Multi layered Stacked Ensemble Method with Feature Reduction Technique for Multi-Label Classification

Hemavati¹, V Susheela Devi² and R Aparna³

¹ Department of CSE, Siddaganga Institute of Technology, Tumakuru, Karnataka, India
² Department of CSA, Indian Institute of Science, Bangaluru, Karnataka, India.
³ Department of CSE, Siddaganga Institute of Technology, Tumakuru, Karnataka, India
E-mail: hemavd@sit.ac.in, susheela@iisc.ac.in, raparna@sit.ac.in

Abstract. Nowadays, multi-label classification can be considered as one of the important challenges for classification problem. In this case instances are assigned more than one class label. Ensemble learning is a process of supervised learning where several classifiers are trained to get a better solution for a given problem. Feature reduction can be used to improve the classification accuracy by considering the class label information with principal Component Analysis (PCA). In this paper, stacked ensemble learning method with augmented class information PCA (CA-PCA) is proposed for classification of multi-label data (SEMML). In the initial step, the dimensionality reduction step is applied, then the number of classifiers have to be chosen to apply on the original training dataset, then the stacking method is applied to it. By observing the results of experiments conducted are showing our proposed method is working better as compared to the existing methods.

1. Introduction
Classification is a problem in machine learning and generally it is single label classification i.e., there will be only one class label for each pattern. In multi-label classification, there will be multiple class labels and each object or instance may be mapped with multiple class labels. The classification of multi-label problem is becoming an important area of research for many applications such as text[7][8], image[8][9], movie genres classification etc. An ensemble technique is achieved with the combination of diverse classifiers and they are shown to be effective methods for multi-label classification problems. The real challenge is in building of an effective multi-label ensemble of classifiers that uses the subset of labels from the given label set for each member of the ensemble[10]. As we are aware in our daily life, for buying any product or choosing any hospitals, we usually read the reviews or we take opinions or individual experiences from many people so that we can have better selection over many available choices.

There will always be the problem of which classifier to choose among many algorithms such as Naive Bayes(NB), Support Vector Machine(SVM), Decision Tree(DT) etc. In every case, it
is necessary to choose the right classifier with minimal error. By using an ensemble of different classifiers and combining their outputs by different ensemble methods[2][12], we can reduce the risk of random selection which may select one of the poorly performing classifiers. However it is important to note that the combination of multiple classifiers may not always perform best. There is no guarantee but it may reduce the risk of poorly performing model selection. The curse of dimensionality is one of the problem to be considered for classification.

In this paper we are using class augmented PCA method to reduce the dimensionality for multi-label data. This method is proposed in [27]. The following sections of the paper will describe related work, proposed method, evaluation metrics and analysis of the experiment results.

2. Related work
2.1. Multi-label Classification
Multi-label classification can be formally described as for given data set X where each instance \(X_i \in \mathbb{R}^n\) has a associated label \(Y_i \subseteq Y\), where \(Y = (l_1, l_2, ..., l_h)\) set of labels, then the model has to learn a function \(f\), for mapping a given instance \(X_j \in X\), to a subset of labels i.e. \(f(X_j) = Y_j\) such that \(Y_j \in Y\). There are two different types of approaches to work with multi-label classification problems. They are Problem Transformation[5] and Algorithm Adaptation[4] approach.

2.1.1. Problem Transformation  This approach basically transforms the multi-label data to a single label data, then we can make use the traditional single label algorithm for classification. The final output will be an ensemble of the results given by each classifier. There are many commonly used methods of problem transformation are Binary Relevance Method[18][27] in which the dataset is duplicated for each unique label as single label instances. The new labels will indicate the presence or absence of the true label for the instance. The number of classifiers used here will be the same as the number of labels. This method is criticized as it assumes that there are no label dependencies. Label Power Set Method[4] considers each unique combination of labels found in the training set as a single label. The test output will be one of these unique label combinations. The problems associated with such an approach is the complexity and class imbalance, as the number of data points with a specific label combination can be sparse in a given dataset.

The following are some of the basic classification algorithms used with binary relevance approach on multi-label data. The performance of the proposed method is compared with these algorithms.

**Binary Relevance k-Nearest Neighbours (BRkNN)** [24] For multi-label classification with the binary relevance approach for each label Knn model is used separately.

**Binary Relevance Decision Tree (BRDT)** [20] Basically used for improved generalization capability of the model, it finds one decision tree for each label.

**Binary Relevance Naive Bayes(BRNB)** [18][23] For each label associated with label set \(Y\), binary Naive Bayes classifier is learned \(H : d \rightarrow \{0, 1\}\), for classification of new data point. This method outputs a set of labels which is the union of different single label Naive Bayes classifiers. Apart from these algorithms there are some variants of stacking concept has been proposed in the literature. In this paper we are trying to extend the concept of stacking with multi layered approach feature reduction technique.
2.1.2. Algorithm Adaptation

This approach adapts traditional classification algorithms to work directly on multi-label data without considering transformation method. One such algorithm is AdaBoost.MH [6] which is a boosting algorithm for multi-label problems aimed at minimizing the hamming loss. In ML-kNN (Multi-label kNN) [11], statistical information such as prior probabilities and posterior probabilities are obtained from the training data. The former indicates the number of data points having a specific label and the latter gives the number of training instances with a specific label, whose $k$ nearest neighbors have exactly a given number of that label. Finally Bayesian rule (maximum a-posterior) is used on these statistics to predict the algorithm output. The posterior probabilities for the test input can also be used to form a ranking vector for the labels predicted. All needed information is obtained using frequency counting from the training data. Random k-Label sets (RAkEL) [10] This method provides an improved performance over the LP [4] method mentioned earlier. The major difference comes in the fact that the label sets are first broken down into smaller finite subsets. Then a Label Power set classifier is used for each of these subsets. The decisions of all the classifiers are combined for the final prediction.

2.2. Dimensionality reduction algorithms

There are different dimensionality reduction methods are available like for example, MDDM, multi-label informed feature selection, kernal dependency estimation etc., in this paper we are using class augmented pca for feature reduction.

2.3. Ensemble methods and algorithms

Ensemble methods are distinguished as dependent and independent ensemble frameworks [13]. In the former case the result of the base classifier can be used in constructing the next level classifier. In AdaBoost [14] AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. Each instance in the training dataset is weighted. The initial weight is set to: $W(X_i) = 1/n$. Where $X_i$ is the $i^{th}$ training instance and $n$ is the number of training instances. Whereas meta-learning methods are mostly more suitable for those classifiers which are consistently correctly classifying or misclassifying certain instances. Some of the weighting methods are majority voting, performance weighting, distribution summation, Bayesian combination, entropy weighting etc., described in [13].

In this paper we are using meta learning method called Stacking [17]. Stacking is basically used to achieve the maximum generalization accuracy [17]. It is usually used to combine the models built by different classifiers for creating the meta dataset that is, instead of using the input features from the original dataset, it uses the predicted results of the base classifiers as input features and class labels will be taken as it is from the original dataset. After this, the meta classifier will produce final prediction from the meta dataset.

3. The proposed method

Multi layered stacked ensemble for classification of multiclass problems has been described in [18] and Stacked MLKnn for multi-label data in [22] for lazy learning approach. In this paper we are extending, the stacked ensemble method for multi-label data (SEMML) with feature reduction technique. Fig.1 shows the flow architecture of our method SEMML. This initially we apply the class augmented PCA for feature reduction, then we apply the problem transformation method.
Here we considered binary relevance[5][18] which will try to transform the multi-label problem to a simple binary class problem, so that the base learner can be applied. Then different base classifiers have to be chosen to apply on the original dataset, to generate the new dataset for the next level, the predicted results need to be stacked and the actual label set from the original dataset will be augmented. Now this new dataset will be the input for the metalearned to get the final prediction.

### 3.1. Algorithm for stacked ensemble method for classification of multi-label data

**Algorithm 1**: Algorithm for stacked ensemble method for classification of multi-label data

```plaintext
1: Input: \( D = \{X_i, Y_i\}_{i=1}^n \), where \( X_i \in \mathbb{R}^n, Y_i \subseteq Y \) (set of all labels).
2: Output: Stacked Ensemble classifier
3: begin
4: Step 1: Apply the CA, PCA(X) as described in section 3.2.
5: Step 2: Transform the multi-label data to binary class problem
6: \( D_{BR} \leftarrow BR(D_{CA,PCA}) \)
7: Step 3: Apply the base classifiers on \( D_{BR} \)
8: for \( i \leftarrow 1 \) to \( m \)
9: Train the base classifier \( BC_i \) based on \( D_{BR} \).
10: end for
11: Step 4: Construct new dataset from the output of base classifier
12: for \( j \leftarrow 1 \) to \( m \)
13: Construct new dataset \( D_2 \) that should contain
14: \( D_2 = (X'_i, Y_i) \)
15: where \( X'_i = \{BC_1(X_i), BC_2(X_i), ..., BC_m\} \)
16: and \( Y_i \) from original dataset \( D \).
17: end for
18: Step 5: Train the metalearned based on new dataset \( D_2 \) generated
19: in Step 4 for predicting the final prediction.
20: learn \( ML \) based on \( D_2 \).
21: Step 6: return \( ML \)
end
```
In Algorithm 1, **Step 1** will take the input feature set and transform it to reduced feature set by considering the class label information. Then in **Step 2**, by applying the transformation method i.e., binary relevance method, the problem is decomposed into binary class problem and \( n \) number of base classifiers will be applied on transformed data \( D_{BR} \). **Step 4** constructs the new dataset \( D_2 \) by stacking the output of each base classifier used in **Step 3** and augmenting the class labels from the original dataset \( D \). In **Step 5** the meta learner is trained based on new dataset \( D_2 \) constructed in **Step 4**. Here the meta learner may be meta decision tree (MDT)[25] for predicting the final prediction.

3.2. **Mathematical Model**

Let \( D \) be the original input data consisting of \( n \) points denoted as follows
\[
D = (X_1, Y_1), (X_2, Y_2)....(X_n, Y_n), \text{ where } X_i \in X \text{ and } Y_i \in Y
\]

3.2.1. **CA,PCA for multi-label data**

PCA determines the orthogonal vectors from the projected data. The eigen vectors of these projected data may not be sufficient for classification as there is no consideration of label information, by augmenting the new axis which is orthogonal to the existing axis.

The following steps summarizes the usage of PCA

- Apply the PCA to the feature set of the original dataset
  \[
  X^P = E_{PCA1} \times X
  \]
  \( (1) \)

- Encode the label information i.e., \( Y_i \) to be the vector of 0’s and 1’s. Then the following equations to be solved.
  \[
  \frac{1}{N} \sum_{i=1}^{N} CY_{ij} = 0 \quad (2)
  \]
  \[
  \frac{1}{N-1} \sum_{i=1}^{N} CY_{ij}^2 = \frac{N_j}{N} \sigma^2 \quad (3)
  \]

- Augment the data with encoded labels.
  \[
  X_{AUG} = [X^P \ CY]
  \]
  \( (4) \)

- Maximize the importance of class label information
  \[
  X_{i,j}^{norm} = \frac{X_{i,j}^P - mean_j}{stddev_j}
  \]
  \( (5) \)

- Now apply the PCA once again to determine the transformation Matrix.
  \[
  X^{Trans} = E_{pca2} \times X^{norm}
  \]
  \( (6) \)

**Base Level:** Apply the transformation method on \( D \)

\[
D_{BR} = BR(D_{CA,PCA})
\]

Let \( BC = \{BC_1, BC_2,...., BC_m\} \) be set of base classifiers.

Train the base classifiers on \( D_{BR} \). The base classifier \( BC_1 \) maps the data points into subset of classes in \( Y \).

\[
BC_1 = D_{BR} \rightarrow Y.
\]

(8)
Stacked ensemble: Let \( D_2 \) be the new data set to be constructed by stacking the predicted output of the different base classifiers.

\[
D_2 = (X'_i, Y_i)
\]  

(9)

where \( X'_i = \{BC_1(X_i), BC_2(X_i)...BC_m(X_i)\} \) and \( Y_i \) from original dataset \( D \).

Consider the following example:

Table 1. Example of initial training dataset

|  | X | Y |
|---|---|---|
| X1 | 1 | 0 | 0 | 1 |
| X2 | 1 | 0 | 1 | 1 |
| X3 | 0 | 0 | 1 | 0 |
| X4 | 1 | 1 | 1 | 0 |
| - | - | - | - | - |
| \( X_n \) | 1 | 1 | 0 | 0 |

Table 2. Example of stacking new features

| NewX | Y |
|------|---|
| \((X_1, 1 0 1 1, 1 1 0 1)\) | 1 | 0 | 0 | 1 |
| \((X_2, 1 1 1 0, 1 0 0 1)\) | 1 | 0 | 1 | 0 |
| \((X_3, 1 1 1 0, 1 0 0 1)\) | 0 | 0 | 1 | 0 |
| \((X_4, 1 1 1 0, 1 0 0 1)\) | 1 | 1 | 1 | 0 |
| \(-, -, -\) | - | - | - | - |
| \((X_n, 1 1 0 0, 1 0 1 0)\) | 1 | 1 | 0 | 0 |

Table I shows the training dataset where each instance \( X_i \) is associated with a label \( Y_i \) and Table II shows the new dataset generated from initial training dataset after applying base classifiers. New \( X_i \) will basically consist of concatenation of \( BC_1(X_i), BC_2(X_i)...BC_m(X_i) \), etc., where \( BC_1(X_i) \) is the predicted labels of \( X_i \) using classifier1. Here we considered two base classifiers, there could be more than two base classifiers and note that if we are using binary relevance then the number of base classifiers should be equal to the number of labels. The dimension of new dataset is going to change according to the number of labels present, in this case four for each classifier. while constructing the next level new dataset we can either consider the prediction probabilities or binary prediction in the above example for simplicity we have considered binary prediction. If maximum number of labels is \( l \) and there are \( m \) classifiers, the number of elements in New \( X_i \) will be \( l * m \). This new dataset is used for meta learning to get the final prediction.

Meta Learning

The metalearning classifier is a learning algorithm which is used to induce a meta-level model for combining the predictions of base-level models. Let \( ML \) be the metalearning classifier. In this paper we have considered meta decision tree(MDT) as metalearning classifier.

\[
ML = MDT(D_2)
\]  

(10)
Once after constructing the dataset \( D_2 \), the metalearning classifier is applied to build the model based on dataset \( D_2 \), now for the test data, at the base level transforms the test data to form new \( X_t \), the final prediction for the new data point \( X_t \), ML is used to decide the final label relevance based on \( D_2 \).

**4. Evaluation Metrics**

In multi-label classification methods, the output is a set of labels, so the normal evaluation metrics used for analyzing single label classification algorithms cannot be applied directly.

Consider a multi-label data set consisting of \( N \) points \((X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)\) and let \( f \) be our multi-label classifier. Let \( f(X_i) \) denote the predicted class labels, on the instance \( X_i \) by \( f \). Assume there are \( h \) labels possible which are denoted by the set of all labels \( Y \). The following list includes some of the commonly used evaluation metrics explained using these assumptions. The metrics used for evaluation differs according to the nature of the target problem.

**Precision**\[^3\] \[^5\] for an instance is the ratio of correct prediction of the labels to the total prediction of the labels. It is averaged over all instances of the data set.

\[
\frac{1}{N} \sum_{i=1}^{N} \frac{|f(X_i) \cap Y_i|}{|f(X_i)|}
\] \hspace{1cm} (11)

**Recall**\[^3\] \[^5\] for an instance is the ratio of correct prediction of the labels to the total number of original labels. It is then averaged over all instances of the data set.

\[
\frac{1}{N} \sum_{i=1}^{N} \frac{|f(X_i) \cap Y_i|}{|Y_i|}
\] \hspace{1cm} (12)

**F1-Measure**\[^3\] \[^5\] for an instance is the ratio of the harmonic mean of precision and recall. It is then averaged over all instances of a data set.

\[
\frac{1}{N} \sum_{i=1}^{N} \frac{2 \times |f(X_i) \cap Y_i|}{|Y_i| + |f(X_i)|}
\] \hspace{1cm} (13)

**Hamming Loss**\[^3\] \[^5\] gives a measure of the number of times labels are miss classified for an instance. It is found by taking the symmetric difference or XOR between the predicted labels and the original labels.

\[
\frac{1}{N} \sum_{i=1}^{N} \frac{|f(X_i) \oplus Y_i|}{|Y_i| + |f(X_i)|}
\] \hspace{1cm} (14)

The performance of algorithm gets better with the lower in value of hamming loss i.e., zero indicates a perfect classification. The measures such as Precision, Recall, F1 measure or any measure which can be used to evaluate binary classification can be computed on a per label basis and used for multi-label evaluation after averaging over all labels possible.

**5. Experimental Results**

To find out the performance of the SEMML we have chosen 10 standard multi-label datasets from the Mulan and Meka Repository[26] and each column of Table 1 shows the properties of datasets. We have tested on all these different domain datasets for the effectiveness of the proposed method.
Table 3. description of the datasets

| Dataset Name | #instances | #Features | #Labels |
|--------------|------------|-----------|---------|
| yeast        | 2417       | 103       | 14      |
| scene        | 2407       | 294       | 6       |
| emotions     | 593        | 72        | 6       |
| music        | 592        | 77        | 6       |
| corel5k      | 5000       | 499       | 374     |
| enron        | 1702       | 1001      | 53      |
| bibtex       | 7395       | 1836      | 159     |
| flags        | 194        | 26        | 12      |
| birds        | 322        | 260       | 20      |
| genbase      | 662        | 1186      | 27      |

Table 4, Table 5 and Table 6 shows the experimental results for the different algorithms. In this paper we experimented with traditional as well as proposed methods on different datasets listed in Table 3. The algorithms used are BRNB, BRDT, BRKnn, MLKnn described in Section II. These algorithms are compared with our proposed method SEMML.

In all cases cross validation is used to avoid overfitting problem. Here we have used the MEKA tool for implementation of BRNB, BRDT, BRKnn and MLKnn is implemented separately with scikitmultilearn package. The proposed method SEMML has been implemented with MEKA as well as scikitmultilearn package. The results of all these algorithms have been listed in the tables with respect to the performance metrics Average precision, F1score and Hamming loss which are described in Section 4. As we can observe in Figure. 2 to Figure. 4 and Table 4 to Table 6, the performance of SEMML is better with respect to average precision, F1score and Hamming loss in comparison with all other algorithms. In all cases except one, the average precision of SEMML is much higher than the other algorithms. In the majority of datasets, the F1score of SEMML is better than the other algorithms. Hamming Loss of SEMML as shown in Figure. 4 and Table 6 is lower than all algorithms in most cases. This shows the superiority of our algorithm, since Hamming loss is one of the important criteria for multi-label performance evaluation. In Figure. 5, the performance of SEMML is far superior to the other algorithms for all the metrics.

![Average Precision of different algorithm on each dataset](image-url)
Table 4. average precision of different multi-label classification algorithms

| Dataset | BRNB | BRDT | BRKnn | MLKnn | SEMML |
|---------|------|------|-------|-------|-------|
| yeast   | 0.654| 0.562| 0.573 | 0.580 | 0.738 |
| scene   | 0.635| 0.654| 0.729 | 0.711 | 0.851 |
| emotions| 0.735| 0.637| 0.571 | 0.581 | 0.746 |
| music   | 0.755| 0.615| 0.595 | 0.689 | 0.778 |
| corel5k | 0.174| 0.167| 0.160 | 0.279 | 0.294 |
| enron   | 0.163| 0.187| 0.170 | 0.168 | 0.191 |
| bibtex  | 0.143| 0.114| 0.180 | 0.175 | 0.198 |
| flags   | 0.622| 0.774| 0.542 | 0.562 | 0.801 |
| birds   | 0.231| 0.324| 0.218 | 0.475 | 0.488 |
| genbase | 0.895| 0.798| 0.840 | 0.827 | 0.814 |
| Average score | 0.477 | 0.465 | 0.437 | 0.495 | 0.572 |

Table 5. F1-score of different multi-label classification algorithms

| Dataset | BRNB | BRDT | BRKnn | MLKnn | SEMML |
|---------|------|------|-------|-------|-------|
| yeast   | 0.516| 0.584| 0.602 | 0.350 | 0.65  |
| scene   | 0.581| 0.602| 0.306 | 0.656 | 0.695 |
| emotions| 0.625| 0.60 | 0.465 | 0.631 | 0.602 |
| music   | 0.627| 0.533| 0.467 | 0.615 | 0.673 |
| corel5k | 0.156| 0.156| 0.162 | 0.190 | 0.201 |
| enron   | 0.143| 0.169| 0.176 | 0.165 | 0.182 |
| bibtex  | 0.152| 0.168| 0.158 | 0.074 | 0.188 |
| flags   | 0.537| 0.691| 0.521 | 0.681 | 0.682 |
| birds   | 0.263| 0.311| 0.225 | 0.454 | 0.441 |
| mediamill| 0.215| 0.298| 0.243 | 0.354 | 0.385 |
| genbase | 0.859| 0.774| 0.798 | 0.802 | 0.811 |
| Average score | 0.425 | 0.444 | 0.375 | 0.452 | 0.5009 |

Table 6. Hamming loss of different multi-label classification algorithms

| Dataset | BRNB | BRDT | BRKnn | MLKnn | SEMML |
|---------|------|------|-------|-------|-------|
| yeast   | 0.293| 0.254| 0.241 | 0.274 | 0.214 |
| scene   | 0.178| 0.143| 0.218 | 0.235 | 0.115 |
| emotions| 0.232| 0.251| 0.697 | 0.466 | 0.306 |
| music   | 0.231| 0.231| 0.696 | 0.253 | 0.203 |
| corel5k | 0.312| 0.321| 0.312 | 0.294 | 0.270 |
| enron   | 0.273| 0.234| 0.535 | 0.321 | 0.278 |
| bibtex  | 0.278| 0.278| 0.489 | 0.432 | 0.378 |
| flags   | 0.143| 0.114| 0.180 | 0.175 | 0.198 |
| birds   | 0.227| 0.274| 0.160 | 0.165 | 0.198 |
| genbase | 0.193| 0.174| 0.214 | 0.124 | 0.118 |
| Average score | 0.235 | 0.227 | 0.358 | 0.251 | 0.214 |
6. Conclusion
In this paper we have experimented the stacked ensemble method with feature reduction technique CA_PCA and compared our method with other existing algorithms in multi-label classification. The ensemble methods with feature reduction technique for multi-label data provide good results compare to different traditional algorithms with respect to the evaluation metrics used. Other ways of carrying out stacking can be experimented with as part of future work.
References

[1] J. Read, B. Pfahringer and G. Holmes, (2008), Eighth IEEE ICDM, pp. 995-1000, doi: 10.1109/ICDM.2008.74.
[2] Lior Rokach, Alon Schclar, and Ehud Itach. (2014). Expert Syst. Appl. 41, 16, 7507–7523. DOI:https://doi.org/10.1016/j.eswa.2014.06.015
[3] M. Zhang and Z. Zhou, (2014), in IEEE Transactions on Knowledge and Data Engineering, vol. 26, no. 8, pp. 1819-1837, doi: 10.1109/TKDE.2013.39.
[4] Zhang, ML., Li, YK., Liu, XY. et al. (2018). Front. Computer. Science. 12, 191–202, https://doi.org/10.1007/s11704-017-7031-7.
[5] Sorower MS. (2010), Oregon State University, Corvallis. Dec;18:1-25.
[6] Schapire, R.E., Singer, (2000), BoosTexter, ML 39, 135–168, https://doi.org/10.1023/A:1007649029923.
[7] S. Godbole and S. Sarawagi. (2004), In 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining.
[8] D. Lewis, Y. Yang, et.al. (2004), RCV1: JMLR, 5:361–397.
[9] Vens, C, Struyf, J, Schietgat, L. et al.(2008) 73, pp.185. https://doi.org/10.1007/s10994-008-5077-3
[10] Tsoumakas G., Vlahavas I. (2007), ECML 2007. Lecture Notes in Computer Science, vol 4701. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74958-5_38
[11] M.L. Zhang and Z.H. Zhou. (2005), volume 2, pages 718- 721. The IEEE Computational Intelligence Society,

... (2010), Oregon State University, Corvallis. Dec;18:1-25.
[6] Schapire, R.E., Singer, (2000), BoosTexter, ML 39, 135–168, https://doi.org/10.1023/A:1007649029923.
[7] S. Godbole and S. Sarawagi. (2004), In 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining.
[8] D. Lewis, Y. Yang, et.al. (2004), RCV1: JMLR, 5:361–397.
[9] Vens, C, Struyf, J, Schietgat, L. et al.(2008) 73, pp.185. https://doi.org/10.1007/s10994-008-5077-3
[10] Tsoumakas G., Vlahavas I. (2007), ECML 2007. Lecture Notes in Computer Science, vol 4701. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-74958-5_38
[11] M.L. Zhang and Z.H. Zhou. (2005), volume 2, pages 718- 721. The IEEE Computational Intelligence Society,