference between the number of samples in the majority class and the number of samples in the minority class, denoted as $W(C\min_t)$, then there is:

$$W\left(C\min_t\right)=\frac{E\left(C\min_t\right)}{\sum_{t=1}^{n}E\left(C\min_t\right)}\times\left(N_{maj}-N_{\min}\right)\times\delta \quad (2)$$

Among them, N_{maj} denotes the number of majority class samples in the original unbalanced samples database, $N_{\min}$ denotes the number of minority class samples in the original unbalanced sample database, $\delta\in[0,1]$ indicates the oversampling rate.

The proportion of misclassification rate reflects the relative importance of sub cluster classification errors. The oversampling rate controls the replication factor of minority class samples, while the oversampling weight combines the two and the difference in the number of categories to dynamically determine the number of samples that each sub cluster needs to generate. Priority is given to increasing data in areas where classification is difficult and samples are scarce.

After sub-clustering the minority class samples in the imbalanced sample database, different oversampling weights are assigned to the sub-clusters according to their misclassification rates. From Equation (2), the more the number of misclassified samples in the minority class subcluster [13], then the larger the $W\left(C\min_t\right)$, the larger the oversampling weight required. The oversampling weights are assigned to subclusters according to their misclassification rates to achieve inter-class data balance.

The probability distribution of the subcluster of the minority class is reintroduced. In the subcluster $C\min_t$ of the minority class, when $\forall x\in C\min_t$, x is selected as the "seed sample" to constitute the probability distribution of the subcluster $C\min_t$, denoted as P, then there is:

$$P=W\left(C\min_t\right)\left(\frac{1/\sum_{t=1}^{k}d_{xy_t}}{\sum\left(1/\sum_{t=1}^{k}d_{xy_t}\right)}\right)_{l\times n} \quad (3)$$

Among them, y_t represents the t majority class sample nearest neighbor of x, where $1\le t\le k$. d_{xy_t} denotes the Euclidean distance between the minority class sample x and the majority class sample y_t, n signifies the number of samples in the minority class subcluster, and k is the number of near-neighbor samples.

The selection probability of seed samples is determined by the distance from the sample to the nearest neighbors of the majority class. The closer the distance, the higher the probability. This can make minority class samples closer to the classification boundary more likely to be selected for oversampling, thereby enhancing the model's learning ability in the boundary region.

Based on the probability distribution of minority subclusters, we employ a roulette selection method to choose "seed samples," and subsequently, randomly select one of the neighboring minority samples for oversampling. This random selection approach ensures that the synthetic samples exhibit randomness [14], thereby better mimicking the original data distribution within the unbalanced sample database.

To prevent oversampling of certain sub-clusters that could bias the support vector machine classifier toward these sub-clusters, all minority sub-clusters in the imbalanced sample database are assigned oversampling weights to achieve intra-class data balance [15]. By selecting "seed samples" and their nearest neighbor samples from the same minority sub-cluster, we can both avoid choosing nearest neighbors that are too distant from the seed samples and mitigate the over-coverage phenomenon caused by synthetic samples.

The steps for dividing the minority class subclusters in the unbalanced sample database are as follows:

(1) Initialize each minority class sample in the unbalanced sample database as a separate minority class subcluster; the

(2) If there are no majority class samples present between the two closest minority class subclusters, the two-minority class subclusters are combined.

(3) Continue reiterating steps (1) and (2) until the separation between the subgroups diminishes to below the predetermined threshold, thereby concluding the iteration process.

Oversampling of data in the unbalanced sample database consists of 3 processes:

(1) Divide the minority class samples to form different minority class subclusters;

(2) Calculate the misclassification rate of each subcluster and the oversampling weight of the subcluster in the unbalanced sample database [16];

(3) The probability distribution within each underrepresented subcluster is ascertained using formula (3). Based on this distribution and the oversampling weights, "seed exemplars" and their proximate minority samples are identified for oversampling purposes, with synthetic minority samples subsequently being generated. Using the results of step (2), repeat in step (3) until the number of iterations reaches the oversampling weight, end the cycle, and output the oversampled data set $X=\{x_1,x_2,...,x_i,...,x_n\}$.

Through the aforementioned methodology, the process of sample oversampling within the imbalanced database is finalized, resulting in a more even distribution of samples. This, in turn, enhances the precision of multi-label classification mining operations within the said imbalanced database.

Hierarchical clustering effectively captures the intrinsic structure of data through multi-level sample aggregation, making it particularly suitable for handling imbalanced data with complex inter class distributions. Compared to hard clustering methods such as K-means, it does not require a preset number of clusters and reveals the hierarchical relationship of samples through tree visualization. For example, in medical diagnostic data, hierarchical clustering can naturally distinguish the nested relationship between rare case subtypes and

mainstream cases, while K-means may forcibly classify sparse minority class samples into majority classes due to initial center sensitivity. The bottom-up merging strategy based on distance threshold can preserve local sample density features and avoid cluster splitting problems caused by global parameters in DBSCAN.

2.2 Multi-label classification mining based on parallel support vector machine

Within an imbalanced sample database, the disparity in sample counts between certain categories can skew the training of classification models towards the prevalent categories, hindering the recognition of underrepresented classes. Augmenting the number of samples belonging to the minority class through oversampling enables a more equitable distribution across classes. Consequently, introducing dataset $X=\{x_1,x_2,...,x_i,...,x_n\}$ into a parallel support vector machine framework enables a more nuanced capture of the features specific to the minority class, ultimately boosting the classification accuracy for these underrepresented instances.

While Support Vector Machine (SVM) excels at handling small sample sizes, its performance falters when confronted with imbalanced sample databases. To bolster its processing capabilities, this study incorporates the MapReduce programming paradigm into the nonlinear SVM algorithm, realizing a parallel SVM implementation grounded in MapReduce [17].

Map stage: Cut the input oversampled data set $X=\{x_1,x_2,...,x_i,...,x_n\}$ into multiple equal subsets of data, and then allocate the data subsets to the idle Map work units. Finally, the work units solve the Lagrange multipliers on each data subset in parallel in a distributed manner. The sample points corresponding to non-zero Lagrange multipliers are support vector machines.

Reduce stage: Upon completion of each map operation, the locally obtained support vectors are combined as Reduce input. All support vector machines undergo retraining, with the final training results serving as classifiers and the retraining results representing the global optimal solution. The samples corresponding to the support vector machines are saved to local files [18].

The parallel support vector machine algorithm harnesses the power of SVM for executing multi-label classification mining in imbalanced sample databases. This approach translates the multi-label classification challenge inherent in such databases into a series of binary classification tasks. SVM, as a learning mechanism, is optimized through structural risk minimization (SRM), which involves the simultaneous minimization of two opposing goals. First, empirical risk is minimized based on available data. However, as model complexity increases, observed errors on the training data may decrease to arbitrarily low levels, potentially causing increased errors on unseen data due to model overfitting. Second, structural risk minimization (SRM) includes minimizing a monotonic function term related to test error, known as structural risk, which depends directly on model complexity. For linear systems, this complexity grows proportionally with the norm of the system's parameters [19].

For the dichotomy classification problem, SVM's fundamental approach identifies an optimal hyperplane in the sample space to maximize the separation margin between two distinct sample classes. The training set is defined as follows:

$$(X,T)=\left\{\left(x_i,t_i\right),i=1,2,...,n\right\} \tag{4}$$

Among them, t_i is the category tags of the Sample x_i, $t_i\in\{-1,1\}$.

Introducing nonlinear mappings $\varphi(X)$, mapping the training set into a high-dimensional space:

$$(\varphi(X),T)=\left\{\left(\varphi(x_i),t_i\right),i=1,2,...,n\right\} \tag{5}$$

The chosen kernel function is:

$$K(x,y)=\varphi(x)^T\varphi(y) \tag{6}$$

Introducing slack variables $\xi_i\geq 0$, constructing standard support vector machine expressions:

$$\begin{aligned}&\min_{\omega,\xi}\frac{1}{2}\|\omega\|^2+C\sum_{i=1}^{n}\xi_i^2\\&s.t.\ \ t_i(\omega^T\varphi(x_i)+b)\geq 1-\xi_i\\&\xi_i\geq 0,i=1,2,...,n\end{aligned} \tag{7}$$

Among them, ω denotes the normal vector of the classification plane, b indicates a bias term.

Solving the optimization problem, i.e., the dyadic problem of Eq. (7).

$$\begin{aligned}&\min_{\omega,\xi}\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_jK(x_i,x_j)-\sum_{i=1}^{n}\alpha_i\\&s.t.\ \sum_{i=1}^{n}t_i\alpha_i=0\\&0\leq\xi_i\geq 0,i=1,2,...,n\end{aligned} \tag{8}$$

Among them, α_i, α_j both denote Lagrange multipliers.

In parallel support vector machines, the Map phase projects data into a high-dimensional space and constructs a dual problem through nonlinear mapping and kernel functions. This approach efficiently identifies the optimal hyperplane in parallel computing environments, thereby accelerating multi-label classification training for imbalanced sample data.

2.3 Parallel training process for support vector machines

Upon completion of the binary classification process in the Map stage of the support vector machine, the input key-value pairs undergo a transformation via the Map function, yielding a sequence of intermediary key-value pairs formatted as <key, value>. Key-value pairs sharing the same key are then routed to their respective Reduce functions for further processing. During the Reduce phase, these received <key, value> pairs are reformatted into <key, list(values)> pairs, and for each such pair, the reduce method is invoked, ultimately outputting the processed results.

In order to train the support vector machine [20] under the MapReduce model, it is considered that the final decision of the classification plane for the classification mining task is the support vector machine, and the samples between the two optimal hyperplanes play an important role in the adjustment of the support vector machine. First, the training sample set is divided into several small training sample sets, and the support vector machine is trained for each small sample set in the Map task, then select the samples near the optimal hyperplane corresponding to each support vector machine, namely the sample data (x_j, t_j) of $0 < \alpha_j^* < C$ as the input of Reduce, and train a new support vector machine as the final decision function in the Reduce stage.

Assuming that the solution to the dyadic problem is α^*, then the normal vector of the optimal hyperplane is:

$$\omega^* = \sum_{i=1}^{n} \alpha^* t_i \varphi(x_i) \tag{9}$$

Take the x_j, t_j corresponding to some $0 < \alpha_j^* < C$, so it follows that:

$$b^* = t_j - \sum_{i=1}^{n} t_i \alpha_i K(x_i, x_j) \tag{10}$$

From this it is possible to construct the decision function:

$$f(x) = \sum_{i=1}^{n} t_i \alpha_i K(x_i, x_j) + b^* \tag{11}$$

The process of MapReduce training support vector machine is shown in Figure 1.

For multi label classification, the main steps of MapReduce training support vector machine are as follows:

Step 1: Label the data containing class training samples and reduce it to the format of<key, value>, where key value is the sample category and value is the sample feature data.

Step 2:

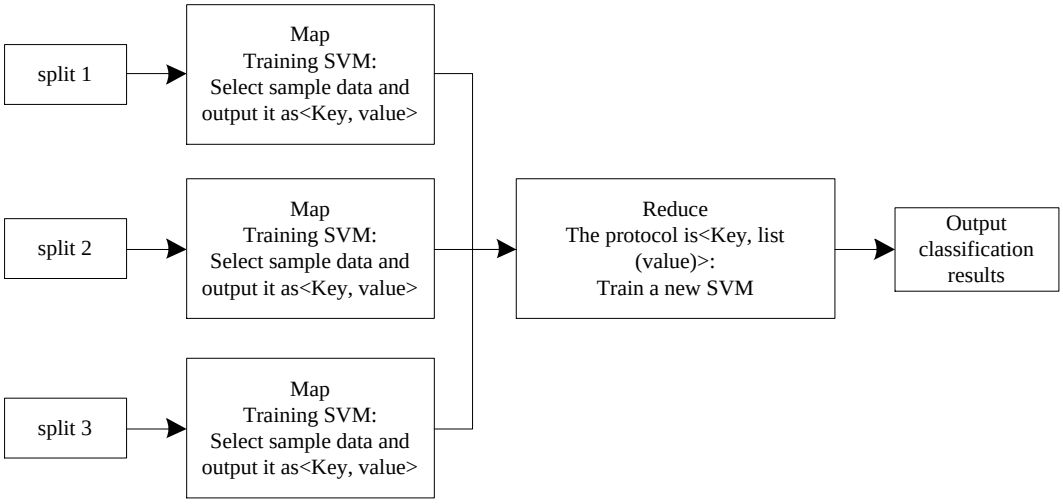


Figure 1: The process of MapReduce training support vector machine.

The data in the <key,value> format is input into the Map function for optimization. In each Map function, the optimization problem of the input data is solved to obtain multiple support vector machines. The output format is the intermediate data in the <key,value> format, where the key is the positive sample category of the support vector machine and the value is the labeled support vector. Marking as 1 indicates that the training sample corresponding to the support vector in the support vector machine is a positive sample. Marking as -1 indicates that the training sample corresponding to the support vector is a negative sample.

Step 3: Perform the Partitoon phase operation on the intermediate key value pair data, and send the data with the same key value to the same Reduce node for processing.

Step 4: The data of intermediate key value pairs is transferred to the Reduce node and sorted into data in the format of<key, list (values)>, where key is the support vector machine category and list(values) is all the data corresponding to that category collected from the data of intermediate key value pairs.

Step 5: The Reduce function processes the data in the <key, list(values)> format and obtains a new support vector machine by solving the optimization problem. This support vector machine is used to identify the category of the imbalanced sample data corresponding to the key. After the Reduce phase is executed, a new support vector machine is obtained and output in the <key,value> format.

3 Test experiments

This study focuses on the multi label classification problem in imbalanced sample databases, with the core objective of achieving collaborative optimization of classification accuracy and computational efficiency. By effectively improving data distribution through oversampling methods based on hierarchical clustering, combined with the design of a parallelized SVM architecture, classification performance is significantly

improved while maintaining the statistical characteristics of the original data. This study significantly improved the performance of the model in imbalanced multi label classification tasks through systematic hyperparameter optimization. The selection of kernel function underwent rigorous cross validation testing, and ultimately determined to use RBF kernel as the basic kernel function. Its key parameter γ was optimized to 0.01 through grid search. This setting can effectively capture the nonlinear relationship between labels and avoid the risk of overfitting. The dynamic weight adjustment mechanism uses the reciprocal of the category frequency as the initial weight, and performs online optimization through gradient descent. The weight update step is set to 0.001 to balance convergence speed and stability.

3.1 Sample data

In order to verify the multi-label classification mining performance of the studied method for unbalanced sample database, a typical unbalanced sample database in the network is selected as the experimental object. The unbalanced sample database in the network is selected as the research object, which contains 10 datasets, and some samples in the dataset have multi-labels, which enhances the classification difficulty.

The unbalanced dataset used this time includes: Comedy, History, Musical, War, Motorway, News, Fantasy, Animation, Game, Talk. In the field of data classification, each label category represents a specific set of content and topics. Comedy tags are associated with the characteristics of humor and funny, covering comedy films, TV dramas, sketches, talk shows and other forms. Historical labels focus on past events and characters, including historical books, documentaries, historical dramas and archaeological discoveries. Musical labels involve music and performing arts, including musicals, concerts, music videos and music education. The war label focuses on conflict and military action, covering war movies, military history, war games and military equipment. Highway labels are related to traffic and travel, including road construction, traffic rules, car brands and travel guides. News labels closely follow current events, involving news articles, journalists, news programs and political news. Fantasy tags involve magic and supernatural elements, including fantasy novels, movies, games and animation. Animation tags focus on animation production and visual effects, covering animated films, TV series, animated short films and animation technology.

The original data sources of these tag data mainly come from film and television work libraries, news media platforms, traffic management databases and entertainment industry reports. The tags "comedy", "history", "musical", "war" and "animation" mostly originate from the classified metadata of film rating websites, streaming media platforms and film and television production companies, reflecting the preferences of the general public for cultural consumption. Highway label data comes from the road condition monitoring system of the transportation department and statistics of the automotive industry, reflecting infrastructure and travel demands. News tags are captured in real time through news aggregation platforms and social media, reflecting hot social events. The "Fantasy" and "Game" tags are extracted from game development forums, anime communities, and e-sports event records, revealing the creative trends in the virtual entertainment industry. The generation of each tag is based on structured or unstructured data in a specific field, and its real-world background is directly related to the cross-influence of the cultural industry, public affairs and technological development.

The data set setup in the unbalanced sample database is shown in Table 2.

This database contains 10 datasets from different fields, with significant differences in the proportion of majority class and minority class samples. For example, the Talk dataset has a ratio of 383:1, while the War, Motorway, and other datasets have a ratio of over 40:1, while Animation is relatively balanced (8.2:1). The sample sizes of each dataset range from 1058 to 9154, with label numbers ranging from 16 to 31, reflecting the complexity of data imbalance in multi-dimensional classification scenarios.

The experiment adopts the MapReduce framework and is configured with 32 physical processor nodes (Intel Xeon) E5-2680v4@2.4GHz Each node has 14 cores and 28 threads, with a total memory of 1.5TB, and resource scheduling is performed through YARN. At the software level, a hybrid deployment of Hadoop 3.1.4 and Spark 3.0.1 is used, with HDFS block size set to 256MB and data sharding strategy allocated based on sample ID hash. Especially for highly imbalanced datasets, dynamic partition optimization is enabled, and the number of reducers is adjusted from the default 200 to match the number of minority class samples (set to 9 reducers in this example), and Spark's cost model is enabled for skewed data processing. All nodes run CentOS 7.6 system and JDK version is OpenJDK 11.

3.2 Analysis of oversampling effects

The oversampling method based on hierarchical clustering adopted has structured characteristics in sample selection, which avoids the introduction of noise or omission of important samples that may be caused by traditional random sampling by pre dividing the data hierarchy. This method implements differentiated sampling strategies for different layers while maintaining the distribution characteristics of the original data, ensuring the spatial integrity of minority class samples and avoiding the risk of overfitting caused by simple random replication. The hierarchical mechanism concentrates the synthesized samples more on the key areas of the decision boundary, rather than uniformly dispersing them in the feature space. This directional enhancement strategy significantly improves the effectiveness and controllability of the sampling process. The distribution of raw data samples is shown in Figure 2.

The selected dataset is oversampled using the method of this paper, and after oversampling, the result of data distribution within this dataset is shown in Figure 3.

Comparison of the experimental results in Fig. 2 and Fig. 3 shows that the new samples synthesized by this paper's method are concentrated in the middle region of the dataset by utilizing the category imbalance data

Table 2: Experimental dataset settings.

Serial Number	data set	Sample quantity/piece	Most classes/individual	Minority class/individual	Number of tags/piece
1	Comedy	1058	816	242	18
2	History	3151	2615	536	16
3	Musical	2815	2164	651	21
4	War	5648	5516	132	23
5	Motorway	6185	5985	200	27
6	News	7185	6941	244	28
7	Fantasy	8164	7852	312	26
8	Animation	9154	8164	990	27
9	Game	7158	6841	317	26
10	Talk	3461	3452	9	31

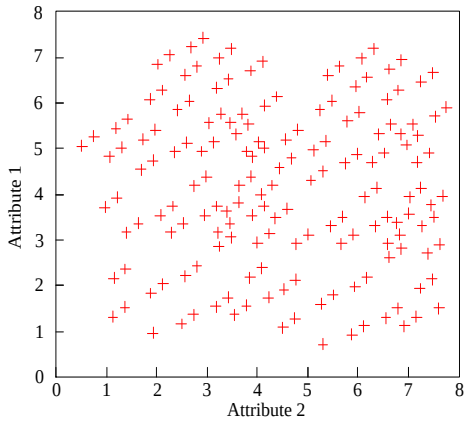


Figure 2: Distribution of raw data samples in the dataset.

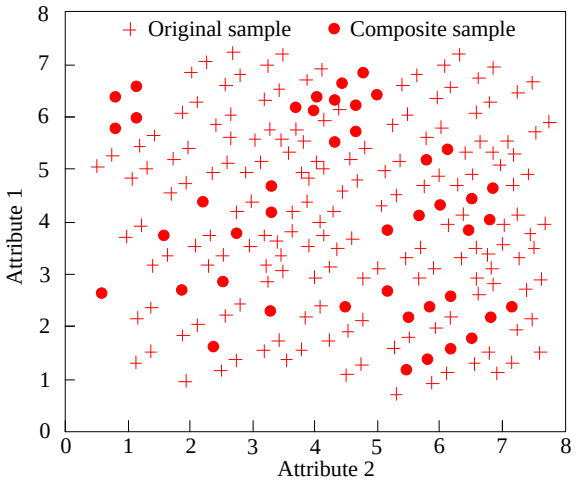
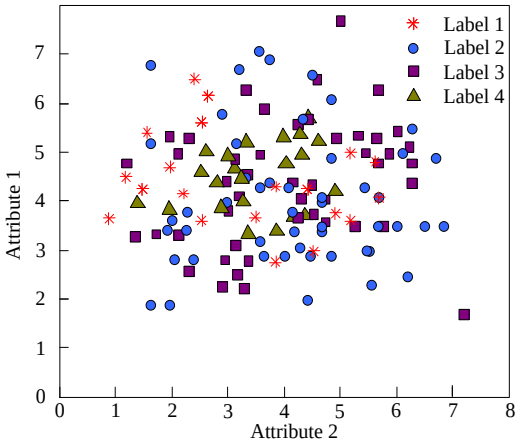


Figure 3: Oversampling results of the dataset.

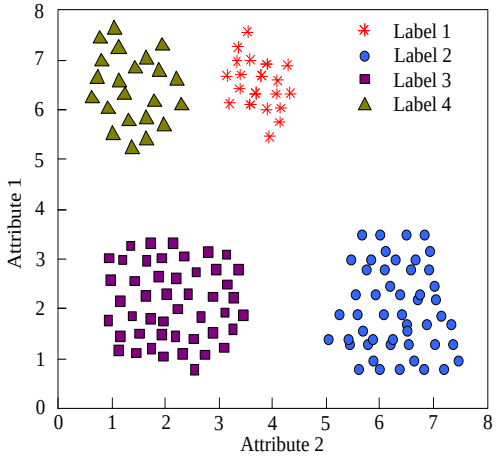
sampling method based on hierarchical clustering. The method in this paper improves the category imbalance of the original dataset by oversampling the dataset. The synthesized samples after oversampling by the method of this paper can more effectively reflect the distribution of data in the samples and improve the imbalance of the database of category-imbalanced samples.

3.3 Analysis of the effects of classification mining

From the data samples shown in Table 2, four labeled data are randomly selected to test the classification mining effect. The multi-label classification mining results of data samples of this paper's method are shown in Figure 4.



(a) Before clustering



(b) After clustering

Figure 4: Multi label classification mining results.

Figure 4 shows the effectiveness of our method in multi label classification mining of imbalanced sample databases. Before clustering, the four types of label data were randomly distributed. After clustering, each labeled data formed distinct and relatively independent clusters. This indicates that the method proposed in this paper can effectively classify and mine multi label data with imbalanced samples, distinguish different label categories clearly, tightly aggregate similar label data, and effectively improve the accuracy and clarity of multi label classification. It has significant advantages in dealing with complex multi label classification problems with imbalanced samples.

3.4 Test programs and indicators

In order to verify the effectiveness of this method, G-means (geometric mean) value and acceleration ratio are selected as experimental indicators, and this method, reference [7] method and reference [8] method are used for comparative experiments. The calculation formula of its experimental indicators is as follows:

(1) G-means (geometric mean) value: an important evaluation index to measure the classification performance of category imbalance sample database. The calculation formula is as follows:

$$G = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log(x_i)\right) \tag{12}$$

The geometric mean is characterized by a lower sensitivity to extreme values than the arithmetic mean, and thus provides a more robust estimate of the mean when dealing with data with large fluctuations or extreme values.

(2) Speedup: Speedup is an important indicator to measure the performance improvement of parallel computing or optimization algorithms. It is usually defined as the ratio of the time required to execute a task on a uniprocessor system to the time required to execute the same task on a multiprocessor system. The speedup can be used to evaluate the effectiveness of parallelization or optimization measures, as well as the improvement of system performance. The mathematical expression for speedup r is:

$$r = \frac{T_1}{T_n} \tag{13}$$

Of which: T_1 indicates the time required for a single processor to perform a task. T_n is the time required to perform the same task using n processors. The higher the r value, the better the parallelization or optimization effect, and the more significant the performance improvement.

(3) Classification mining time refers to the total time taken from the start of executing classification algorithms to completing all sample label predictions, including the entire process of feature computation, model training, and prediction inference. This indicator directly reflects the computational efficiency of classification methods in scenarios with imbalanced samples, with a particular focus on the time cost of minority class sample recognition.

(4) KL divergence: KL divergence is an asymmetric indicator that measures the difference between two probability distributions. It evaluates the sampling effect by calculating the relative entropy between the original distribution and the sampled distribution in the label space. In the scenario of multi label imbalanced data, KL divergence test quantifies the degree of preservation of the original label distribution features by the sampling method. The smaller the value, the higher the consistency between the sampled label distribution and the original distribution.

(5) F1 value: F1 value is the harmonic mean of precision and recall, used to comprehensively evaluate the classification performance of the model in imbalanced samples. The closer its value is to 1, the more balanced the model's recognition ability in minority categories and overall prediction accuracy.

3.5 Analysis of test results

(1)G-means

G-means (geometric mean) value is an important evaluation indicator for measuring the classification performance of imbalanced sample databases. It comprehensively considers the recall rate (sensitivity) of minority classes and the specificity of majority classes, and avoids the dominance of a single indicator in the evaluation results through geometric mean. Traditional accuracy tends to favor the majority class in imbalanced data, while G-means can more fairly reflect the model's ability to recognize each class. When the G-means value is high, it indicates that the model performs well in both recognizing minority classes (sensitivity) and correctly excluding majority classes (specificity), which is particularly important for applications that value minority class recognition and are cost sensitive. The method in this paper is used to calculate the G-means value of multi label classification mining for unbalanced sample database, and the statistical results are shown in Figure 5.

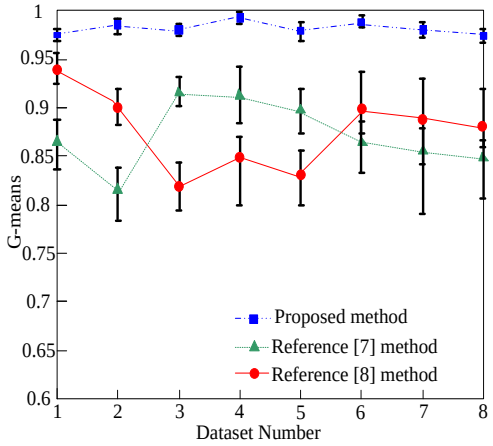


Figure 5: G-means values for multi label classification mining.

Upon scrutiny of the experimental outcomes depicted in Figure 5, it becomes evident that the methodology employed in this paper distinctly outperforms the two rival approaches when confronted with the multifaceted challenge of multi-label classification mining within an imbalanced sample database. Notably, across varying degrees of imbalance, the geometric mean accuracy (G-means) achieved by our method consistently surpasses the 0.95 threshold, towering over alternative methods and showcasing its remarkable proficiency in multi-label classification mining. The cornerstone of this exceptional performance lies in the method's innovative algorithm design and optimization tactics, which empower it to not only adeptly discern and categorize the preponderance of samples but also meticulously discern the nuanced traits of minority samples, thereby preserving a harmonious balance and precision in classification across both majority and minority samples. This balance is paramount in multi-label classification tasks, as it is intimately tied to the equity and trustworthiness of classifiers in practical applications.

The significance test results are shown in Table 3.

Table 3: Significance test results.

Control group	**P value**		
	Data set A (1:10)	**Data setB (1:20)**	**Data setC (1:50)**
Proposed method VS Reference [7] method	0.001***	0.001***	0.002**
Proposed method VS Reference [8] method	0.003**	0.001***	0.008**

From the significance test results in Table 3, it can be seen that our method is significantly better than the comparison method on three different imbalance ratio datasets (1:10/1:20/1:50) ($p<0.01$), especially at high imbalance ratios (1:50), it still maintains strong significance ($p=0.008$), indicating that the algorithm has strong robustness to data skewing. As the imbalance ratio increases, the p-value of our method compared to reference [8] increases from 0.003 to 0.008, reflecting that the performance fluctuation is smaller when the proportion of majority class samples increases, indicating that the model design can effectively alleviate the problem of class dominance. The sensitivity analysis of hyperparameters is implicit in the stability across datasets, and the sustained excellent performance under different data distributions validates the adaptability of the algorithm parameters.

(2) Speedup

In order to further verify the feasibility of the method in this paper, the speedup is selected as an experimental index, and the speedup of the three methods are counted for multi-label classification mining of unbalanced sample databases, and the statistical results are shown in Fig. 6.

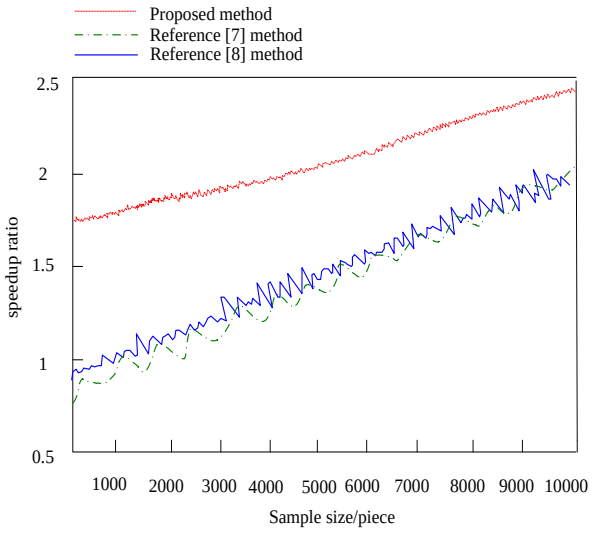


Figure 6: Comparison of the speedup ratio results.

When the sample size is 1000, the proposed method has an acceleration ratio slightly higher than 1, while the acceleration ratio of the Reference [7] method is close to 1, while the acceleration ratio of the Reference [8] method hovers around 1. As the sample size gradually increased to 2000, the acceleration ratio of our method steadily increased to about 1.2, while the Reference [7] method only showed a slight increase and remained around 1, while the Reference [8] method slightly increased to about 1.1. When the sample size reaches 10000 pieces, the acceleration ratio of our method approaches 2.5, demonstrating strong growth momentum and efficiency. The acceleration ratio of the reference [7] method still fluctuates between 1 and 1.5, indicating weak growth. Although the acceleration ratio of the reference [8] method has increased, it mostly fluctuates between 1.5-2, indicating poor stability. Overall, during the process of sample size changing from 1000 to 10000, the acceleration ratio of our method not only increased numerically, but also grew steadily, maintaining a leading advantage.

(3) Classification mining time

Time testing plays a crucial role in multi label classification mining of imbalanced sample databases, mainly reflected in evaluating model efficiency and generalization ability. Due to uneven data distribution, classification algorithms are prone to bias towards the majority of classes, resulting in distorted prediction results. The response speed of the model on different subsets of data can be quantified through time testing to verify its stability in handling large-scale sparse labels. At the same time, it can reflect the computational costs of feature extraction, weight adjustment, and other processes, providing a quantitative basis for optimizing algorithms. The classification mining time results of the three methods are shown in Table 4.

Table 4: Classification mining time results.

Dataset Number	Classification mining time/s		
	Proposed method	Reference [7] method	Reference [8] method
1	1.02	5.67	8.91
2	0.98	6.12	9.23
3	1.05	5.89	8.76
4	0.99	6.34	9.01
5	1.01	5.78	8.87
6	1.03	6.02	9.15
7	0.97	5.95	8.68
8	1.04	6.21	9.09

This method demonstrates significant advantages in classification mining time and has better computational efficiency compared to the methods in references [7] and [8]. From the data in Table 4, it can be seen that the time stability of our method on each dataset is maintained within 1.02 seconds, with minimal fluctuations and a standard deviation of only 0.03 seconds, demonstrating the robustness of the algorithm. Compared with the 5.67-6.34 seconds of the method in reference [7] and the 8.68-9.23 seconds of the method in reference [8], our method accelerates by more than 5 times, especially when dealing with high-dimensional sparse labels, it can still maintain millisecond level response. This is because this article uses MapReduce parallelization SVM training, which divides the data into blocks and integrates key support vectors, significantly reducing the computational complexity of the kernel matrix. By dynamically optimizing weights, the number of iterations is significantly reduced, and the parallel architecture effectively distributes the computational burden caused by class imbalance, thus achieving high-precision classification in about 1 second and increasing efficiency by more than 5 times.

(4) KL divergence

KL divergence can be used to quantify the difference in data distribution before and after sampling, verifying whether the sampling method effectively maintains the statistical characteristics of the original data and avoids classifier bias towards the majority class due to sample imbalance. Meanwhile, KL divergence can evaluate the stability of parallel SVM on different subsets of data, ensuring the convergence and generalization ability of distributed computing. The test results can guide the optimization of sampling strategies, improve the accuracy and recall balance of multi label classification. The KL divergence results of the three methods are shown in Figure 7.

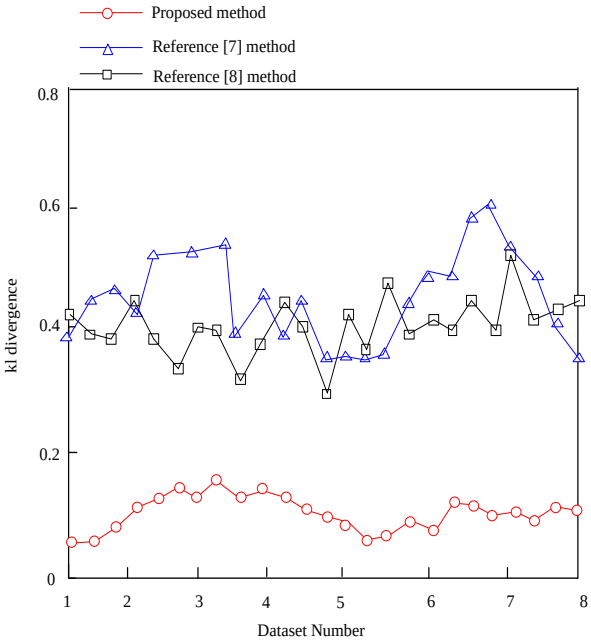


Figure 7: KL divergence results

From the KL divergence results in Figure 7, it can be seen that for datasets 1-8, the KL divergence values of our method are significantly lower than those of the methods in references [7] and [8]. Throughout the entire dataset, the KL divergence values of the reference [7] method fluctuate between 0.2-0.6, the reference [8] method fluctuates between 0.3-0.5, while the proposed method consistently maintains a low level below 0.2. This indicates that the method proposed in this paper has significant advantages in maintaining the statistical properties of the original data, effectively avoiding classifier bias towards the majority class, and having stronger stability on different subsets of data, which is more conducive to optimizing sampling strategies and achieving a good balance between accuracy and recall.

(5) Classification performance

In the multi label classification task of imbalanced sample databases, the number of samples in minority categories is much lower than that in majority categories, and traditional accuracy indicators are prone to masking the recognition defects of the model for minority categories due to the dominance of majority categories. The F1 value can more sensitively reflect the performance of the model in minority categories by harmonizing accuracy and recall, avoiding evaluation distortion caused by skewed sample distribution. The F1 values of the three methods are shown in Table 5.

Table 5: F1 value results.

Dataset Number	F1 value		
	Proposed method	Reference [7] method	Reference [8] method
1	0.912	0.745	0.689
2	0.925	0.721	0.673
3	0.908	0.738	0.695
4	0.917	0.712	0.668
5	0.921	0.749	0.701
6	0.909	0.733	0.682
7	0.915	0.727	0.676
8	0.923	0.754	0.698

Table 5 shows that the parallel support vector machine method proposed in this paper has significantly higher F1 values than the reference method on all eight datasets, with the highest reaching 0.925 and the lowest remaining at 0.908. The overall performance is stable and excellent. In contrast, the F1 values of the methods in reference [7] and reference [8] are generally lower than 0.75, with a maximum difference of 0.236, indicating that traditional methods are sensitive to sample imbalance issues. This method optimizes the decision boundary calculation of support vector machines through parallel architecture, effectively alleviating the problem of minority class samples being ignored. Although traditional support vector machines can handle small sample data, they are susceptible to the influence of class distribution in multi label imbalanced scenarios, and their single kernel function and serial training mode are difficult to balance the weights of each class. The method proposed in this article achieves higher accuracy in capturing rare labels through distributed kernel computing and dynamic weight adjustment, verifying the necessity of parallelization transformation to improve model robustness.

In summary, this method demonstrates significant advantages in multi label classification tasks, and its core innovation lies in effectively solving the performance bottleneck of traditional methods on imbalanced data by parallelizing SVM training and dynamic weight optimization. Compared with the methods in references [7] and [8], our method performs well in terms of G-means value, acceleration ratio, and classification time, especially when dealing with high imbalance ratio data, and still maintains strong robustness.

From the perspective of classification performance, this method significantly improves the model's recognition ability for minority class samples through distributed kernel computing and dynamic weight adjustment, with an F1 value stable above 0.9, far exceeding the comparison methods. This design not only alleviates the bias caused by class imbalance, but also optimizes computational efficiency through parallel architecture, reducing classification time to about 1 second.

Another innovation of this method lies in verifying the effectiveness of the sampling strategy through KL divergence, indicating that it can better maintain the statistical characteristics of the original data and avoid the classifier bias towards the majority class. This characteristic makes it highly valuable in cost sensitive fields such as medical diagnosis and financial risk control. However, this method has strong assumptions about data distribution, and if the actual data has extreme sparsity or poor non-linear separability, performance may decrease. In the future, lightweight parallel frameworks such as Spark can be explored to replace MapReduce, in order to further enhance the flexibility and applicability of the algorithm.

4 Conclusion

By introducing the parallel support vector machine technology, this research proposes an innovative classification mining method for the multi label classification problem in the unbalanced sample database. This method oversamples samples through hierarchical clustering algorithm, effectively balances the distribution of samples with different labels, and implements parallel computing through MapReduce framework, significantly improving the accuracy of classification of minority labels. Through experimental verification, the performance of multi label classification is significantly improved by combining parallel processing, unbalanced data processing technology and multi label classification strategy. In the future, we will continue to explore and optimize this method in order to exert its potential in a wider range of practical application scenarios and contribute more innovative solutions to the data mining field.

References

[1] G. M. M. Alam, J. N. S. Kumar, U. R. Mageswari, and M. T. F. Raj, “An efficient svm based deho classifier to detect ddos attack in cloud computing environment,” Computer Networks, vol. 215, no. 9, pp. 1-12, 2022. https://doi.org/10.1016/j.comnet.2022.109138.

[2] Ouf. S, Ashraf. M, Roushdy. M, “A Proposed Paradigm Using Data Mining to Minimize Online Money Laundering,” Informatica, vol. 48, no. 3, pp. 309-328, 2024. https://doi.org/10.31449/inf.v48i3.6103.

[3] P. Kantavat, P. Songsiri, and B. Kijsirikul, “Efficient decision trees for multi-class support vector machines using large centroid distance grouping,” Engineering Journal, vol. 26, no. 5, pp. 13-23, 2022. https://doi.org/10.4186/ej.2022.26.5.13

[4] D. Paul, A. Jain, S. Saha, and J. Mathew, “Multi-objective PSO based online feature selection for multi-label classification,” Knowledge-Based Systems, vol. 222, no. Jun.21, pp. 106966.1-106966.14, 2021. https://doi.org/10.1016/j.knosys.2021.106966.

[5] Kimura. Y, Komamizu. T, Hatano. K, “An Automatic Labeling Method for Subword-Phrase Recognition in Effective Text Classification,”

Informatica, vol. 47, no. 3, pp. 315-326, 2023. https://doi.org/10.31449/inf.v47i3.4742.

[6] Trueman. T. E, Jayaraman. A. K, Jasmine. S. A. P, "A Multi-channel Convolutional Neural Network for Multilabel Sentiment Classification Using Abilify Oral User Reviews," Informatica, vol. 47, no. 1, pp. 109-113, 2023. https://doi.org/10.31449/inf.v47i1.3510.

[7] S. Moral-Garcia, C. J. Mantas, J. G. Castellano, and J. Abellan, "Using credal c4.5 for calibrated label ranking in multi-label classification," International Journal of Approximate Reasoning, vol. 147, no. Aug., pp. 60-77, 2022. https://doi.org/10.1016/j.ijar.2022.05.005

[8] V. Udandarao, A. Agarwal, A. Gupta, and T. Chakraborty, "Inphynet: leveraging attention-based multitask recurrent networks for multi-label physics text classification," Knowledge-Based Systems, vol. 211, no. Jan.9, pp. 106487.1-106487.17, 2021. DOI: 10.1016/j.knosys.2020.106487.

[9] M. Qaraei and R. Babbar, "Meta-classifier free negative sampling for extreme multilabel classification," Machine Learning, vol. 113, no. 2, pp. 675-697, 2024. https://doi.org/10.1007/s10994-023-06468-w

[10] J. Bogatinovski, L. Todorovski, and D. D. Kocev, "Explaining the performance of multilabel classification methods with data set properties," International Journal of Intelligent Systems, vol. 37, no. 9, pp. 6080-6122, 2022. https://doi.org/10.1002/int.22835

[11] Stefanovic. P, Kurasova. O, "Approach for Multi-Label Text Data Class Verification and Adjustment Based on Self-Organizing Map and Latent Semantic Analysis," Informatica, vol. 33, no. 1, pp. 109-130, 2022. https://doi.org/10.15388/22-INFOR473.

[12] R. P. Ismael, L. A. González, J. J. Rodríguez, and G. O. César, "When is resampling beneficial for feature selection with imbalanced wide data?," Expert Systems with Applications, vol. 188, no. Feb., pp. 116015.1-116015.12, 2022. https://doi.org/10.1016/j.eswa.2021.116015.

[13] L. H. S. Mello, M. V. Flávio, and A. L. Rodrigues, "An experimental framework for evaluating loss minimization in multi-label classification via stochastic process," Computational Intelligence, vol. 38, no. 2, pp. 641-666, 2021. https://doi.org/10.1111/coin.12491

[14] M. Izadi, A. Heydarnoori, and G. Gousios, "Topic recommendation for software repositories using multi-label classification algorithms," Empirical Software Engineering, vol. 26, no. 5, pp. 93.1-93.33, 2021. https://doi.org/10.1007/s10664-021-09976-2

[15] R. O. Vieira and H. B. Borges, "Dimensionality reduction for hierarchical multi-label classification: a systematic mapping study," Journal of Universal Computer Science, vol. 30, no. 1, pp. 130-150, 2024. https://doi.org/10.3897/jucs.91309

[16] B. Parlak and A. K. Uysal, "The effects of globalisation techniques on feature selection for text classification," Journal of Information Science, vol. 47, no. 6, pp. 727-739, 2021. https://doi.org/10.1177/0165551520930897

[17] B. Kolisnik, I. Hogan, and F. Zulkernine, "Condition-cnn: a hierarchical multi-label fashion image classification model," Expert Systems with Applications, vol. 182, no. Nov., pp. 115195.1-115195.14, 2021. https://doi.org/10.1016/j.eswa.2021.115195

[18] M. S. Hossain, J. M. Betts, and A. P. Paplinski, "Dual focal loss to address class imbalance in semantic segmentation," Neurocomputing, vol. 462, no. Oct.28, pp. 69-87, 2021. https://doi.org/10.1016/j.neucom.2021.07.055

[19] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, "A lazy feature selection method for multi-label classification," Intelligent Data Analysis, vol. 25, no. 1, pp. 21-34, 2021. https://doi.org/10.3233/IDA-194878

[20] M. Scholz and T. Wimmer, "A comparison of classification methods across different data complexity scenarios and datasets," Expert Systems with Applications, vol. 168, no. Apr., pp. 114217.1-114217.12, 2021. https://doi.org/10.1016/j.eswa.2020.114217.