# A Survey on Multilabel Data ClassificationAlgorithms

<sup>1</sup>Vyas Pooja V

<sup>1</sup>M. E.Student<sup>1</sup>ComputerEngineering <sup>1</sup>Alpha College of Engineering and Technology (GTU), Ahmedabad,India <sup>1</sup>pooja.vyas121@gmail.com

Abstract—Classification of multi-label data is challenging task for machine learning community. In real world scenario, an object can be represented by more than one label at the same time unlike traditional single label classification.Multilabel classification has two methods: Pro blem Transformation Methods and Algorithm Adaptation Methods. The former method transforms multi-label data into single-label data and then classification is done whereas later method extends the already developed algorithm to directly cope up with multi-label data. In this paper various methods for Multilabel data classification will be discussed and compared with their advantages and disadvantages.

**Keywords---**Classification,Multilabel,Single label, Problem transformation,Algorithm adaptation.

## I. INTRODUCTION

Classification problems are aim to identify the characteristics that indicate the group to which each case belongs. This pattern can be used both to understand the existing data and to predict how new instances will behave. Data mining creates classification models by examining already classified data and inductively finding predictive patterns. The applications of classification includes, Protein function classification, Categorization according to economic activities, Music categorization, Semantic annotation of Image, Semantic annotation of Video data Semantic annotation of Audio data.[1]

Classification can be divided in two types: Single label classification and Multilabel classification

# A. Single Label Classification

The process to learn from a set of instances each associated with a unique class label from a set of disjoint class labels is known as single-label classification.Traditional supervised learning is one of the mostly-studied machine learning paradigms, where each example is associated with a single label. Formally, let X denote the instance space and Y denote the label space, the task of traditional supervised learning is to learn a function f:  $X \rightarrow Y$  from the training set  $\{(xi, yi) \mid 1 \le i \le m\}$ . Here,  $xi \in X$  is an instance characterizing the properties (features) of an object and  $yi \in$ Y is the corresponding label characterizing its semantics. one fundamentalassumption adopted Therefore, bv traditional supervised learning is that each examplebelongs to only one concept, i.e. having *unique* meaning [2]

# B. Multi Label Classification

The process is to learn from a set of instances where each instance belongs to one or more classes in Label space, known as multi-label classification. Multilabel classification is needed because, Although traditional supervised learning is successful, there are many learning tasks where the above simplifying assumption does not fit well, as real-world objects might be complicated and have multiple meanings simultaneously. Conventional classification considers only one label as an output but in reality an instance may be ambiguous i.e. related to more than one labels

Below are some example applications describing the above point: (1) A Text document may belong to more than one semantics like festival and environmental at the same time or an image may belong to trees, seaand mountain as well?, (2) In Text categorization, a news document could cover several topics such as sports, London Olympics, ticket sales and torch relay, (3) In music information retrieval, a piece of symphony could convey various messages such as piano, classical music, Mozart and Austria(4) In automatic video annotation, one video clip could be related to some scenarios, such as urban and building, and so on. [2], [4]

# II. MULTILABEL CLASSIFICATION METHODS

Multi-label classification methods can be grouped in two categories as proposed in [2]: (1) ProblemTransformationMethod (2) Algorithm Adaptation Method

First method transform multi-label classification problem into one or more single-label classification problem. It is

Algorithm independent method. Second method extends existing specific algorithm to directly handle multi-label data. Figure 1 gives the classification of the various problem transformation and algorithm adaptation methods

# A. Problem Transformation Methods

There exist several simple problem transformation methods that transform multi-label dataset into single-label dataset so that existing single-label classifier can be applied to multi-label dataset [5] All Problem transformation Methods will be examined using the multi-label example dataset of Table I. It consists of five instances which are annotated by one or more out of five labels, 11, 12, 13, 14, and 15. To describe different methods, attribute field is not so important, so it is omitted in the discussion. In Below section various methods are discussed which are popularly used in the literature

Instance	Label Set
1	{ 11,12 }
2	{ 11,12,13 }
3	{ 14 }
4	{ 11, 12, 15 }
5	{ 12,14 }

Table 1: Multilabel Dataset



Fig.1: Classification of Multilabel learning algorithms

#### 1) Simple Problem Transformation Methods:

There existSeveral simple problem transformation methods that transform multi-label dataset into single-label dataset so that existing single-label classifier can be applied to multi-label dataset [5] The Copy Transformation method replaces each multi-label instance with a single class-label for eachClass-label occurring in that instance. A variation of this method is dubbed copy-weight, which associates a weight to each produced instances. In these methods no information loss is there. But it increases the instances. So that it requires more processing on them. The Select Transformation method replaces the Label-Set (L) of instance with one of its member. Depending on which one member is selected from L, there are several versions exist, such as, select-min , which selects least frequent label from this set, select-max, which selects most frequent label from the set, select-random which randomly selects any label from the set . These methods are very simple but it loses some information of the labels belonging to the examples. The Ignore Transformation method simply ignores all the instances which has multiple labels and takes only singlelabel instances in training. There is major information loss

in this method. Because in real-world datasets, the vast number of examples are associated with more than one label. All simple problem transformation method does not consider label dependency. So they produce less efficient classification than some advanced methods, which considers label correlations into account to make their classifiers effective and accurate.



	-	_	$\geq$	$\mathcal{T}$			-	_	_	
Ex #	Label		Ex #	Label	1	Ex #	λ4	] [	Ex #	> Label set
1	14		1	l1		1	14	11	3	l1
2	14		2	13	1	2	13	1'		
3	11		3	l1	1	3	<b>l1</b>	1		
4	14	1	4	12		4	12	1		

Fig. 3:Select max, min, random and ignore

2) Binary Relevance:

1) A Binary Relevance is one of the

Most popular transformation methods which learns q binary classifiers, one for each label. BR transforms the original dataset into q datasets, where each dataset contains all the instances of original dataset. If particular instance contains label Lj  $(1 \le j \le q)$ , then it is labeled positive otherwise negative. Fig. 4 shows dataset that are constructed using BR for dataset of Table I. [3]



Fig. 4: Transformation using Binary Relevance

From these datasets, it is easy to train a binary classifier for each dataset but it cannot handle label dependency in the data.

## 3) Label Power set:

This method takes into account the label dependency. It considers each unique occurrence of set of labels in multilabel training dataset as one class for newly transformed dataset. For a new instance to classify, LP outputs the mostprobable class, which is actually a set of labels.

1	,	 -	
Ex #	Label set	Ex #	Label
1	{ <b>l1</b> , <b>l4</b> }	1	l1,4
2	{ <b>l3</b> , <b>l4</b> }	2	lar
3	{ <b>l1</b> }	_	13,4
4	$\{12, 13, 14\}$	3	lı
		 4	l2,3,4

#### Fig. 5: Transformation using Label Power set

But, its computational complexity depends on the number of distinct label-sets that exists in the training set. It has very high complexity especially for large values of m and q. and another limitation is that LP cannot predict unseen label-sets [3]

#### 4) Ranking by Pair wise Comparison:

Ranking by pairwise comparison [3]transforms the multilabel datasets into q(q-1)/2 binary label datasets, one for each pair of labels (Li, Lj),  $1 \le I < j < q$ . Each dataset contains those instances of original dataset that are annotated by at least one of the corresponding labels, but not by both. (Fig. 6). A binary classifier is then trained on each dataset. For a new instance, all binary classifiers are invoked and then ranking is obtained by counting votes received by each label.

Ex #	Label set
1	{ <b>l1</b> , <b>l4</b> }
2	{ <b>l3</b> , <b>l4</b> }
3	{ <b>l1</b> }
4	{12,13,14}

		/				$\geq$	$\sim$			_	_
Ex #	Label	Ex #	Label	Ex #	Label	Ex #	Label	Ex #	Label	Ex #	Label
1	l1.¬2	1	l1, ¬3	2	l¬1,4	1	l-2,3	1	l¬2,4	1	l¬3,4
3	11.72	2	l¬1,3	3	l1, ¬4			2	l¬2,4	-	-
4	l¬1.2	3	l1,¬3	4	l¬1,4						
	/-	4	l-1,3								
			100								

Fig. 6: Ranking by pair wise comparison

Different problem transformation methods has problem of Increased instances, Information loss due to Label independency, High complexity and also cannot predict unseen label sets and not consistent sometimes.

# 5) Random k label sets for Multilabel classification:

Label power set (LP), considers each distinct combination of labels that exist in the training set as a different class value of a single-label classification task. The computational efficiency and predictive performance of LP is challenged by application domains with large number of labels and training examples. In these cases the number of classes may become very large and at the same time many classes are associated with very few training examples. To deal with

these problems, this paperproposes breaking the initial set of labels into a number of small random subsets, called label setsand employing LP to train acorresponding classifier. The label sets can be either disjoint or overlappingdepending on which of two strategies is used to construct them. The proposed method is called RAkEL (RAndom k labELsets), where *k* is a parameter that specifies the size of the subsets. Empirical evidence indicates that RAkEL manages to improve substantially over LP, especially in domains with large number of labels and exhibits competitive performance against other high-performing multi-label learning methods. The main idea in this work is to randomly break a large set of labels into a number of small-sized label sets, and for each of them train a multi-label classifier using the LP method. For the multi-label classification of an unlabeled instance, the decisions of all LP classifiers are gathered and combined. For simplicity, we only consider labelsets of the same size, k. A labelset  $R \subseteq L$  with k = |R| is called klabelset. Therefore, the proposed approach is dubbed RAkEL (Random k labelsets). This paper examines the construction of two different types of labelsets: a) disjoint (RAkELd), and b) overlapping (RAkELo) Experiments were conducted on 8 multi-label datasets. That are scene dataset,tmc2007 dataset ,medical dataset, dataset, yeast enron dataset, bibtex dataset, reuters (rcv1) dataset. This paper has presented a new multi-label classification method, called RAkEL, that learns an ensemble of LP classifiers, each one targeting a different small random subset of the set of labels. The motivation was the computational efficiency and predictive performance problems of the simple and effective standard LP method, when faced with domains with large number of labels and training examples. they examined both disjoint and overlapping subsets and found that both lead to improved results over the standard LP method, especially in domains with many labels. they also found that overlapping subsets lead to better results compared to disjoint ones, due to the classifier fusion process that takes place for each label. due to randomization it is inconsistent in some cases.[12]

#### B. Algorithm Adaptation Methods

In classification problems, Multilabel classification has gained interest among researchers and lots of work had been done in this regard to provide better systems using different approaches.

#### 1) Multi-Label k-Nearest Neighbor (ML-kNN) [8]:

Basically this algorithm adapts *k-nearest neighbor* techniques to deal with multi-label data, where *maximum a posteriori* (MAP) rule is utilized to make prediction by reasoning with the labeling information in the neighbors. It is a *first-order* approach which reasons the relevance of each label separately. ML-kNN has the advantage of inheriting merits of both *lazy learning and Bayesian reasoning*: a) decision boundary can be adaptively adjusted due to the *varying neighbors* identified for each unseen instance; b) the classimbalance issue can be addressed

due to the prior probabilities estimated for each class

labels.In this paper, a lazy learning algorithm named ML-KNN, which is the multi-label version of KNN, is proposed. Based on statistical information derived from the label sets of an unseen instance's neighboring instances, i.e. the *membership counting* statistic, ML-KNN utilizes MAP principle to determine the label set for the unseen instance. Experiments on three real-world multi-label learning problems, i.e. Yeast gene functional analysis, natural scene classification and automatic web page categorization, show that ML-KNN outperforms some well-established multi-label learning algorithms. In this paper, the distance between instances is simply measured by *Euclidean metric*. Therefore, it is interesting to see whether other kinds of distance metrics could further improve the performance of ML-KNN. On the other hand, investigating more complex statistical information other than the *membershipcounting* statistic to facilitate the usage of maximum a posteriori principle is another interesting issue for future work.

## 2) Ranking Support Vector Machine (Rank-SVM)[6]:

Rank-SVM is a Second-order strategy where the task of multi-label learning is tackled by considering pairwise relations between labels, such as the ranking between relevant label and irrelevant label. a stacking-style procedure would be used to determine the thresholding where it assume a linear model with q-dimensional stacking vector storing the learning system's real-valued outputs on each label and then linear least squares problem is Solved. target output of the stacking model which bipartitions labelset into relevant and irrelevant labels for each training example with minimum misclassifications. Basically, this algorithm adapts maximum margin strategy to deal with multi-label data, where a set of linear classifiers are optimized to minimize the empirical ranking loss and enabled to handle nonlinear cases with kernel tricks. Main contribution is the definition of a ranking based SVM that extend the use of the kernel to many problems in the area of Bioinformatics and Text Mining. It defines the margin over hyperplanes for relevant-irrelevant label pairs. Rank-SVM inherits the benefits of kernel learning to be responsible to nonlinear classification problems, it is a second order approach. there could be *kernel selection problem* which can be taken into account in further research.

#### 3) Backpropogation multilabel learning(BPMLL)[7] :

BP-MLL is an extension of the popular back-propagation algorithm for multi-label learning. The main modification is the introduction of a new error function that takes multiple labels into account. Given multi-label training set,

$$S = \{(xi, Yi) \mid 1 \le i \le m\}$$

The global training error E on S is defined as:

$$E = \sum_{i=1}^{m} Ei = \sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(k,l)\in Y_i\times\overline{Y_i}} \exp(-(c_k^i - c_l^i))$$

Where, Ei is the error of the network on (xi, Yi) and cij is the actual network output on xi on the jth label. In experiments the learning rate is set to be 0.05. Hidden units of the network is set to be 20 percent of the number of input units. Tenfold cross validation is performed on this dataset. In detail, the original data set is randomly divided into 10 parts, each with approximately the same size. In each fold, one part is held out for testing and the learning algorithm is trained on the remaining data. The above process is iterated 10 times so that each part is used as the test data exactly once, where the averaged metric values out of 10 runs are reported for the algorithm. It is a *second-order* approach which defines the error function with respect to relevantirrelevant label pairs. BP-MLL aim to minimize *surrogate* ranking loss, and isimplemented in exponential form BP-MLL hasstrong ability in handling nonlinear classification cases while may get stuck in local minima dueto the gradient descent-based minimization procedure.

## 4) Improved BPMLL [9] :

This paper, propose some improvements of the error function used in [9]. The introduced modification is integration of the threshold value into the error function used in BP-MLL. The last output neuron's value is interpreted as the threshold. The meaning of the remaining output neurons is the same as in case of using the BP-MLL method. The modifications concern the form of the global error function used in BP-MLL. The modified classification system is tested in the domain of functional genomics, on the yeast genome data set. Experimental results show that proposed modifications visibly improve the performance of the neural network based multilabel classifier. The results are statistically significant. Overall, including the threshold values into the error function and considering differences between the rank values and the thresholds proved to be a promising direction for improvement of the multilabel classifier performance.

#### 5) Multilabel Radial basis function (ML-RBF) [10]:

ML-RBF is derived from the traditional radial basis function (RBF) methods. Briefly, the first layer of an Ml-RBF neural network is formed by conducting clustering analysis on instances of each possible class, where the centroid of each clustered groups is regarded as the prototype vector of a basis function. After that, second layer weights of the MI-rbf neural network are learned by minimizing a sum- of-squares error function. Specifically, information encoded in the prototype vectors corresponding to all classes are fully exploited to optimize the weights corresponding to each specific class. Radial basis function (RBF) consist of two layers of neurons. each hidden neuron (basis function) in the first layer is associated with a prototype vector while each output neuron corresponds to a possible class. Generally a two-stage procedure is employed to train an RBF neural network, where the basis functions are learned by performing clustering analysis on training instances and second-layer weights are optimized by solving a linear problem. Firstly, the hidden layer of MI-RBF neural network is constituted by performing k-means clustering on training instances of each possible class, After that, the weights between hidden and output layers are determined through minimizing the sum-of-squares error function. Finally, the test multi-label instance is fed to the trained neural network for prediction.

6) Multilabel

probabilistic neural network (MLPNN)[11] : This PNN was proposed aiming at executing automatic classification of economic activities, which is the focus of the article. This paper has presented a slightly modified version of the standard structure of the probabilistic neural network., so that they could deal with the multi-label problem faced in this work. They have chosen the PNN classifier because of its implementation simplicity and high computational speed in the training stage, when compared to other algorithms. The original PNN algorithm was designed for single-label problems. Thus, they slightly modified its standard architecture, so that it is now capable of solving multi-label problem addressed in this work. In their modified version, instead of four, the network is composed of only three layers: the input layer, the pattern layer and the summation layer. In the first set of experiments, they have used 11 multi-labeled databases of text from yahoo.com domain. Initially, each database passed by a process of simple feature selection based on the number of documents that contains a specific term to reduce the dimensionality of each one. Actually, only 2% terms with highest document frequency were selected and the others were removed. Then, each document was represented by a vector, where each dimension represents the number of times a word appeared in the document. In addition, each database has 2000 samples to training and 3000 to test, and the average number of classes is 30 Second experiment is done on Economic activities database dataset containing 3264 documents of free text business descriptions of Brazilian companies categorized into a subset of 764 CNAE categories. This dataset was obtained from real companies placed in Vitoria County in Brazil. The problem of classifying huge number of economic activities description in free text format every day is a huge challenge for the Brazilian governmental administration. They performed a comparative study of PNN and other classifiers on a Yahoo and economic activities databases. A direction for a future work includes a study to improve the PNN's performance, such as, to examine the correlation, to use techniques to feature selection and selection of the best training samples. Furthermore, researches to turn the PNN working in online environment, keeping the reduced dimension.

 $\frac{1}{n} \sum_{i=1}^{n} \left[ \frac{|\mathbf{y}_i \cap \mathbf{z}_i|}{|\mathbf{y}_i \cup \mathbf{z}_i|} \right]$ 

# III. EVALUATION MEASURES USED IN MULTILABEL LEARNING

Evaluation of some learning algorithm is a measurement of how far the learning system predictions are from the actual class labels, tested on some unseen data. To capture the notion of partially correct, one strategy is to evaluate the average difference between the predicted labels and the actual labels for each test example, and then average over all examples in the test set. This approach is called example based evaluations [2].

If a classifier is able to learn **the ranking** of the predicted labels, then the following metrics are common to evaluate the performance of the algorithm:

*One Error* (*O*): One error measures how many times the top ranked predicted label is not in the set of true labels of the instance.

$$o = \frac{1}{n} \sum_{i=1}^{n} I(\arg\min_{\lambda \in L} r_i(\lambda \notin Y_i^l))$$

where, I is an indicator function and  $r(\lambda)$  is the predicted rank of class label  $\lambda$  for an instance xi. The top ranked predicted label is the label the classifier is most confident on and getting it wrong would clearly be an indication of overall lower performance of the classifier. Ideally, we would expect the perfect performance when one error = 0; practically the smaller the value of one error, the better the performance. One error is the classification error for single label classification.

*Coverage (C):* For some applications, it is often important to get all the true labels predicted even with a few extra false positive predicted labels (e.g. fraud detection). Coverage is the metric that evaluates how far on average a learning algorithm need to go down in the ordered list of prediction to cover all the true labels of an instance. Clearly, the smaller the value of coverage, the better the performance.

$$c = \frac{1}{n} \sum_{i=1}^{n} \max_{\lambda \in Y_i} r_i(\lambda) - 1$$

*Ranking Loss (RL):* Instead of comparing two label subsets, ranking loss evaluates the average proportion of label pairs that are incorrectly ordered for an instance.

$$RL = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|Y_i^l| |\overline{Y_i^l}|} |(\lambda_a, \lambda_b) : r(\lambda_a) > r(\lambda_b), (\lambda_a, \lambda_b) \in Y_i^l X \overline{Y_i^l}||$$

Similar to one error, the smaller the ranking loss, the better the performance of the learning algorithm.

Average Precision (AP): For each relevant label, average precision computes the proportion of relevant labels that are ranked before it, and finally averages over all relevant labels.

$$AP = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{|Y_i^l|} \sum_{\lambda \in Y_i^l} \frac{|\{\lambda, ' \in Y_i^l\} : r_i(\lambda, ' \leq r_i(\lambda))|}{r_i(\lambda)}$$

If a classifier is able to classify the predicted labels, then the following metrics are common to evaluate the performance of the algorithm:

Accuracy (A): Accuracy for each instance is defined as the proportion of the predicted correct labels to the total number (predicted and actual) of labels for that instance. Overall accuracy is the average across all instances.

*Precision (P):* Precision is the proportion of predicted correct labels to the total number of actual labels, averaged over all instances.

$$\frac{1}{n} \sum_{i=1}^{n} \left[ \frac{|\mathbf{y}_i \cap z_i|}{z_i} \right]$$

Hamming Loss (HL): Hamming Loss reports how many times on average, the relevance of an example to a class label is incorrectly predicted. Therefore, hamming loss takes into account the prediction error (an incorrect label is predicted) and the missing error (a relevant label not

predicted), normalized over total number of classes and total number of examples.

$$HL = \frac{1}{kn} \sum_{i=1}^{n} \sum_{i=1}^{k} [I(l \in Z_{i} \land l \notin Y_{i}) + I(l \notin Z_{i} \land l \in Y_{i})]$$

Where I is the indicator function. Ideally, we would expect hamming loss, HL = 0, which would imply no error; practically the smaller the value of hamming loss, the better the performance of the learning algorithm. If Yi = 1 and Zi = 1, then we have a single label multi-class classification problem. It is easy to note that, in that case, hamming loss is 2/k times of the classification error

*Recall (R):* Recall is the proportion of predicted correct labels to the total number of predicted labels, averaged over all instances.

$$\frac{1}{n}\sum_{i=1}^{n}\left[\frac{|\mathbf{y}_{i} \cap z_{i}|}{\mathbf{y}_{i}}\right]$$

*F1-Measure (F):* Definition for precision and recall naturally leads to the following definition for F1-measure i.e. harmonic mean of precision and recall

$$F_1 = \frac{1}{n} \sum_{i=1}^n 2 \frac{|Y_i \cap Z_i|}{|Y_i| + |Z_i|}$$

As in single label multi-class classification, the higher the value of accuracy, precision, recall and F1- score, the better the performance of the learning algorithm

IV.	COMPARISON OF PROBLEM TRANSFORMATION AND
	ALGORITHM ADAPTATION METHODS

Methods	Advantage	Disadvantage
Copy, Copy- weight	<ul> <li>Simple</li> <li>No information loss</li> </ul>	<ul> <li>Increased instances</li> </ul>
Max, Min, Random	<ul><li>Simple</li><li>Decreased instances</li></ul>	<ul> <li>Information loss</li> </ul>
Ignore	<ul> <li>Very simple</li> </ul>	<ul> <li>Major information loss</li> </ul>
BR	Less complex	<ul> <li>Label independency</li> </ul>
LP	<ul> <li>Consider label dependency</li> <li>Depends on the number of distin ct label sets</li> </ul>	<ul> <li>High complexity</li> <li>Can't predict unseen label sets</li> </ul>
RPC	<ul> <li>Improved ranking performance,Cla ssification and ranking</li> </ul>	<ul> <li>Need to quey</li> <li><sup>2</sup> binary</li> <li>models (time complexity)</li> </ul>

RAkEL	•	Can predict unseen label sets	•	Randomly creates label s ets, so not consistent
RANK- SVM	•	effectively handles non- linear classificati on problem	•	kernel selection problem
BP-MLL	1	strong ability in handling nonlinear classification	•	may get stuck in local minima
ML-KNN		Effectively address class imbalance problem .		Needs more powerful distance metric
ML-RBF		Correlations between different classes are appropriately addressed	00	Need to enhancing the generalization abilities
ML-PNN		Implementation simplicity High computational speed		Need correlations for better feature selection

Table. 2: Comparison of Methods

# V. CONCLUSION

Here, we discussed classification problems in data mining. there are two types of classification problems. traditional single label classification and multilabel classification problems. where in today's world the problem of multilabel classification must be addressed, because an object or an instance may belong to more than one labels at the same time rather than a single label. so our focus is on multilabel classification. Also there are two methods for solving Multilabel classification problems a) problem transformatio n method and b) algorithm adaptation method. We discussed both methods here with their prime advantages and disadvantages. Also various evaluation measures for measuring algorithm's performance have been discussed.

#### REFERENCES

- [1] Han, Jiawei, Micheline Kamber, and Jian Pei. *Data mining: concepts and techniques.* Morgan kaufmann, 2006.
- [2] Sorower, Mohammad S. A literature survey on algorithms for multi-label learning. Technical report, Oregon State University, Corvallis, OR, USA , 2010
- [3] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Mining multi-label data." *Data mining and knowledge discovery handbook*. Springer US, 2010.
- [4] Zhang, M., and Z. Zhou. "A Review on Multi-Label Learning Algorithms.",2013
- [5] Tsoumakas, Grigorios, and Ioannis Katakis. "Multilabel classification: An overview." International Journal of Data Warehousing and Mining (IJDWM) 3.3 ,2007

- [6] Elisseeff, André, and Jason Weston. "A kernel method for multi-labelled classification." Advances in neural information processing systems. 2001.
- [7] Zhang, Min-Ling, and Zhi-Hua Zhou. "Multilabel neural networks with applications to functional genomics and text categorization." Knowledge and Data Engineering, IEEE Transactions on 18.10,2006
- [8] Zhang, Min-Ling, and Zhi-Hua Zhou. "ML-KNN: A lazy learning approach to multi-label learning." Pattern Recognition 40.7, 2007
- [9] Grodzicki, Rafał, Jacek Mańdziuk, and Lipo Wang. "Improved multilabel classification with neural networks." Parallel Problem Solving from Nature– PPSN X. Springer Berlin Heidelberg, 2008.
- [10] Zhang, Min-Ling. "ML-RBF: RBF neural networks for multi-label learning." Neural Processing Letters,2009[11] Ciarelli, Patrick Marques, et al. "Multi-label text
- [11] Ciarelli, Patrick Marques, et al. "Multi-label text categorization using a probabilistic neural network." International Journal of Computer Information Systems and Industrial Management Applications 1,2009
- [12] Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Random k-labelsets for multilabel classification." *Knowledge and Data Engineering, IEEE Transactions on* 23.7, 2011