A Comparison of Multi-Label Text Classification Models in Research Articles Labeled With Sustainable Development Goals

ROBERTO CARLOS MORALES-HERNÁNDEZ1, JOAQUÍN GUTIÉRREZ JUAGÜEY1, AND DAVID BECERRA-ALONSO2
1Centro de Investigaciones Biológicas del Noroeste, La Paz 23096, Mexico
2Department of Quantitative Methods, Universidad Loyola Andalucía, 41704 Seville, Spain
Corresponding author: David Becerra-Alonso (dbecerra@uloyola.es)

ABSTRACT The classification of scientific articles aligned to Sustainable Development Goals is crucial for research institutions and universities when assessing their influence in these areas. Machine learning enables the implementation of massive text data classification tasks. The objective of this study is to apply Natural Language Processing techniques to articles from peer-reviewed journals to facilitate their classification according to the 17 Sustainable Development Goals of the 2030 Agenda. This article compares the performance of multi-label text classification models based on a proposed framework with datasets of different characteristics. The results show that the combination of Label Powerset (a transformation method) with Support Vector Machine (a classification algorithm) can achieve an accuracy of up to 87% for an imbalanced dataset, 83% for a dataset with the same number of instances per label, and even 91% for a multiclass dataset.

INDEX TERMS Classification algorithm, multi-label text classification, problem transformation method, scientific articles, sustainable development goals, text classification.

I. INTRODUCTION

For scientists and decision-makers at research centers and universities, identifying the alignment of scientific products with goals or policies becomes crucial to recognize their contribution and impact. Classification methods help to categorize research articles based on selected content features. In this context, machine learning enables large-scale data handling, analysis, and automation of the scientific evaluation process, such as the current massive production of research articles and the complex interdisciplinary nature of modern science [1]. Text classification is a Natural Language Processing (NLP) technique of text analysis to sort and categorize data into different types, forms, or any other distinct pre-defined class. Text classification is a relevant task in NLP and a powerful tool in research articles making scientific knowledge easier to reach, discover, and reuse [2]. The categorization of textual data is perhaps the dominant multi-label application [3].

Classification problems can be grouped in three types: Binary, Multi-class, and Multi-label. In binary classification, the task is classifying the data into one of two classes. In multi-class, each instance belongs to only one class of more than two possible classes. Multi-label classification refers to problems in which an instance can belong to one or more predefined labels. Unlike binary or multi-class, multi-label text classification presents more challenge because each textual document can be assigned a several labels. The initial training phase in supervised learning algorithms depends on labeled samples to adjust the parameters of model [4]. When the model has been trained, it can predict the set of labels from unseen instances. The learning from multi-label data has been tackled by approaches with a transformation or adaptation to conventional classification methods (binary or multi-class).

Three methods can be applied to learn from multi-label data: Problem Transformation, Problem Adaptation, and
Ensemble. Problem transformation techniques produce one or more binary or multi-class datasets to be managed by classification algorithms. Problem Adaptation method extends specific learning models to handle multi-label classification directly [5]. Ensemble methods combine classifiers to perform multi-label predictions in which some of the transformation or adoption methods are assembled by themselves [4]. Applications on multi-label problem transformation approach solve several problems from bioinformatics [6], [7] to tag recommendations [8], including genomic [7], [9] student learning experiences [10], microblogging text [11], and recently, health problems such as COVID-19 [12], [13]. Binary Relevance (BR), Label Powerset (LP), and Classification Chains (CC) are problem transformation techniques to convert the multi-label data to single-label (binary or multiclass label) data.

After the multi-label problem has been transformed in multiple independent single-label task, classification algorithms such as Naïve Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVM), or Random Forest (RF) can be deployed. This approach remains useful to this day [14], [15], [16], with NB, LR, SVM, and RF commonly applied [16], [17], [18], [19].

Experimental comparative analysis and performance for some transformation methods in combination with different classification algorithms is presented by Tsoumakas & Katakis [20]. The analysis includes 3 dataset, three multi-label problem transformation methods, and four classification algorithms. Experiments continue comparing multi-label methods and solutions; for instance, [21] performed comparative experimental results of certain multi-label classification methods in 11 datasets, with 4 classification algorithms and 16 evaluation metrics. Yapp [21] confirmed that there is a correlation between the base classifier and multi-label transformation method.

This work proposes a comparative study of one multi-class classifier called One-Versus-Rest (OvR) and three multi-label problem transformation methods (BR, LP, and CC), applied to four classification algorithms: NB, LR, SVM, and RF, on balanced and imbalanced datasets. The aim is to classify scientific papers with labels related to sustainable development goals (SDG), motivated to obtain a model that helps institutions and researchers to align scientific products to these goals. The study compares the performance of the method and algorithm combinations, on different balanced and imbalanced datasets and evaluating classification effectiveness based on specific metrics.

The scientific article datasets were obtained from a free account of Dimensions: a bibliographic database categorized with United Nation’s Sustainable Development Goals (SDG), managed by Digital Science [22].

In summary, the contributions are as follows:

- A comparison experiment designed with: Three multi-label problem transformation methods (Binary Relevance, Label Powerset, and Classification Chains) and one multi-class classifier One-Vs-Rest. All of them implemented with four classification algorithm models: NB, LR, SVM, and RF.
- A comparison of 5 dataset scenarios modifying the number of instances and discarding some SDG labels to evaluate classification models performance.

The rest of this paper is organized as follows: Section 2 includes related literature. Section 3 presents transformation methods and classification algorithms. Section 4 details the results. Finally, the conclusions are presented in Section 5.

II. RELATED WORK

Currently, comparative studies of classic text classifier models continue to be carried out. Pranckevičius et al 2017 [23] has made a comparison of NB, RF, SVM and LR as multi-class classifiers with text reviews; Yapp [21] made a comparison with base classifiers (SVM, NB, k-NN) with transformations methods such as BR, CC, RAKEL, among others, through 11 datasets.

Likewise, in relation to multi-label classification with respect to SDG, there are recent projects that apply Deep Learning for classification [24], [25], [26], [27], and a studies that compare base classifiers (SVM, NB) with Word2Vec, BERT, ELMo for datasets with SDG labels [28].

In NLP, multi-label text classification process categorizes one object to one or more classes for some specific purpose. In supervised learning, multi-label text classification process can be defined in a framework with six stages: (1) Information retrieval, (2) dataset creation, (3) exploratory data analysis, (4) preprocessing, (5) model building (selection of both the transformation method and the classification algorithm), and (6) performance measurement metrics.

In information retrieval and dataset creation, some scientific article classification searches are performed in public standardized datasets [29], [30], [31]. On the other hand, there are studies where research paper databases are created for multi-label classification [1], [28], [32]. The dataset creation can be done with a collection of articles from sources such as Scopus, Web of Science, and Dimensions, which allow collection through tools such as APIs or web portals, providing metadata of interest [33], [34].

Regardless of the origin of the databases, the Exploratory Data Analysis (EDA) can be generalized to obtain four metric categories: Basic traits (number of instances, attributes, and number of labelsets), label distribution data (label cardinality and label density), label relationship metrics (number of labels and label distributions), and metrics related to label imbalance [4]. Label cardinality, label density [20], and label correlations are characteristics frequently measured because they have proved to influence the performance of the transformation methods LP, BR, and CC [35]. For example, with some experiments is observed that multi-label methods such
as BR, CC, and LP, may be more affected by low density values than by high cardinality values [36].

A relevant dataset preprocessing entails the extraction of text to create clean word sequences [37]. In this phase, several methods are implemented such as: stop words removal, lower-case conversion, symbols filtering, lemmatization, stemming, instances with missing values, randomization rows, tokenization, and vectorization. The most common procedure removes unnecessary words in documents with a technique called stop words [38]. Although, there is not a single combination rule for preprocessing tasks, lowercase conversion and stop words are important methods to improve text classification [39].

Another relevant preprocessing task is tokenization, which allows dividing a text stream into words, terms, phrases, or any other element called tokens. Once documents are tokenized, a text feature extraction method is applied to obtain the most distinguishing features of a text, reducing dimensionality [40], [41], [42], [43]. Some of the broadly feature extraction techniques in research article classification approaches are: 1) One Hot Encoding, 2) Bag of Word (BOW) or Term Frequency (TF), and 3) Term Frequency and Inverse Document Frequency (TF-IDF), and semantic based approaches are: 1) Glove, 2) FastText, and 3) Word2Vec [30].

The TF-IDF is widely used in the fields of information retrieval and text mining to weight the relation for each word in a document collection [44]. TF refers to the occurrences of a word (term) in a document. Each term gets a weight depending on the number of appearances in the document. Document Frequency (DF) indicates how many times a specific term occurrence in the collection of documents. High DF values mean the term appears frequently in all documents. IDF is a mechanism for attenuating the effect of terms that occur often in the collection of documents to be meaningful for relevance determination [45]. The high IDF values mean rare words in all documents, increasing their importance [44]. In combination, the product TF * IDF results in a composited weight for each term in every document, with the highest values where it occurs several times in a document collection, lower when the token appears fewer times in a document or occurs in many documents, and the lowest when the term appears in all documents [45]. The result is a vector space model to represent a document as a vector of features and weights, whereas the words form features. A generation approach of simple features such as keywords (unigrams) or phrases (bigrams, n-grams) refers to preparing for classification models [46], [47], [48], [49], [50].

After the text is vectorized, the next stage is the model building described by two subprocesses. The former is the multi-label transformation method selection to convert the multi-label task into a binary or multi-class problem, followed by the classification algorithm, which returns predictions per sample in the form of a set of one or more labels evaluated against the ground truth label set [28]. Contrary to the single-label classification in which an instance has only two possible outcomes, correct or incorrect [51], the multi-label classification can deliver partially correct labeling.

The next classification algorithms work with binary or multi-class data to make predictions:

- **Naïve Bayes (NB)** is a probabilistic technique that assigns a probability to each instance-class pair, represented by a vector of binary weights. Thus, it attempts to predict the class corresponding to each instance [52].
- **Logistic Regression (LR)** is a regression model where an iterative procedure adjusts a logistic (sigmoid) mathematical function to estimate the parameter values and model the probabilities describing the possible outcomes of a single trial. The target variable is categorical and uses binomial logistic regression, where the target values are yes/no or pass/fail with sigmoid function mapping any value between 0 and 1 [53].
- **Support Vector Machine (SVM)** is one of the most robust and implemented supervised learning methods because of its excellent generalization capability, optimal solution, and discriminative power. It has been interesting for the data mining, pattern recognition, information retrieval, and machine learning groups in the last years [54]. SVM is a linear classification model that maximizes the margin between data instances and a hyperplane, acting as a division boundary [55]. It can be used in classification or regression.
- **Random Forest (RF)** is an ensemble learning model composed of different decision trees for regression, classification, and other tasks. For the classification task, the class selected by the majority of trees is the output of the random forest (majority voting). The RF purpose is to avoid overfitting and correlations among trees in a forest [56].

However, for multi-label classification, an approach is transforming datasets in such a way that binary or multi-class classifiers can process them. Generally, the output produced by those classifiers must be backtransformed to perform the multi-label prediction [4]. Multi-label classification problems can be tackled using problem transformation methods and multi-class described below.

**Problem Transformation Methods**: The multi-label classification problem is transformed into one or more single-label classification problems, combined to solve the original multi-label learning. These methods include Label Powerset (LP) [3], Binary Relevance (BR) [57], and Classifier Chains (CC) [58].

One-Vesus-Rest (OvR): This method is employed to solve multi-class problems with binary classifiers, where an instance of multi-class problems has only one label associated [59].

Comparison research studies, based on size training datasets and n-grams, using NB, LR, SVM, and RM with different multi-label transformation methods are still relevant [18], [21], [23]. Yapp et al. [21], showed that the best classifier depends on the method: SVM with BR, CC, and
others, k-NN and NB with hierarchy of multilabel classifiers, and decision trees and SVM with the ensemble of Classifier Chains. Pranckevičius et al. [23] showed that the LR multi-class classification method resulted in the highest (max 58.50%) classification accuracy in comparison to NB, RF, and SVM for a product-review dataset. Also, observation indicated that increasing the training dataset from 5,000 to 75,000 reviews per class showed an insignificant accuracy growth of 2% in NB, RF, and SVM classifiers. Other comparison classification models made by Medina et al. [28], classifying SDG labeled documents, concluded that much less computationally expensive methods such as SVM and LG were almost as effective as the deep learning model BERT in terms of the F1 metric.

Several metrics are usual to evaluate the performance of multi-label classifiers. Accuracy, Hamming loss, Precision, Recall, F1-Score, and even consumed computational time are measurements to estimate the classifier effectiveness. Accuracy is the number of correct answers among all classified instances, usually a percentage. Hamming loss is a label-based indicator considering the prediction error (incorrectly predicted label) and the missing error (unpredicted relevant label), normalized over the total number of classes and instances [60]. Precision is the number of instances labeled correctly belonging to the positive class divided by the total number of labeled instances, both correctly and incorrectly, belonging to the class [56]. Recall is the proportion of relevant instances that were correctly retrieved. F1-Score is the harmonic mean of precision and recall.

Finally, the deep learning classifiers are widely adopted and specialized, such as DocBERT [61], SicBERT [62], Hybrid Bert [63], or scientific publications classification with BERT [2]. However, the computational resources necessary for these algorithms make unattainable projects in institutions without high computing power; thus, the computationally efficient classifiers such as SVM, LR, NB, and RF are still used in several projects.

III. METHODS

In this section the proposed framework to compare multi-label classification models is presented. Figure 1 shows a typical pipeline to apply classification methods in six phases. All these phases were followed as it will be shown in the next sections.

A. INFORMATION RETRIEVAL

After a revision of multiple resources (Scopus, Web of Science, Microsoft Academic), Dimensions, as a bibliographic database produced by Digital Science, offers a feasible categorization scheme for the 17 SDG [22]. For collecting data, queries to its web portal allows article data exports to CSV files with 31 variables, including metadata such as Title, Abstracts, and classification labels from the SDG. Table 1 lists the 17 SDG labels from UN Agenda 2030 [64].

With organic agriculture as the knowledge domain, the query keywords focus on the main words of the document

### TABLE 1. Goals in agenda 2030 for sustainable development from united nations.

| SDG1 | SDG9 |
|------|------|
| No Poverty | Industry, Innovation and Infrastructure |
| SDG2 | SDG10 |
| Zero Hunger | Reducing Inequality |
| SDG3 | SDG11 |
| Good Health and Well-being | Sustainable Cities and Communities |
| SDG4 | SDG12 |
| Quality Education | Responsible Consumption and Production |
| SDG5 | SDG13 |
| Gender Equality | Climate Action |
| SDG6 | SDG14 |
| Clean Water and Sanitation | Life Below Water |
| SDG7 | SDG15 |
| Affordable and Clean Energy | Life on Land |
| SDG8 | SDG16 |
| Decent Work and Economic Growth | Peace, Justice, and Strong Institutions |
| SDG17 | Partnerhips for the Goals |

### TABLE 2. Keywords for organic agriculture search.

| Organic agriculture | Organic 1.0 |
|---------------------|-------------|
| organic principles | food |
| organic vision for Europe in 2030 | movements |
| organic standard | organic certification |
| organic produce | sustainable markets |
| smallholding family farmers | organic field |
| soil fertility | organic culture |
| transitioning from Organic 2.0 to 3.0 | organic production |
| organic baby food | organic consumption |
| certified organic land | organic movement |

Organic 3.0 for truly sustainable farming & consumption published by IFOAM Organics International [65]. Table 2 shows the search terms used to collect scientific papers in Dimensions.

B. DATASETS CREATION

For this study, the selected period for the Dimensions database queries was from January 2015 to August 2021. The UN published the 2030 Agenda for sustainable development with the definition of the 17 SDGs in the year 2015. Another reason to collect research papers since 2015 is that publications about organic agriculture 3.0 began this year (Table 2).

Seven annual datasets (2015-august 2021) were created from selective-text features from the Dimensions database exported to CSV files. These datasets are built...
FIGURE 1. Framework for comparative multi-label text classification.
with 20 variables: Publication ID, consolidated Title with Abstract, and 17 SDG columns in binary value (0/1) for the knowledge domain Organic Agriculture 3.0.

C. DATA ANALYSIS FOR MULTI-LABEL DATASET
EDA allows extracting meaningful knowledge from datasets. It includes visual data mining and statistical techniques that support decisions in preprocessing phase. Metrics for data characterization can be grouped into: Basic traits, label distribution data, label relationship metrics, and metrics related to label imbalance [4].

Number of samples per year, number of output labels, label cardinality, and label density are basic traits parameters considered in this study. Metrics for label distribution data are label cardinality and label density. Label cardinality is the average number of labels of the instances in dataset, and label density is the average number of labels of the instances in dataset divided by absolute number of labels. According with Tsoumakas et al. [20], both parameters have a potential influence in the performance of multi-label models.

Label cardinality:
\[
\frac{1}{N} \sum_{i=1}^{N} |Y_i|
\]
where \( N \) is the total number of data samples; \( Y_i \) is a set of data labels for \( i \)th data samples.

Label density:
\[
\frac{1}{L} \sum_{i=1}^{N} \frac{|Y_i|}{|L|}
\]
where \( L \) is the total number of existing classes.

Although the problem transformation method BR fails to consider the correlation of the labels, LP and CC, by design, do implicitly. Therefore, analyzing the correlations between labels allows recognize potential behaviors of classification models performance according to [66].

D. DATA PREPROCESSING
Before splitting for training and test with a 2:1 ratio, the dataset undergoes feature extraction and feature selection, such as: symbol filtering, eliminating instances with missing values, stop words removal, randomization rows, tokenization and vectorization.

Revising hundreds of Title-Abstract documents clean of rear symbols (i.e., ã, à, æ’, Â, and ®) helps to reduce documents size and reduce the possible incidence for the classification model according to [67].

Words and other symbols are eliminated through the stop words library and particular filters for a dimensionality reduction of the principal variable (Title-Abstract).

This study proposes to configure different dataset scenarios by modifying the instance number, including imbalanced, multi-class, extreme imbalanced with a label proportion 10:1, and balanced datasets with adjustments in the number of ODS to be considered. Table 3 shows these five scenarios applied to every yearly dataset for the experimental study.

| Scenario | Dataset characteristics |
|----------|-------------------------|
| SC1      | Complete imbalanced dataset with all 17 SDG labels |
| SC2      | Imbalanced dataset removing noisy labels with values of tens |
| SC3      | Balanced dataset with equal number of instances for each SDG label |
| SC4      | Extreme imbalanced of 10 to 1 for one label versus the remaining labels |
| SC5      | Multi-class dataset (instances with one SDG label) |

The TF-IDF technique is used for feature extraction from input data as a preprocessing task for the classification algorithm. TF-IDF captures important words from documents in a dataset (corpus), whereas Tfidfvectorizer is a technique used in scikit-learn library as a tokenizer and vectorizer for text classification [68].

E. MODEL BUILDING
In FIGURE 1, two components make the model building stage: multi-label transformation methods and classification algorithms. The multi-label transformation methods are solutions to convert from multi-label instances to a single label problem. With these conversions, the classification models (NB, LR, SVM, and Random Forest) of multi-label learning change to one or more single-learning tasks.

1) PROBLEM TRANSFORMATION METHODS
Scikit-multilearn is a multi-label classification software module that builds on top of the scikit-learn Python framework. The scikit-multilearn tools for multi-label transformation methods include Problem Adaptation, Problem Transformations, and Ensemble Methods. In this study, BR [57], LP [20], and CC [58] are implemented from this module. Besides, these problem transformation methods (BR, LP and CC) are configured with the default values, and none of their respective hyperparameters are optimized. This criterion is to enable a fair comparison among the methods.

An additional approach for multi-label classifier is applied with OvR classifiers from the Scikit-learn library. OvR is a meta-estimator that builds multiple classifiers to identify whether each sample belongs to one class or not, and finally combines all the decisions made [68].

2) CLASSIFICATION ALGORITHMS
This study implements four classification algorithms, NB [69], LR [70], SVM [71], and RF [16] for measuring their reliable performance and low resource requirements. Each classifier algorithm will be combined with each transformation technique OvR, BR, LP and CC, considering the five datasets scenarios defined in TABLE 3. The classification algorithms NB, LR, SVM, and RF, are executed with default
hyperparameter values to enable a fair comparison among the methods.

**F. MODEL EVALUATION**

Three multi-label classification metrics are selected to evaluate the experiment multi-label classification models:

- **Accuracy** is defined as the ratio of observations predicted correctly to the total number of observations:

  \[
  \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}
  \]

  where TP = True Positives; TN = True Negatives; FP = False Positives, and FN = False Negatives.

- **Hamming loss** refers to an average binary classification error [72]. For two samples is defined when if \( \hat{y} \) is the predicted value for the \( j \)-th label of a given sample, \( y \) is the corresponding true value, and \( n_{\text{labels}} \) is the number of classes or labels, then the Hamming loss between two samples as showed in equation (4).

  \[
  L_{\text{Hamming}}(y, \hat{y}) = \frac{1}{n_{\text{labels}}} \sum_{j=0}^{n_{\text{labels}}-1} 1(\hat{y}_j \neq y_j) \tag{4}
  \]

- **F1-Score (micro)** is the harmonic mean (weighted) of Recall and Precision:

  \[
  F_1 - \text{Score} = 2 \times \frac{\text{PRECISION} \times \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \tag{5}
  \]

  where Recall is the ratio of true positives to the sum of true positives and false negatives across all labels, and Precision refers to the percentage of predicted labels that are relevant [73].

  Note that for Hamming loss, the lower the values are the better the performance. Both accuracy and F1-Score (micro), higher values refer to better classification results.

The used data-processing infrastructure used for data-processing is equipped with Intel Dual-Core with 2.3 GHz, card graphics of 1536MB, 16GB of RAM (DDR4 2133 MHz), MacOS (64 bits) operating system, and programming in Python v3.8.

**IV. RESULTS AND DISCUSSION**

In this section, experimental information about datasets, pre-processing, multi-label classification results, and classification evaluations are presented.

Although the classification experiments were carried out with all the datasets from 2015 to 2021, to demonstrate the results in this paper, **2018 has been selected** for having the largest number of instances (31,434).

**A. DATASETS**

Table 4 presents the annual datasets distribution of scientific papers with SDG labels for this work. Except for 2018, the other annual datasets have instances around 25,000. One of the objectives is to compare the year 2018 (with 20% more instances) to the other annual databases, to determine if the performance of the models improves with the increase in samples. Density, and cardinality of the collected datasets are consistent with extremely low variations, thus their influence in multi-label transformation methods is considered the same.

Each classifier is run under five different dataset scenarios, varying yearly datasets in instances and SDG labels, as described in Table 5. For scenarios 2, 3 and 4, SDG labels with less than 1,000 instances are discarded and considered as noisy labels.

**B. MULTI-LABEL CLASSIFICATIONS RESULTS**

Five dataset scenarios, four transformation methods and four classification algorithms are evaluated.

1) **SCENARIO 1 (SC1): IMBALANCED 2018 DATASET WITH ALL 17 SDG LABELS**

In SC1, all 31,434 instances are considered with all 17 SDG labels. Table 6 shows that LP with SVM combination produces the best Accuracy score with 81%, but OvR with SVM has slightly best scores in F1-Score and Hamming loss. Even at execution time (ExecTime) measured in seconds, LP-SVM consumes 93% less than the second-best model (CC-SVM). It is notable how OvR and BR, even being similar class binarization strategies, have a clear difference in runtime. Medina 2019 [28], applied OvR with NB and SVM, obtaining micro F1-Score values of 0.69 and 0.74, respectively., whereas this proposed study had a better score of 0.86 with the dataset imbalanced and 17 SDG labels.

2) **SCENARIO 2 (SC2): IMBALANCED 2018 DATASET WITH 11 SDG LABELS**

In SC2, six SDG labels (SDG 1,5,8,9,10, and 17), with less than 1,000 instances are removed. The objective is to reduce
TABLE 6. Classification results for SC1, imbalanced 2018 dataset with 17 SDG (all instances).

| Method | One-vs-Rest (OvsR) | Label Powerset (LP) | Binary Relevance (BR) | Classifier Chains (CC) |
|--------|---------------------|---------------------|-----------------------|------------------------|
| Classifier Model | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF |
| Accuracy | 0.046 | 0.628 | 0.764 | 0.395 | 0.571 | 0.782 | 0.810 | 0.728 | 0.043 | 0.625 | 0.743 | 0.309 | 0.047 | 0.720 | 0.788 | 0.330 |
| F1-Score (Micro) | 0.081 | 0.773 | 0.864 | 0.570 | 0.612 | 0.828 | 0.855 | 0.770 | 0.075 | 0.772 | 0.850 | 0.470 | 0.081 | 0.769 | 0.842 | 0.500 |
| Hamming loss | 0.062 | 0.025 | 0.017 | 0.040 | 0.048 | 0.021 | 0.018 | 0.028 | 0.063 | 0.025 | 0.018 | 0.045 | 0.062 | 0.029 | 0.020 | 0.044 |
| Exec Time (s) | 7 | 15 | 5 | 161 | 94 | 21,866 | 103 | 1,820 | 1,146 | 10,547 | 429 | 11,685 | 853 | 13,423 | 1,580 | 2,445 |

TABLE 7. Results for SC2 imbalanced 2018 dataset with 11 SDG labels > 1,530 instances.

| Method | One-vs-Rest (OvsR) | Label Powerset (LP) | Binary Relevance (BR) | Classifier Chains (CC) |
|--------|---------------------|---------------------|-----------------------|------------------------|
| Classifier Model | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF |
| Accuracy | 0.046 | 0.689 | 0.811 | 0.460 | 0.578 | 0.795 | 0.868 | 0.812 | 0.046 | 0.689 | 0.811 | 0.467 | 0.051 | 0.765 | 0.850 | 0.488 |
| F1-Score (Micro) | 0.080 | 0.824 | 0.898 | 0.670 | 0.636 | 0.854 | 0.901 | 0.850 | 0.080 | 0.824 | 0.898 | 0.670 | 0.088 | 0.830 | 0.893 | 0.690 |
| Hamming loss | 0.108 | 0.036 | 0.022 | 0.057 | 0.077 | 0.031 | 0.022 | 0.034 | 0.108 | 0.036 | 0.023 | 0.057 | 0.108 | 0.038 | 0.024 | 0.055 |
| Exec Time (s) | 5 | 13 | 6 | 128 | 52 | 1,965 | 41 | 375 | 252 | 5,966 | 283 | 7,869 | 254 | 6,883 | 1,055 | 1,965 |

TABLE 8. Classification results for SC3, balanced 2018 dataset subsampling to 1,530 instances for each label from 11 SDG.

| Method | One-vs-Rest (OvsR) | Label Powerset (LP) | Binary Relevance (BR) | Classifier Chains (CC) |
|--------|---------------------|---------------------|-----------------------|------------------------|
| Classifier Model | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF |
| Accuracy | 0.006 | 0.604 | 0.759 | 0.428 | 0.686 | 0.775 | 0.833 | 0.779 | 0.011 | 0.604 | 0.759 | 0.424 | 0.012 | 0.698 | 0.806 | 0.433 |
| F1-Score (Micro) | 0.010 | 0.760 | 0.860 | 0.620 | 0.740 | 0.830 | 0.870 | 0.820 | 0.020 | 0.760 | 0.860 | 0.620 | 0.020 | 0.760 | 0.860 | 0.620 |
| Hamming loss | 0.112 | 0.046 | 0.030 | 0.063 | 0.054 | 0.035 | 0.025 | 0.063 | 0.112 | 0.046 | 0.030 | 0.063 | 0.111 | 0.051 | 0.032 | 0.062 |
| Exec Time (s) | 2 | 7 | 4 | 42 | 10 | 154 | 7 | 83 | 41 | 164 | 39 | 1,457 | 48 | 286 | 126 | 454 |

TABLE 9. Results for 2018 dataset for SC4, undersampling labels with one label with a relation 10 to 1 with the rest labels.

| Method | One-vs-Rest (OvsR) | Label Powerset (LP) | Binary Relevance (BR) | Classifier Chains (CC) |
|--------|---------------------|---------------------|-----------------------|------------------------|
| Classifier Model | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF |
| Accuracy | 0.485 | 0.533 | 0.711 | 0.534 | 0.495 | 0.649 | 0.756 | 0.648 | 0.485 | 0.353 | 0.711 | 0.554 | 0.484 | 0.607 | 0.755 | 0.557 |
| F1-Score (Micro) | 0.614 | 0.702 | 0.833 | 0.695 | 0.548 | 0.715 | 0.834 | 0.704 | 0.614 | 0.702 | 0.833 | 0.694 | 5.739 | 0.693 | 0.830 | 0.693 |
| Hamming loss | 0.067 | 0.051 | 0.032 | 0.053 | 0.090 | 0.111 | 0.034 | 0.059 | 0.067 | 0.051 | 0.032 | 0.053 | 0.068 | 0.060 | 0.034 | 0.054 |
| Exec Time (s) | 7 | 3 | 4 | 14 | 6 | 44 | 3 | 18 | 7 | 21 | 11 | 181 | 11 | 32 | 16 | 64 |

TABLE 10. Classification results for multiclass 2018 dataset with 17 SDG labels for SC5.

| Method | One-vs-Rest (OvsR) | Label Powerset (LP) | Binary Relevance (BR) | Classifier Chains (CC) |
|--------|---------------------|---------------------|-----------------------|------------------------|
| Classifier Model | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF | NB | LR | SVM | RF |
| Accuracy | 0.046 | 0.628 | 0.764 | 0.395 | 0.571 | 0.782 | 0.908 | 0.728 | 0.043 | 0.625 | 0.743 | 0.309 | 0.047 | 0.720 | 0.788 | 0.330 |
| F1-Score (Micro) | 0.081 | 0.773 | 0.864 | 0.570 | 0.612 | 0.828 | 0.907 | 0.770 | 0.075 | 0.772 | 0.850 | 0.470 | 0.081 | 0.769 | 0.842 | 0.500 |
| Hamming loss | 0.062 | 0.025 | 0.017 | 0.040 | 0.048 | 0.021 | 0.011 | 0.028 | 0.063 | 0.025 | 0.018 | 0.045 | 0.062 | 0.029 | 0.020 | 0.044 |
| Exec Time (s) | 7 | 15 | 5 | 161 | 94 | 21,866 | 103 | 1,820 | 1,146 | 10,547 | 429 | 11,685 | 853 | 13,423 | 1,580 | 2,445 |

possible noisy data considering the from labels with few occurrences that could affect classification models performance. Table 7 presents the results with improvements for all classification models except NB. Again, LP-SVM had the best scores in all metrics. This combination has the best Accuracy with 87% and the lowest Hamming loss with 0.022.
FIGURE 2. Performance for the classification models in 2018 dataset with five scenarios.
In this scenario, a reduced imbalance dataset could have let SVM algorithm reduce sensitivity to noise in training data and perform better for classification. It is notable how, by reducing the dataset by 3%, the execution times are reduced considerably, for example, in LP-SVM it is reduced by 68%, CC-LR by 59%, and BR-LR by 55%.

3) SCENARIO 3 (SC3): BALANCED 2018 DATASET WITH 1,530 INSTANCES PER LABEL

In this case, all 11 SDG labels (SDG2,3,4,6,7,11,12,13,14,15, and 16) were adjusted to 1,530 samples to evaluate the performance of multi-label classification models. LP-SVM presented the best performance compared to the rest of the models (IV-B5). However, most of the models show a lower performance in accuracy, F1-Score and Hamming loss compared to SC1 and SC2. The 55% decrease in training samples, compared to the SC2, is one of the reasons why the performance of most of the models is lower. Nevertheless, in accuracy, LP-SVM improved 3% compared to SC1, and only decreased 4% compared to SC2, with around 45% of training instances compared to the previous scenarios. Execution time reduces dramatically with respect SC1, for instance, LP-LR, with the highest execution time in SC1 (27,493s), with 45% of the total samples, the time is reduced to 154s (−99.4%).
4) SCENARIO 4 (SC4): EXTREME IMBALANCED (10 TO 1) FROM ONE LABEL VS OTHER LABELS – 2018 DATASET
The SC4 objective is to analyze OvR and BR performance when the ‘One’ has the same number of instances as ‘the Rest’ (adding all instances) avoiding bias but extremely reducing the training samples. The dataset is adjusted to 11 SDG labels, eliminating the SDG 1,5,8,9,10, and 17. The first test had the label SDG 2 with 1,800 instances and the sum of the rest is equal to 1,800. The process is repeated with SDG 3, and so on. Table 9 shows the average results of all these tests.

The combination OvR-NB had the more significant but still insufficient improvement in accuracy (from 5% in SC1 to 49%). With an average of 2,045 training instances, LP-SVM had its worst performance compared to the other scenarios considered above (76% accuracy). According to Hernández Santiago et al., 2016, in imbalanced sets for SVM, the separation hyperplane is skewed towards the majority class, decreasing the classification precision, since the minority class can be considered as noise and therefore ignored by the classifier. However, with just 10% of training examples, compared to SC1, LP-SVM are the best combined model for the extreme imbalanced dataset 1800-180, with a highest accuracy of 76% (just −5% compared to SC1).

5) SCENARIO 5 (SC5): INSTANCES WITH ONLY ONE SDG LABEL (MULTI-CLASS)
The approach was keeping the scientific articles with just one label (28,322) for a multi-class classification task. LP-SVM presented the highest performance compared to the rest of the scenarios with an accuracy of 91%, and Hamming-Loss with 0.011 (see TABLE 10). Although OvR with all the classification algorithms presented the best execution times, the time-accuracy relationship was always in favor of LP.

C. COMPARATIVE GRAPHS
Figure 2(a) shows SVM as the best classification algorithm in combination with any transformation method regarding accuracy. SVM with LP is the best combination with a maximum accuracy of 89% (in SC5: Multi-class dataset).

LP and CC, which are transformation methods that consider the possible correlations among the SDG variables, presented the best performance values.

Figure 2(b) confirms the excellent SVM performance with F1-Score (micro) graphics in five scenarios. Even in SC4,
with a radical imbalanced dataset and few instances, compared with other scenarios, SVM presented an appropriate F1-Score performance.

Figure 2(c) shows the Hamming loss metrics, where NB had the worst performance of the classifiers. The findings indicate LP-SVM is the best model (min 0.013, max 0.034) in Hamming loss values.

Figure 2(d) shows the computational complexity through the execution time, where LR and RF had large values (due to the computer’s RAM memory consumption) and without good classification performance. Finally, SVM with LP presented a consistent and reduced computation time, strengthening the overall performance of the combination.

### TABLE 12. F1-Score (micro) of the worst model (BR-NB) vs. the best (LP-SVM) in the five scenarios for the years collected (2015 to August 2021).

| Year | SC1  | SC2  | SC3  | SC4  | SC5  |
|------|------|------|------|------|------|
| 2015 | 0.143| 0.159| 0.012| 0.606| 0.119|
| 2016 | 0.099| 0.109| 0.010| 0.610| 0.078|
| 2017 | 0.092| 0.116| 0.010| 0.606| 0.076|
| 2018 | 0.075| 0.080| 0.020| 0.614| 0.065|
| 2019 | 0.077| 0.094| 0.010| 0.602| 0.053|
| 2020 | 0.081| 0.100| 0.010| 0.602| 0.068|
| 2021 | 0.099| 0.121| 0.010| 0.601| 0.073|

| Model | Year | SC1  | SC2  | SC3  | SC4  | SC5  |
|-------|------|------|------|------|------|------|
| BR-NB | 2015 | 0.864| 0.884| 0.853| 0.829| 0.904|
|       | 2016 | 0.857| 0.882| 0.850| 0.844| 0.901|
|       | 2017 | 0.850| 0.870| 0.840| 0.835| 0.900|
|       | 2018 | 0.855| 0.901| 0.870| 0.834| 0.907|
|       | 2019 | 0.841| 0.866| 0.850| 0.829| 0.892|
|       | 2020 | 0.853| 0.868| 0.850| 0.830| 0.898|
|       | 2021 | 0.143| 0.159| 0.012| 0.606| 0.119|

| Model | Year | SC1  | SC2  | SC3  | SC4  | SC5  |
|-------|------|------|------|------|------|------|
| LP-SVM| 2015 | 0.017| 0.022| 0.028| 0.035| 0.011|
|       | 2016 | 0.018| 0.023| 0.029| 0.032| 0.012|
|       | 2017 | 0.019| 0.024| 0.030| 0.032| 0.012|
|       | 2018 | 0.018| 0.022| 0.025| 0.034| 0.011|
|       | 2019 | 0.020| 0.026| 0.030| 0.034| 0.013|
|       | 2020 | 0.018| 0.024| 0.030| 0.034| 0.012|
|       | 2021 | 0.020| 0.027| 0.033| 0.033| 0.013|

### TABLE 13. Hamming loss performance between the worst model (BR-BN) and the best (LP-SVM), on five scenarios (datasets from 2015 to 2021).

1) COMPARATIVE PERFORMANCE BETWEEN THE WORST AND THE BEST CLASSIFICATION MODEL WITH ALL YEARLY DATASETS (2015-AUGUST 2021)

This section allows comparing the results of Accuracy, F1-Score and Hamming loss of the worst classification model, BR-NB, against the best LP-SVM, in the 5 scenarios and for all the years of the databases (2015-august 2021).

Accuracy results (all years and all scenarios) for both the worst and the best model are presented in Table 11 and Figure 3. The results show that, in accuracy, even with its best performance (51% in scenario 4), BR-NB has a difference of almost 40% with respect to the best result of LP-SVM (scenario 5).

Comparison of F1-Score (micro) results from the best model LP-SVM Vs. LR-NB are presented in Table 12 and
Figure 4. LP-SVM presents a uniformity in its behavior during the different years and scenarios, not only in F1-Score as shown in Figure FIGURE 5, but also in Accuracy and Hamming Loss.

Table 13 and Figure 5, confirm the irregular performance in the different scenarios in all the years for the BR-NB model. The irregularity is present in Hamming loss, Accuracy and in F1-Score. On the contrary, LP-SVM presents a uniform performance in the 3 metrics presented in this section.

D. LIMITATIONS
Classification algorithms run with default parameters, which means a fair scheme, but it also could mean not reaching the maximum performance of the algorithms. Scenario 4 (SC4) with extreme imbalances could have a small number of instances for training (2,023) that could affect the model results.

V. CONCLUSION
This study proposed a comparison review of multi-label text classification models based on their performance. The results support the framework that implements combinations of transformation methods with classification algorithms with acceptable classification performance.

The findings in the 2018 dataset indicate that SVM with LP is the best model with the highest accuracy (average 83%). Even in SC4, with around 2,000 training instances, LP-SVM has acceptable values of accuracy (~76%). Compared to SC1, this model increases its accuracy by 7% when, in scenario 2, the noisy data is removed. LP-SVM presents a uniform performance in the metrics through the different years and scenarios.

In dataset 2018, independent of the dataset scenario and transformation method, Support Vector Machine is the best classification algorithm with the best overall classification accuracy performance with a difference of 20% (best 91% and worst 71%). The results present LP as the best transformation method, independently of the dataset scenario (balance/imbalanced) and even with any classification algorithm NB, LR, SVM, and RF. LP has accuracy values of min 50% with NB, and max 91% with SVM.

On the other hand, NB combined with any transformation method has the lowest performance with a range of difference between the best (LP-NB) and the worst (BR-NB) result in accuracy of 68 points.

In execution times, LP-SVM is confirmed with good performance values, with an average of 42s considering the five scenarios. OvR with any classification algorithm, on average, presents the lowest execution values. However, its classification results are low. Moreover, BR and CC, on average, are the highest execution times and good performance values in the classification; nevertheless, LP performs better.

Following the comparative analysis, LP and CC exploit the weak relationships between labels, presenting better results in all scenarios than OvR and BR.

The results of the comparative tables between the best and the worst model of all the years collected (2015-August 2021) allow validating the reliable behavior of the applied reference framework.

The experimental results have shown that using the default hyperparameters for multi-label transformation and classification algorithms, a competitive classification of scientific articles with title and abstract as main data features and SDG labels is obtained.

In future work, hyperparameter adjustments in models, and the use of bigram/trigram in word vectorization could represent a path to improve performance. Also, continuing with efficient and reduced resources, Word2Vec [74] instead of TF-IDF vectorizer, could capture the relevant information from Title and Abstract for multi-label classification.

With the created organic agriculture domain 3.0 datasets, data mining and knowledge discovery could be applied using machine learning to exploit the information for the benefit of researchers and academic institutions.

REFERENCES
[1] A. Ebadi, S. Tremblay, C. Goutte, and A. Schifflauerová, “Application of machine learning techniques to assess the trends and alignment of the funded research output,” J. Informetrics, vol. 14, no. 2, May 2020, Art. no. 101018, doi: 10.1016/j.joi.2020.101018.
[2] A. García-Silva and J. M. Gomez-Perez, “Classifying scientific publications with BERT—Is self-attention a feature selection method?” in Proc. Eur. Conf. Inf Retr., Cham, Switzerland: Springer, 2021, pp. 161–175, doi: 10.1007/978-3-030-72113-8_11.
[3] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Mining multi-label data,” in Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Boston, MA, USA: Springer, 2010, pp. 667–685, doi: 10.1007/978-0-387-09823-4.
[4] F. Herrera, F. Charte, A. J. Rivera, and M. J. D. Jesus, Multilabel Classification, Cham, Switzerland: Springer, 2016, doi: 10.1007/978-3-319-41111-8.
[5] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, “An extensive experimental comparison of methods for multi-label learning,” Pattern Recognit., vol. 45, no. 9, pp. 3084–3104, Sep. 2012, doi: 10.1016/j.patcog.2012.03.004.
[6] A. Maxwell, R. Li, B. Yang, H. Weng, A. Ou, H. Hong, Z. Zhou, P. Gong, and C. Zhang, “Deep learning architectures for multi-label classification of intelligent health risk prediction,” BMC Biomed., vol. 18, no. S14, pp. 121–131, Dec. 2017, doi: 10.1186/s12859-017-1898-z.
[7] M.-L. Zhang and Z.-H. Zhou, “Multilabel neural networks with applications to functional genomics and text categorization,” IEEE Trans. Knowl. Data Eng., vol. 18, no. 10, pp. 1338–1351, Oct. 2006, doi: 10.1109/TKDE.2006.162.
[8] L. Li, Y. Yao, F. Xu, and J. Lu, “MATAB: Keywords enhanced multi-label learning for tag recommendation,” in Proc. Asia-Pacific Web Conf., in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, vol. 9313, 2015, pp. 268–279, doi: 10.1007/978-3-319-25255-1_22.
[9] S. Nazmi, X. Yan, and A. Homaifar, “Multi-label classification using genetic-based machine learning,” in Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC), Oct. 2018, pp. 675–680, doi: 10.1109/SMC.2018.00123.
[10] N. Yanes, A. M. Mostafa, M. Ezz, and S. N. Almuayyyl, “A machine learning-based recommender system for improving students learning experiences,” IEEE Access, vol. 8, pp. 201218–201235, 2020, doi: 10.1109/ACCESS.2020.3036336.
[11] M. A. Parwez, M. Abulaish, and J. Jahiruddin, “Multi-label classification of microblogging texts using convolution neural network,” IEEE Access, vol. 7, pp. 68678–68691, 2019, doi: 10.1109/ACCESS.2019.2919494.
[12] B. J. Gutierrez, J. Zeng, D. Zhang, P. Zhang, and Y. Su, “Document classification for COVID-19 literature,” in Proc. EMNLP, 2020, pp. 3715–3722, doi: 10.18653/v1/2020.findings-emnlp.332.
A. P. Santos and F. Rodrigues, “Multi-label hierarchical text classification,” Complex Intell. Syst., vol. 7, no. 1, pp. 235–247, Feb. 2021, doi: 10.1007/s41477-020-00199-4.

A. Abdullahi, N. Azah, S. Kamal, and Z. Ali, “An improved multi-label classifier chain method for automated text classification,” Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 3, p. 2021, 2021. Accessed: Jul. 5, 2021. [Online]. Available: https://www.ijacs.thesai.org

M. Pushpa and S. Karpagavalli, “Multi-label classification: Problem transformation methods in Tamil phoneme classification,” Proc. Sci. Comput., vol. 115, pp. 572–579, Aug. 2017, doi: 10.1016/j.procs.2017.09.116.

X. Wu, Y. Gao, and D. Jiao, “Multi-label classification based on random forest algorithm for non-intrusive load monitoring system,” Processes, vol. 7, no. 6, p. 337, Jun. 2019, doi: 10.3390/pr7060337.

W. Dudzik, J. Nalepa, and M. Kawulok, “Evolving data-adaptive support vector machines for binary classification,” Knowl.-Based Syst., vol. 227, Sep. 2021, Art. no. 107221, doi: 10.1016/j.knosys.2021.107221.

K. Shah, H. Patel, D. Sanghvi, and M. Shah, “A comparative analysis of logistic regression, random forest and KNN models for the text classification,” Augmented Hum. Res., vol. 5, no. 1, p. 12, Dec. 2020, doi: 10.1007/s41133-020-00032-0.

S. Xu, “Bayesian Naive Bayes classifiers to text classification,” J. Inf. Sci., vol. 44, no. 1, pp. 48–59, Feb. 2018, doi: 10.1177/0165551516677946.

G. Tsoumakas and I. Katakis, “Multi-label classification: An overview,” Int. J. Data Warehousing Mining, vol. 3, pp. 1–13, Jul. 2007, doi: 10.1002/dwm.101.

E. K. Y. Yapp, X. Li, W. F. Lu, and P. S. Tan, “Comparison of base classifiers for multi-label learning,” Neurocomputing, vol. 394, pp. 51–60, Jun. 2020, doi: 10.1016/j.neucom.2020.01.102.

J. Wastl, S. Porter, H. Draux, B. Fane, and D. Hook, “Contextualizing development goals,” M.S. thesis, Dept. Linguistics Philol., Uppsala Univ., Jun. 2020, doi: 10.5281/zenodo.4103512.

J. M. Hilbe, Practical Guide to Logistic Regression. Boca Raton, FL, USA: CRC Press, 2016. [Online]. Available: http://www.jstastof.org/71i/603/

J. Cervantes, F. García-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” Neurocomputing, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.
