# COMPARATIVE EVALUATION OF FOUR MULTI-LABEL CLASSIFICATION ALGORITHMS IN CLASSIFYING LEARNING OBJECTS

Asma Aldrees[1] and Azeddine Chikh[2] and Jawad Berri[3]

Information System Department,
College of Computer and Information Sciences
King Saud University, Riyadh, Kingdom of Saudi Arabia
[1]asma.aldrees@gmail.com
[2]az_chikh@ksu.edu.sa
[3]jberri@ksu.edu.sa

***ABSTRACT***

*The classification of learning objects (LOs) enables users to search for, access, and reuse them as needed. It makes e-learning as effective and efficient as possible. In this article the multi-label learning approach is represented for classifying and ranking multi-labelled LOs, whereas each LO might be associated with multiple labels as opposed to a single-label approach. A comprehensive overview of the common fundamental multi-label classification algorithms and metrics will be discussed. In this article, a new multi-labelled LOs dataset will be created and extracted from ARIADNE Learning Object Repository. We experimentally train four effective multi-label classifiers on the created LOs dataset and then, assess their performance based on the results of 16 evaluation metrics. The result of this article will answer the question of: what is the best multi-label classification algorithm for classifying multi-labelled LOs?*

***KEYWORDS***

*Learning object, data mining, machine learning, multi-label classification, label ranking.*

## 1. INTRODUCTION

The advancement and increasing availability in Internet technologies have changed many activities in life. One of the important activities is Learning which is being supported by these various technologies. The form of online distance learning is gaining a strong attention by learners of all ages with different interests. Learners have found digital learning media to be extremely convenient while learning as it involve the various human senses and different cognitive activities. It is the combination of the web and learning.

E-learning has emerged as a promising domain to facilitate and enhance learning through information technologies. Gerard (2006) [1] suggested that course units in computer-based instruction could be made smaller and combined in various ways for customization and use by

David C. Wyld et al. (Eds) : CCSEA, CLOUD, DKMP, SEA, SIPRO - 2016
pp. 107–124, 2016. © CS & IT-CSCP 2016 DOI : 10.5121/csit.2016.60210

learners. Learning objects (LOs) are an application of this type of course-units, and through the past years, they have gained the attention in the education area. Nowadays, LO is a concept used very often in different domains regarding learning management systems where it can be described as an essential, major unit that can be shared, reused and retrieved.

LOs should be tagged with metadata description and stored in a digital library, called Learning Object Repository (LOR), for future reuse. . Within the huge number of LOs, the demand to identify and classify them has arisen and become a critical issue in e-learning in order to make it faster and easier to the learners. To achieve this classification, each LO must be tagged with metadata about it to be easily located and later retrieved from repositories. These metadata are descriptive information of the LO, such as its topic, type, and keywords, that allow easy search of LOs.

LOs are mainly annotated with multiple labels, so we would like to be able to retrieve LOs based on any of the associated tags, not only one tag. Therefore, the single-label classification cannot model this multiplicity.

The focus of this paper is on multi-label classification methods [2] [3] [4] for searching LOs based on their tagged metadata. It aims to offer a sorting system that allows recovering and classifying LOs and offering individualized help based on choosing the best and effective classification technique for them.

A second contribution of this paper is creating a new multi-label dataset within a vast number of LOs and their associated metadata from one of the available repositories. The labels in this dataset are automatically generated as metadata and assigned to the LOs,. Metadata generation is a research field, which has been heavily worked on, in the recent years. This contribution will be explained in details in the next sections.

This paper is structured as follows: section 2 explains the main concepts and characteristics that establish LOs as the critical base within the context of web-based learning. Section 3 presents the background material on the multi-label learning, including the: classification techniques and evaluation measures. Also, in this section we will select the effective techniques to be used and compared in this experiment. Section 4 provides the details of the dataset used in this paper In Section 5; we will show the experimental results of comparing the adopted four multi-label classification techniques. Finally, conclusions and future work are drawn in Section 6.

## 2. CONTEXT OF LEARNING OBJECTS

The concept of LO has received considerable attention, for the first time, and described in 1967 by Gerard [5]. The term LO derived from the idea of Object Oriented Programming (OOP), in which, the parts of code are reused for multiple software applications. This concept suggests that, the ideal way to build a program is to assemble it from standardized, small, interchangeable chunks of code [6].

E-learning is defined as "learning facilitated and supported through the use of Information Technology (IT)". An E-learning experience is made up of the interaction of a number of learning components such as: courses, assessments, teaching materials, study materials, etc.
LOs are a relatively new way of presenting these learning contents. The idea appears to have a

transformation from traditional, direct instruction courseware design approaches, to a more effective and economical strategies for management and reuse of learning resources in computer-based networked environments.

The functionality of LOs can be described as [7]: "Firstly breaking educational material down into modular 'chunks' (objects), where each object can then have its defining properties described (or tagged) using metadata constructs".

Examples of LOs include: multimedia content, instructional content, learning objectives, instructional software and software tools, and persons, organizations, or events referenced during technology supported learning [8].

Recently, many research efforts concentrated on defining LOs. Currently, it appears difficult to arrive at a single definition of a LO that would align communities with diverse perspectives.

To aggregate up what a LO is, we summarize the general specifications of LO among all definitions:

- LOs are a new way of thinking about learning content. Conventionally, content comes in a several small chunks. LOs are smaller units of learning, which indicates that LO is a small component of the lesson.

- LOs are self-contained - each LO is independent, which means that each LO can be considered particularly without connection to other LO.

- LOs are reusable - a single LO may be used in multiple contexts for multiple purposes. That means the LO is the basis for a new LO or expands existing ones.

- LOs can be aggregated - they can be grouped into larger collections of content, including traditional course structures.

- LOs are tagged with metadata - every LO should has descriptive information making it to be easily retrieved. Quite important feature allowing using and reusing LOs.

LOs are annotated and tagged with many metadata descriptions. The most notable standards of metadata for LOs are: the Electrical and Electronic Engineers metadata (IEEE-LOM) [9]; Dublin Core Metadata (DCM) [10]; Instructional Management System (IMS) Global Learning Consortium [11]; Advanced Distributed Learning (ADL) [12]; and Canadian Core Initiative metadata (Can-Core) [13]. Since 2002, LOM has been the standard for describing the syntax and semantics of LOs. It's usually encoded in XML.

The purpose of LOM is to support the reusability, discoverability of LOs and to enable their interoperability. They include the element names, definitions, data types, vocabularies, and taxonomies. LOM focus on the minimum set of features needed to allow the LOs to be searched and managed.

LOs are placed and stored inside LORs, in an attempt to facilitate their reusability so that they can be more easily stored and retrieved on the basis of a description of their content. LORs

support simple and advanced search through the LOs. In simple search, they return the results according to the input keywords given by the user. The advanced search allows the user to specify some specific metadata features to filter LOs in order to meet his specific needs. There are many existing, available LORs, for example, but not limited to; Multimedia Educational Resources for Learning and Online Teaching (MERLOT) [14]; European digital library (ARIADNE) [15]; National Science, mathematics, engineering, and technology education Digital Library (NSDL) [16]; Health Education Assets Library (HEAL) [17]; Education Network Australia (EDNA) [18]; … etc.

In this paper a large dataset will be created, from the ARIADNE repository. It will be composed of a sufficient number of LOs and their related LOM metadata.

## 3. MULTI-LABLE LEARNING

In the machine learning domain, the traditional single-label classification methods has a large amount of research. These methods are concerned with learning a set of examples that are associated with a single label $l$ from a known finite set of disjoint labels $L$. However, there is a significant and real problem within the classification, while an example belongs to more than one label. This problem is known as multi-label classification problem. [2,19]. In the multi-label classification, the examples are associated with a set of labels $Y \subseteq L$.

The multi-label learning has two major tasks: multi-label classification (MLC) and multi-label ranking (MLR). In the case of MLC, the idea is to build a predictive algorithm that will provide a list of relevant labels for a given unseen example. On the other hand, the idea in the task of MLR is to provide a ranking of the selected relevant labels for the given unseen example.

Initially, MLC was mainly motivated by application in the domains of text categorization and medical diagnosis. However, nowadays, MLC has attached and is increasingly required by many new application domains, such as semantic annotation of images [20] and video [21]; protein function classification [22]; music categorization into emotions [23]; and Yeast gene functional classification [24].

There are different techniques that have been proposed to be applied to MLC problems, [25]. The next two subsections will describe the common and representative techniques of MLC and their evaluation metrics.

### 3.1. Multi-label Classification Techniques

MLC methods are divided in two categories as proposed in [2]: (1) Problem Transformation Methods; and (2) Algorithm Adaptation Methods.

#### 3.1.1. Problem Transformation Methods

It transforms the MLC problem into one or more single-label classification problems. It is an algorithm independent method. Many methods belong tothis category, such as:

- **Binary methods**

- **Binary Relevance (BR):** it is a well-known and the most popular problem transformation method [26]. BR is also known as One-Against-All (OAA). It transforms the multi-label problem into Q-binary problems, by considering the prediction of each label as independent binary classifier. Therefore, BR establishes Q-binary classifiers, one for each label $l \in L$ (whereas: Q = |L|). It transforms original multi-labeled dataset into Q single-label datasets, where each single-label dataset contains all the instances of the original multi-labeled dataset, and trains a classifier on each of these datasets. The instances are labeled positively if they have the existing label, otherwise they are labeled negatively. For the classification of a new instance, it gives the set of labels that are positively predicted by the Q classifiers. Although it is conceptually simple and relatively fast, it is recognized that BR ignores the possible correlations among labels.

❖ **Pair-wise methods**

- **Ranking via Pair-wise Comparison (RPC):** the basic idea of this method is transforming multi-label datasets into q(q-1)/2 binary-label datasets, covering all pairs of labels. (Where q is the number of labels, q = |L|). Each dataset contains the instances of the original multi-labeled dataset that are annotated by at least one of the corresponding labels, but not both. For classifying a new instance, all binary classifiers are invoked. Each classifier votes and predicts one of the two labels. After all classifiers are evaluated, the labels are ranked according to their sum of votes. Then, MLR is used to predict the relevant labels for the intended instance [27].

- **Calibrated Label Ranking (CLR):** it is the extended version of the RPC method [28]. It introduces one additional virtual label V (calibrated label), which is a split point between relevant and irrelevant labels. Thus, CLR solves the MLC problem with the RPC method. Each instance is considered positive if it belongs to the particular label, otherwise it is considered negative for the particular label and positive for the virtual one. The ranking is obtained by summing the votes of all labels; including V. CLR applies both for MLC and MLR tasks.

❖ **Label-combination methods**

These methods remove the limitation of the previous methods by taking into account the correlation and dependencies among labels.

- **Label Power-set (LP):** it is a simple and less-common problem transformation method [2,29]. The idea behind LP is considering each distinct label-set that exists in a multi-labeled dataset as one (single) label to transform the original dataset into a single-label dataset, so any single-label classifier can be applied to it. Given a new instance, the single-label classifier of LP gives the most probable class label, which is actually a set of labels. While the classifier can produce a probability distribution over all class labels, LP can provide the raking task among all. To apply the label ranking, for each label it calculates the sum of probability of class labels that contain it. So, LP can perform MLC and MLR tasks. Although, it takes into account the label correlations, it suffers from the increasing complexity that depends on the large number of distinct label-sets. The number of distinct label-sets is typically smaller, but it is still larger than the total number of labels q (q= |L|), and poses a critical complexity problem, especially for large values of instances and labels.

- **Pruned Set (PS)** [30]: This method follows the same paradigm of LP. But it extends it to resolve its limitations through pruning away the label-sets that are occurring less time than a user-defined threshold. It removes the infrequent label-sets. Then, it replaces these label-sets by the existing disjoint label-sets that are occurring more times than the threshold.

- **Classifier Chains (CC)** [31]: it involves Q-binary classifiers as in a BR method. It resolves the BR limitations, by taking into account the label correlation task. The classifiers are linked along a chain where each classifier deals with the BR problem associated with the label. Each link in the chain is expressed with the 0/1 label associations of all previous links.

❖ **Ensemble methods**

The ensemble methods are developed on top of the common problem transformation and algorithm adaptation methods.

They construct a set of classifiers and then classify new data points by taking a weighted vote of their predictions. They are used for further augment predictive performance and high accuracy results. They aim to aggregate the predictions of several base estimators built with a given learning algorithm.

- **Random k-label sets (RAkEL)** [32]: it constructs an ensemble of LP classifiers. It breaks the large label-sets into m models or subsets, which are associated with random and small-sized k-label-sets. It takes label correlation into account and also avoids LP's problems within the large number of distinct label-sets. Given a new instance, it queries models and finds the average of their decisions per label. Also, it uses the threshold value t to obtain the final prediction. The final decision is positive for a specific label if the average decision is greater than the given threshold t. Thus, this method provides more accuracy of results.

- **Ensembles of Pruned Sets (EPS)** [30]: it combines the PS method in an ensemble scheme. PS is specifically suited to an ensemble due to its fast build times. Also, it counters any over-fitting effects of the pruning process and allows the creation of new label sets at classification time. Applying the ensembles on PS method increases the predictive performance of the algorithm.

- **Ensembles of Classifier Chains (ECC)** [31]: it uses the CC method as a base classifier. It trains m models of CC classifiers C1,C2 ,…,Cm . Each Ck model is trained with a random chain ordering of labels L and a random subset of the datasets D. Each model is likely to be unique and able to predict different label-sets. After that, these predictions are summed by label so that each label receives a number of votes. A threshold value is applied to select the most relevant labels, which form the final predicted multi-label set.

### 3.1.2. Problem Adaption Methods

It extends and adapts the existing specific learning algorithm to directly handle the multi-label problem. It is an algorithm dependent method. Many methods belong to this category, such as:

- **Multi-Label k Nearest Neighbors (MLKNN)** [25]:it is an extension of the popular k-nearest neighbors (KNN) lazy learning algorithm using a Bayesian approach. It uses the

Maximum A Posteriori principle (MAP) to specify the relevant label-set for the new given instance, based on prior and posterior probabilities for the frequency of each label within the k nearest neighbors. It also has the capability to produce the ranking of the labels.

- **Multi-Label Decision-Tree (ML-DT)** [33]**:** it is an adaptation of the well-known C4.5 algorithm to handle multi-label data. The process is accomplished by allowing multiple labels in the leaves of the tree; the formula for calculating the entropy is modified for solving multi-label problems. The modified entropy sums all the entropies for each individual label. The key property of ML-DT is its computational efficiency:

$$\text{Entropy (D)} = \sum_{i=1}^{q} -p_j \ log_2 \ p_j - \left(1 - \ p_j\right) log_2 \ (1 - \ p_j)$$

Where D is the set of instances in the dataset and $p_j$ is the fraction of instances in D that belongs to the label j.

- **Back-Propagation Multi-Label Learning (BPMLL):**it is a neural network algorithm for multi-label learning. It's derived from the popular basic Back-propagation algorithm. The main modification to the algorithm is the introduction of a new error function that takes multiple labels into account [34].

- **Multi-label Boosting (ADABOOST.MH & ADABOOST.MR)** [35]**:** these are the two extensions of AdaBoost algorithm to handle multi-label data. While AdaBoost.MH is designed to minimize Hamming-loss, AdaBoost.MR is designed to minimize the Ranking-loss and find a hypothesis that ranks the relevant labels at the top.

- **Ranking Support Vector Machine (Rank-SVM)** [24]**:** it is a ranking approach for multi-label learning that is based on SVM. It is used to minimize the Ranking-loss. The main function they use is the average fraction of incorrectly ordered pairs of labels.

- **Multi-label Naïve Bayesian (ML-NB)** [36]**:** it extends the Naïve Bayesian algorithm to adapt it with the multi-label data. It deals with the probabilistic generation among the labels. It uses MAP to specify the more probable labels and assign them to the new given instance.

### 3.1.3. The Adopted Classification Techniques

We intend to select the most effective and reliable techniques for our experiment. So, looked at the related works that provide a comparison between these algorithms:

1. The authors in [23] compare MLC algorithms: binary relevance (BR), label power-set (LP), random k-label sets (RAKEL) and MLKNN. The RAKEL algorithm is more efficient and gives the best results.

2. The authors in [37] evaluate MLC algorithms RAKEL and MLKNN. Also, RAKEL records the best and effective results.

3. The authors in [38] show that MLKNN provides the best results in almost all analyzed cases.

4. The authors in [39] indicate that, the ECC is the best performance in all measures followed by RAKEL and EPS. The authors observe that, all ensemble methods provide the best results for almost all evaluation metrics.

5. The authors in [40] introduce a survey on the MLC algorithms and states that MLKNN gives better results than other algorithms.

6. The authors in [29 give a detailed description and survey about the MLC algorithms. Then, compare between them by using two different datasets. RAKEL achieves the best results followed by MLKNN. The authors mention that the ensemble methods are the closest algorithms of the best results.

7. The authors in [41] show that the MLKNN performs the best compared to the other algorithms followed by RAKEL algorithm.

From above, we can observe that:

- The algorithm transformation methods: the ensemble methods address the best and most accurate results.

- The algorithm adaptation methods: the MLKNN usually gives higher and best results compared to the other algorithms in the same category.

Therefore, we adopted in our experiment the following MLC techniques:

The Ensemble Methods, from the algorithm transformation category including:

1- Ensemble of Classifier Chains (ECC)
2- Random k-label sets (RAkEL)
3- Ensemble of Pruned Sets (EPS), and
4- Multi-Label k-Nearest Neighbors (MLKNN) from Algorithm Adaptation category.

### 3.2. Evaluation Metrics

The evaluation of multi-label algorithms requires different measures than those used in single-label classification. Several measures have been proposed for evaluating multi-label classifiers [2,26]. These measures are categorized in three groups: example-based; label-based; and ranking-based metrics. Example-based-measures, evaluate bipartitions over all instances of the evaluation dataset. Label-based measures breakdown the evaluation process into separate evaluations for each label. Furthermore, the ranking-based measures evaluate the ranking of labels with respect to the original multi-labelled dataset. Below, these three types will be described.

However, we need to define some aspects before defining those measures:

- The instances of multi-label dataset $(x_i , Y_i)$, i = 1…. m, where $Y_i \subseteq L$ is the set of true labels and $L = (l_j : j = 1 \ldots q)$ is the set of all labels.

- Given a new instance $x_i$ , the set of labels that are predicted by an MLC algorithm is denoted as $Z_i$.

- $r_i(l)$ is denoted as the LR method for the label *l*.

### 3.2.1. Example-Based Measures

- **Hamming Loss:** it evaluates how many times the label of the instance is misclassified, i.e., label which doesn't belong to the instance is predicted or a label belonging to the instance is not predicted. The smaller the value of HL the better the performance:

$$\text{HL} = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \,\Delta\, Z_i|}{|L|}$$

  Where Δ stands for the symmetric difference of two sets, which is the set-theoretic equivalent of the exclusive disjunction (XOR operation) in Boolean logic.

- **Subset Accuracy:** it evaluates the percentage of correctly predicted labels among all predicted and true labels:

$$\text{Subset Accuracy } = \frac{1}{m}\sum_{i=1}^{m} I\,(Z_i = Y_i)$$

  Where I( true) = 1 and I( false) = 0. It is very strict measure where it requires the predicted set of labels to be an exact match of the true set of labels, and ignores predictions that may be almost correct or totally wrong.

The following measurements are:

- **Precision** $= \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i|}$
- **Recall** $= \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Y_i|}$
- **$F_1$-Measure**$= \frac{1}{m}\sum_{i=1}^{m}\frac{2\,|Y_i \cap Z_i|}{|Z_i|+|Y_i|}$
- **Accuracy** $= \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \cap Z_i|}{|Z_i \cup Y_i|}$

### 3.2.2. Label-Based Measures

These measures are calculated for all labels by using two averaging operations, called macro-averaging and micro-averaging [42]. These operations are usually considered for averaging precision, recall and F-measure. We consider a binary evaluation measures B(tp, tn, fp, fn) which is calculated according to the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). The macro and micro-averaged versions of B, can be calculated as follows:

$$\mathbf{B}_{\text{macro}} = \frac{1}{q}\sum_{l=1}^{q} B\,(tp_l, fp_l, tn_l, fn_l)$$

$$\mathbf{B}_{\mathbf{micro}} = \frac{1}{q}\sum_{l=1}^{q} B\left(\sum_{l=1}^{q} tp_l,\ \sum_{l=1}^{q} fp_l, \sum_{l=1}^{q} tn_l, \sum_{l=1}^{q} fn_l\right)$$

### 3.2.3. Ranking-Based Measures

- **One Error:** it calculates how many times the top-ranked label is not in the set of relevant labels of the instance. The smaller the value of 1-error the better the performance:

$$\text{1-Error} = \frac{1}{m}\sum_{i=1}^{m} \delta(\operatorname{argmin}_{l \epsilon L}\, r_i(l))$$

Where:

$$\delta(l) = 1\ if\ l \notin L, 0\ otherwise$$

- **Coverage**: it evaluates how far we need, to go down the ranked list of labels to cover all the relevant labels of the instance. The smaller the value of coverage the better the performance:

$$\text{Coverage} = \frac{1}{m}\sum_{i=1}^{m} \max_{l \epsilon Y_i} r_i(l) - 1$$

- **Ranking Loss:** it evaluates the number of times that irrelevant labels are ranked above relevant labels. The smaller the value of RL the better the performance**:**

$$RL = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|Y_i||\bar{Y}_i|} \left|\{(l_a, l_b): r_i(l_a) > r_i(l_b), (l_a, l_b) \epsilon\ Y_i x \bar{Y}_i\}\right|$$

where $\bar{Y}_i$ is the complementary set of $Y_i$ with the respect to L.

- **Average Precision:** calculates the average fraction of labels ranked above a particular label $l \epsilon Y_i$ that actually are in $Y_i$. The bigger the value of AP the better the performance**:**

$$\text{AvgPrec} = \frac{1}{m}\sum_{i=1}^{m} \frac{1}{|Y_i|} \sum_{l\ \epsilon\ Y_i} \frac{\left|\{l' \epsilon\ Y_i: r_i(l') \leq r_i(l)\}\right|}{r_i(l)}$$

## 4. EXPERIMENTAL WORK

The LO dataset was created from the ARIADNE repository [15]. It was obtained by using a Web scrapping technique, which is a technique of extracting information from websites. In this experiment, we used the scrapping extension tool attached to Google Chrome browser, called Web Scraper [43]. The dataset should contain a sufficient number of LOs and their related LOM annotations. ARIADNE repository shows the content-related metadata for each browsed LO such as title, description, keywords and rights.

The LO dataset, we have created, contains 658 LO instances, annotated with one or more of 30 labels. These labels correspond to the searched input keywords applied by the learner and to the automatic generation of labels for each LO instance. All labels are related to the computer science domain, such as; computer networks; computer programming; computer graphics; computer security; electronic engineering…. etc. The LOs are described within 3500 features extracted from their LOM annotations In the next subsections, we will explain the approach we followed to automatically assign multiple labels to each LO instance as well as the process of minimizing the size of the metadata features to improve the quality of the classification technique and save the

time. Finally we propose the main statistics of the created multi-labeled dataset.

## 4.1. Automatic Generation of Metadata (Labels)

Metadata generation is a research field, which has been heavily worked on, in the recent years. Metadata generation method strongly depends on the target metadata types. The focus of this paper is the automatic generation of label metadata and assigning them to the scrapped LO instances. Particularly, this generation is done by keywords metadata. LOs have different keywords. Some of the keywords are different from each other, but their meanings are almost same. Hence, for classification purposes, keywords are categorized, and those categories are used as labels. Label categorization and related keywords are defined and stored in XML file. Then, the parsing function in the java programming language, parses this XML file, and when the LO instance contains any of the listed keywords, the label category of the intended keyword will be assigned to that instance as its additional label. By applying this automatic generation approach, the multiple labels are automatically assigned to each LO [44].

## 4.2. Dimensionality Reduction (DR)

In machine learning domain, dimensionality reduction (DR) is the process of minimizing and reducing the number of features in the dataset. The motivation for DR is summarized as follows: the reduction of the number of features provides an effective and high accuracy outcomes; the training and classification times are reduced due to the minimization of features' numbers; and removing noisy and irrelevant features which can have an influence on classification and a negative impact on accuracy results.

Dimensionality reduction can be divided into two categories: feature selection and feature extraction [45]. Feature selection is the process of selecting the relevant and high-valued features for the use in dataset classification [23]. Feature extraction is the process that constructs and builds new-derived features out of the original ones; they are intended to be informative and non-redundant.

In this experiment the feature selection approach was used to reduce the features' number. We applied the Gain-Ratio attribute evaluator, from WEKA [46], to select the top valuable features. In the MLC problems, the DR can be executed by invoking one of the multi-label algorithms, as mentioned in (MULAN), a Java Library for Multi-Label Learning, [47]. We performed the attribute evaluation using the LP transformation algorithm.

By applying the DR process, the features' number of the dataset has been reduced from 6166 to 3500 features.

### 4.3. Dataset Statistics

The multi-labelled dataset has many statistics, which explains the number of labels in the dataset that can influence the performance of the different multi-label methods. These statistics are [26]:

- **Label cardinality:** it is the average number of labels of the instances in dataset:

$$\text{Label-Cardinality} = \frac{1}{m} \sum_{i=1}^{m} |Y_i|$$

- **Label density:** it is the average number of labels of the instances in dataset divided by L (L=

total number of all labels):

$$\text{Label-Density} = \frac{1}{m} \sum_{i=1}^{m} \frac{|Y_i|}{L}$$

- **Distinct Label-sets:** it provides the number of unique label-sets in the dataset.
- The number of dataset's instances and features, along with features' type: whether they are numeric or nominal.

Table 1. Dataset statistics

| | | | Attributes | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | Domain | Instances | Before DR | Nominal | Numeric | Labels | Cardinality | Density | Distinct |
| ARIADNE | Text | 658 | 6166 | 3500 | 0 | 30 | 2.8586626 | 0.0952887 | 299 |

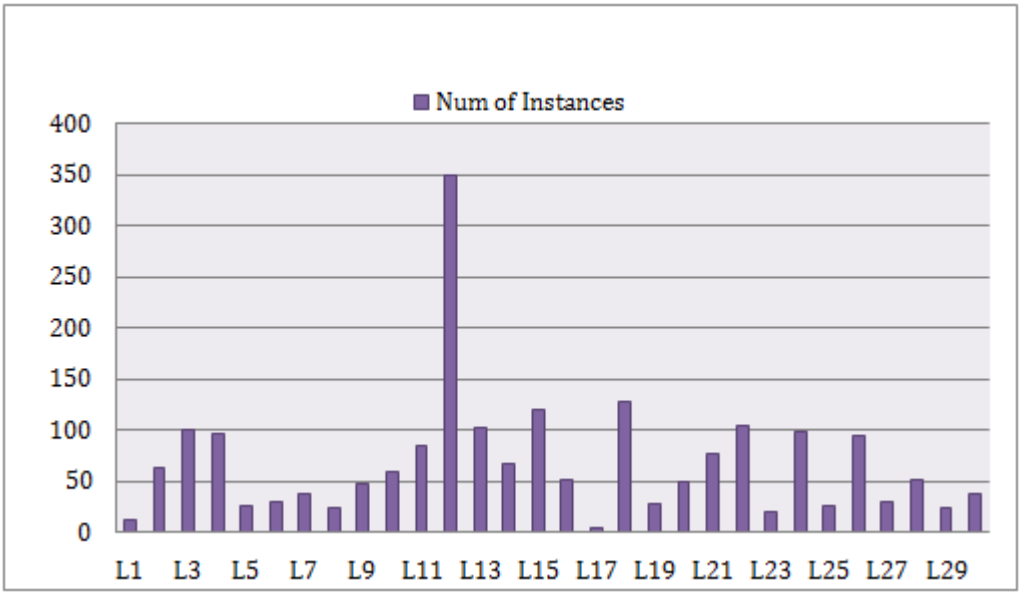


Figure 1. The number of instances per label

## 5. RESULTS AND DISCUSSION

We have applied the adopted classification techniques from MULAN, for obtaining the predicted results of the dataset. For the experiments, we followed the following three steps of the directive that is available in open-source MULAN system:

1. We loaded the multi-label dataset for training. The dataset is composed of two text files required by MULAN for the specification of a multi-label dataset: an XML file specifying the names of the labels (Ariadne.xml), and an ARFF file specifying the actual data (Ariadne.arff).

2. We created an instance of each learning algorithm that we want to train: ECC, RAKEL, EPS and MLKNN, in order to build a model and obtain predictions.

We trained each classifier using the multi-labeled LO dataset that we loaded from ARIADNE repository. For the empirical evaluation of all adopted algorithms, we used the cross-Validate method of the Evaluator class of MULAN library. Each classifier applied the 4-fold cross validation folds for evaluations to divide the dataset into: training-set and test-set.

The transformation-based algorithms transform MLC problem into one or more single-label problems. So, they accept the single-label classifier (base classifier) as a parameter. In this experiment: the J48 single-label classifier is used as a base classifier for all problem transformation algorithms. J48-classifier is the decision-tree classifier in WEKA Software [48].

Each of the adopted MLC algorithms has it's own parameters, needed to be stated prior to training them.

- **ECC has three parameters:**
  1. The number of models: varied from 30-150 models.
  2. Boolean parameter of using confidence while choosing the subset for dataset: false.
  3. Boolean parameter of using sampling-with-replacement: which means, the instances of the dataset could be selected more than one time at each model: in this paper; it was stated false, each instance could be selected only one time among all models.

- **RAKEL has three parameters:**
  1. The number of models: varied from 50-200 models.
  2. The k-subset size: 3
  3. Threshold value: 0.5

  RAKEL is meta-algorithm, and it can accept any multi-label algorithm as a parameter. It is typically used in conjunction with the LP algorithm. In turn LP is a transformation-based algorithm and it accepts a single-label classifier as a parameter. The J48-classifier, which is the decision-tree algorithm from WEKA, will be used for this purpose.

- **EPS has 6 parameters:**
  1. The percentage of dataset sample at each model: 60%
  2. The number of model: varied from 30-200 models
  3. The threshold value: 0.5
  4. The pruned sets parameter p: 3
  5. The pruned set strategy: Using both strategies: strategy A; and strategy B
  6. The pruned sets parameter b: 3

- **MLKNN has 2 parameters:**
  1. The number of neighbors: varied from 5 to 30 neighbors.
  2. The smooth factor: (always = 1).

## 5.1. Discussion

The comparison between the four learning algorithms will be evaluated from two points of view:

1. **The Classification point of view:** Table 2 shows the predictive performance results of the four competing MLC algorithms using the evaluation metrics, mentioned above. We noticed that **ECC** dominates the other algorithms in almost all measures, followed by RAKEL, MLKNN and finally EPS. ECC improves the predictive accuracy and can be used to further augment predictive performance.

2. **The Time-Complexity point of view:** In relation to the time issue, we observed that **ECC** is the most time-consuming algorithm, followed by RAKEL algorithm, EPS, and finally MLKNN, which is the fastest algorithm. Table 3 shows the classification time in seconds that was consumed during the process.

From the previous comparison, we could say that **ECC** performs the best and predicts the highest performance. According to the time issue, we have to use special devices, which has a quite enough memory space and a fast processor speed, to do the classification process. In this experiment, we have used our own Laptops to execute the results. Our Laptops have low features compared to more professional devices.

Table 2. Performance results

| | ECC | RAKEL | EPS | MLKNN |
|---|---|---|---|---|
| **Example - Based Measures** | | | | |
| Hamming Loss | **0.043918** | 0.045595 | 0.064851 | 0.062868 |
| Subset Accuracy | **0.326746** | 0.323715 | 0.256790 | 0.244696 |
| Example-Based Precision | **0.798503** | 0.791586 | 0.794673 | 0.704624 |
| Example-Based Recall | 0.684650 | **0.690402** | 0.506751 | 0.511439 |
| Example-Based F Measure | 0.702024 | **0.703801** | 0.566668 | 0.552293 |
| Example-Based Accuracy | 0.617970 | **0.618045** | 0.478257 | 0.468276 |
| **Label - Based Measures** | | | | |
| Micro-averaged Precision | **0.861877** | 0.840858 | 0.858957 | 0.830026 |
| Micro-averaged Recall | 0.643364 | **0.643919** | 0.386065 | 0.428970 |
| Micro-averaged F-Measure | **0.736260** | 0.729119 | 0.531133 | 0.565065 |
| Macro-averaged Precision | **0.810214** | 0.794930 | 0.437930 | 0.581787 |
| Macro-averaged Recall | 0.578120 | **0.590036** | 0.277954 | 0.313773 |
| Macro-averaged F-Measure | 0.645468 | **0.650262** | 0.318188 | 0.380048 |
| **Ranking - Based Measures** | | | | |
| Average Precision | **0.835098** | 0.796120 | 0.715237 | 0.730319 |
| Coverage | **6.727281** | 8.291398 | 11.435864 | 8.546322 |
| One-Error | **0.104822** | 0.126117 | 0.136751 | 0.153427 |
| Ranking Loss | **0.083637** | 0.113088 | 0.170504 | 0.127692 |

Table 3. Classification time

| Multi-label Classification Algorithm | Time of Classification in Sec |
|---|---|
| ECC | 4012.90 seconds (The slowest) |
| RAKEL | 1650.02 seconds |
| EPS | 181.136 seconds |
| MLKNN | 2.159 seconds (The fastest) |

## 6. CONCLUSIONS AND FUTURE WORK

The services of locating and searching educational contents, specifically LOs, present the core of the development of educational systems. This search area has been active in the recent years. In this paper, we have built an efficient MLC system for classifying the LOs. We have used four effective MLC techniques and compared between them, to notice which classification algorithm is the best for classifying the multi-labelled LOs. The classification was performed on the collection of 658 LO instances and 30 class labels. Therefore this system offers a methodology that illustrates the application of multi-label learning of LOs for classification and ranking tasks. We have concluded that, the **ECC** algorithm was very effective and it was proposed as the best classification algorithm for multi-labelled LOs, followed by RAKEL, MLKNN and finally EPS. From the performance results, it's obvious that the ensemble methods provide the best results for almost all evaluation metrics.

As future work, we intend to: Increase the dataset size, consisting of a very large number of LOs and labels; use the hierarchical MLC approach, which has a great potential in this domain; employ other metadata features to obtain the best classification for LOs; and study the multi-class and multi-instance approaches. They are new studied areas associated with the multi-label learning domain

## REFERENCES

[1] Gerard, R. W. (2006). Shaping the Mind: Computers in Education. In R. C. Atkinson and H. A. Wilson (eds.), Computer-Assisted Instruction: A Book of Readings. Orlando, Fla.: Academic Press, Health Education Assets Library.

[2] Tsoumakas, G., and Katakis, I. (2007). Multi-label classification an overview. International Journal of Data Warehousing and Mining. 3(3), 1-13

[3] Zhu, S.; Ji, X.; Xu, W.; and Gong., Y. (2005). Multi-labelled classification using maximum entropy method. In Proceedings SIGIR, 1-8.

[4] Barutcuoglu, Z.; Schapire, R.; and Troyanskaya, O. (2006). Hierarchical multi-label prediction of gene function. Bioinformatics 22, 880–836.

[5] Gerard, R.W. (1967). Shaping the mind: Computers in education. In National Academy of Sciences, Applied Science and Technological Progress. 207-228

[6] Lehman, R. (2007). Learning Object Repositories. New Directions for adult and continuing education, Spring. Wiley Periodicals, Inc, 113, 57-65. Retrieved from www.interscience.wiley.com

[7] Semmens, P. N. (2004). The potential for learning objects to support flexible learning in higher education. IEEE Computer Society Technical Committee on Learning Technology newsletter 6(2), 1-5. Retrieved from http://www.ieeetclt.org/issues/april2004/learn_tech_april2004.pdf .

[8] Polsani, P.(2003). Use and abuse of reusable learning objects. Journal of Digital Information, 3(4), 1-10. Retrieved from http://www.info2.uqam.ca/~nkambou_r/DIC9340/seances/seance10et12/Standards%20et%20LO/http___jodi.ecs.soton.ac.pdf.

[9] IEEE Learning Technology Standards Committee (LTSC). (2013, October 6). Retrieved from http://ieee-sa.centraldesktop.com/ltsc/

[10] DCMI Home: Dublin Core® Metadata Initiative (DCMI). Retrieved from http://dublincore.org/

[11] IEEE. (2006). IMS Meta-data best practice guide for IEEE1484.12.1-2002 Standard for Learning Object Metadata. Retrieved from http://www.imsglobal.org/metadata/mdv1p3/imsmd_bestv1p3.html .

[12] The ADL SCORM specification v 1.2. (2003). ADL – Advanced Distributed Learning Initiative, SCORM –Shareable Content Object Reference Model , 1(2), 9. Retrieved from http://www.adlnet.org

[13] CanCore: Homepage. Retrieved from http://cancore.athabascau.ca/en/index.html

[14] MERLOT II - Home. Retrieved from https://www.merlot.org/merlot/index.htm

[15] Ariadne Foundation. Retrieved from http://www.ariadne-eu.org/

[16] NSDL Library. Retrieved from https://nsdl.oercommons.org/

[17] EHSL - HEAL Collection. Retrieved from http://library.med.utah.edu/heal/

[18] Educational Network Australia. Retrieved from http://www.network-ed.com.au/

[19] Comp R. Cerri, R. R. Silva, and A. C. Carvalho,. “Comparing Methods for Multilabel Classification of Proteins Using Machine Learning Techniques”, BSB 2009, LNCS 5676, 109-120, 2009.

[20] Boutell, M. R., Luo, J., Shen, X., & Brown, C. M. (2004). Learning multi-label scene classification. Pattern Recognition, 37(3), 1757–1771. Retrieved from https://www.rose-hulman.edu/~boutell/publications/boutell04PRmultilabel.pdf

[21] Qi, G., Hua, X., Rui, Y., Tang, J., Mei, T., & Zhang, H. (2007). Correlative multi-label video annotation. In Proceedings of the 15th international conference on Multimedia (MULTIMEDIA '07). ACM, New York, 17-26.

[22] Diplaris, S., Tsoumakas, G., Mitkas, P. A., & Vlahavas, I. P. (2005). Protein Classification with Multiple Algorithms. Springer, 10th Panhellenic Conference on Informatics, PCI 2005, Volas, Greece, November 11-13, 2005. Proceedings,3746, 448-456.

[23] Trohidis, K., Tsoumakas, G., Kalliris, G., & Vlahavas, I. P. (2008). Multi-Label Classification of Music into Emotions. In: Proc. 9th International Conference on Music Information Retrieval (ISMIR 2008), Philadelphia, 6.

[24] Elisseeff, A., & Weston, J. (2001). A kernel method for multi-labelled classification. In: Advances inNeural Information Processing Systems, 1-7.

[25] Zhang, M., & Zhou, Z. (2007). ML-KNN: A lazy learning approach to multi-label learning. ScienceDirect, Pattern Recognition, 40(7), 2038-2048.

[26] Tsoumakas, G., Katakis, I., & Vlahavas, I. P. (2010). Mining Multi-label Data,1-20.

[27] Hüllermeier, E., Fürnkranz, J., Cheng, W., & Brinker, K. (2008). Label ranking by learning pairwise preferences. Artificial Intelligence, 34(5), 1897–1916.

[28] Fürnkranz, J., Hüllermeier, E., & Brinker, K. (2008). Multi-label classification via calibrated label ranking. ACM- Digital Library, Machine Learning, 73(2), 133 - 153.

[29] Sorower, M. S. (2010). A Literature Survey on Algorithms for Multi-label Learning, 25.

[30] Read, J., Pfahringer, B., & Holmes, G. (2008). Multi-label Classification Using Ensembles of Pruned Sets. Proc 8th IEEE International Conference on Data Mining, Pisa, Italy, 995-1000.

[31] Read, J., Pfahringer, B., Holmes, G., & Frank, E. (2009). Classifier chains for multi-label classification. Springer,Machine Learning, 5782, 254-269.

[32] Tsoumakas, G., Katakis, I., & Vlahavas, I. (2010). Random k-Labelsets for Multi-Label Classification. Knowledge and Data Engineering, IEEE Transactions, 23(7), 1079 - 1089.

[33] Clare, A., & King, R. D. (2001). Knowledge Discovery in Multi-label Phenotype Data.in: Proceedings of the 5th European Conference on PKDD, 42-53.

[34] Zhang, M., & Zhou, Z. (2006). Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering, 18(10), 1338-1351.

[35] Schapire, R. E., & Singer, Y. (2000). BoosTexter: a boosting-based system for text categorization. Machine Learning, 39(2/3), 135-168.

[36] Wei, Z., Zhang, H., Zhang, Z., Li, W., & Miao, D. (2011). A Naive Bayesian Multi-label Classification Algorithm With Application to Visualize Text Search Results. International Journal of Advanced Intelligence, 3(2), 173-188.

[37] Batista, V., Pintado, F. P., Gil, A. B., Rodríguez, S., & Moreno, M. (2011). A System for Multi-label Classification of Learning Objects. Springer, Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011 Advances in Intelligent and Soft Computing, 87, 523-531.

[38] Santos, A. M., P Canuto, A. M., & Neto, A. F. (2011). A Comparative Analysis of Classification Methods to Multi-label Tasks in Different Application Domains.International Journal of Computer Information Systems and Industrial Management Applications, 3, 218-227.

[39] El Kafrawy, P., Mausad, A., & Esmail, H. (2015). Experimental Comparison of Methods for Multi-Label Classification in Different Application Domains. International Journal of Computer Applications, 114(19), 1-9.

[40] Prajapati, P., Thakkar, A., & Ganatra, A. (2012). A Survey and Current Research Challenges in Multi-Label Classification Methods. International Journal of Soft Computing and Engineering (IJSCE), 2(1), 248-252.

[41] Tawiah, C. A., & Sheng, V. S. (2013). Empirical Comparison of Multi-Label Classification Algorithms. Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 1645-1646.

[42] Yang, Y. (1999). An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1, 67–88.

[43] Web Scraper - Chrome Web Store. Retrieved from https://chrome.google.com/webstore/detail/web-scraper/jnhgnonknehpejjnehehllkliplmbmhn?hl=en

[44] Meyer, M., Rensing, C., & Steinmetz, R. (2007). Categorizing Learning Objects Based On Wikipedia as Substitute Corpus. Proceedings of the First International Workshop on Learning Object Discovery & Exchange, 64-71.

[45] Sun, L., Ji, S., & Ye, J. (2014). Multi-label dimensionality reduction (1st ed.). USA: Chapman and Hall/CRC.

[46] Weka 3 - Data Mining with Open Source Machine Learning Software in Java. Retrieved from http://www.cs.waikato.ac.nz/ml/weka/

[47] Mulan: A Java library for multi-label learning. Retrieved from http://mulan.sourceforge.net/index.html

[48] J48-Decision Tree Classifier- WEKA. Retrieved from http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html