IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received December 6, 2019, accepted December 18, 2019, date of publication December 25, 2019,
date of current version January 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962201

Lattice and Imbalance Informed Multi-Label Learning

PAYEL SADHUKHAN[1] **AND SARBANI PALIT**[2]
[1]Machine Intelligence Unit (MIU), Indian Statistical Institute, Kolkata 700108, India
[2]Computer Vision and Pattern Recognition (CVPR), Indian Statistical Institute, Kolkata 700108, India

Corresponding author: Payel Sadhukhan (payel0410@gmail.com)

ABSTRACT In a multi-label dataset, an instance is given a single representation across all possible labels. Despite the mutual sharing of instances among the labels, the membership of the instances vary from label to label. This diversifies the intrinsic class geometries of the labels. Multi-label datasets are often found to be class-imbalanced as well. The varying membership of the instances coupled with the imbalance phenomenon gives rise to varying imbalance ratios across the labels. We address these two key aspects in this work, Lattice and Imbalance Informed Multi-label Learning (LIIML) in a two step procedure. Firstly, we obtain the imbalance ratios and the intrinsic positive and negative class lattices of each label. We capitalize on these two information to obtain a dedicated feature set for each label. In the second step, to handle the class-imbalance further, we employ a scheme of imbalance-adaptive misclassification cost across the labels. We have evaluated the competence of the proposed method in a generic as well as class-imbalanced framework. The elaborate empirical study establishes the competence of the proposed method in both the contexts.

INDEX TERMS Class imbalance learning, multi-label learning, relative neighborhood graph, cost sensitive learning, classification.

I. INTRODUCTION

Contemporary datasets differ from the class of traditional datasets in a number of aspects – multi-label nature of the data being one of them. In multi-label datasets a single instance in a given input space can belong to one or more of the possible class labels. The need for efficient processing of multi-label data is backed by the availability of datasets with multi-label characteristics from several real-world applications. Beginning with text categorization by [1] and [2], data with multi-label characteristics have emerged from different genres namely images [3], [4], music [5], bioinformatics [6], chemical data analysis [7], tag recommendation systems [8] and video [9]. Consequently, multi-label classification and learning grabbed the attention of the data science community. Let a multi-label dataset be denoted by $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{Y}_i), i = 1, 2, \ldots, n\}$ and the label set cardinality be $\mathcal{L}$. $\mathcal{Y}_i = \{y_{i1}, y_{i2}, \ldots, y_{iL}\}$. Let us assume that each label has exactly two classes positive (1) and negative (0) that is $\mathcal{Y}_{ij}$ can be either 1 or 0, $j = 1, 2, \ldots, \mathcal{L}$. An instance $\mathbf{x}_i$ has to be rightfully classified into either positive (1) or negative (0) class for $\mathcal{L}$ labels.

In a multi-label dataset, a single set of instances though sharing a same representation across the labels can belong to different classes for different labels. This leads to a variable class partition of the same instance set across different labels. To tackle this aspect of multi-label datasets, selecting dedicated and discriminating features for different labels has been a popular and choice. A number of works has been done following this paradigm whose details can be found in [10].

Class imbalance is the quantitative disproportion between the number of instances belonging to the possible classes of a dataset. For binary classification problems, the class with higher and lesser share of instances are called majority class and minority class respectively. Imbalance ratio is the ratio of number of instances in the majority class to the number of instances in the minority class. Let us assume that we have exactly two classes - positive (1) and negative (0) for each label. In multi-label datasets, the positive class is under-represented for most of the labels in a dataset. This issue is further compounded by the varying class membership of the instances across different labels. A natural outcome is

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

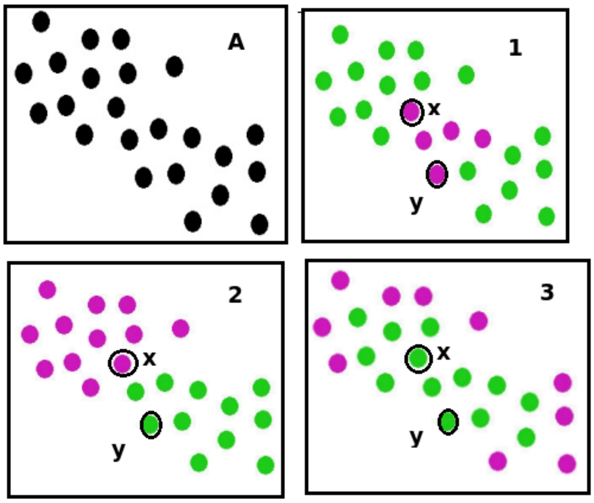


FIGURE 1. This illustration depicts the phenomenon of variable class geometries and varying imbalance ratios in a toy multi-label dataset. The dataset comprises of 23 two-dimensional points. Fig A shows the set of feature points. We assume this dataset to have 3 binary labels - 1, 2 and 3. Each instance can belong to the positive or the negative class with respect to labels 1, 2 and 3 individually. We have used pink and green color to represent positive and negative class-memberships of a point. Figures 1, 2 and 3 shows the classification of the given instances with respect to labels 1, 2 and 3 in order. These figures show that the membership of these instances vary from label to label. For example, instance marked x is positive for labels 1 and 2 while negative for label 3. On the other hand, instance y is positive with respect to label 1 only. The consequence of this varying membership is two-fold. Firstly, as figures 1, 2 and 3 indicate, we get varying positive and negative class geometries for different labels. We may also note that out of the 23 given points, labels 1, 2 and 3 have 5, 12 and 10 positive points respectively. Accordingly, the their imbalances are 3.6, 0.92 and 1.3 in order. So, we get a set of varying imbalance ratios across different labels by virtue of the differential class membership of the instances. Labels 1 and 3 have somewhat similar imbalance ratios (imbalance ratio of label 1 and 3 are 0.92 and 1.3). Despite that, the class geometries of these two labels are fairly diverse. These observations serve as the motivation of this work. We address the issue of differential class geometry as well as varying imbalance ratio to have a fruitful learning of multi-label datasets.

differential degree of class imbalance for different labels. For example, in yeast dataset [11] (with 14 labels), the minimum imbalance ratio and maximum ratio is 1.32 (for label 12) and 50.74 (for label 14) respectively. A single framework with a single set of parameters may not work well across the two diversified labels. Figure 1 illustrates this phenomenon on a toy multi-label dataset.

In this work, we employ feature extraction followed by an imbalance-adaptive cost sensitive classification to learn the multi-label datasets. We propose that while handling class-imbalance of a dataset we should not overlook or distort its original class geometry. A standard solution of handling the class-imbalance problem is by undersampling the majority class or oversampling the minority class. These two techniques modify the set of representaive points of a dataset and can lead to the distortion of the it's original class-geometries. We propose a two-step procedure for obtaining a geometry preserving and imbalance-aware multi-label learner. In the first part of our work, we extract a dedicated feature set from the intrinsic class lattice of the labels. We obtain the overall structure of a dataset from its Relative Neighborhood Graph (RNG). Next, we detect the regions of homogeneous class memberships of RNG and select the label specific lattice points from those. To preserve the original class geometry, we select a differing number of positive and negative lattice points for the labels. When we have significant difference in the class cardinalities of positive and negative classes, selecting equal number of lattice points from both may not preserve the class geometries. It can lead to a distorted representation which in turn may affect the learning. For a label which is well balanced across positive and negative classes, selecting equal number of lattice points for both classes can work well. But for an imbalanced label (with more number of negative points), we have to select more number of negative lattice points than that of positives for proper representation. The ratio of the number of negative lattices to that of the positive lattices for a label is dependent on its imbalance ratio. This helps us preserving the original class geometries. Next, as in LIFT [12], we compute a distance based feature extraction for the points. The extracted features are used to model the set of classifiers (one for each label) and predict the test data.

Class-imbalance is a fundamental feature and issue of multi-label datasets. Cost-sensitive learning [13] was one of primal techniques for tackling the issue of class-imbalance and consequently detecting the 'hard-to-learn' positive or minority instances. To nullify the natural bias of the classifier towards the quantitatively abundant majority class, a higher mis-classification penalty is set for the quantitatively scare minority class. The main goal is to bias the classifier towards identifying the minority samples.

In the second part of our work, to address differential class-imbalance further, we adopt a cost-sensitive learning scheme where the misclassification cost is adaptive to the imbalance ratio of the labels. As said earlier, a multi-label dataset has differing values of imbalance ratios across the labels. In such a situation, selecting a single misclassification cost for the minority class across will not yield proper learning. Instead, we select a set of misclassification cost values of the minority (positive) class, one for each label. Between two labels with differing degree of imbalance, we set a higher misclassification penalty for the one with higher imbalance than that of the other.

We summarize the contributions of this paper as follows.

- We propose a scheme which works on two perspectives of multi-label learning –i] dedicated feature extraction and ii] handling differential class-imbalance.
- For feature extraction, intrinsic class geometries of the labels are explored. The concept of Relative Neighborhood Graph is used for capturing the class geometries.
- To tackle the differential class-imbalance ratios of the labels, we adopt a simple yet effective imbalance-adaptive misclassification penalty across the labels.
- The efficaciousness of the feature extraction scheme of LIIML is demonstrated empirically on 11 real-world multi-label datasets against generic multi-label learners. It indicates the competitiveness of the proposed scheme.

- From the perspective of class-imbalance, the proposed imbalance-adaptive misclassification cost scheme has given remarkable improvement in multi-label performance. In the empirical study, we have compared with class-imbalance dedicated multi-label learner COCOA, RNNOML and three more generic multi-label learners in addition to class-imbalance learners SMOTE, USAM and RML.
- The imbalance-adaptive misclassification cost's effectiveness is also demonstrated on two extant first-order works, namely Binary Relevance [14] and LIFT [12].

In the next section, we present the literature review. In Section 3 and 4, we present the approach and algorithm of our work respectively. Section 5 and Section 6 present the experimental study and the experimental results respectively. The article is wrapped by the Conclusions in Section 7.

II. RELATED WORK

Multi-label learning methods are broadly classified into two types –*Problem transformation approach (PT)* and *Algorithm Adaptation Approach (AA)* [15]. On the other hand, studies such as the one in [16] differentiate the multi-label learners into three groups, namely *Problem transformation*, *Algorithm Adaptation* and *Ensemble of multi-label classifiers*.

Problem transformation approaches modify or decompose a multi-label dataset to fit it in a framework of regular decomposition. Depending on the number of decompositions and number of labels involved in a classifier, this class of learners are further sub-divided into *first order*, *second order* and *higher order* paradigms [17]. In first-order PT, only one label is involved in a classifier while for second-order and higher-order approaches, two and more number of labels are involved in a classifier respectively. Notable problem transformation approaches are namely Binary Relevance [14], power set of labels [3], pruned problem transformation [18] and calibrated label ranking [19]. Binary relevance, (BR) is the most primitive form of *PT* approach, where a series of binary classifiers is generated, one for each label. Though computationally sound (linear with label cardinality), BR is criticized for its inability to capture label correlations [20]. The solution proposed in [3] accommodated label correlations by employing Label Powerset. It generated $2^8 = 256$ classifiers for learning 8 labels. Despite involving label correlations, the scheme lacked computational feasibility as it generated an exponential number of classifiers. A more feasible approach was given in RAKEL [21], which considered random subsets of labels. Calibrated Label ranking [19] scheme provided multi-label outputs on the basis of pair-wise classification, considering a synthetic label to distinguish the relevant and irrelevant groups of labels. Ensemble of classifiers like RAKEL [21], ensembles of classifier chains [22]–[24] and ensembles of pruned sets are also popular and effective in learning multi-label datasets. In addition to these, a number of feature selection and extraction methods transform the features in context of each label and follow first-order approach to complete the learning. They are discussed in the detail in the next paragraph. In Algorithm-Adaptation approach, an existing classifier is adapted in the context of multi-label scenario. Quite a number of classifier paradigms like k-nearest neighborhood classifier [25], naive bayes algorithm [26], back-propagation of neural networks [27] are adapted to facilitate multi-label learning. In [25], the k-nearest neighbors of a test point are identified. Following that, their label configurations and the principles of maximum posteriori are used to determine the label predictions of the test instance. In [27], the usual back-propagation algorithm is used with small modifications to accommodate the multi-label characteristics. The error function of the back-propagation algorithm is replaced with a ranking loss minimization function which operates on the fact that a relevant label of an instance is ranked higher than another label to which the instance dose not belong. Another scheme [28] uses the cross-entropy error function in back-propagation neural network for facilitating multi-label learning. Table 1 outlines the basic principles of a number of state-of-the-art multi-label methods.

Apart from the above, multi-label datasets are analyzed from newer perspectives like feature or dataset preprocessing and class distribution of the labels. Label specific feature extraction was proposed in LIFT [12]. In LIFT, following the clustering of the positive and negative classes of each label, the authors extract a label-specific feature set. Feature selection is also done by a number of works on the basis of class characteristics of the labels. Works dealing with feature extraction and selection of multi-label datasets include [29] and [30]. A detailed account and comparative analysis of the extant works in multi-label feature extraction and selection in first-order framework can be found in [10]. Joint feature selection and classification (JFSC) [31] and [32] performs label-correlated feature selection of multi-label datasets. Class distribution of the various of a multi-label dataset is a probable data mine and gives a number of pertinent information. As said earlier, multi-label datasets are class-imbalanced and they are differently imbalanced also. COCOA [33] has addressed the imbalance issue in their work by considering an ensemble of classifiers using pair-wise label correlation. A few more works, [34] and [35] have used cost-adaptive paradigm to address multi-label problems. In [36], authors have integrated the data gravitational model with multi-label lazy learner for improving the minority class performances of imbalanced multi-label datasets. In our very recent work [37], we have used a reverse-nearest neighborhood oversampling to curb the problem of differential imbalance in multi-label datasets. Several other techniques like convex relaxation [38], ensembles of random graph [39] and graph classification [40] are also employed to address multi-label classification.

The proposed method LIIML is a hybrid method which involves i]. a dedicated feature-extraction for the labels (like LIFT [12] and JFSC [31]) and also ii]. adapts the cost-sensitive learning in multi-label context. From first

TABLE 1. This table describes the approaches of state-of-the-earth multi-label learners BR, RAKEL, CLR, MLKNN, Naive Bayes and BPMLL. The last column outlines the working principles of the proposed method, LIIML.

Binary Relevance	RAKEL	CLR	MLKNN	Naive Bayes	BPMLL	**LIIML**
Operates on a single-label framework. Decomposes a n-label problem into single label binary sub-problems.	It makes random, smaller and overlapping partitions of the label set. It explores label correlation by considering powerset of label subsets.	It incorporates pair-wise correlation of the labels and uses a synthetic label to separate the relevant and irrelevant labels.	Label configuration of the neighboring instances is integrated with MAP for predicting the label set of the unknown instance.	Naive bayes algorithm is used in context of multi-label datasets. To mitigate the 'class conditional independence' assumption, a PCA+GA based features selection is also done.	It incorporates Back propagation algorithm of neural networks in multi-label context by using a novel error function. Its value is modulated on the information that a relevant label is ranked higher than that of an irrelevant one for an instance.	**Firstly, implements the Relative Neighborhood Graph of the instance set and extracts a label-specific feature set. Secondly, uses cost-sensitive learning paradigm to handle differential class-imbalance problem of multi-label datasets.**

perspective, LIIML is a first-order PT method and from the latter it is an algorithm-adaptation method.

III. APPROACH

A. MULTI-LABEL NATURE OF DATA, ITS CONSEQUENCES AND OUR THOUGHTS

A multi-label dataset is characterized by the membership of a set of feature points to more than one label. A class-imbalanced dataset is typified by the quantitative disproportion in the number of instances representing its classes. Multi-label datasets are often found to be class-imbalanced. In this work, we deal with binary multi-label dataset where each label can take exactly one of the two classes (1 - positive class and 0 - negative class). Typically, class 1 and class 0 are the minority and majority classes respectively. The class membership of each instance varies from label to label. An instance which is positive for some label A can be negative for some other label B. A similar phenomenon for all the instances will lead to a different combinations of positive and negative sets for each label (even though the union of the positive and negative set of instances is same for all labels). From spatial perspective, this leads to variable positive and negative class geometries and variable class boundaries for the labels. Figure 1 illustrates this phenomenon. The quantitative consequence of this phenomenon is the variable degree of class imbalance across the labels. The key idea of this work is to design an imbalance-informed scheme which also takes into account the differential class geometries of different labels. For each label, we will extract an imbalance-informed feature set from the positive and negative class geometries of that label. The learning is wrapped up with a set of cost-sensitive, first-order learners, one for each label.

B. EXTRACTING THE CLASS GEOMETRIES OF LABELS

Our first goal is to extract the positive and negative class geometry of each label. To perceive the geometry, we generate a Relative Neighborhood Graph (RNG) of the entire set of training dataset where the edge weights are the distance between the points. RNG shows the connectivity of a data point or vertex to its adjacent neighborhood and the interconnectivity the points gives the overall anatomy of the feature points. A RNG of G is an undirected graph defined from G where $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected whenever there is no third point $\mathbf{x}_k$ such that $d(\mathbf{x}_i, \mathbf{x}_k) < d(\mathbf{x}_i, \mathbf{x}_j)$ and $d(\mathbf{x}_j, \mathbf{x}_k) < d(\mathbf{x}_i, \mathbf{x}_j)$. For a given set of points, it's MST is a subgraph of it's RNG. We may note that this Tree will be same for all the labels. But the membership of the vertexes or the data points vary from label to label and leads to a differential positive and negative class structures for the labels. Let $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be the training data and $\mathcal{Y}_i = \{y_{i1}, \ldots, y_{i\mathrm{L}}\}$ be the class label vector associated with instance $\mathbf{x}_i$. We have assumed that there are $\mathcal{L}$ labels in the dataset.

We will extract the positive and negative class geometries (with respect to each label) from the RNG. To extract the above-said, we need to look at the membership of the vertexes to each label. For a label, the membership of a vertex can be either positive (1) - if it belongs to that label or negative (0) - if it does not belong to that label. The class-memberships of the data points will likely vary across the labels.

Let us consider an edge e_{ij} between two vertices, v_i and v_j. If the class-membership of $\mathbf{x}_i$ and $\mathbf{x}_j$ with respect to label k are same (both 0 or both 1), we term edge e_{ij} as a homogeneous edge. If y_{ik} and y_{jk} (the memberships of $\mathbf{x}_i$ and $\mathbf{x}_j$) are both 1, we term e_{ik} to be a positive homogeneous edge. If the class-memberships (y_{ik} and y_{jk}) are 0, we call it negative homogeneous edge. So, for each label, we will have a set of homogeneous edges which is a subset of the RNG edges. We can further partition this homogeneous edge set into two mutually exclusive sets of positive homogeneous edges and negative homogeneous edges. For each label, we will have a set of positive homogeneous edges and a set of negative homogeneous edges which is described in the next paragraph. We will extract the positive and negative class lattices of the label from its respective sets of homogeneous edges. Homogeneous edges lie in the regions of same class memberships.

A homogeneous edge (belonging to a certain class) with smaller weight will likely be a better representative of that class than another with higher weight. It is because, with increasing edge weight, the vertexes (associated with the edge) become sparser in the feature space and eventually overlap with the vertexes associated with a different class. But a vertex associated with a shorter edge will has another vertex near its vicinity which affirms its class-membership. Hence, to get the positive class lattice (for a label), we arrange the positive homogeneous edges in increasing order of their weights. For a certain label k, we select a N_{Pk} number of positive homogeneous edges in increasing order of their weights and compute their midpoints. The set of N_{Pk} midpoints represents the positive lattice of label k. Similarly, we compute N_{Nk} lattice points to represent the negative lattice of label k. In case of extreme imbalance when we do not get any positive homogeneous edge, we select the positive points themselves to represent the positive lattices.

We determine the values of N_{Pk} and N_{Nk} in light of the degree of class-imbalance of label k. Let the degree of imbalance of label k be imb_k. For the negative class (generally the majority class) of label k, we select the value of N_{Nk} as N_{Pk}*$(\log_2(\text{imb}_k) + 1)$. The logarithm function allows us to add deviations in the figures in a controlled manner. Let us consider two scenarios to analyse the aspect. If there is no imbalance in two classes of label k, that is $\text{imb}_k = 1$, $(\log_2(\text{imb}_k)+1)$ value will be 1 and we will select N_{Pk} points as the negative class. On the contrary, if imb_k is 16 (dataset is highly imbalanced with respect to label k), $(\log_2(\text{imb}_k) + 1)$ will be 5 and we will select $5\times$ N_{Pk} points to represent the negative class. We can also select different figures of N_{Pk} and N_{Nk}. We present a discussion in Remarks 1 at the end of this section.

C. EXTRACTING THE FEATURES

Now, we extract the features for each label. For that, we obtain the distance of a data point from the sets of positive and negative lattice points of a label. In order to make the positive information stand out in a pool of negative data, we multiply the distances from the positive lattices with the respective class imbalance ratio of that label. The above computed distances gives the imbalance-informed mapping of a data point for that label. The set of N_{Pk} positive distances an N_{Nk} negative distances give the transformed mapping of $\mathbf{x}$ with respect to label k.

D. HANDLING IMBALANCE FURTHER - COST SENSITIVE CLASSIFICATION

Cost-sensitive learning is one of the ways of handling imbalanced data. As stated earlier, in multi-label datasets, the degree of imbalance varies across labels. After generating the imbalance-informed representations for each data point, we proceed with a cost-sensitive linear SVM based classification. Let the misclassification cost of a minority instance to the majority class for label k be denoted by Cost_k. To improve the detection of minority class (generally the positive class), for label j, Cost_k value is fixed to $cf \times (\log_2(\text{imb}_k) + 1)$. cf is a cost-factor whose increasing value gives increasing misclassification cost for the minority class. Remark 5 discusses the details on choice of cf. The misclassification cost of a majority instance to the minority class is set to 1 for all labels. The misclassification cost Cost_k value increases with increase in imbalance value of a label and is adaptive to various and diversified ranges of imbalance values in a single dataset. For a label which has no imbalance or the imbalance ratio is 1, the misclassification costs of both classes (no class is minority or majority to be precise) is 1. For an imbalanced label with $\text{Cost}_k > 1$, the misclassification cost of the minority instances to the majority class is greater than 1 and it increases with increase in imbalance value. Hence, we have an imbalance-informed misclassification cost for each label. The $\log_2$ function allows us restrict the misclassification costs within an admissible yet varying limit depending on the imbalance ratios.

Remarks:

1) **Values of N_{Pk} and N_{Nk}:** The number of lattice points for the positive class and the negative class are given by N_{Pk} and N_{Nk} respectively. The feature set cardinality in the extracted feature set will increase with the increase in the number of lattice points. Increasing the number of lattice points will give better discernible and classification capabilities of the learner. But this is accompanied with an increase in computational complexity. While setting the values of N_{Pk} and N_{Nk}, we have to make a trade-off between complexity and performance. Experiment 4 in the empirical study explores this aspect.
2) **Distance function used:** We have used Euclidean distance and Jaccard distance functions for numeric and nominal datasets respectively.
3) **Learning with cost:** The scheme that we have presented here can be carried out in a equal misclassification cost framework as well as in an enhanced cost (for misclassifying minority instances) framework, the latter presented in subsection 3.4. The equal misclassification cost variant shows the intrinsic capability of the scheme. The enhanced cost variant helps us in handling the class-imbalance of the labels better. We report the outcomes of both the variants to investigate the efficaciousness of our method in both contexts. The utility of the enhanced cost can be also be investigated on existing multi-label learners like LIFT and BR. We have presented a study on this aspect.
4) **Choice of cf:** In experiment 4, we have explored the choice of cf value. Increase in cf value gives increased misclassification cost for the minority class.

IV. ALGORITHM

Let the multi-label dataset be denoted by D and the number of class labels for $\mathcal{D}$ be $\mathcal{L}$. $\mathcal{D} = \{(\mathbf{x}_i, \mathcal{Y}_i), |1 \leq i \leq n, \mathcal{Y}_i$ denotes class label vector of $\mathbf{x}_i\}$.

$\mathcal{Y}_\mathrm{i} = \{y_{i1}, y_{i2}, \ldots, y_{il}\}$. y_{ij} is 1 when label j is positive for instance $\mathbf{x}_i$, otherwise the value of y_{ij} is 0. Let each $\mathbf{x}_\mathrm{i} \in \mathbf{R}^\mathbf{p}$. We randomly equi-partition $\mathcal{D}$ into a training set, $\mathcal{D}_{tr}$ and a test set, $\mathcal{D}_{te}$. Let $\mathcal{X}$ be the set of training instances (without the label information).

$$\mathcal{X} = \{\mathbf{x}_i, \text{i=1, 2,}\ldots, \text{n}\} \tag{1}$$

We calculate class-imbalance ratio of each label j, $j = 1, 2, \ldots, l$ denoted by imb_j.

$$\begin{aligned} \text{imb}_\text{j} &= \frac{\text{Number of negative training instances for label } j}{\text{Number of positive training instances for label } j} \\ \Rightarrow \text{imb}_\text{j} &= \frac{||\{\mathbf{x}_i \text{ such that } \mathcal{Y}_{ij} = 0, i = 1, 2, \ldots, n\}||}{||\{\mathbf{x}_i \text{ such that } \mathcal{Y}_{ij} = 1, i = 1, 2, \ldots, n\}||} \end{aligned} \tag{2}$$

In multi-label datasets, we have a single set of observations covering all labels. We construct a Relative Neighborhood Graph (RNG) whose vertexes are represented by the members of $\mathcal{X}$.

$$\text{Tree} = \text{RNG}(\mathcal{X}) \tag{3}$$

To extract more refined information about the class structures, we have to extract a label-specific lattice from these graphs. Firstly, we extract the homogeneous edges of the graph. As explained earlier, homogeneous edge is one whose both vertexes belong to the same class, there are two classes of homogeneous edges, positive and negative.

Let $\mathbf{x}_i$ denotes the i^{th} vertex of the graph and $c_j(\mathbf{x}_i)$ denotes the class-membership of $\mathbf{x}_i$ to label j.

$$c_j(\mathbf{x}_i) = \begin{cases} 1, & \text{if } y_{ij} = 1 \\ 0, & \text{else} \end{cases} \tag{4}$$

Let an edge of *Tree* between two vertices $\mathbf{x}_i$ and $\mathbf{x}_k$ be represented by e_{ik}. w_{ik} denotes the edge-weight of e_{ik}.

$$e_{ik} = \{(\mathbf{x}_i, \mathbf{x}_k), w_{ik}\}, i, \quad k = 1, 2, \ldots, n, \ i \neq k \tag{5}$$

$\mathbf{S}_{pj}$ and $\mathbf{S}_{nj}$ are the sets of positive and negative homogeneous edges of label j respectively.

For each label $j, j = 1, 2, \ldots, \mathcal{L}$,

$$\mathbf{S}_{pj} = \{e_{ik}, c_j(\mathbf{x}_i) = c_j(\mathbf{x}_k) = 1\} \tag{6}$$

Similarly,

$$\mathbf{S}_{nj} = \{e_{ik}, c_j(\mathbf{x}_i) = c_j(\mathbf{x}_k) = 0\} \tag{7}$$

We arrange the elements of $\mathbf{S}_{pj}$ and $\mathbf{S}_{nj}$ in increasing order of their edge-weights to get the ranks of their respective elements. Let, for an edge e_{ik}, its rank in its respective set (the set where it belongs) be denoted by $R(e_{ik})$. We obtain the ranks of the edges because we will select the lattice points from the shorter homogeneous edge weights. Shorter homogeneous edges have lower ranks than longer edges.

The mid-points of edges in $\mathbf{S}_{pj}$ are stored in $\mathbf{M}_{pj}$. $\mathbf{M}_{nj}$ stores the mid-points of $\mathbf{M}_{nj}$. Let N_{Pj} and N_{Nj} denote the number of negative and positive lattice points of label j respectively.

$$\mathbf{M}_{pj} = \bigcup_{\substack{e_{ik} \in \mathbf{S}_{pj} \\ R(e_{ik}) \leq N_{pj}}} \frac{\mathbf{x}_i + \mathbf{x}_k}{2} \tag{8}$$

$$\mathbf{M}_{nj} = \bigcup_{\substack{e_{ik} \in \mathbf{S}_{nj} \\ R(e_{ik}) \leq N_{nj}}} \frac{\mathbf{x}_i + \mathbf{x}_k}{2} \tag{9}$$

Let the representations of $\mathbf{M}_{pj}$ and $\mathbf{M}_{nj}$ be as follows.

$$\mathbf{M}_{pj} = \{\mathbf{m}_{1j}, \ \mathbf{m}_{2j}, \ldots, \ \mathbf{m}_{k_{pj}}\} \tag{10}$$

$$\mathbf{M}_{nj} = \{\mathbf{m}'_{1j}, \ \mathbf{m}'_{2j}, \ldots, \ \mathbf{m}'_{k_{nj}}\} \tag{11}$$

$\mathbf{m}_{1j}, \ \mathbf{m}_{2j}, \ldots, \ \mathbf{m}_{k_{pj}}$ represent the individual members of $\mathbf{M}_{pj}$.

Similarly, $\mathbf{m}'_{1j}, \ \mathbf{m}'_{2j}, \ldots, \ \mathbf{m}'_{k_{nj}}$ represent the elements of $\mathbf{M}_{nj}$. It is easy to note that the number of elements of $\mathbf{M}_{pj}$ and $\mathbf{M}_{nj}$ depends on data distribution and are likely unequal. We have represented their cardinalities with k_p and k_n respectively.

The transformed mapping of instance $\mathbf{x}_i$ with respect to label j denoted by $\mathbf{z}_{ij}$ is as follows:

$$\mathbf{z}_{ij} = f_j(\mathbf{x}_i) = \{d(\mathbf{x}_i, \mathbf{m}_{1j}), \ldots, d(\mathbf{x}_i, \mathbf{m}_{k_{pj}}), d(\mathbf{x}_i, \mathbf{m}'_{1j}), \ldots, d(\mathbf{x}_i, \mathbf{m}'_{k_{nj}})\} \tag{12}$$

$\mathbf{z}_{ij}$ is a k_p+k_n dimensional vector or feature. Its first k_p components are generated by taking distance from the midpoints of the positive homogeneous edges and multiplying them with the imbalance ratio of label j. The remaining k_n components by taking distance from the negative homogeneous edges.

Let $\mathcal{Z}_j = \{\mathbf{z}_{ij}, i = 1, 2, \ldots, n\}$. $\mathcal{Z}_j$ represents the transformed feature mapping of the training instances in $\mathcal{D}_{tr}$ for label j.

Let Min and Maj be the minority and majority classes of a label respectively. Let $\text{Cost}_j(\text{Min, Maj})$ and $\text{Cost}_j(\text{Maj, Min})$ denote the misclassification costs of a minority instance to the majority class for label j and vice versa. For each label, $\text{Cost}_j(\text{Min, Maj})$ is equal to the product of a cost factor (cf) and logarithm of the it's imbalance ratio. In this work, we have fixed the value of cf to 1.

For each label j,

$$\text{Cost}_j(\text{Min, Maj}) = \text{cf} \times \max((\log_2(\text{imb}_j) + 1), 1) \tag{13}$$

$$\text{Cost}_j(\text{Maj, Min}) = 1, \quad i = 1, 2, \ldots, n \tag{14}$$

For each label j, we train a learner $\mathcal{W}_j$ by invoking $\mathcal{Z}_j$ and the above defined cost function for label j. For classifying a test instance $\mathbf{t}$ with respect to label j, we first obtain its transformed mapping for label j and invoke $\mathcal{W}_j$ to predict it's class. We have used linear SVM classifier implementation of LIBSVM ([41]) for modeling and classification.

TABLE 2. **Description of Datasets. This table is universal for all experiments in this work. *D* the number of instances for all experiments. *L.Card* and *L.Uniq* gives the values of average number of instances per label and the number of unique label combinations respectively. att.type gives the information about nominal or numeric nature of the features. The number of labels and features associated with these datasets in Experiment 1 are denoted by *L* and *F* respectively. *L.Card* and *L.Uniq* represents label cardinality and number of unique labels in regular setting. (*L.Card*)$_I$ and (*L.Uniq*)$_I$ denotes these two values in the class-imbalanced framework (Experiment 2, 3, 4). L_I, F_I gives the number of labels and features of these datasets in Experiment 2, 3 and 4 (experiment on imbalance). min IR, max IR and avg IR gives the minimum imbalance ratio, maximum imbalance ratio and average imbalance ratio associated with the labels of the datasets with respect to the label information of L_I.**

Dataset	domain	att.type	D	L / L_I	F/ F_I	$L.Card$/$(L.Card)_I$	$L.Uniq$/$(L.Uniq)_I$	min IR	max IR	avg IR
Corel5k	image	nominal	5000	374/ 44	499/ 499	3.522/ 2.214	3175/ 1037	3.46	50.00	17.86
Enron	text	nominal	1702	53/ 24	1001/ 50	3.378/ 3.113	753/ 547	1.00	43.48	5.35
medical	bio-NLP	nominal	978	45/ 14	1449/ 144	1.245/ 1.075	94/ 42	2.67	43.48	11.24
Slashdot	text	nominal	3782	22/ 14	1079/ 53	1.181/ 1.134	156/ 118	5.46	35.71	10.99
Tmc2007	text	nominal	28596	22/ 15	500/ 500	2.158/ 2.100	1341/ 637	1.45	34.48	5.85
CAL500	music	numeric	502	174/ 24	68/ 68	26.044/ 25.058	502/ 502	1.04	24.39	3.85
RCV1 Subset1	text	numeric	6000	101/ 42	472/ 472	2.880/ 2.458	1028/ 574	3.34	50.00	15.15
RCV1 Subset2	text	numeric	6000	101/ 39	472/ 472	2.634/ 2.170	954/ 489	3.22	47.62	15.87
Emotions	music	numeric	592	6/ 6	72/ 72	1.869/ 1.869	27/ 27	1.25	3.00	2.15
Scene	image	numeric	2407	6/ 6	294/ 294	1.047/ 1.074	15/ 15	3.52	5.62	5.56
Yeast	biology	numeric	2417	14/ 13	103/ 103	4.237/ 4.233	198/ 189	1.32	12.50	2.78

A. COMPLEXITY ANALYSIS

We analyse the complexity of the proposed scheme of feature extraction and class-imbalance handling separately below.

- **Feature Extraction:** We compute the Relative Neighborhood Graph of the given set of points. For N given points, the complexity for RNG formation is $\mathcal{O}(N\log N)$. Let the number of labels be L. The complexity for computing the lattice points for L labels is $\mathcal{O}(N \cdot L)$. For L labels, extraction of features from the lattice points require operations of order $\mathcal{O}(N \cdot L)$. So, the overall complexity of feature extraction for N points and L labels is $\mathcal{O}(N\log N)$ (if $L \leq \log N$) or $\mathcal{O}(N \cdot L)$ (if $L > \log N$).
- **Class imbalance handling:** In our method, we add a dedicated misclassification cost for each label. We set the misclassification cost according to the imbalance ratios of the labels. For L labels and N points, we calculate the misclassification ratios by going through the class labels of N points just once. Hence, for N points and L labels the complexity for calculating the class imbalance ratio and misclassification cost is $\mathcal{O}(N \cdot L)$.

V. EXPERIMENTAL SETUP

In this section, we have presented a detailed empirical study where four sets of experiments are carried out. Motivation of each experiment and its experimental layout are presented in the next three subsections.

A. FIRST EXPERIMENT: FEATURE EXTRACTION

In the first experiment, we demonstrate the relative competencies of the proposed and compared methods in a generalized multi-label framework. Eleven regular multi-label datasets are used. The detailed statistics of the datasets is given in Table 2. These datasets are obtained from MULAN [42] and MEKA [43] repositories.

For the comparative analysis, we have considered five multi-label learners from different genres.

- *Binary Relevance (BR) ([14]):* It is a first-order approach which considers one classifier for each label. Basically we transform the multi-label classification problem into L binary classification problems for L labels.
- *Calibrated Label Ranking (CLR) ([19]):* It is second-order approach which considers pairwise correlation of labels. It also considers a synthetic label to distinguish the set of relevant and irrelevant labels of the instances.
- *Random k-Labelsets (RAKEL) ([21]):* A higher order approach, which considers a number of subsets of labels and learns the full correlations within the subsets. We have considered the overlapped version of RAKEL as it considers more number of subsets and captures greater degree of correlation among labels. We have used paper recommended settings of $k = 3$ and number of subsets $m = 2q$.
- *Ensembles of Classifier Chains [24]:* It is a higher order approach which uses binary classifiers for each label. Label correlation is facilitated by the inclusion of predictions of preceding labels into the succeeding ones. Ensembles with randomized label order is considered to distribute the learning of correlations. We have considered ensemble size 100.
- *Multi-label learning with label specific features (LIFT) ([12]):* This work is based on a feature extraction scheme, where a dedicated set of features is learned for each label. The label-specific features are used to invoke L binary classifiers, one for each label. As recommended in the paper, r value is set to 0.1.

In this experiment, we have considered equal misclassification cost of minority and majority classes in LIIML to test the inherent efficacy of the proposed method. We have taken the number of positive lattice points to be 100 and varied the cardinality of negative lattice set according the imbalance of the labels.

Evaluating metrics: Six metrics namely *Hamming Loss, Coverage, One Error, Ranking Loss, Average Precision* and *Macro-averaging AUC* are employed to evaluate the relative efficacies of the comparing and proposed methods. Let $\mathbf{x}_i$, i $=$ 1,2,…,N be the set of N test instances and $\mathcal{Y}_i$ be the L-dimensional label vector of $\mathbf{x}_i$. Let $\overline{\mathcal{Y}_i}$ be the complement label set of $\mathbf{x}_i$. Let α_i be the label prediction vector for $\mathbf{x}_i$. We denote the label specific predicted score of $\mathbf{x}_i$ for label j by $f_j(\mathbf{x}_i)$.

- **Hamming Loss:** It measures the fraction incorrect predictions for all instances across the entire label set. Lower the value achieved by a learner, better is its performance.

$$\text{Hamming Loss} = \frac{1}{\text{NL}} \sum_{i=1}^{N} \mathcal{Y}_i \oplus \alpha_i \tag{15}$$

- **Average Precision:** It calculates the fraction of labels ranked higher (predicted) than a particular labels correctly by a learner. $\mathrm{r}_i(j)$ denotes the rank of label j for $\mathbf{x}_i$ instance predicted by a learner.

$$\begin{aligned}\text{Average Precision} = \frac{1}{\text{N}} \sum_{i=1}^{\text{N}} \frac{1}{|\mathcal{Y}_i|} \\ \times \sum_{\gamma \in \mathcal{Y}_i} \frac{|\{\gamma \in \mathcal{Y}_{i:} r_i(\gamma) \leq r_i(\gamma')\}|}{\mathrm{r_i}(\gamma)}\end{aligned} \tag{16}$$

- **One-Error:** One error counts the number of instances for which the predicted top-rank label is not present in the actual label set. Lower the value of one-error, better is the performance of the learner.

$$\text{One Error} = \frac{1}{n} \sum_{i=1}^{n} [\arg \max\nolimits_{\mathrm{label}_j \in \alpha_i} f_j(\mathbf{x}_i) \notin \mathcal{Y}_i] \tag{17}$$

- **Coverage:** Let us considered an ordered list of predicted labels for each instance, where the top-ranked label is label is numbered one. Coverage evaluates the number of steps we need to move down the list to get the set of all true labels of the instance. It can easily intuited that lesser the value of coverage better is the performance. Let l_j be the j^{th} label. Let $\mathrm{rank}(\mathbf{x}_i, l_j)$ be the rank of j^{th} label w.r.t. instance $\mathbf{x}_i$.

$$\text{Coverage} = \frac{1}{L}(\frac{1}{n} \sum_{i=1}^{n} \max\nolimits_{l_j \in \alpha_i} \mathrm{rank}(\mathbf{x}_i, l_j) - 1) \tag{18}$$

- **Ranking Loss:** Ranking Loss calculates the average percentage of mis-ordered pair of labels. Lower value of ranking loss is desirable for a classifier.

$$\begin{aligned}&\text{Ranking loss} \\ &= \frac{1}{n} \sum_{i=1}^{t} \frac{|\{(l_k, l_j),\ f_k(\mathbf{x}_i) \leq f_j(\mathbf{x}_j), (l_k, l_j) \in Y_i \times \overline{Y}_i\}|}{|Y_i||\overline{Y}_i|}\end{aligned} \tag{19}$$

- **Macro-averaging AUC:** Let AUC_j be the AUC score for label j. We calculate the average AUC score of all labels in Macro-averaging AUC. Higher the value of Macro-averaging AUC, better is the performance of the learner.

$$\text{Macro-averaging AUC} = \frac{1}{L} \sum_{i=1}^{L} \mathrm{AUC}_i \tag{20}$$

B. SECOND EXPERIMENT: CLASS-IMBALANCE

We present the empirical study on class-imbalance aspect of multi-label datasets in this subsection. The same eleven multi-label datasets used in the first experiment are used in this section but with some preprocessing. In all of these datasets, we have removed the labels whose imbalance ratio (number of negative instances / number of positive instances) is more than 50 or the number of positive instances is less than 20. A similar protocol has been suggested in [33]. For the nominal datasets, we have performed reduction in the feature set according the same recommendation. The attribute information of the datasets with respect to this experiment are presented on Table 2. Since this work deals with differential class imbalance ratio of multi-label datasets, we have also showed the minimum (min IR), maximum (max IR) and average imbalance (avg IR) statistics of each dataset in Table 2. These datasets are obtained from MULAN [42] and MEKA [43] repositories.

For comparative analysis, we consider RAKEL ([21]), LIFT ([12]) and CLR ([19]) of multi-label learners which are used in the first experiment. In addition to that, we have also included COCOA [33] and RML [44]. COCOA specifically addresses class-imbalance problem in multi-label datasets. Additionally, we have also included Reverse-nearest neighborhood based oversampling for multi-label dataset (RNNOML) [37] in this empirical study. MLKNN invokes a set of k-nearest neighbor based classifiers for multi-label datasets. Besides these, we have included a couple of methods - namely SMOTE ([45]) and Random Undersampling (USAM) which are dedicated to general class-imbalance problem and used them in multi-label setting. The proposed method is run in a cost-sensitive learning framework, where an imbalance adaptive misclassification cost is assigned for each label.

For evaluating their performances we have employed Macro-averaging F_1 and Macro-averaging AUC. They are described below.

- **Macro-averaging $\mathbf{F}_1$:** It calculates the average of F_1 values across all labels. Let tp_j, tn_j, fp_j and fn_j denote the number of true positive, true negative, false positive and false negative predictions for label j respectively. We calculate F_1 for label j,

$$\mathrm{F}_{1\mathrm{j}} = \frac{2 \times tp_j}{2 \times tp_j + fp_j + fn_j} \tag{21}$$

$$\text{Macro-averaging } \mathrm{F}_1 = \frac{1}{\mathrm{L}} \sum_{\mathrm{j}=1}^{\mathrm{L}} \mathrm{F}_{1\mathrm{j}} \tag{22}$$

- **Macro-averaging AUC:** Let AUC_j be the AUC score for label j. We calculate the average AUC score of all

TABLE 3. This table shows the predictive performance of the proposed method (LIIML) and the competing methods on 11 multi-label datasets in Experiment 1. The performances are reported on six evaluating metrics — ↑ beside a metric indicates that higher value is better and ↓ indicates superiority with a lower score, we have indicated the best outcome in a background highlighted rectangle. On *Hamming Loss* and *One Error*, LIIML has achieved the best scores on 10 and 9 datasets respectively. On *Ranking Loss* and *Coverage*, the performance of LIIML is mot as good as on the previous two metrics. LIIML gets 5 best scores on *Ranking Loss* and only 4 best scores on *Coverage*. LIIML achieves 8 best scores on *Average Precision* metric and 6 best scores on *Macro-averaging AUC*.

	Corel5k	Enron	Medical	Slashdot	Tmc2007	CAL500	emotions	RCV1subset1	Rcv1subset2	scene	yeast
Hamming Loss ↓											
BR	0.013± 0.002	0.059±0.002	0.049±0.007	0.050±0.001	0.072±0.001	**0.137±0.002**	0.199±0.004	0.031±0.002	0.029±0.001	0.110±0.003	0.200±0.002
CLR	0.011±0.002	0.054±0.001	0.067±0.018	0.049±0.001	0.068±0.001	**0.137±0.002**	0.193±0.005	0.029±0.001	0.025±0.001	0.112±0.002	0.201±0.003
RAKEL	0.012±0.001	0.058±0.001	0.016±0.002	0.048±0.002	0.069±0.002	0.139±0.002	0.198±0.004	0.030±0.002	0.024±0.001	0.097±0.006	0.207±0.003
ECC	0.014±0.001	0.056±0.001	0.014±0.001	0.057±0.001	0.067±0.004	0.182±0.003	0.195±0.003	0.030±0.001	0.030±0.001	0.097±0.004	0.207±0.003
LIFT	0.010±0.001	0.048±0.004	0.014±0.002	**0.040±0.003**	0.061±0.001	0.138±0.004	0.193±0.005	0.027±0.004	**0.022±0.012**	0.084±0.003	0.197±0.003
LIIML	**0.009±0.003**	**0.045±0.006**	**0.012±0.005**	**0.040±0.004**	**0.057±0.004**	0.138±0.003	**0.176±0.006**	**0.025±0.002**	**0.022±0.010**	**0.081±0.004**	**0.193±0.005**
One-error ↓											
BR	0.848±0.007	0.498±0.010	0.162±0.005	0.503±0.008	0.347±0.002	0.357±0.021	0.286±0.016	0.587±0.010	0.574±0.008	0.346±0.006	0.256±0.007
CLR	0.718±0.008	0.278±0.009	0.160±0.007	0.434±0.006	0.244±0.003	0.123±0.014	0.243±0.007	0.421±0.006	0.418±0.005	0.256±0.008	0.230±0.006
RAKEL	0.815±0.011	0.412±0.012	0.159±0.008	0.452±0.006	0.252±0.002	0.287±0.037	0.251±0.008	0.480±0.003	0.427±0.001	0.248±0.006	0.251±0.006
ECC	0.698±0.004	0.295±0.007	**0.100±0.005**	0.418±0.009	0.233±0.002	0.137±0.023	0.221±0.010	0.422±0.008	0.418±0.006	0.256±0.004	**0.227 0.004**
LIFT	0.706±0.005	0.264±0.004	0.203±0.004	0.423±0.003	0.213±0.002	0.126±0.005	0.261±0.004	0.416±0.006	0.422±0.004	0.213±0.004	0.229±0.010
LIIML	**0.660±0.003**	**0.232±0.006**	0.198±0.005	**0.413±0.010**	**0.176±0.004**	**0.099±0.003**	**0.193±0.006**	**0.406±0.005**	**0.407±0.008**	**0.202±0.005**	0.229±0.004
Ranking Loss ↓											
BR	0.655±0.005	0.307±0.008	0.038±0.002	0.218±0.005	0.216±0.002	0.516±0.007	0.159±0.024	0.280±0.003	0.252±0.004	0.167±0.007	0.314±0.004
CLR	**0.114±0.002**	0.080±0.004	0.048±0.004	**0.094±0.003**	0.050±0.002	0.181±0.003	0.165±0.005	**0.041±0.002**	**0.042±0.002**	0.084±0.005	0.171±0.004
RAKEL	0.626±0.008	0.242±0.005	0.058±0.004	0.212±0.005	0.137±0.003	0.439±0.017	0.212±0.008	0.243±0.016	0.221±0.012	0.105±0.004	0.241±0.003
ECC	0.292±0.003	0.134±0.005	0.098±0.004	0.131±0.005	0.076±0.004	0.184±0.004	0.231±0.006	0.240±0.004	0.218±0.003	0.110±0.003	0.240±0.004
LIFT	0.127±0.004	0.087±0.005	**0.035±0.003**	0.100±0.004	0.052±0.002	0.182±0.004	0.152±0.008	0.052±0.004	0.054±0.004	**0.066±0.004**	0.168±0.004
LIIML	0.135±0.003	**0.076±0.004**	0.036±0.007	0.104±0.010	**0.042±0.003**	**0.179±0.003**	**0.123±0.006**	0.071±0.004	0.068±0.010	0.070±0.003	**0.166±0.004**
Coverage ↓											
BR	0.898±0.008	0.596±0.005	**0.049±0.004**	0.239±0.005	0.381±0.004	0.967±0.011	0.301±0.008	0.447±0.004	0.382±0.005	0.157±0.006	0.638±0.004
CLR	**0.268±0.005**	**0.229±0.003**	0.054±0.006	**0.109±0.004**	0.127±0.003	**0.751±0.007**	0.285±0.006	**0.103±0.004**	**0.106±0.005**	0.084±0.004	0.463±0.005
RAKEL	0.886±0.005	0.524±0.008	0.054±0.003	0.214±0.004	0.278±0.005	0.970±0.027	0.318±0.011	0.414±0.008	0.354±0.005	0.107±0.006	0.557±0.011
ECC	0.564±0.006	0.351±0.005	0.072±0.004	0.132±0.006	0.173±0.003	0.807±0.007	0.315±0.006	0.187±0.003	0.209±0.007	0.086±0.004	0.465±0.007
LIFT	0.314±0.007	0.242±0.004	0.052±0.003	0.125±0.004	0.129±0.002	0.758±0.004	0.289±0.005	0.141±0.006	0.142±0.005	0.069±0.007	0.467±0.007
LIIML	0.310±0.003	0.245±0.006	0.051±0.005	0.122±0.005	**0.121±0.004**	0.757±0.003	**0.265±0.006**	0.134±0.005	0.121±0.005	**0.065±0.003**	**0.452 ±0.006**
Average Precision ↑											
BR	0.101±0.004	0.452±0.010	0.821±0.005	0.573±0.005	0.647±0.002	0.278±0.006	0.785±0.012	0.386±0.002	0.434±0.003	0.761±0.005	0.674±0.006
CLR	0.273±0.003	0.675±0.005	0.832±0.018	0.672±0.004	0.798±0.002	0.498±0.005	0.808±0.005	**0.629±0.003**	**0.641±0.004**	0.851±0.004	0.756±0.003
RAKEL	0.122±0.005	0.541±0.006	0.824±0.005	0.617±0.005	0.736±0.002	0.352±0.005	0.794±0.006	0.436±0.008	0.483±0.005	0.842±0.006	0.724±0.005
ECC	0.280±0.003	0.651±0.006	0.840±0.011	0.675±0.009	0.785±0.002	0.482±0.005	0.798±0.006	0.607±0.006	0.616±0.005	0.854±0.004	0.752±0.005
LIFT	**0.282±0.006**	0.682±0.003	0.839±0.006	0.672±0.005	0.814±0.002	0.497±0.007	0.809±0.005	0.564±0.008	0.579±0.006	0.877±0.005	0.758±0.006
LIIML	0.268±0.003	**0.684±0.006**	**0.843±0.005**	**0.680±0.010**	**0.839±0.005**	**0.499±0.003**	**0.848±0.006**	0.575±0.005	0.604±0.010	**0.882 ±0.004**	**0.768±0.006**
Macro-averaging AUC ↑											
BR	0.519±0.002	0.579±0.007	0.843±0.006	0.656±0.008	0.722±0.002	0.502±0.009	0.833±0.009	0.608±0.005	0.509±0.004	0.803±0.005	0.565±0.004
CLR	0.678±0.006	**0.698±0.008**	0.921±0.011	0.834±0.014	0.903±0.002	0.516±0.005	0.848±0.009	0.898±0.004	0.863±0.003	0.916±0.006	0.645±0.005
RAKEL	0.522±0.002	0.598±0.006	0.912±0.014	0.688±0.010	0.799±0.002	0.504±0.007	0.832±0.006	0.636±0.005	0.628±0.005	0.883±0.006	0.616±0.005
ECC	0.569±0.003	0.646±0.007	0.907±0.015	0.767±0.009	0.881±0.002	0.508±0.004	0.824±0.007	0.774±0.005	0.763±0.005	0.931±0.006	0.644±0.004
LIFT	**0.688±0.010**	0.692±0.006	0.921±0.009	**0.848±0.005**	0.907±0.002	0.520±0.006	0.842±0.006	**0.901±0.007**	**0.879±0.006**	0.942±0.006	0.663±0.007
LIIML	0.593±0.003	0.648±0.006	**0.931±0.005**	0.822±0.010	**0.918±0.004**	**0.524±0.003**	**0.858±0.006**	0.813±0.005	0.818±0.003	**0.946±0.010**	**0.676±0.003**

labels in Macro-averaging AUC. Higher the value of Macro-averaging AUC, better is the performance of the learner.

$$\text{Macro-averaging AUC} = \frac{1}{L}\sum_{i=1}^{L}\text{AUC}_i \tag{23}$$

C. THIRD EXPERIMENT: COMPETENCE OF THE IMBALANCE ADAPTIVE MISCLASSIFICATION COST

We analyze the utility of the proposed scheme of imbalance-adaptive misclassification cost in this study. We consider two first-order methods LIFT and B_R in their default settings where the mis-classification costs of the classes are equal. We compare their performances with an enhanced cost version of each of them, LIFT-cost and B_R respectively, where the cost of misclassification of the minority instances is set according to the proposed scheme. We evaluate the difference in performances using Macro-averaging F_1 metric.

D. FOURTH EXPERIMENT: PARAMETER OPTIMIZATION

In this experiment, we have studied the effect of variation of cost factor and number of lattice points on class-imbalance focused multi-label learning. Cost factor is varied between 0.5, 1, 2 and 4. Variation of the number of positive and negative lattice points is also explored.

E. STATISTICAL SIGNIFICANCE TEST

We have conducted Wilcoxon Signed Rank Sum Test to measure the statistical significance of the difference in performance given by the proposed method, LIIML with respect to a competing method. In this work, we have a number of experiments and each is evaluated with more than one metric. Experiment 1 and 2 are the key ones of this work. We have constituted the statistical tests for these two experiments. We report the p value at which the performance of the two methods are different. Lower the p value, more significance is the difference or more certain we are about rejecting the null hypothesis. The null hypothesis assumes that the performance of two methods are same. p value 0.05 or 5% significance level is the standard threshold for rejecting or accepting a null hypothesis. We have used p value 0.05 as the threshold for statistical significance of difference.

VI. RESULTS AND ANALYSIS

In this section, we summarize the results of LIIML with that of the state-of-the-art multi-label learners. Table 3 record the results of Experiment 1, which is dedicated to evaluate the efficacy of the methods in a regular setting. In Table 4 we present the outcomes of Experiment 2, where we have evaluated enhanced-cost versions of LIIML in a class-imbalanced setting. The outcomes of Experiment 3 is presented in Table 5. Outcomes of experiment 4 are portrayed graphically in Figures 2-5.

TABLE 4. This table records the observations of experiment on class-imbalance aspect of multi-label dataset (Experiment 2). Result are reported for 2 metrics (Macro-averaging F_1 and Macro-averaging AUC), 11 datasets and 9 methods (including LIIML). For both the metrics, a higher value means better result (as indicated by the ↑). On Macro-averaging F_1 and Macro-averaging AUC, LIIML has achieved best scores among all methods on 8 and 7 datasets respectively.

	Corel5k	Enron	Medical	Slashdot	TMC	CAL500	RCV1 Subset1	RCV2 Subset2	Emotions	Scene	Yeast
Macro-averaging F_1 ↑											
USAM	0.143±0.004	0.263±0.011	0.673±0.013	0.260±0.008	0.606±0.003	0.216±0.006	0.357±0.006	0.342±0.005	0.594±0.012	0.621±0.007	0.431±0.008
SMOTE	0.131±0.003	0.263±0.005	0.671±0.018	0.323±0.006	0.608±0.003	0.238±0.005	0.312±0.003	0.308±0.004	0.586±0.021	0.618±0.005	0.431±0.003
RML	**0.215±0.007**	0.308±0.008	0.670±0.016	0.315±0.002	0.568±0.003	0.209±0.006	0.313±0.004	0.302±0.004	0.586±0.003	0.623±0.006	0.429±0.004
COCOA	0.197±0.003	0.327±0.007	0.690±0.011	0.326±0.009	0.668±0.003	0.208±0.010	0.362±0.006	0.339±0.008	0.665±0.014	0.731±0.010	0.455±0.014
LIFT	0.072±0.003	0.290±0.005	0.588±0.005	0.382±0.007	0.416±0.003	0.076±0.006	0.212±0.004	0.162±0.005	0.639±0.006	0.758±0.006	0.377±0.005
RAKEL	0.091±0.006	0.249±0.004	0.577±0.012	0.248±0.004	0.642±0.003	0.195±0.002	0.274±0.005	0.267±0.005	0.611±0.011	0.684±0.007	0.421±0.006
CLR	0.051±0.003	0.223±0.005	0.653±0.011	0.234±0.006	0.626±0.003	0.086±0.006	0.226±0.004	0.226±0.005	0.593±0.015	0.631±0.012	0.414±0.007
RNNOML	0.200±0.003	0.345±0.004	**0.768±0.005**	0.444±0.005	0.630±0.004	0.246±0.003	**0.478±0.003**	**0.475±0.002**	0.683±0.002	0.745±0.005	0.488±0.004
LIIML	0.179±0.003	**0.367±0.006**	0.721±0.005	**0.477±0.010**	**0.672±0.005**	**0.255±0.006**	0.459±0.008	0.457±0.003	**0.692±0.004**	**0.766±0.005**	**0.509±0.004**
Macro-averaging AUC ↑											
USAM	0.574±0.005	0.605±0.011	0.852±0.014	0.620±0.004	0.801 ± 0.004	0.515±0.003	0.675±0.011	0.673±0.009	0.707±0.015	0.792±0.007	0.579±0.006
SMOTE	0.601±0.006	0.619±0.007	0.872±0.006	0.685±0.006	0.789±0.003	0.514±0.004	0.623±0.006	0.621±0.005	0.701±0.008	0.771±0.012	0.589±0.008
RML	–	–	–	–	–	–	–	–	–	–	–
COCOA	0.717±0.003	0.735±0.005	0.955±0.004	0.732±0.004	0.930±0.002	0.554±0.004	0.890±0.003	0.884±0.003	0.842±0.007	**0.944±0.004**	**0.712±0.003**
LIFT	**0.742±0.003**	0.756±0.004	0.946±0.010	0.830±0.006	0.911±0.003	0.529±0.006	0.898±0.005	0.893±0.005	0.844±0.007	0.942±0.007	0.680±0.005
RAKEL	0.550±0.003	0.637±0.003	0.831±0.005	0.613±0.003	0.862±0.002	0.525±0.003	0.730±0.004	0.721±0.005	0.795±0.011	0.892±0.003	0.641±0.005
CLR	0.742±0.001	0.662±0.004	0.801±0.008	0.697±0.008	0.905±0.003	**0.559±0.003**	0.892±0.005	0.883±0.002	0.793±0.009	0.897±0.005	0.652±0.004
RNNOML	0.726±0.004	0.735±0.006	**0.977±0.007**	0.822±0.003	0.921±0.004	0.558±0.004	0.912±0.003	0.914±0.005	0.841±0.005	0.918±0.006	0.661±0.003
LIIML	0.718±0.006	**0.765±0.003**	0.967±0.004	**0.838±0.005**	**0.923±0.006**	0.524±0.002	**0.920±0.002**	**0.918±0.004**	**0.847±0.003**	**0.944±0.004**	0.694±0.003

TABLE 5. This table records the results of applying the proposed cost-sensitive learning paradigm on two first order approaches LIFT and BR. We have used Macro-averaging F_1 and Macro-averaging AUC as the evaluating metrics. The original results (without added cost), results with enhanced cost and the corresponding improvement on each dataset are reported in the table. The instances where the enhanced cost version achieves an improvement of greater than 20% over the original result are highlighted through darkening of their backgrounds. — ↑ indicates higher is better and ↓ indicates lower is better, best outcome is indicated in bold-face.

	Corel5k	Enron	Medical	Slashdot	TMC	CAL500	RCV1 Subset1	RCV2 Subset2	Emotions	Scene	Yeast
Macro-averaging F_1 ↑											
LIFT-with cost	0.071	0.397	0.704	0.463	0.643	0.245	0.387	0.379	0.682	0.758	0.490
LIFT	0.154	0.290	0.588	0.381	0.417	0.076	0.212	0.167	0.639	0.755	0.377
% of improvement	**116%**	**36%**	19%	**21%**	**54%**	**222%**	**82%**	**127%**	7%	0.4%	**30%**
Macro-averaging AUC ↑											
BR-with cost	0.206	0.357	0.797	0.459	0.583	0.275	0.482	0.486	0.651	0.674	0.476
BR	0.029	0.194	0.742	0.346	0.365	0.076	0.258	0.247	0.556	0.635	0.337
% of improvement	**610%**	**84%**	7%	**33%**	**60%**	**261%**	**87%**	**97%**	17%	6%	**42%**

TABLE 6. This table corresponds to the outcomes of Experiment 1 (Table 3). It reports the p value at which LIIML's performance is statistically superior to that of a comparing method for a given metric. Each row corresponds to a metric and each column to a method. Lower the p value, more significant is superiority. We have selected $p = 0.05$ as the threshold for statistical significance. Outcomes at which $p < 0.05$ are indicated in boldface. LIIML achieves statistical superiority with respect to BR, CLR, RAKEL, ECC and LIFT on 5, 2, 5, 4 and 4 cases respectively.

Methods→ Metrics↓	BR	CLR	RAKEL	ECC	LIFT
Hamming Loss	**0.004**	**0.004**	**0.003**	**0.003**	**0.009**
One Error	**0.016**	**0.016**	**0.008**	0.062	**0.005**
Ranking Loss	**0.003**	0.928	**0.003**	**0.003**	0.857
Coverage	**0.004**	0.447	**0.003**	**0.003**	**0.009**
Average Precision	**0.003**	0.286	**0.003**	0.109	**0.026**
Macro-averaging AUC	**0.003**	0.424	**0.003**	**0.005**	0.285

Experiment 1: Table 3 shows the performances of LIIML and the competing methods on the 11 multi-label datasets. On *Hamming Loss*, the proposed scheme has achieved lowest error value on 10 out of 11 (**90.90%**) datasets. On *One error, Coverage* and *Ranking Loss*, either of LIIML have achieved best scores on 9 (**81.81%**), 4 (36.36 %) and 5 (45.45%) datasets respectively. On *Average Precision*, LIIML has achieved best scores on 8 (**72.72%**) datasets. LIIML's performance is superior to other datasets on *Macro-averaging AUC* metric across 6 (**54.54%**) datasets. On Table 3, each method has 66 observations (11 datasets × 6 metrics). We have summarized the cumulative observations of Table 3 as follows.

- LIIML has out performed BR on 63 out of 66 cases (**91.66%**).
- LIIML's performance is better than CLR on 46 out of 66 (**69.69%**) pairwise observations. We note that LIIML could not outperform CLR on Ranking Loss and Coverage metrics. The working principles of CLR is based on ranking of labels and this aspect has likely contributed to it's efficiency.
- LIIML has outperformed RAKEL on 65 occasions (**98.48%**).
- ECC achieves better performance as compared to LIIML on 5 cases. Hence, LIIML has outperformed ECC on 61 (**92.42%** cases).
- LIIML has performed better than LIFT on 50 (**75.76%**) occasions. For 2 pair-wise observations, the scores of LIFT and LIIML are tied. On remaining 14 cases, LIFT has outperformed LIIML.

Experiment 2: Table 4 shows the performance of the proposed and comparing methods on Macro-averaging F_1 and Macro-averaging AUC. LIIML achieves the best score on a total of 8 out of 11 cases (**72.72%** cases) on Macro-averaging F_1. On one remaining dataset *Corel5k*, RML has obtained the best results of Macro-averaging F_1. On 2 datasets, RNNOML has performed best. On Macro-averaging AUC, LIIML performs better than all other methods on 7 out of 11 datasets (**63.63%**). The remaining 4 best

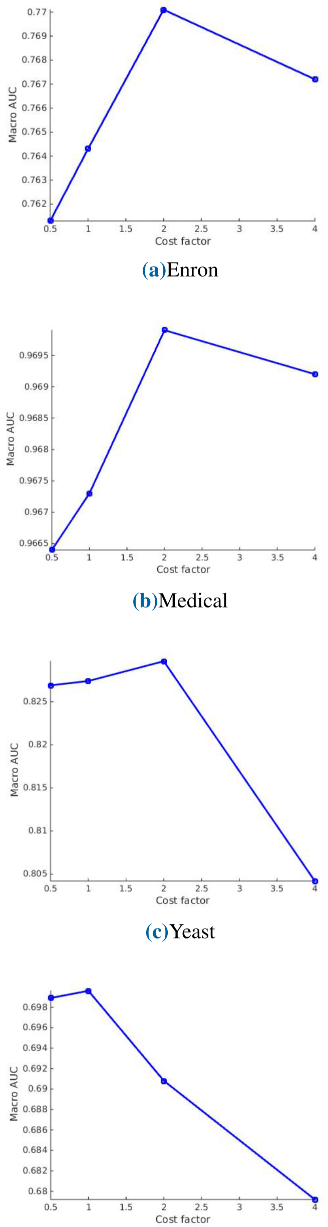


FIGURE 2. Macro-averaging AUC results of four datasets subject to varying and increasing misclassification costs for the minority class. We have varied the cost factor between 0.5, 1, 2 and 4. It can be observed that increasing the cost factor value upto 2 improves the learning of minority classes of each label. The graphs of these figures indicate a loss of performance on cost factor beyond 2 on all the four datasets. On using a value beyond 2, the classifier is getting over-biased towards the minority class. The optimal cost factor value is 2 for three datasets and 1 for one dataset.

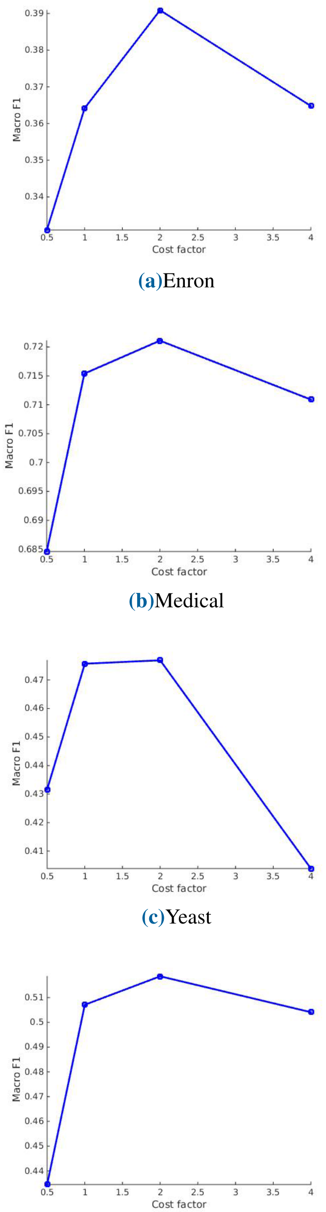
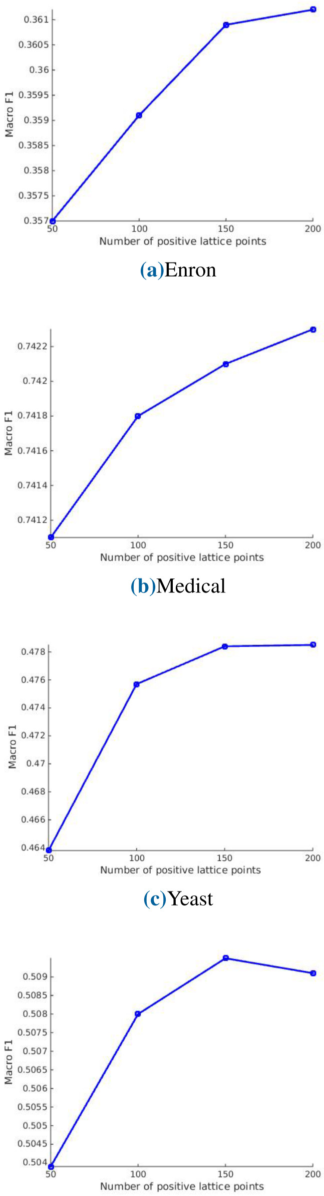


FIGURE 3. Macro-averaging F_1 results of four datasets subject to varying and increasing misclassification costs for the minority class. We have varied the cost factor between 0.5, 1, 2 and 4. The observation and analysis of this figure's data is in congruence with our findings from Figure 2. On all four cases, Macro-averaging F_1 value increases on increasing the cost factor value upto 2. It marks the optimal value for learning the given datasets. An increase beyond cost factor value 2 is observed to cause a loss of minority performance.

scores of Macro-averaging AUC are shared by COCOA (2), CLR (1) and RNNOML (1).

Experiment 3: Table 5 records the *Macro-averaging* F_1 scores of LIFT and BR in regular cost framework and enhanced misclassification cost framework. The results indicate certain effectiveness of the enhanced cost scheme in handling class-imbalance and recognition of the positive (minority) class of the multi-label datasets. *Macro-averaging* F_1 performance of LIFT has improved by over 20% on 8 out of 11 datasets using the misclassification cost-enhancement learning. On BR, the improvement using this scheme is also pronounced as we witness the improvement in results by over 20% on 8 out of 11 datasets also. For two datasets *Corel5k* and *CAL500* the percentage of improvement is more than 100 (w.r.t both BR and LIFT).



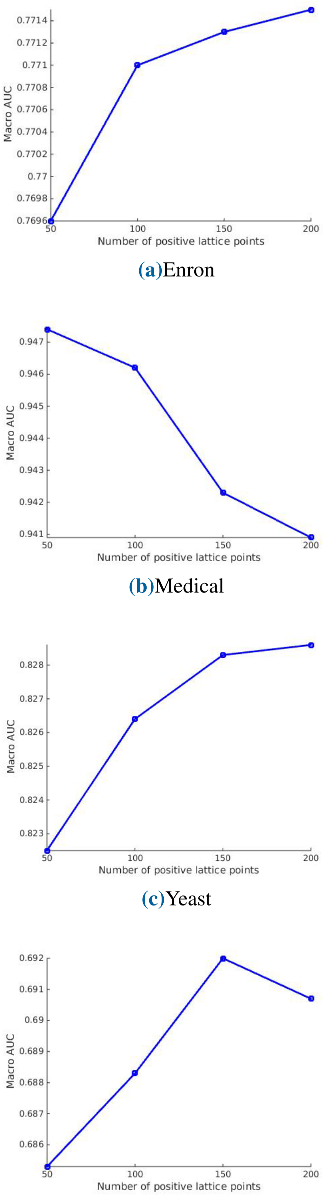
(a) Enron

(b) Medical

(c) Yeast

(d) Slashdot

FIGURE 4. Macro-averaging F_1 results of four datasets subject to varying number of lattice points. We have varied the number of positive lattices between 50, 100, 150 and 200. Number of negative lattices is proportional to the number of positive lattice points and vary accordingly. The figures indicate that increasing the lattice points result in improvement in macro-averaging F_1 performance.



(a) Enron

(b) Medical

(c) Yeast

(d) Slashdot

FIGURE 5. Macro-averaging AUC results of four datasets subject to varying number of lattice points. We have varied the number of positive lattices between 50, 100, 150 and 200. Number of negative lattices is proportional to the number of positive lattice points and vary accordingly. For three datasets (Enron, Yeast and Slashdot) Macro-averaging AUC scores increase with increasing the lattice points. We get an exception with Medical dataset, where the performance degrades as number of lattices.

Experiment 4: Figures 2 and 3 shows the variation of Macro-averaging AUC and Macro-averaging F_1 scores on varying ranges of cost factor. Increasing the value of the cost factor promotes the recognition of minority class instances at the cost of majority class performance dataset. Macro-averaging AUC and Macro F_1 scores also indicate the same. Increasing the cost factor beyond 2 results in sharp fall of Macro-averaging F_1 and Macro-averaging AUC scores for all four datasets. Number of lattice points is important for perceiving a functional geometry of the data points. Considering a lower number of lattice points will give a distorted geometry. This in turn causes a fall in performance. On the other hand increasing the cardinality of the lattice point set is computationally more intensive. Th findings demonstrated in Figures 4 and 5 are somewhat in agreement with the above. The only exception is Medical dataset, which has shown falls in performance with increasing number of lattice points.

Statistical Tests: Tables 6 and 7 show the results of statistical significance test on outcomes of Experiment 1 and

TABLE 7. This table corresponds to the outcomes of Experiment 2 (Table 4). It reports the p value at which LIIML's performance is statistically superior to that of a comparing method for a given metric. Each row corresponds to a method and each column to a metric. A lower p value more significant difference in performance. We have selected $p = 0.05$ as the threshold for statistical significance. Outcomes at which $p < 0.05$ are indicated in boldface. On Macro-averaging F_1, LIIML achieves statistical superior performance over 7 out of 8 methods. On Macro-averaging AUC metric, LIIML's performance is statistically superior to 4 out of 7 methods.

Metrics→ Methods↓	Macro-averaging F_1	Macro-averaging AUC
USAM	**0.003**	**0.003**
SMOTE	**0.003**	**0.003**
RML	**0.004**	–
COCOA	**0.006**	0.184
LIFT	**0.003**	0.075
RAKEL	**0.003**	**0.004**
CLR	**0.003**	**0.018**
RNNOML	0.424	0.308

Experiment 2 respectively. For analyzing results of Experiment 1 (Table 6), we have performed 30 (5 × 6) tests for 5 comparing methods and 6 metrics. LIIML has achieved statistically superior performance on 22 (73.33%) occasions. For analyzing results of Experiment 2 (Table 7), we have conducted 15 tests for 8 comparing methods (RML did not output Macro-averaging AUC scores) and 2 metrics. On 11 (73.33%) cases, LIIML has achieved statistically superior performance.

VII. CONCLUSION

In this paper, we have proposed a novel multi-label learner which takes into account the class imbalance and class geometries of the multi-label datasets. The simple and naive framework of imbalance-adaptive misclassification cost framework has given a new direction in the field of class-imbalanced multi-label learning. The detailed experimental evaluation indicates the performance of the proposed method as comparable to superior across different evaluating metrics. The novelty of LIIML is two fold. First one is the feature extraction method which takes into account both class imbalance and class geometries of the labels. The other key contribution is the imbalance-adaptive cost sensitive learning. It is an effective yet simple tool for handling diversified imbalance ratios of the labels. Moreover, it has a general framework which can be used with any first order multi-label approach. Our future studies are directed at exploring and finding out the optimal number of lattice points across different labels to represent the class geometries of a multi-label dataset.

REFERENCES

[1] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proc. Eur. Conf. Mach. Learn.* Springer, 1998, pp. 137–142.

[2] S. Godbole and S. Sarawagi, "Discriminative methods for multi-labeled classification," in *Proc. 8th Pacific-Asia Conf. Knowl. Discovery Data Mining*, 2004, pp. 22–30.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004.

[4] G. Nasierding, G. Tsoumakas, and A. Z. Kouzani, "Clustering based multi-label classification for image annotation and retrieval," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2009, pp. 4514–4519.

[5] T. Li and M. Ogihara, "Toward intelligent music information retrieval," *IEEE Trans. Multimedia*, vol. 8, no. 3, pp. 564–574, Jun. 2006.

[6] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, Apr. 2006.

[7] M. A. Mammadov, A. M. Rubinov, and J. Yearwood, "The study of drug–reaction relationships using global optimization techniques," *Optim. Methods Softw.*, vol. 22, no. 1, pp. 99–126, Feb. 2007.

[8] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proc. ECML/PKDD Workshop Discovery Challenge*, 2008, pp. 1–9.

[9] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *Proc. 15th ACM Int. Conf. Multimedia (MM)*. New York, NY, USA: ACM, 2007, pp. 17–26.

[10] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "A comparison of multi-label feature selection methods using the problem transformation approach," *Electron. Notes Theoretical Comput. Sci.*, vol. 292, pp. 135–151, Mar. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1571066113000121

[11] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst., Natural Synth. (NIPS)*. Cambridge, MA, USA: MIT Press, 2001, pp. 681–687. [Online]. Available: http://dl.acm.org/citation.cfm?id=2980539.2980628

[12] M.-L. Zhang and L. Wu, "Lift: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[13] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[14] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognit.*, vol. 45, no. 9, pp. 3084–3104, Sep. 2012.

[15] E. Gibaja and S. Ventura, "Multi-label learning: A review of the state of the art and ongoing research," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 4, no. 6, pp. 411–444, 2014.

[16] J. M. Moyano, E. L. Gibaja, K. J. Cios, and S. Ventura, "Review of ensembles of multi-label classifiers: Models, experimental study and prospects," *Inf. Fusion*, vol. 44, pp. 33–45, Nov. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253517307169

[17] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[18] J. Read, "A pruned problem transformation method for multi-label classification," in *Proc. New Zealand Comput. Sci. Res. Student Conf. (NZCSRS)*, vol. 143150, 2008, p. 41.

[19] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, Nov. 2008.

[20] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[21] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[22] J. Read, L. Martino, and D. Luengo, "Efficient monte carlo optimization for multi-label classifier chains," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 3457–3461.

[23] W. Cheng, E. Hüllermeier, and K. J. Dembczynski, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 279–286.

[24] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Mach. Learn.*, vol. 85, no. 3, pp. 333–359, Dec. 2011.

[25] 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, Jul. 2007.

[26] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, Sep. 2009.

[27] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[28] J. Nam, J. Kim, E. L. Mencía, I. Gurevych, and J. Fürnkranz, ‘‘Large-scale multi-label text classification—Revisiting neural networks,’’ in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, 2014, pp. 437–452.

[29] S. Xu, X. Yang, H. Yu, D.-J. Yu, J. Yang, and E. C. Tsang, ‘‘Multi-label learning with label-specific feature reduction,’’ *Knowl.-Based Syst.*, vol. 104, pp. 52–61, Jul. 2016, doi: 10.1016/j.knosys.2016.04.012.

[30] F. Li, D. Miao, and W. Pedrycz, ‘‘Granular multi-label feature selection based on mutual information,’’ *Pattern Recognit.*, vol. 67, pp. 410–423, Jul. 2017.

[31] J. Huang, G. Li, Q. Huang, and X. Wu, ‘‘Joint feature selection and classification for multilabel learning,’’ *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

[32] J. Zhang, C. Li, D. Cao, Y. Lin, S. Su, L. Dai, and S. Li, ‘‘Multi-label learning with label-specific features by resolving label correlations,’’ *Knowl.-Based Syst.*, vol. 159, pp. 148–157, Nov. 2018.

[33] M.-L. Zhang, Y.-K. Li, and X.-Y. Liu, ‘‘Towards class-imbalance aware multi-label learning,’’ in *Proc. 24th Int. Conf. Artif. Intell. (IJCAI)*, 2015, pp. 4041–4047. [Online]. Available: http://dl.acm.org/citation.cfm?id=2832747.2832812

[34] Y.-P. Wu and H.-T. Lin, ‘‘Progressive random k-labelsets for cost-sensitive multi-label classification,’’ *Mach. Learn.*, vol. 106, no. 5, pp. 671–694, May 2017, doi: 10.1007/s10994-016-5600-x.

[35] K.-H. Huang and H.-T. Lin, ‘‘Cost-sensitive label embedding for multi-label classification,’’ *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1725–1746, Oct. 2017, doi: 10.1007/s10994-017-5659-z.

[36] O. Reyes, C. Morell, and S. Ventura, ‘‘Effective lazy learning algorithm based on a data gravitation model for multi-label learning,’’ *Inf. Sci.*, vols. 340–341, pp. 159–174, May 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025516000086

[37] P. Sadhukhan and S. Palit, ‘‘Reverse-nearest neighborhood based over-sampling for imbalanced, multi-label datasets,’’ *Pattern Recognit. Lett.*, vol. 125, pp. 813–820, Jul. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167865519302193

[38] B. Goldluecke and D. Cremers, ‘‘Convex relaxation for multilabel problems with product label spaces,’’ in *Computer Vision—ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Germany: Springer, 2010.

[39] H. Su and J. Rousu, ‘‘Multilabel classification through random graph ensembles,’’ *Mach. Learn.*, vol. 99, no. 2, pp. 231–256, May 2015.

[40] X. Kong and P. S. Yu, ‘‘Multi-label feature selection for graph classification,’’ in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2010, pp. 274–283.

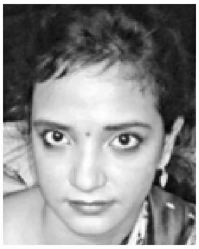
[41] C.-C. Chang and C.-J. Lin, ‘‘LIBSVM: A library for support vector machines,’’ *ACM Trans. Intell. Syst. Technol.*, vol. 2, pp. 27:1–27:27, May 2011.

[42] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, ‘‘Mulan: A java library for multi-label learning,’’ *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jun. 2011.

[43] J. Read, P. Reutemann, B. Pfahringer, and G. Holmes, ‘‘MEKA: A multi-label/multi-target extension to Weka,’’ *J. Mach. Learn. Res.*, vol. 17, no. 21, pp. 1–5, 2016. [Online]. Available: http://jmlr.org/papers/v17/12-164.html

[44] J. Petterson and T. S. Caetano, ‘‘Reverse multi-label learning,’’ in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Red Hook, NY, USA: Curran Associates, 2010, pp. 1912–1920. [Online]. Available: http://papers.nips.cc/paper/3920-reverse-multi-label-learning.pdf

[45] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ‘‘SMOTE: Synthetic minority over-sampling technique,’’ *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jul. 2018.



PAYEL SADHUKHAN received the degree in computer science from Indian Statistical Institute (ISI), India, the bachelor’s degree in electronics and communication in 2011, and the M.Tech. degree. She has worked on crater detection during her M.Tech. degree. She is currently pursuing the Ph.D. degree in computer science with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata, India, where she is working on metamorphosing datasets.

SARBANI PALIT received the B.Tech. and Ph.D. degrees from the Indian Institute of Technology Kharagpur, India, and the master’s degree from the University of California at Santa Barbara, USA. She held a postdoctoral position at the University of California at Santa Barbara. Since 1999, she has been a Faculty at the Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata. Her primary research interests are signal processing, machine learning, and cryptography.