



**ARTICLE**

## A Novel Framework for Learning and Classifying the Imbalanced Multi-Label Data

P. K. A. Chitra<sup>1</sup>, S. Appavu alias Balamurugan<sup>2</sup>, S. Geetha<sup>3</sup>, Seifedine Kadry<sup>4,5,6</sup>, Jungeun Kim<sup>7,\*</sup> and Keejun Han<sup>8</sup>

<sup>1</sup>Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, 603203, India

<sup>2</sup>Department of Computer Science and Engineering, Periyar Maniammai Institute of Science & Technology (Deemed to be University), Thanjavur, Tamil Nadu, 613403, India

<sup>3</sup>School of Computer Science and Engineering, Chennai, Tamil Nadu, 600048, India

<sup>4</sup>Department of Applied Data Science, Noroff University College, Kristiansand, 4612, Norway

<sup>5</sup>Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, P.O. Box 346, Ajman, United Arab Emirates

<sup>6</sup>Department of Electrical and Computer Engineering, Lebanese American University, Byblos, 10150, Lebanon

<sup>7</sup>Department of Software, Kongju National University, Cheonan, 31080, Republic of Korea

<sup>8</sup>Division of Computer Engineering, Hansung University, Seoul, 02876, Republic of Korea

\*Corresponding Author: Jungeun Kim. Email: jekim@kongju.ac.kr

Received: 15 July 2022 Accepted: 14 December 2022 Published: 13 September 2024

### ABSTRACT

A generalization of supervised single-label learning based on the assumption that each sample in a dataset may belong to more than one class simultaneously is called multi-label learning. The main objective of this work is to create a novel framework for learning and classifying imbalanced multi-label data. This work proposes a framework of two phases. The imbalanced distribution of the multi-label dataset is addressed through the proposed Borderline MLSMOTE resampling method in phase 1. Later, an adaptive weighted  $l_{21}$  norm regularized (Elastic-net) multi-label logistic regression is used to predict unseen samples in phase 2. The proposed Borderline MLSMOTE resampling method focuses on samples with concurrent high labels in contrast to conventional MLSMOTE. The minority labels in these samples are called difficult minority labels and are more prone to penalize classification performance. The concurrent measure is considered borderline, and labels associated with samples are regarded as borderline labels in the decision boundary. In phase II, a novel adaptive  $l_{21}$  norm regularized weighted multi-label logistic regression is used to handle balanced data with different weighted synthetic samples. Experimentation on various benchmark datasets shows the outperformance of the proposed method and its powerful predictive performances over existing conventional state-of-the-art multi-label methods.

### KEYWORDS

Multi-label imbalanced data; multi-label learning; Borderline MLSMOTE; concurrent multi-label; adaptive weighted multi-label elastic net; difficult minority label



## 1 Introduction

An essential variation of typical supervised learning is known as multi-label learning. Unlike traditional supervised learning, the labels in multi-label learning are not mutually exclusive. However, it might be associated. In the case of multi-label learning, every example relates to multiple class labels concurrently [1]. Each data sample is represented predictor vector associated with multiple labels simultaneously. Let  $D = \{(x_i, y_j) | 1 \leq i \leq n, 1 \leq j \leq k\}$  be the training data of—labelmulti-label, which consists of  $n$  features and  $k$  labels in the label space. Single instance  $(x_i, y_i)$  in a multi-label data represents an  $n$ -dimensional predictive vector  $(x_{i1}, x_{i2}, \dots, x_{in})$  of real values and  $y_i$  is the associated  $k$ -dimensional label space  $(y_{i1}, y_{i2}, \dots, y_{ik})$  of binary values,  $y_{ij} = 1$  indicates label  $j$  is present in the label set of  $x_i$ , else  $y_{ij} = 0$ . Given training data set  $D = \{(x_i, y_j) | 1 \leq i \leq n, 1 \leq j \leq k\}$ , the multi-label learning role is to figure out a function  $f(\cdot): X \rightarrow 2^Y$  which predicts a correct label set  $y \subseteq Y$  for a provided undetected multi-label example instance. In many real-world applications, multi-label learning is commonly used, which includes automatic image annotation [2], text categorization [3], bioinformatics [4], music categorization [5], and drug side effect prediction [6]. For the past decade, the research in multi-label learning developed various multi-label learning approaches. Still, many challenges exist in multi-learning. The number of labels for prediction, which is exponential in size, is considered a fundamental challenge to multi-label data. This involves exploiting label correlation among labels, over-fitting due to high dimensional predictive space, and highly imbalanced training sets.

Most modern existing multi-label methods try to address the first problem generally, and a couple of efforts are made to resolve the second and third issues. The performance of many standard learning algorithms is degraded due to the class imbalance problem [7]. This is the case where the classes are not present equally. The number of positive examples for each category is less than its negative counterparts. This may lead to performance degradation of the learning method. As the learning algorithms are biased to prefer the majority class, the imbalanced data can negatively affect the learning algorithms [8]. With the neglect of imbalanced class distribution, conventional state-of-the-art learning algorithms perform poorly and produce unsatisfactory suboptimal results [9,10]. The irrelevant classes can be well identified with the poor-performing learning algorithms, while the minority is reversed. The class imbalance ratio of the existing data sets used for this research has been mentioned in [11]. The use of ensemble techniques to improve accuracy in single-label classifiers has been discussed in [12]. The easiest way to handle imbalanced data is through sampling. The sampling methods are of two types: (1) under-sampling; (2) over-sampling. Random sampling leads to overfitting if the sampling ratios are not appropriately set [13]. The synthetic minority over-sampling procedure (*SMOTE*) makes synthetic minority cases through interpolation amongst proper training samples and  $k$ -nearest neighborhoods. Many enhancements were done over *SMOTE* [14–16] to improve the performance of the learning process for imbalanced binary and multi-class data.

*SMOTE* was adopted by [17] to handle imbalance conditions in multi-label data. First, the instances with minority labels are selected as the seed, and their nearest neighbors are identified. Next, the features of the artificial instances are created based on randomly chosen neighbors. Then the synthetic instances are created with the feature values and the label information obtained from samples with minority labels and neighbors of minority labels. This process works with the  $k$ -Nearest Neighbor (*kNN*) method to find the nearest neighbors for a minority label. Finally, distances are computed based on Euclidean measure, and as a result, static  $k$  numbers of neighbors are returned for a minority label. Regression is a simple but powerful statistical method to discover the linear and non-linear relationships between predictors and the response variable. Logistic Regression (*LR*) is a robust and computationally fast discriminative method designed to model and capture each class's posterior

probabilities. *LR* is a conventional statistical technique addressing binary problems. However, *LR* overfits the training samples when the number of predictors vastly outstrips the number of samples. The elastic net can carry out automatic variable selection and continual contraction all at once and chooses the group of associated variables. This elastic net serves especially when the number of predictors is much larger than the variety of observations in the sample data.

Machine learning and AI methods have started to show their dominance in many fields, reference [18] showed the usage of the above two methods in the field of handwritten alphabet recognition, which plays a crucial role in pattern recognition, computer vision, and image processing. Deep Learning has been a boon in automated effective image processing. Deep learning-based automated weed in crops was presented by [19]. They used ten various rabi crops for their experimentation and proved the use of deep learning in the field. AI and deep learning-based methods could be used to handle the sheer amount of data that is being generated nowadays, and it was addressed by [20]. They managed high-dimensional data and explored their research in various application areas to justify the dominance of machine learning and deep learning. Deep learning and machine learning could also be used to address real-time social needs and have been presented by [21] to automatically detect garbage areas in remote locations.

This work addresses the imbalanced characteristic of logistic regression and the extension of logistic regression to penalized multi-label logistic regression in this paper. The adaptive weighted elastic net is used in the second phase to handle the synthetic samples produced in the first phase. This work offers an approach to make the logistic regression model work on imbalanced data. This framework introduces a new pre-processing method called *Borderline MLSMOTE* in phase 1. The newly created balanced multi-label dataset will combine the training dataset and generated synthetic samples. One noteworthy feature of the offered pre-processing approach is that it assists in expanding the minority labels in areas where the concurrent appearance of the minority and majority labels is too high. In the second phase, the  $l_{21}$ -norm regularized weighted logistic regression (adaptive elastic-net) is used to handle the over-fitting and variable selection simultaneously to make the learning and prediction over the balanced data. The objectives of this research are:

- A framework of two phases to handle and predict imbalanced multi-label data has been presented.
- *Borderline MLSMOTE* has been introduced to handle difficult concurrent minority labels.
- Adaptive weighted  $l_{21}$  norm regularization is presented and introduced to handle the problem of overfitting and variable selection in high-dimensional multi-label data.
- To conduct in-depth experiments on eighteen benchmark multi-label datasets to demonstrate that the proposed framework will handle imbalanced data more effectively than existing multi-label learning techniques.

### 1.1 Organization of the Paper

[Section 1](#) describes the introduction of multi-label learning, imbalanced data, logistic regression, and the paper's contributions. [Section 2](#) presents related works in imbalanced learning, multi-label learning, and logistic regression. [Section 3](#) discusses the background of multi-label learning, measures for imbalance in multi-label data, a review of logistic regression, and the proposed system. [Section 4](#) provides the speculative setup needed for the construction of the paper. [Section 5](#) defines outcomes as well as discussion. Finally, [Section 6](#) concludes with the end and future improvement of the work addressed in this paper.

## 2 Methods

Before the suggested frame was introduced, a few simple understandings regarding the methods were outlined briefly. Inside the frame, the *Border MLSMOTE* sampling technique is used at the pre-processing data level to resolve the imbalanced dataset by creating a synthetic dataset. Then adaptive  $l_{21}$ -norm regularised multi-label logistic regression is used to predict the balanced multi-label data. Finally, the results show the significance of the proposed framework.

### 2.1 Multi-Label Learning

Let  $D = \{(x_i, y_j) | 1 \leq i \leq n, 1 \leq j \leq k\}$  be the multi-label training data with  $p$  observations. The  $i^{\text{th}}$  multi-label sample instance  $(x_i, y_i)$ ,  $x_i$  is an  $n$ -dimensional predictive vector  $(x_{i1}, x_{i2}, \dots, x_{in})$  of real values, and  $y_i$  is the associated  $k$ -dimensional label space  $(y_{i1}, y_{i2}, \dots, y_{ik})$  of binary values. For an unseen sample  $x$ , the classifier  $f(\cdot)$  predicts  $(y_{i1}, y_{i2}, \dots, y_{ik})$  as a label vector for the unseen sample  $x$ . Classification whereby each example could be connected with an assortment of class labels. Each label comprises just a binary value. The multi-label classification maps an example instance to multiple labels. It is a generalization of supervised single label learning which map an example instance  $q_i \in \{Q\}$  to a label set  $\ell \in \{\mathcal{L}\}$  and is given in Eq. (1).

$$\hat{h} : \chi \rightarrow 2^{\mathcal{L}}, \quad \text{i.e.,} \quad MLC(\hat{h}) : q_i \mapsto \prod_{\ell \in \mathcal{L}} \ell \quad \text{where } |\ell| \geq 2. \quad (1)$$

Problem transformation alters multi-labeled data into single labels; conventional single-label classification approaches are used on transformed data. The transformed single-labeled data are binary classifiers; conventional single-label classifiers are enough to produce the model. Adaptation in the algorithm in multi-label learning frees single-labeled classifiers to alter them to choose multi-labeled data. Thus, algorithm variation strategies are effective and free from information loss.

### 2.2 Measures of Imbalance in Multi-Label Data

Multi-label classification is more complex than single-label classification when addressing class imbalance. IRperLabel and MeanIR were presented by [22] to measure imbalance in multi-label data. The IRperLabel (Imbalance Rate per Label) measures each label in the dataset. It provides individual imbalance levels in the dataset. Eq. (2) shows the definition of IRperLabel:

$$IRperLabel = \frac{\underset{\ell \in \mathcal{L}}{\operatorname{argmax}} \left( \sum_{i=1}^{|D|} h(\ell, Y_i) \right)}{\sum_{i=1}^{|D|} h(\ell, Y_i)}, \quad \text{where } h(\ell, Y_i) = \begin{cases} 1 & \ell \in Y_i \\ 0 & \ell \notin Y_i \end{cases} \quad (2)$$

MeanIR represents the average imbalance in multi-label data and is shown in Eq. (3). It shows the average of *IRperLabel* for all labels.

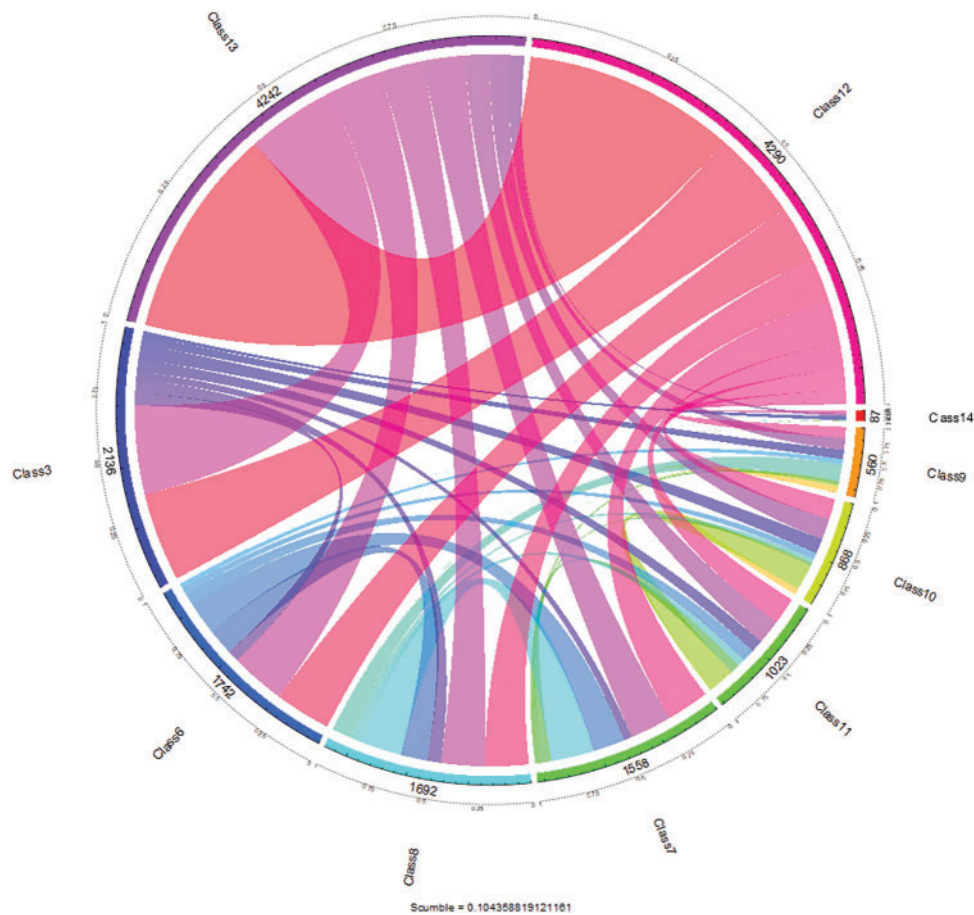
$$MeanIR = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} IRperLabel(\ell_i) \quad (3)$$

Apart from imbalance, concurrence among the imbalanced labels also needs to be considered before balancing. Concurrence among labels represents joint appearances of relevant and irrelevant labels of the same instance. This is measured using SCUMBLE (Score of Concurrence among iMBalanced LabEls). It concerns the amount of imbalance variance among relevant and irrelevant labels of each instance. The concurrence measure of each instance (SCUMBLE) in the dataset is calculated, and the average of all the instances  $SCUMBLE_i$  measure is given in Eqs. (4) and (5).

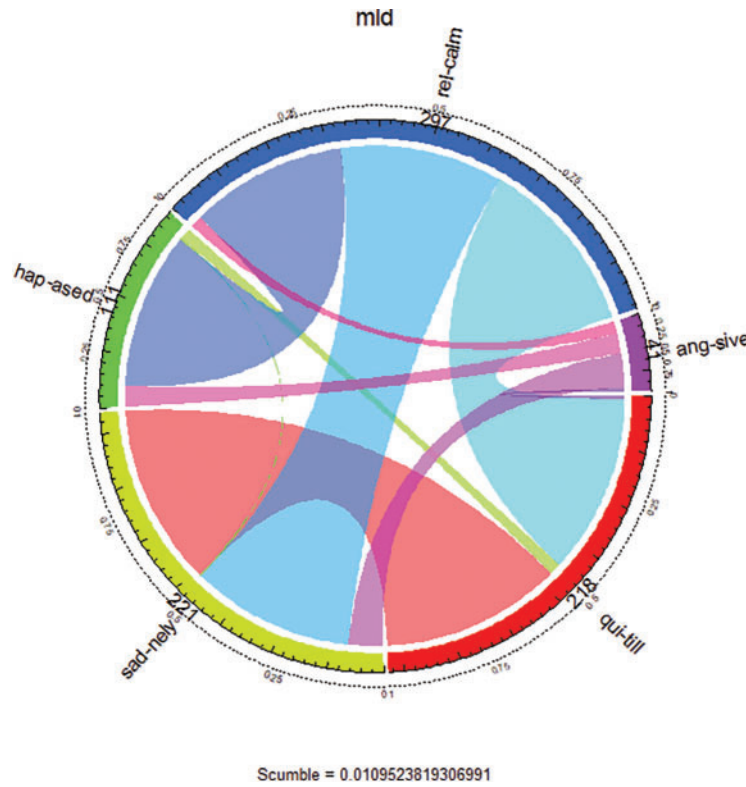
$$SCUMBLE_i = 1 - \frac{1}{\frac{1}{|D|} \sum_{i=1}^{|D|} IRperLabel_i} \left( \prod_{i=1}^{|D|} IRperLabel_i \right) \quad (4)$$

$$SCUMBLE(D) = \frac{1}{|D|} \sum_{i=1}^{|D|} SCUMBLE_i \tag{5}$$

In multi-label data, a group of positive or negative labels might exist, which necessitates the algorithms to be designed in such a way as to handle a group of labels instead of a single one. Label concurrence of yeast data is depicted in Fig. 1. Arc represents labels in the dataset. The length segment of the arc describes the number of instances associated with each label. Minority labels in yeast data are class 14, class 9, class 10, and class 11; these labels appear together with one or more irrelevant labels, and these four relevant labels are difficult. Fig. 2 shows labels concurrence of emotional data. This dataset contains six labels, and the picture indicates the number of samples related to each label and interactions among labels. For example, angry-aggressive is a minority label in the emotions dataset. It appears together with other majority labels like relaxing-calm and sad-lonely.



**Figure 1:** Label concurrence view of yeast data



**Figure 2:** Label concurrence view of emotional data

### 2.3 Logistic Regression and Elastic Net

The logistic regression (LR) learns the relationship of predictive variables to binary 1 (“success”/“Presence”) or 0 (“failure”/“Absence”) valued response variables. When the response data is binary, logistic regression is used to find the relationship between the predictor and responses. The LR extracts some weighted predictors from predictor space and then combines them linearly. The conditional probability of LR is the joint prediction of labels in the label space given as in Eqs. (6) and (7).

$$pP(L|X, \beta) = L(\Theta) = \prod_{i=1}^k P(\ell_i|X) \quad (6)$$

$$= \prod_{i=1}^k \frac{e^{\ell_i \beta_i^T X}}{e^{\beta_i^T X} + e^{-\beta_i^T X}} \quad (7)$$

where  $\beta = (\beta_0, \beta_1, \dots, \beta_n)$  are the coefficient for the  $i^{\text{th}}$  logistic regression to be determined and  $\beta_0$  is the intercept;  $X = (x_{i1}, x_{i2}, \dots, x_{ik})$ ,  $i = 1, 2, 3, \dots, k$  is the input predictors of the training instances;  $Y = (y_{i1}, y_{i2}, \dots, y_{im})$  is the regression object vectors, i.e., labels/targets. The predictor variables are continuous, while the response variable is binary (yes/no or present/absent). The regression coefficients are estimated using maximum likelihood estimation. The negative log-likelihood of  $k$  observations is as in Eq. (8) under the independence assumption.

$$P(L|X, \beta) = L(\Theta) = -\frac{1}{k} \sum_{i=1}^k (\ell_i f(X', \beta) + (1 - \ell_i) \log(1 - f(X', \beta))) \quad (8)$$

To handle the ill-condition and the over-fitting problems, Eq. (8) is added with a penalty term. The penalized function of the logistic regression is given in Eq. (9). The penalty term is defined in Eq. (10).

$$P(L|X, \beta) = \underset{\Theta}{\operatorname{argmin}} L(\Theta) + R(\Theta) \quad (9)$$

$$L(\Theta) = -\frac{1}{k} \sum_{i=1}^k [\log p(\ell_i|x_i; \beta) + (1 - \log p(\ell_i|x_i; \beta))] \quad (10)$$

$$R(\Theta) = \left(1 + \frac{\lambda_2}{m}\right) \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^m \|x_i \beta_i - y_i\|_2^2 + \lambda_2 \sum_{i < m} \|\beta_i\|_2^2 + \lambda_1 \sum_{i < m} \|\beta_i\|_1 \quad (11)$$

$R(\Theta)$  in Eq. (11) is the elastic net regularization that learns a linear model to predict the response vector from the predictor by minimizing the squared loss with  $\ell_2$  regularization and  $\ell_1$  norm constraint.  $\lambda_1, \lambda_2$  are the hyperparameters that decide the trade-off between the term regularization and error. These hyperparameters are the tuning parameters that provide the goodness of fit and the complexity of the model. An Elastic net is a mixture of  $l_1$  (lasso) and  $l_2$  (ridge) penalties. The lasso part of the elastic net performs variable selection, while the ridge part in the penalization stabilizes the solution paths. The amount of predictor variables exceeds the range of all samples/observations in the case of high-dimensional data analysis. When  $\lambda_2 = 0$ , elastic net reduces to lasso. Elastic net is effective when dimensional space  $n$  is much greater than the observations  $p$  ( $p > n$ ). In our research, the predictors greatly exceed the samples as they increase with the label space. Thus we choose to develop a statistical model based on the elastic net for multi-label data processing. The elastic net gives computational flexibility and performance stability.

## 2.4 Proposed Algorithm

CAL500, Enron, Corel5k, Rcv1 (subset1), Rcv1 (subset2), Mediamill, tmc2007, Corel6k, and eurlex-sm have SCUMBLE values more than 0.1, and they are extremely difficult MLDs. They take a high degree of concurrence amongst labels with distinct imbalance levels. Existing sampling methods ‘won’t be suitable for handling such labels with high concurrence values. On the other hand, emotions, Medical, Scene, Slashdot, Bibtex, and Genbase, have low SCUMBLE values, and the imbalanced processing with the conventional methods over these datasets will benefit. The proposed framework uses a two-stage approach to balance and predicts the imbalanced multi-label data. The first stage uses the pre-processing data to balance the imbalanced multi-label dataset. The multi-label SMOTE (*MLSMOTE*) implemented is modified to treat labels in the training set as a one-vs.-rest way to create new synthetic samples. Then a new balanced multi-label dataset of original and synthetic samples is given as input for adaptive weighted  $l_{21}$ -norm regularized logistic regression to make the learning and prediction over the balanced dataset. Fig. 3 describes the flow of the data balancing approach.

### 2.4.1 Borderline MLSMOTE

SMOTE system uses heuristics to pick samples with minority labels; ergo, their operation was better than other pre-processing approaches. However, to accomplish a better forecast, the sample minority labels that are high interactions with all majority labels must be obtained instead of sample majority labels. These minority labels using high concurrent values are considered borderline, and the neighboring ones tend to be more inclined to become misclassified compared to the main ones, much against the uncontrollable ones. Therefore, those concurrent minority labels tend to be more crucial for classification.

---

```

Input: Training Dataset D, sampling ratio  $\alpha$ , number of nearest neighbours k
Output: balanced dataset  $D_{new}$ 
Begin
  Initialize  $D_{new} = \phi$ ,  $D_{relevant} = \phi$ 
  L = getLabels(D) // get label set
  MeanIR = Calculate MeanIR(D,L) by Eq. 3
  SCUMBLEi = calculates CUMBLE(Di,D) by Eq. 4 // SCUMBLE for each instance
  SCUMBLE =  $\frac{1}{|D|} \sum_{i=1}^{|D|} SCUMBLE_i$  // SCUMBLE of dataset
  For each instance Di in D // all samples in dataset
    Begin
    if SCUMBLEi > SCUMBLE then // Border line labels
      For each label li in L // each label
        Begin
        IRperLabel = CalculateIRperLabel(Di,li)
        if IRperLabel > MeanIR then
          Drelevant = getAllInstances(D, li) // Create Relevant Bag
        End if
      End for
    End for
  For each sample in Drelevant
    Begin
    distances = calculateDistance(sample, Drelevant) // nearest neighbours of borderline
    labels
    neighbours = getAllNeighbours(distances,k)
    ran_neighbour = getRandomNeighbour(neighbours)
    End for
  For each feat in sample // New sample generation
    Begin
     $\Delta = \text{ran\_neighbour.feat} - \text{sample.feat}$ 
     $\gamma = \Delta * \text{uniform}(0,1)$ 
    featnew = sample.feat +  $\gamma$ 
    Dnew.feat = featnew
    End for
  N = |Drelevant.labels| + |neighbours.labels|
  Lnew = N > (k+1) / 2
  Dnew.labels = Lnew
  Return Dnew
End

```

---

**Figure 3:** Phase 1-balancing the data set through *Borderline-MLSMOTE*

The instances with those concurrent labels tend to be somewhat more prone to become misclassified. Focusing resampling on those samples with labels that are concurrent makes it more beneficial compared to doing whole minority labels. However, the samples from the samples that are concurrent can contribute little to this classification. Our approaches are all predicated on the synthetic minority over-sampling Technique. The synthetic sample creation pre-processing creates synthetic minority samples to oversample the concurrent minority class. For each single concurrent minority label, its own k closest nearest neighbor is calculated, and then some samples are randomly chosen in line with this sampling speed. Now, the brand-new synthetic samples have been generated concerned with concurrent minority labels along with their own chosen nearest neighbor. Unlike the multi-label SMOTE system, our suggested pre-processing only reinforces the borderline (concurrent) minority samples. The generated synthetic samples are subsequently added to the unique training set. The most



widely selected neighborhood size  $k = 5$  [23] is used in this work. The flow of the proposed Borderline MLSMOTE pre-processing is shown in Fig. 3.

#### 2.4.2 Adaptive Weighted Logistic Regression for Multi-Label Data

The adaptive weighted elastic net is used to learn and predict the balanced data set created in the first phase. The weighted regularisation ensures the handling of synthetic samples created in phase 1. The adaptive elastic-net guarantees variable selection by adding  $l_2$  regularization with an adaptive lasso to address multi-collinearity problems. Reference [24] outperformed well than an adaptive lasso and elastic-net in terms of accuracy while maintaining a higher value of true positive rate and a lower value of false negative rate with the selected predictors. The elastic net proposed in [25] is modified to an adaptive weighted elastic net via logistic regression to address three issues: over-fitting, biased estimation, multi-collinearity, and low false-positive rate. The regularization part of the elastic-net logistic regression is modified by adding weight terms to the lasso and ridge parts. The Eq. (12) is a modified adaptive elastic net.

$$AdaR(\Theta) = \left(1 + \frac{\lambda_2}{m}\right) \operatorname{argmin}_{\beta} \sum_{i=1}^m \|x_i \beta_i - y_i\|_2^2 + \frac{m}{2} \lambda_2 \sum_{i < m} w_i \|\beta_i\|_2^2 + m \lambda_1 \sum_{i < m} w_i \|\beta_i\|_1 \quad (12)$$

where  $w_i > 0$ ,  $i = 1, 2, \dots, m$  are the weighted penalty coefficients. The key idea in adaptive elastic net is the weight parameter only. This weight makes the adaptive elastic net perform different amounts of shrinkage to different predictors and penalizes the smaller coefficient predictors more severely. The coefficient estimations for the elastic net are done first to construct the adaptive weights of the lasso and ridge part of the adaptive elastic net. This is presented in Eq. (13).

$$w_i = (|\beta_i(\mathbf{R}(\Theta))|)^{-\gamma}, i = 1, 2 \dots K \quad (13)$$

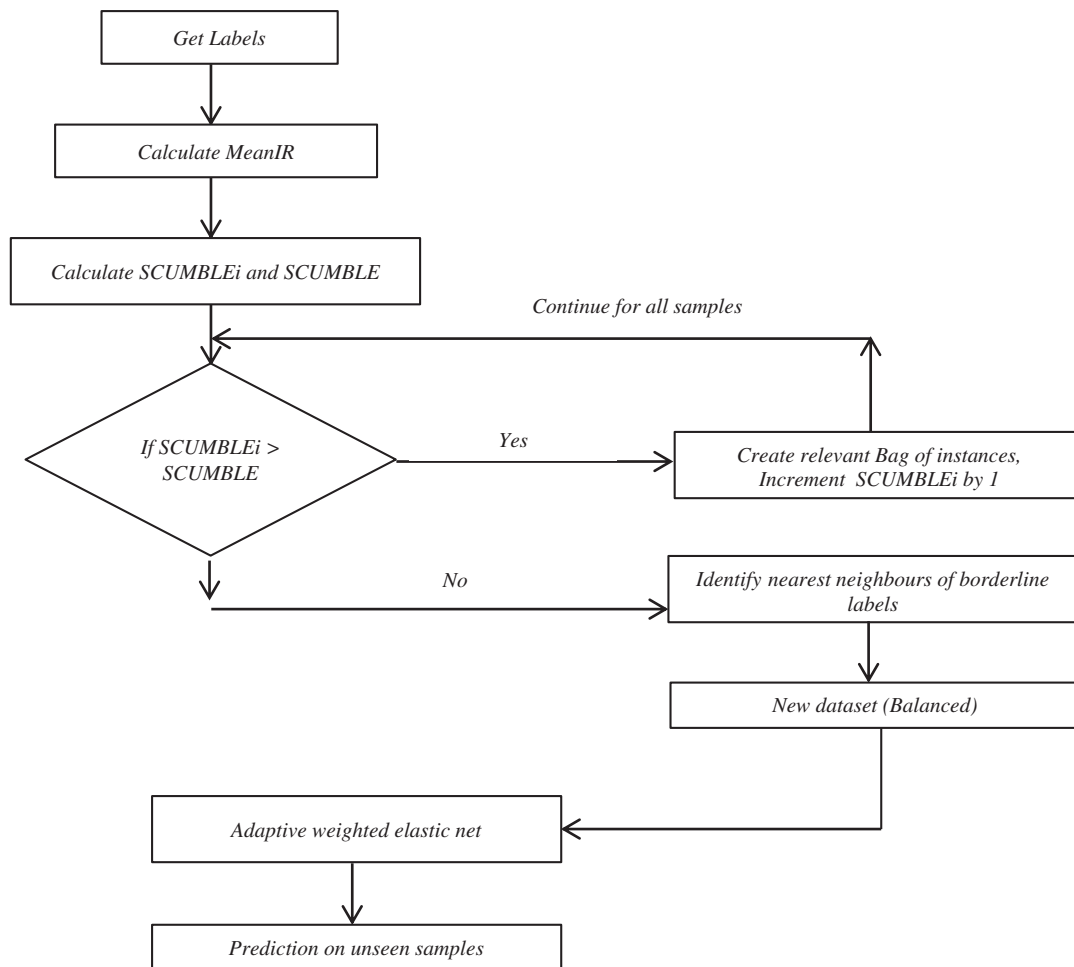
$\gamma$  is a positive constant. In this paper, we use  $\gamma = 1$ . The  $\mathbf{R}(\Theta)$  in Eq. (11) gives the initial estimator for weight in Eq. (13). The learning through the adaptive weighted elastic net is presented in Fig. 4. The proposed framework is illustrated in Fig. 5.

---

**Input:** Balanced Training Dataset  $D_B$ ,  $\lambda$   
**Output:** Model Coefficients  $\beta = [\beta_1, \beta_2 \dots \beta_k]$   
Initialize parameter  $\beta_0 = 0$ ;  
**For each** label  $l$  **in** dataset  $D_B$   
**Begin**  
    Compute the adaptive weight using eqn. 13  
    Compute penalty term using eqn.12  
    Compute the posterior probability using eqn.10  
    Add the penalty term to the posterior probability and compute the argmin using eqn.9  
**End for**  
**Return** Coefficients

---

**Figure 4:** Phase 2-learning of balanced multi-label data through the adaptive weighted elastic net



**Figure 5:** Proposed framework

### 3 Experimental Setup

This section describes the list of multi-label datasets employed with this experimentation and the evaluation metrics used to evaluate the learning algorithms.

#### 3.1 Data Sets Used

The datasets used for the experimentation are taken from various domains like music, text, image, video, and biology. All the benchmark data used here are available in the MULAN data repository. The data sets are shown in [Table 1](#).

#### 3.2 Evaluation Metrics

The efficiency assessment of multi-label techniques will be far more ambitious than single-label classification since it involves several labels.

**Table 1:** Description of the data sets used

S. No.	Data Set	Domain	#n	#d	#l	$L_C$	$L_D$	$IR_{\min}$	$IR_{\text{avg}}$	$IR_{\max}$	SCUMBLE
1.	Corel5k	Image	5000	499	44	2.214	0.050	3.460	17.857	50.000	0.393
2.	Mediamill	Video	43907	120	29	4.010	0.138	1.748	7.092	45.455	0.355
3.	CAL500	Music	502	68	174	25.058	0.202	1.040	3.846	24.390	0.336
4.	Enron	Text	1702	1001	53	3.378	0.130	1.00	5.348	43.478	0.302
5.	Corel16k	Image	13766	500	161	2.867	0.018	1.808	34.1552	126.80	0.279
6.	Rcv1(subset 1)	Image	6000	472	42	2.458	0.059	3.344	15.152	50.000	0.223
7.	Rcv1(subset 2)	Image	6000	472	39	2.170	0.056	3.215	15.873	47.619	0.209
8.	Eurlex-sm	Text	19348	250	27	1.492	0.055	1.447	5.848	34.483	0.182
9.	tmc2007	Text	28596	500	15	2.100	0.140	1.447	5.848	34.483	0.175
10.	Yeast	Biology	2417	103	13	4.233	0.325	1.328	2.778	12.500	0.104
11.	Bibtex	Text	7395	1836	159	2.402	0.015	0.450	12.4983	20.4314	0.094
12.	Medical	Text	978	1449	45	1.275	0.077	2.674	11.236	43.478	0.046
13.	Genbase	biology	662	1186	27	1.252	0.046	1.4494	37.3146	171.00	0.028
14.	Slashdot	Text	3782	53	14	1.134	0.081	5.464	10.989	35.714	0.013
15.	Emotions	Music	593	72	6	1.869	0.311	1.247	2.146	3.003	0.010
16.	Scene	image	2407	294	6	1.074	0.179	3.521	4.566	5.618	0.0003

Note: #n–Number of samples, #d–Number of features, #l–Number of labels,  $L_C$ –Label Cardinality,  $L_D$ –Label Density,  $IR_{\min}$ –Imbalance Ratio minimum,  $IR_{\max}$ –Imbalance Ratio maximum,  $IR_{\text{avg}}$ –Imbalance Ratio average.

1) *Hamming Loss*  $\downarrow$  requires the error of prediction, overlooking missing errors into consideration, and testimonials the typical example-label set misclassification. Therefore, a lower  $h\_loss$  value shows higher classifier performance. The hamming\_Loss is specified as in Eq. (14).

$$h\_loss(x_i, y_i) = \frac{1}{q} \sum_{i=1}^{|q|} \frac{1}{|I|} |h(x_i \Delta y_i)| \quad (14)$$

Or

Let  $tp_i$ ,  $tn_i$ ,  $fp_i$  and  $fn_i$  are the count of true positive, true negative, false positive, and false negative of  $i$ th sample, respectively, and the  $h\_loss$  can be defined as in Eq. (15).

$$h\_loss(x_i, y_i) = \frac{1}{q} \sum_{i=1}^{|q|} \frac{fp_i + fn_i}{tp_i + fp_i + tn_i + fn_i} \quad (15)$$

2) *One\_error*  $\downarrow$  outlines the lack of high-positioned labels *vs.* the proper label in the instant collection. This step chooses the good value between 1 and 0. The smaller the value of *one\_error*, the classifier does effectively, and it is defined as in Eq. (16).

$$One\_error(\hat{h}) = \frac{1}{N} \sum_{i=1}^N \left\| \underset{\lambda \in \ell}{argmax}(x_i \lambda) \right\| \quad (16)$$

This step is comparable to the classification error just in a single-label classification problem.

3) *Ranking Loss*  $\downarrow$  defines the quality of reversely ordered label sets for the specified example, plus it is also as in Eq. (17).

$$\text{Ranking Loss} = \frac{1}{N} \sum_{i=1}^N \frac{|\lambda_a, \lambda_b|}{|x'_i| |x_i|} \quad (17)$$

When the ranking loss is smaller, the learning algorithm's performance is better.

4) *Average Precision*  $\uparrow$  computes the normal percentage of proper labels in every label set. Fundamentally, the quality has performed all applicable labels. This is provided from Eq. (18).

$$\text{Average Precision} = \frac{1}{N} \sum_{i=1}^N \frac{1}{|x'_i|} \sum_{\lambda \in x'_i} \frac{|\lambda' \in x'_i|}{r_i(\lambda)} \text{ where } r_i(\lambda') \leq r_i(\lambda) \quad (18)$$

Better the significance of moderate precision improved the learning algorithm's performance, and when standard precision = 1, the learning algorithm shows optimum performance.

5) *Subset Accuracy*  $\uparrow$  is characterized by the Jaccard similarity coefficient amongst label sets  $\hat{h}(x_i)$  and  $y_i$ . This step will be a type across all instances and is represented as in Eq. (19).

$$\text{Subset Accuracy}(\hat{h}) = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{h}(x_i) \cap y_i|}{|\hat{h}(x_i) \cup y_i|} \quad (19)$$

6) *Coverage*  $\downarrow$  is the portion of covered labels in the instance collection. Tiny the present, the higher the performance, i.e., far better label coverage. We must use the example set to cover the remaining uncovered labels if this measure is high. This is given in Eq. (20).

$$\text{Cover}(\hat{h}) = \frac{1}{N} \sum_{i=1}^N \max_{\lambda \in \ell} \text{rank}(x_i, \lambda) - 1 \quad (20)$$

## 4 Results and Discussion

### 4.1 Performance Comparison of Proposed Method against Competing Methods

The performance of a multi-label classifier is assessed in the shape of several test metrics. These classification results are evaluated with five multi-label measures: Hamming Loss, Subset Accuracy, Average Precision, One Error, Ranking Loss, and Coverage. The Hamming Loss is a sample-based step that assesses the gaps between the predicted and the provided label set. Lower the Hamming, the greater the predictions. Average Precision is an example-based step and a usual performance metric. Finally, One\_Error and Ranking Loss are ranking-based metrics. The different comparison methods are chosen because they have relatively high performance and efficiency.

Table 2 presents the overall performance of the proposed system. Table 2 shows that the suggested framework works quite competing methods utilized for the experimentation. The proposed method performs well on most data sets and is like other rival techniques with hamming loss and accuracy. The CAL500 exhibits 92% subset accuracy and 99% precision. The hamming loss for this dataset is 0.1. The Emotions and Medical datasets' accuracy ranges are 87% and 93 %, respectively. The precision value on 'Emotion's data reaches a good 99%. In Medical data, the misclassification rate was reduced to 0.03 only. The Corel5k is a dataset with a high concurrence problem as its SCUMBLE is 0.39. Therefore, the accuracy is 83%. But for the same dataset, the MLSR produces 89% accuracy. However, the Precision is 93% with the proposed framework, and the hamming loss is 0.01. For Rcv1 (subset1) and Rcv1 (subset 2) data, the subset accuracy is 97% and 80%, respectively. The Hamming Loss measures 0.09 for Rcv1 (subset1) and 0.01 for Rcv1 (subset2). The hamming loss for Rcv1 (subset2) is much less than Rcv1 (subset1) as the SCUMBLE value of Rcv1 (subset2) is less than Rcv1 (subse1). This shows that

the dataset with low concurrent value makes the learner perform well. The hamming loss for the Scene dataset is 0.02 with 99% accuracy and 92% precision value. The Scene data has a low SCUMBLE value, i.e., 0.003, among other datasets taken for our experimentation. Even though many datasets got benefited from the proposed framework, the Core5k gives the second-best Hamming Loss of 0.1007. The MLSR performs best with the Corel5k dataset, resulting in 0.1005 Hamming Loss and 89% accuracy. But still, the proposed system with Corek5k gives the second-best 83% accuracy. In general, the dataset with high concurrence labels is much benefited from the proposed system. Finally, experimentation on 17 datasets shows that the proposed framework performs better than the other nine popular methods.

**Table 2:** Performance comparison of the proposed method against competing methods

Data set	Performance measures	IBLR	ECC	BR	RAkEL	ML-kNN	CLR	MLSR	MDDM	MORP	Proposed
CAL500	Hamming loss ↓	0.1420	0.2015	0.2518	0.2566	0.2198	0.1987	0.1928	0.1638	0.1637	<b>0.1036</b>
	Subset accuracy ↑	0.5601	0.5564	0.5089	0.4743	0.5331	0.5598	0.5038	0.5172	0.5728	<b>0.9283</b>
	Average precision ↑	0.7538	0.7843	0.7580	0.6992	0.7698	0.7999	0.6982	0.6538	0.6628	<b>0.9937</b>
	Ranking loss ↓	0.1677	0.2037	0.2208	0.3006	0.2096	0.1776	0.2293	0.2182	0.2273	<b>0.1272</b>
	One_error ↓	0.1423	0.2749	0.2306	0.2980	0.2053	0.2699	0.2335	0.2284	0.2102	<b>0.1087</b>
	Coverage ↓	2.0169	2.0607	2.1063	2.5465	2.0623	1.8922	1.8920	1.9872	2.0371	<b>1.5269</b>
Emotions	Hamming loss ↓	0.2010	0.1993	0.2555	0.2566	0.2246	0.2125	0.2802	0.2173	0.2302	<b>0.1566</b>
	Subset accuracy ↑	0.5390	0.5175	0.5207	0.4743	0.5073	0.5196	0.4338	0.5544	0.5114	<b>0.8743</b>
	Average precision ↑	0.6947	0.7338	0.7624	0.6992	0.7885	0.7819	0.6517	0.3630	0.7169	<b>0.9992</b>
	Ranking loss ↓	0.3015	0.2706	0.1967	0.3006	<b>0.1713</b>	0.1819	0.3545	0.7706	0.2709	0.2006
	One_error ↓	0.3930	0.3070	0.3576	0.3980	0.2969	0.2985	0.4772	0.8819	0.4014	<b>0.2980</b>
	Coverage ↓	1.5869	2.4199	1.9360	2.5465	1.8449	1.9022	2.7284	4.3086	2.3542	<b>1.0465</b>
Medical	Hamming loss ↓	0.0774	0.0658	0.0729	0.0629	0.0573	0.0728	0.0652	0.0526	0.0682	<b>0.0392</b>
	Subset accuracy ↑	0.5068	0.2182	0.2198	0.2278	0.5092	0.2135	0.5788	0.6526	0.5997	<b>0.9342</b>
	Average precision ↑	0.4735	0.4625	0.4523	0.4992	0.4426	0.4019	0.3927	0.3263	0.3893	<b>0.9019</b>
	Ranking loss ↓	0.2839	0.2019	0.2734	0.2435	0.2283	0.2361	0.2021	0.2172	0.2253	<b>0.1526</b>
	One_error ↓	0.1353	0.1392	0.1428	0.1028	0.1426	0.1325	0.1397	0.1399	0.1428	<b>0.1028</b>
	Coverage ↓	1.2937	1.1927	1.2739	1.0173	1.4572	1.3920	1.5739	1.4983	1.5028	<b>1.0527</b>
Enron	Hamming loss ↓	0.2058	0.3389	0.3291	0.3009	0.3857	0.3134	0.2945	0.2437	0.2754	<b>0.1971</b>
	Subset accuracy ↑	0.0648	0.0632	0.0592	0.0611	0.0598	0.0534	0.0485	0.0472	0.04173	<b>0.9571</b>
	Average precision ↑	0.4836	0.4982	0.4383	0.4927	0.4892	0.4492	0.4856	0.4139	0.5193	<b>0.9072</b>
	Ranking loss ↓	0.1233	0.1927	0.1873	0.1865	0.1856	0.1943	0.1827	0.2183	0.1855	<b>0.0859</b>
	One_error ↓	0.4098	0.9311	0.6921	0.9372	0.7372	0.8922	0.7822	0.8927	0.7212	<b>0.2310</b>
	Coverage ↓	1.9237	1.8729	1.8882	1.9833	1.8727	1.8192	2.1932	2.4821	2.4382	<b>1.3471</b>
Scene	Hamming loss ↓	0.0798	0.0728	0.0732	0.0739	0.0725	0.0732	0.0526	0.0529	0.0592	<b>0.0228</b>
	Subset accuracy ↑	0.7892	0.7256	0.7927	0.8019	0.6299	0.8917	0.6982	0.6625	0.6383	<b>0.9989</b>
	Average precision ↑	0.5638	0.5927	0.5832	0.5725	0.5829	0.5535	0.6093	0.6192	0.6082	<b>0.9282</b>
	Ranking loss ↓	0.3923	0.3846	0.3626	0.4017	0.3975	0.4103	0.5039	0.5102	0.5097	<b>0.2273</b>
	One_error ↓	0.1801	0.1578	0.1623	0.1937	0.1643	0.1547	0.2092	0.2176	0.2097	<b>0.1283</b>
	Coverage ↓	2.0371	1.9277	1.8235	2.0458	1.8290	1.9366	1.9827	1.8872	1.5279	<b>1.3092</b>
Yeast	Hamming loss ↓	0.1836	0.1936	0.2846	0.1926	0.1872	0.2017	0.2423	0.2978	0.2783	<b>0.1110</b>
	Subset accuracy ↑	0.3760	0.3911	0.4811	0.3812	0.4621	0.4706	0.5312	0.4928	0.4802	<b>0.9019</b>
	Average precision ↑	0.4834	0.3978	0.4378	0.4586	0.3978	0.4837	0.3982	0.3927	0.3826	<b>0.8528</b>
	Ranking loss ↓	0.1990	0.1830	0.2002	0.1930	0.1932	0.1845	0.2018	0.2418	0.2614	<b>0.0992</b>
	One_error ↓	0.8921	0.8821	0.8819	0.8913	0.8229	0.7271	0.7897	<b>0.6190</b>	0.8392	<b>0.1590</b>
	Coverage ↓	1.7456	1.9822	1.7393	2.0913	1.7392	2.0381	1.0289	1.2738	1.2473	<b>1.0947</b>
Slashdot	Hamming loss ↓	0.1937	0.2028	0.2947	0.2374	0.2247	0.2750	0.1827	0.1725	0.1889	<b>0.1027</b>
	Subset accuracy ↑	0.3958	0.3358	0.3028	0.3658	0.3579	0.3590	0.4728	0.4526	0.4923	<b>0.8827</b>
	Average precision ↑	0.5992	0.5954	0.5239	0.5539	0.5057	0.5258	0.4928	0.4829	0.4262	<b>0.8927</b>
	Ranking loss ↓	0.3038	0.4456	0.4937	0.5102	0.5329	0.5683	0.3937	0.3728	0.3576	<b>0.2920</b>
	One_error ↓	0.1940	0.2249	0.2673	0.2564	0.2847	0.2759	0.1928	0.1829	0.1393	<b>0.1728</b>

(Continued)

**Table 2 (continued)**

Data set	Performance measures	IBLR	ECC	BR	RAkEL	ML-kNN	CLR	MLSR	MDDM	MORP	Proposed
	Coverage ↓	1.2478	1.5749	1.5559	1.2854	1.0894	1.9238	1.7283	1.9327	1.3845	<b>1.0489</b>
Corel5k	Hamming loss ↓	0.1024	0.1027	0.1076	0.1085	0.1019	0.1034	<b>0.1005</b>	0.1082	0.1054	0.1007
	Subset accuracy ↑	0.6737	0.6622	0.6238	0.6492	0.6292	0.6282	<b>0.8918</b>	0.7142	0.6826	0.8354
	Average precision ↑	0.5529	0.53538	0.5628	0.5281	0.5938	0.5621	0.8836	0.5934	0.6262	<b>0.9382</b>
	Ranking loss ↓	0.2048	0.2428	0.2738	0.2839	0.2510	0.2472	0.1930	0.2734	0.2438	<b>0.1639</b>
	One_error ↓	0.1129	0.1927	0.1537	0.1409	0.1728	0.1548	0.1097	0.1382	0.1273	<b>0.1026</b>
	Coverage ↓	1.1289	1.1637	1.1689	1.1927	1.1538	1.1782	1.1849	1.3903	1.1927	<b>1.0937</b>
Rcv1 (subset 1)	Hamming loss ↓	0.1826	0.1973	0.1934	0.1916	0.1423	0.1234	0.1812	0.1221	0.1935	<b>0.0916</b>
	Subset accuracy ↑	0.6934	0.6812	0.6810	0.6101	0.6091	0.6917	0.6842	0.6817	0.6271	<b>0.9716</b>
	Average precision ↑	0.6937	0.7032	0.6348	0.6634	0.7012	0.6342	0.7845	0.7743	0.7623	<b>0.9238</b>
	Ranking loss ↓	0.2035	0.2045	0.2859	0.2754	0.2347	0.2854	0.3154	0.2901	0.3017	<b>0.1560</b>
	One_error ↓	0.2954	0.2018	0.2292	0.3018	0.3045	0.2920	0.3047	0.3081	0.3001	<b>0.1946</b>
	Coverage ↓	1.3929	1.0380	1.5830	1.4924	1.3840	1.3957	2.5498	2.0739	2.3819	<b>1.1471</b>
Rcv1 (subset 2)	Hamming loss ↓	0.0672	0.0628	0.0623	0.0614	0.0693	0.0658	<b>0.0116</b>	0.0159	0.0128	0.0119
	Subset accuracy ↑	0.5116	0.5232	0.4182	0.5383	0.4019	0.5152	0.7763	0.6528	0.6382	<b>0.8028</b>
	Average precision ↑	0.5373	0.5947	0.5372	0.6027	0.5532	0.5382	0.7928	0.6836	0.6124	<b>0.8352</b>
	Ranking loss ↓	0.3927	0.3846	0.3253	0.3012	0.3549	0.3328	0.3027	0.4282	0.4153	<b>0.2523</b>
	One_error ↓	0.1935	0.1823	0.1528	0.1102	0.1538	0.1628	0.0979	0.2093	0.2183	<b>0.0284</b>
	Coverage ↓	1.0928	1.0376	1.0283	1.0192	1.0826	1.0263	1.1028	1.1927	1.1263	<b>1.0378</b>
tmc2007	Hamming loss ↓	0.0927	0.0266	0.0239	0.0264	0.0292	0.0291	0.0527	0.0639	0.0592	<b>0.0128</b>
	Subset accuracy ↑	0.6918	0.7019	0.6827	0.6725	0.6283	0.6629	0.5986	0.5024	0.5756	<b>0.8425</b>
	Average precision ↑	0.6837	0.7038	0.6352	0.6392	0.6643	0.6293	0.6037	0.5833	0.5537	<b>0.8284</b>
	Ranking loss ↓	0.1916	0.1567	0.2873	0.2273	0.2671	0.1982	0.1283	0.1772	0.1980	<b>0.1330</b>
	One_error ↓	0.2039	0.1527	0.1826	0.1926	0.1527	0.2081	0.1392	0.1562	0.1982	<b>0.1258</b>
	Coverage ↓	1.9280	1.2327	1.5281	1.3259	1.6299	1.4523	1.0192	1.1728	1.1839	<b>1.0211</b>
Mediamill	Hamming loss ↓	0.1836	0.1936	0.2046	0.1926	0.1872	0.2017	0.1923	0.2078	0.2083	<b>0.1510</b>
	Subset accuracy ↑	0.6076	0.6191	0.6081	0.5181	0.5062	0.5076	0.6312	0.6928	0.6827	<b>0.8092</b>
	Average precision ↑	0.5834	0.5978	0.5378	0.5586	0.5978	0.5837	0.6982	0.5927	0.5826	<b>0.8528</b>
	Ranking loss ↓	0.1990	0.1830	0.2002	0.1930	0.1932	0.1845	0.2018	0.2418	0.2614	<b>0.1092</b>
	One_error ↓	0.3921	0.3821	0.3819	0.3913	0.3229	0.3271	0.3897	0.3190	0.3392	<b>0.2347</b>
	Coverage ↓	1.7456	1.9822	1.7393	2.0913	1.7392	2.0381	1.0289	1.2738	1.2473	<b>1.1947</b>
Corel16k	Hamming loss ↓	0.0203	0.0352	0.0283	0.0384	0.0183	0.0972	0.0255	0.0836	0.0352	<b>0.0152</b>
	Subset accuracy ↑	0.5282	0.5420	0.5920	0.5429	0.5027	0.5829	0.5421	0.59267	0.6826	<b>0.8429</b>
	Average precision ↑	0.5927	0.5018	0.5172	0.5284	0.6038	0.5978	0.5828	0.5527	0.5293	<b>0.9028</b>
	Ranking loss ↓	1.1393	0.1439	0.1595	0.1982	0.2038	0.2573	0.2384	0.2987	0.2674	<b>0.1073</b>
	One_error ↓	0.1338	0.1872	0.1772	0.1572	0.1027	0.1762	0.1823	0.1563	0.1487	<b>0.0362</b>
	Coverage ↓	1.0013	1.2829	1.1093	1.1256	1.157	1.1982	1.1862	1.1682	1.1579	<b>1.0072</b>
Genbase	Hamming loss ↓	0.0352	0.0283	0.0384	0.0183	0.0572	0.0255	0.0836	0.0352	0.0352	<b>0.0182</b>
	Subset accuracy ↑	0.5027	0.5829	0.4421	0.4926	0.5426	0.5429	0.5082	0.6420	0.5920	<b>0.8429</b>
	Average precision ↑	0.6981	0.6412	0.6725	0.6627	0.6293	0.5928	0.6172	0.6427	0.6912	<b>0.8827</b>
	Ranking loss ↓	0.3393	0.3039	0.3095	0.3982	0.2938	0.3503	0.3284	0.3987	0.3804	<b>0.1073</b>
	One_error ↓	0.2038	0.2892	0.2072	0.2592	0.2327	0.2962	0.2923	0.2093	0.2087	<b>0.1062</b>
	Coverage ↓	1.3413	1.3929	1.2903	1.2056	1.2157	1.2982	1.2602	1.1908	1.1987	<b>1.1087</b>
Bibtex	Hamming loss ↓	0.07095	0.07913	0.07252	0.06826	0.07292	0.07162	0.0493	0.0501	0.0498	<b>0.0528</b>
	Subset accuracy ↑	0.7451	0.7519	0.7143	0.7242	0.7092	0.7172	0.7259	0.7583	0.7019	<b>0.8947</b>
	Average precision ↑	0.6091	0.6267	0.6192	0.6815	0.6269	0.6735	0.5628	0.6193	0.5823	<b>0.8628</b>
	Ranking loss ↓	0.0577	0.0571	0.06921	0.05930	0.06408	0.0598	0.0683	0.06342	0.0610	<b>0.0452</b>
	One_error ↓	0.01920	0.02043	0.01837	0.02994	0.01937	0.02834	0.01128	0.02893	0.01983	<b>0.0109</b>
	Coverage ↓	1.4937	1.4028	1.5018	1.5937	1.5826	1.4927	<b>1.3272</b>	1.4028	1.3923	1.3321
Eurlex-sm	Hamming loss ↓	0.1036	0.1306	0.1146	0.1026	0.1172	0.1097	0.1042	0.1029	0.1027	<b>0.1001</b>
	Subset accuracy ↑	0.5760	0.5911	0.5811	0.5812	0.5621	0.5706	0.5312	0.5928	0.5802	<b>0.8919</b>
	Average precision ↑	0.5834	0.5978	0.5878	0.5486	0.5978	0.5937	0.5702	0.5987	0.5856	<b>0.8528</b>

(Continued)

**Table 2 (continued)**

Data set	Performance measures	IBLR	ECC	BR	RAkEL	ML-kNN	CLR	MLSR	MDDM	MORP	Proposed
	Ranking loss ↓	0.1990	0.1830	0.2002	0.1930	0.1932	0.1845	0.2018	0.2418	0.2614	<b>0.0992</b>
	One_error ↓	0.1038	0.1072	0.1172	0.1012	0.1090	0.1062	0.1083	0.1053	0.1047	<b>0.0982</b>
	Coverage ↓	1.8456	1.7822	1.6793	2.2013	1.7992	2.1381	1.7689	1.7838	1.7473	<b>1.1817</b>

#### 4.2 Performance Comparison of Proposed Method against Recent Methods

This paper proposes a novel framework for learning and classifying the imbalanced multi-label data in two phases so that the logistic regression model will work better on imbalanced data. Phase 1 has a pre-processing method named *Borderline MLSMOTE*, which expands minority labels in areas where the concurrent appearance of the minority and majority labels is too high. Phase 2 has an adaptive weighted  $l_{21}$ -norm regularized weighted logistic regression to address over-fitting and variable selection. Phase 1 uses data pre-processing to balance imbalanced multi-label data, where Multi-label SMOTE (*MLSMOTE*) is modified to treat labels in the training set as one-vs.-rest to create new synthetic samples. Elastic net is modified to the adaptive weighted elastic net to address over-fitting, biased estimation, multi-collinearity, and low false-positive rate. The key challenge to multi-label data lies when the number of labels for prediction is exponential. This involves exploiting label correlation among labels, over-fitting due to high dimensional predictive space, and highly imbalanced training sets. The proposed *Borderline MLSMOTE* is compared with other methods to demonstrate the superiority of the proposed method and is presented in Table 3. The proposed method combines two phases, and phase 2 works on pre-processed data. Most of the other comparison methods only focus on classification, which seems unfair as they work on original datasets.

**Table 3:** Comprehensive comparison results between the proposed method and recent algorithms

	Yeast				Enron				Scene			
	HL	RL	AP	OE	HL	RL	AP	OE	HL	RL	AP	OE
Shu et al. [26]	0.183	00.158	00.782	0.193	0.161	0.141	0.831	0.257	0.019	0.052	0.791	0.273
Wu et al. [27]	0.222	00.207	00.709	0.224	0.299	0.278	0.451	0.600	0.021	0.051	0.761	0.314
Zhang et al. [28]	0.192	00.164	00.771	0.222	0.166	0.149	0.818	0.277	0.036	0.519	0.388	0.972
<b>Proposed method</b>	<b>0.111</b>	<b>0.099</b>	<b>0.853</b>	<b>0.159</b>	<b>0.152</b>	<b>0.127</b>	<b>0.843</b>	<b>0.197</b>	<b>0.014</b>	<b>0.031</b>	<b>0.921</b>	<b>0.187</b>

Note: HL–Hamming Loss, RL–Ranking Loss, AP–Average Precision, OE–One Error.

In contrast, the proposed method works on new datasets processed by *Borderline MLSMOTE*. Incorporating multiple cluster centers for multi-label learning (IMCC) [38] creates more samples out of neighborhood clustering centers to expand the training set and realize data enhancement. Feature-induced labeling information enrichment for multi-label learning (MLFE) [39] employs the structure information of attribute area to improve label details. Joint Ranking SVM and Binary Relevance with Robust Low-Rank Learning for Multilabel Classification (RBRL) [40] show Ranking SVM and Binary relevance with low-rank solid learning. Three standard data sets have been chosen to validate the legitimacy of the proposed method, yeast (gene function prediction using 2417 samples and 14 labels), image (image classification with 2000 samples and five labels) along with also social (5000 samples with 39 labels). The performance of the proposed framework has been examined with three recent state-of-the-art procedures and confirmed against four metrics. Table 3 shows the comparison of the proposed method against different recent works. The proposed method outperforms all metrics in

yeast, image, and social dataset. The result shows the proposed method accomplishes a nearly flawless prediction on yeast, image, and social datasets and exhibits the potency of this suggested procedure.

The performance comparison of the proposed system with and without the pre-processing stage is presented in Figs. 6–8. The accuracy increases at reasonable rates for all datasets. Specifically, this framework benefited a lot of data sets with high SCUBMLE values. The datasets with high concurrent value are Corel5k, Mediamill, CAL500, Enron, Corel6k, and both Rcv1 data benefitted greatly from this framework. The accuracy of CAL500 increases to 31%, and the accuracy is 92%. With Emotions and Mediamill the percentage increase is 27.53% and 19.96%. The accuracy of Emotions is 87.43% and 80.92%. The overall increase in accuracy ranges from 7% to 31%. The overall increase in Precision for all datasets ranges from 3 % to 53%. The Corel5k dataset with high concurrence measures benefits greatly from this framework in terms of Precision—the precision value for the Corel5k dataset increases from 40% to 93%. Mediamill showed 50%; with the proposed framework, it gives 85% precision. The Scene dataset offers 87% without phase 1, which now provides 92% precision. The increase in the percentage of Precision after the framework is as follows: CAL500 (28%), Enron (37.36%), Rcv1 (Subset2) (25.14%), Eurlex-sm (24.44%), Corel6k (23.01%), tmc2007 (13.77%).

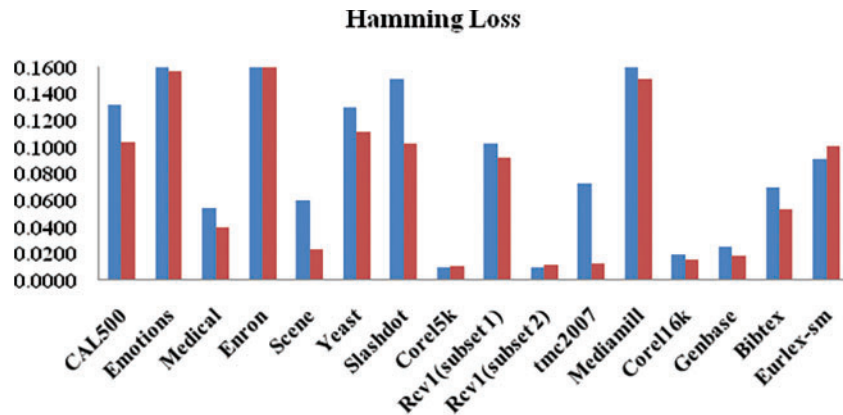


Figure 6: Hamming loss performance of adaptive weighted elastic net with and without pre-processing

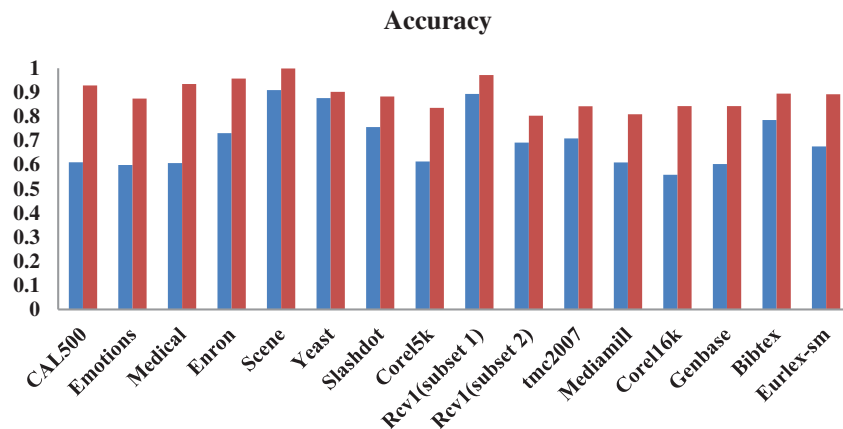
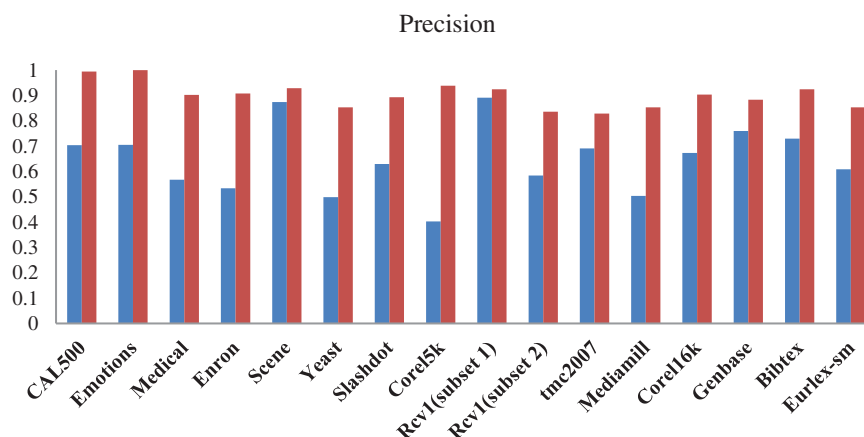


Figure 7: Accuracy of adaptive weighted elastic net with and without pre-processing





**Figure 8:** Performance of adaptive weighted elastic net with and without pre-processing

## 5 Conclusion

A framework to classify and predict imbalanced multi-label data has been introduced in this work. The framework has two phases; (1) an adaptive *Borderline-MLSMOTE* has proposed and pre-processed the biased data, and (2)  $l_{21}$ -norm (Elastic net) regularized adaptive weighted logistic regression has exploited to learn parameters and predict the processed data. This variant of *MLSMOTE* concentrates on minority concurrence labels that contribute to the relief imbalance among multiple labels and promote the influence of minority labels. Experimental effects on various multi-label datasets have shown that the proposed framework enhances the performance over other competing and recent methods in most cases. The results confirm that the dataset with concurrent high labels benefited greatly from the proposed system. The proposed *Borderline-MLSMOTE* method works based on kNN to generate new samples. Identification of the k value may be challenging in *Borderline-MLSMOTE*. Further investigations on the imbalance of hierarchical data can be done on *Borderline-MLSMOTE*. The proposed sampling method is poor in identifying label correlations; additional work on the above issue could throw light on multi-label data.

**Acknowledgement:** This research was partly supported by the Technology Development Program of MSS (No. S3033853) and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A4A1031509).

**Funding Statement:** The authors received no specific funding for this study.

**Author Contributions:** P. K. A. Chitra and S. Geetha contributed in conceptualization and design of the study and supervising the research process, S. Appavu alias Balamurugan and S. Geetha drafted the introduction and discussion sections, S. Geetha, Seifedine Kadry and Jungeun Kim conducted literature review and sourced relevant studies, P. K. A. Chitra and S. Geetha involved in writing the methodology section and developing research instruments, Jungeun Kim and Keejun Han conducted experiments and field work, P. K. A. Chitra and S. Geetha involved in statistical analysis and interpretation of results, P. K. A. Chitr, S. Geetha and Seifedine Kadry reviewed and edited final

manuscript for submission. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** The data used in this research are open source and can be downloaded from <https://mulan.sourceforge.net/datasets-mlc.html> (accessed on 15 September 2022).

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

## References

- [1] M. L. Zhang and Z. H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007. doi: [10.1016/j.patcog.2006.12.019](https://doi.org/10.1016/j.patcog.2006.12.019).
- [2] S. Feng and D. Xu, "Transductive multi-instance multi-label learning algorithm with application to automatic image annotation," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 661–670, 2010.
- [3] Y. C. Chang, S. M. Chen, and C. J. Liau, "Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method," *Expert Syst. Appl.*, vol. 34, no. 3, pp. 1948–1953, 2008. doi: [10.1016/j.eswa.2007.02.037](https://doi.org/10.1016/j.eswa.2007.02.037).
- [4] R. M. M. Vallim, T. S. Duque, D. E. Goldberg, and A. C. Carvalho, "The multi-label OCS with a genetic algorithm for rule discovery: Implementation and first results," in *Proc. 11th Annu. Conf. Genetic Evol. Comput.*, 2009, pp. 1323–1330.
- [5] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. P. Vlahavas, "Multi-label classification of music into emotions," in *Proc. 9th Int. Conf. Music Inform. Retr.*, 2008, vol. 8, pp. 325–330.
- [6] W. Zhang, F. Liu, L. Luo, and J. Zhang, "Predicting drug side effects by multi-label learning and ensemble learning," *BMC Bioinform.*, vol. 16, no. 1, pp. 1–11, 2015.
- [7] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, 2009.
- [8] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, 2002.
- [9] R. Batuwita and V. Palade, "Class imbalance learning methods for support vector machines," *Imbalanced Learn.: Found., Algorithms, Appl.*, vol. 20, no. 3, pp. 83–99, 2013.
- [10] J. H. Xue and P. Hall, "Why does rebalancing class-unbalanced data improve AUC for linear discriminant analysis?," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 5, pp. 1109–1112, 2014.
- [11] M. L. Zhang, Y. K. Li, H. Yang, and X. Y. Liu, "Towards class-imbalance aware multi-label learning," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 4459–4471, 2020.
- [12] N. V. Chawla and J. Sylvester, "Exploiting diversity in ensembles: Improving the performance on unbalanced datasets," in *Int. Workshop Multiple Classif. Syst.*, 2007, pp. 397–406.
- [13] H. Han, W. Y. Wang, and B. H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *Int. Conf. Intell. Comput.*, 2005, pp. 878–887.
- [14] Q. Wang, Z. Luo, J. Huang, Y. Feng, and Z. Liu, "Novel ensemble method for imbalanced data learning: Bagging of extrapolation-SMOTE SVM," *Comput. Intell. Neurosci.*, vol. 2017, no. 3, pp. 1–11, 2017. doi: [10.1155/2017/1827016](https://doi.org/10.1155/2017/1827016).
- [15] R. C. Bhagat and S. S. Patil, "Enhanced SMOTE algorithm for classification of imbalanced big-data using random forest," in *2015 IEEE Int. Adv. Comput. Conf. (IACC)*, Bangalore, India, 2015, pp. 403–408.
- [16] Q. Gu, X. M. Wang, Z. Wu, B. Ningand, and C. S. Xin, "An improved SMOTE algorithm based on genetic algorithm for imbalanced data classification," *J. Digit. Inform. Manage.*, vol. 14, no. 2, pp. 92–103, 2016.
- [17] F. Charte, A. J. Rivera, M. J. delJesus, and F. Herrera, "MLSMOTE: Approaching imbalanced multi-label learning through synthetic instance generation," *Knowl. Based Syst.*, vol. 89, no. 1, pp. 385–397, 2015. doi: [10.1016/j.knosys.2015.07.019](https://doi.org/10.1016/j.knosys.2015.07.019).
- [18] R. Popli, I. Kansal, A. Garg, N. Goyal, and K. Garg, "Classification and recognition of online handwritten alphabets using machine learning methods," *IOP Conf. Series: Mat. Sci. Eng.*, 2021, vol. 1022, Art. no. 012111.

- [19] A. M. Mishra *et al.*, “A deep learning based novel approach for weed growth estimation,” *Intell. Autom. Soft Comput.*, vol. 31, no. 2, pp. 1157–1172, 2022.
- [20] S. Sharma, R. Mittal, and N. Goyal, “An assessment of machine learning and deep learning techniques with applications,” *ECS Trans.*, vol. 107, no. 1, pp. 8979–8988, 2022.
- [21] V. Verma *et al.*, “A deep learning based intelligent garbage detection system using an unmanned aerial vehicle,” *Symmetry*, vol. 14, no. 5, 2022, Art. no. 960.
- [22] F. Charte, A. Rivera, M. J. D. Jesus, and F. Herrera, “A first approach to deal with imbalance in multi-label datasets,” in *Int. Conf. Hybrid Artif. Intell. Syst.*, 2013, pp. 150–160.
- [23] K. Napierala and J. Stefanowski, “Types of minority class examples and their influence on learning classifiers from imbalanced data,” *J. Intell. Inform. Syst.*, vol. 46, no. 3, pp. 563–597, 2016. doi: [10.1007/s10844-015-0368-1](https://doi.org/10.1007/s10844-015-0368-1).
- [24] H. Zou and H. H. Zhang, “On the adaptive elastic-net with a diverging number of parameters,” *Ann. Stat.*, vol. 37, no. 4, pp. 1733–1751, 2009.
- [25] H. Liu and S. Zhang, “MLSLR: Multilabel learning via sparse logistic regression,” *Inf. Sci.*, vol. 281, no. 3, pp. 310–320, 2014. doi: [10.1016/j.ins.2014.05.013](https://doi.org/10.1016/j.ins.2014.05.013).
- [26] S. Shu, F. Lv, Y. Yan, L. Li, S. He, and J. He, “Incorporating multiple cluster centers for multi-label learning,” *Inf. Sci.*, vol. 590, no. 8, pp. 60–73, 2022. doi: [10.1016/j.ins.2021.12.104](https://doi.org/10.1016/j.ins.2021.12.104).
- [27] G. Wu, R. Zheng, Y. Tian, and D. Liu, “Joint ranking SVM and binary relevance with robust low-rank learning for multi-label classification,” *Neural Netw.*, vol. 122, no. 3, pp. 24–39, 2020.
- [28] Q. W. Zhang, Y. Zhong, and M. L. Zhang, “Feature-induced labeling information enrichment for multi-label learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32.