Indonesian Journal of Electrical Engineering and Computer Science
Vol. 24, No. 1, October 2021, pp. 484~490
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v24.i1.pp484-490

Multi-label classification approach for Quranic verses labeling

Adeleke Abdullahi[1], Noor Azah Samsudin[2], Mohd Hisyam Abdul Rahim[3], Shamsul Kamal Ahmad Khalid[4], Riswan Efendi[5]
[1,2,3,4]Department of Software Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Malaysia
[5]Department of Mathematics, Universitas Islam Negeri Sultan Syarif Kasim Riau, Riau, Indonesia

Article Info

Article history:

Received Mar 5, 2021
Revised Aug 6, 2021
Accepted Aug 11, 2021

Keywords:

Holy Quran
Machine learning
Multi-label classification
Multi-label evaluation metrics
Text classification

ABSTRACT

Machine learning involves the task of training systems to be able to make decisions without being explicitly programmed. Important among machine learning tasks is classification involving the process of training machines to make predictions from predefined labels. Classification is broadly categorized into three distinct groups: single-label (SL), multi-class, and multi-label (ML) classification. This research work presents an application of a multi-label classification (MLC) technique in automating Quranic verses labeling. MLC has been gaining attention in recent years. This is due to the increasing amount of works based on real-world classification problems of multi-label data. In traditional classification problems, patterns are associated with a single-label from a set of disjoint labels. However, in MLC, an instance of data is associated with a set of labels. In this paper, three standard MLC methods: binary relevance (BR), classifier chain (CC), and label powerset (LP) algorithms are implemented with four baseline classifiers: support vector machine (SVM), naïve Bayes (NB), k-nearest neighbors (k-NN), and J48. The research methodology adopts the multi-label problem transformation (PT) approach. The results are validated using six conventional performance metrics. These include: hamming loss, accuracy, one error, micro-F1, macro-F1, and avg. precision. From the results, the classifiers effectively achieved above 70% accuracy mark. Overall, SVM achieved the best results with CC and LP algorithms.

Corresponding Author:

Adeleke Abdullahi
Faculty of Computer Science and Information Technology
Universiti Tun Hussein Onn Malaysia
Johor, 86400 Parit Raja, Batu Pahat, Malaysia
Email: abdul2040@yahoo.com

1. INTRODUCTION

The field of machine learning focuses on the study that gives artificial intelligence (AI) systems the capability to improve its performance over a time period through acquiring new knowledge and skills [1]. Conceptually, machine learning is based on training machines to be able to detect patterns and adapt to a new circumstance [2]. Important to machine learning is the problem of classification, the task of identifying to which category/class an observation (instance) belongs [3]. Traditionally, in a typical classification problem, the goal is to predict automatically one of the predefined classes each to a set of samples.

Given an input x, the goal of classification is to learn a mapping from input x to output y where $y \in \{i, \dots, c\}$, c representing number of classes. This is referred to as a single-label classification (SLC) problem. However, in some real-world classification problems, such as in the Quranic text classification task, a data instance may be categorized into multiple classes at the same time.

For example, a verse in the Quran may be targeted towards several issues (or topics) such as related to faith, family, worship, good deeds, paradise, hell among others. This kind of classification problem is termed multi-label classification (MLC) [4]. In MLC, which is an extension of the conventional SLC, data instances are associated with a set of labels $Y \subseteq L$.

Primarily, the concept of MLC originated from text [5] where often documents are associated simultaneously with multiple topics such as news, sports, education, economy etc. The techniques of MLC have been further applied to other classification problems including marketing [6], imaging [7], multimedia [8], and genomics [9]. Although, there have been increasing amount of research works on multi-label classification methods proposed in literatures, however in the Quranic text classification problem, the application of MLC is relatively new. Hence, this paper presents the implementation of multi-label classification methods and algorithms applicable in automating Quranic verses labeling task.

In this work, standard machine learning algorithms (classifiers) are applied for the multi-label task. The experimental work involves the use of binary relevance (BR), classifier chain (CC), and label powerset (LP) algorithms. These MLC methods will be used to classify Quranic verses simultaneously into one or more predefined categories (or class labels) namely: faith ("*iman*"), worship ("*ibadah*"), and etiquettes ("*akhlak*"). The selected categories are from the most fundamental aspects of Islam as recognized by the Quran experts [2].

Generally, a classification task is the problem of predicting class labels for an instance described by a finite set of features. Given a set of n attributes $X = \{X_1, \ldots, X_n\}$, a set of q class labels $L = \{l_1, \ldots, l_q\}$, a training dataset D comprising of N instances: $\{(x_1, l_1), (x_2, l_2), \ldots, (x_N, l_N)\}$, each x_i corresponds to an attribute vector $(x_1, \ldots, x_n)$ that stores values (information) for the set of n attributes in X, and each $l_i \in L$ corresponds to a single class label.

From the work [10], there are two classical approaches (or methods) employed to solve classification problems involving multi-label data: problem transformation (PT) and algorithm adaptation (AA) methods. Problem transformation approach is a simplified way to address MLC problems. It works by selecting for each multi-label data instance a single label from its multi-label subset $Y \subseteq L$.

PT methods are algorithm independent and have been successfully employed to solve classification problems [11], [12]. In other words, the methods work by transforming multi-label classification problem to one or more single-label classification problems. Thereafter, any of the available SLC algorithms such as support vector machines (SVMs), naïve Bayes, k-nearest neighbors (k-NN), neural networks, and decision trees can be implemented directly as baseline classifiers.

On the other hand, algorithm adaptation (also referred to as algorithm dependent) involves extending the single-label classifiers to adapt and be implemented directly in multi-label problems [5], [10]. AA algorithms are specifically developed to solve a given multi-label problem. Hence, they lack flexibility and simplicity [5]. These setbacks are the main reasons why AA methods have been less popular compared to the PT methods. Existing works based on AA approach include probabilistic methods [13], neural networks [14], [15], support vector machines [16], [17], and decision trees [18], [19].

This study employed the PT approach for the Quranic text multi-label classification problem due to its popularity and simplicity. There are several algorithms available for implementation based on the PT approach. The study employed three of the most conventional MLC algorithms: BR [20], [21], CC [22], [23], and LP [24]. Review of these algorithms are documented in the next section.

2. METHODS AND MATERIALS

This work involves the multi-label classification of Quranic verses using three standard MLC methods: binary relevance (BR), classifier chain (CC), and label powerset (LP) algorithms. The MLC algorithms will be used to classify the input verses into one or more of the predefined labels: faith ("*iman*"), worship ("*ibadah*"), and etiquettes ("*akhlak*"). Traditional single-label algorithms such as SVMs are not capable of handling the classification of multiple labels simultaneously. In this paper, four single-label classification algorithms: SVMs, naïve Bayes, k-NN, and decision trees (J48) are implemented as baseline classifiers along with the MLC methods. The research methodology follows the problem transformation approach previously explained. The experimental workflow (as shown in Figure 1) comprises of four phases: input data, pre-processing, prediction, and output results.

2.1. Experimental dataset

The dataset experimented in this work as given in Table 1 consists of 1098 verses (data instances) of the Quranic text. From the class weight distribution, faith ("*iman*") class label has the most class members (input verses). This is as expected since most of the *ayaat* (verses) of the Quran are connected to faith (*iman*). The primary sources of the Quranic textual data are the English translation of the Quran by Abdullah Yusuf

Ali (obtained from www.qurandatabase.org) and the English commentary by Ibn Kathir (obtained from www.allahsword.com). To the best of our knowledge, there is no availability of standard English Quran dataset for machine learning classification tasks.

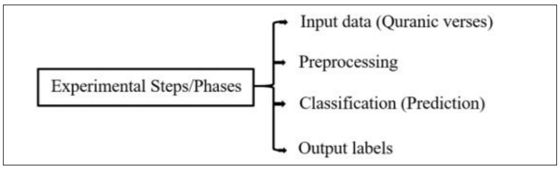

Figure 1. Experimental workflow

Table 1. Percentage composition of class labels

Dataset	No of Instances	Class Weight						
		Faith (*iman)*	Worship (*ibadah)*	Etiquettes *(akhlak)*	Faith-Worship (*iman-ibadah)*	Faith-Etiquettes (*iman-akhlak)*	Worship-Etiquettes (*ibadah-akhlak)*	Faith-Worship-Etiquettes *(iman-ibadah-akhlak)*
Quran	1098	1051.0	115.0	249.0	95.0	64.0	44.0	51.0

2.2. Text pre-processing

Preprocessing is the process of extracting features, normalizing, and transforming textual data suitable for analysis and implementation. The Quranic text is first converted to the standard attribute-relation File format (ARFF), which is the format for machine learning in Meka (an extension of Weka machine learning software). Thereafter, features are generated from the transformed text using StringToWordVector [25] and term frequency-inverse document frequency (TF-IDF) [26], [27]. These are standard filter tools for attributes (features) generation and extraction.

TF-IDF is one of the most widely-used method for accessing and measuring the significance of a word to a document. TF-IDF is a combination of two statistical weighting methods: term frequency (TF) and inverse document frequency (IDF). The term frequency $Tf(t,d)$ of a particular word t as expressed in (1) is defined as the number of times a word t appears in a document d. In addition, inverse-document frequency (expressed in (2)) is a method used to further verify if a term t is common/rare across all documents.

$$Tf(t,d) = 0.5 + \frac{0.5 \times f(t,d)}{Maximum\ Occurrences\ of\ words} \tag{1}$$

$$idf(t,D) = log\,\frac{N}{|\{d \in D : t \in d\}|} \tag{2}$$

where N is the total number of documents in D, $|\{d \in D : t \in d\}|$ is the number of documents where t featured.

2.3. Multi-label classification (MLC) models

Multi-label classification is the task of categorizing (or predicting) a set of data instances into one or more predefined labels using multi-label classification algorithms. In this experimental work, the problem transformation (PT) approach is adopted for the classification task. The study implemented three of the most applied PT methods: binary relevance (BR), classifier chain (CC), and label powerset (LP). In the classification/prediction phase, stratified 10-fold cross validation method [28], [29] is used for the training and testing process. Four traditional supervised learning algorithms: SVMs, NB, *k*NN, and J48, were implemented as baseline single-label classifiers with default parameter values as specified in Meka Toolbox for machine learning projects (obtained from https://sourceforge.net/projects/meka/). The input to the classifier is a Quranic verse represented by a vector of term count, while the outputs from the MLC classifiers are the predefined class labels: faith '*iman*', worship '*ibadah*', etiquettes '*akhlak*'. The multi-label classification methods are explained as follows:

1) Binary relevance (BR) is the most widely-applied problem transformation method. The MLC algorithm works by training multiple single-label binary classifiers. It builds M binary classifiers, one for each label

L (where M = L). In turn, each classifier predicts a yes/no (i.e., 0/1) per class. For a new instance, the BR method outputs all the positively predicted labels l_i by the M classifiers.
2) Classifier chain (CC) is also one of the most popular multi-label classification methods based on problem transformation approach. CC is a direct extension of binary relevance (BR) method. The MLC algorithm takes into consideration label dependency while retaining the simplicity and efficiency of the binary relevance method. CC works similar to BR by training first a classifier for each label L (where M = L). However different from BR, the algorithm makes predictions based on the chain order sequence of labels randomly initiated. The value of the first label in the sequence is predicted, then the predicted value along with its instance will be used as input to predict the value of the next label. This process continues following the randomly ordered chain sequence until the last label is predicted.
3) Label powerset (LP) multi-label classification algorithm is a simple but less popular of the problem transformation methods [30]. The MLC algorithm takes into consideration label correlations that may exist among the class labels. It considers each set of labels in the multi-label training data as one of the labels of a new single-label classification problem. For a new instance, the single-label classifier predicts the most likely label (which in return is a set of labels). The major setback with LP is high complexity [30] as a result of large number of possible label subsets combinations that could exist.

2.4. Evaluation metrics

In multi-label classification task, there are standard performance measures different from those conventionally used in single-label classification problems. Among these include hamming loss, one error, ranking loss, recall, precision, accuracy, and avg. Precision. In the experimental work, six standard performance metrics were employed for evaluating the multi-label classification algorithms. Given an evaluation dataset: (x_i, y_i); $i = 1, \dots, N$ denotes a multi-label data sample, $Y_i \subseteq L$ denotes set of true labels, $L = \{\lambda j : j = 1, \dots, M\}$ denotes set of all labels, Z_i denotes set of predicted labels, and $r_i(\lambda)$ denotes rank predicted for a label λ, the performance measures are explained as follows:

1) Hamming loss [31] is a standard performance metric that takes into consideration prediction errors (i.e., incorrect labels), and also missing errors (i.e., labels not predicted). The metric is used to evaluate the frequency of a misclassified label. The best performance is attained when hamming loss value is equal to zero i.e., the smaller the hamming loss, the better the performance of the MLC method.

$$Hamming\ Loss = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{M} \quad (3)$$

2) Accuracy is used to symmetrically measure how close a set of true labels (Y_i) is to a set of predicted labels (Z_i) [32]. Thus, the higher the accuracy value, the better the performance of the MLC method.

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{Y_i \cap Z_i}{Y_i \cup Z_i}\right| \quad (4)$$

3) One error evaluation metric [33] is used to measure the frequency of the top-ranked label that was not in the set of true labels. As its value tends towards zero, the best performance is reached.

$$One\ Error = \frac{1}{N}\sum_{i=1}^{N}\delta\left(arg_{\lambda \in L} min\, r_i(\lambda)\right) \quad (5)$$

4) Avg. precision measures the average fraction of labels ranked above a particular label $l \in Y_i$, which is actually in Y_i. It is the average of precision taken for all possible labels. The best result is achieved when avg. precision is 1 [34]. Thus, a larger avg. precision's value signifies a better performance.

$$Avg.P = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i \triangle Z_i|}{|Z_i|} \quad (6)$$

5) Micro-averaged F-measure (Micro-F1) [34] represents harmonic mean of micro-precision (Mic-P) and micro-recall (Mic-R).

$$Micro - F1 = \frac{2 \times (mic-P) \times (mic-R)}{(mic-P) + (mic-R)} \quad (7)$$

6) Macro-averaged F-measure (Macro-F1) [34] represents harmonic mean of macro-precision and macro-recall.

$$Macro - F1 = \frac{2 \times (mac-P) \times (mac-R)}{|L|} \quad (8)$$

3. EXPERIMENTAL RESULTS AND ANALYSIS

This section reports the experimental results of the study. Implementation was carried out using three standard multi-label classification methods: BR, CC, and LP. In addition, four traditional single-label classifiers: SVMs, NB, k-NN, and J48 were used as baseline classifiers. Also, six standard evaluation metrics were applied to validate the effectiveness of the classification algorithms.

The results obtained using the MLC methods along with the SLC baseline classifiers were exhaustively compared. Tables 2 to 4 showed the results comparison in terms of hamming loss, accuracy, one error, avg. precision, micro-F1, and macro-F1. In the bold are the best results achieved by the baseline SLC algorithms with respect to each of the evaluation metrics and MLC methods.

Table 2. Multi-label classification results using binary relevance algorithm

Evaluation metrics	BR NB	SVM	k-NN	J48
Accuracy↑	0.778	0.852	0.823	0.838
Hamming loss↓	0.186	0.107	0.129	0.114
One error↓	0.071	0.037	0.051	0.101
Micro-F1↑	0.807	0.875	0.843	0.866
Macro-F1↑	0.839	0.893	0.866	0.88
Avg. Precision↑	0.589	0.578	0.602	0.565

Table 3. Multi-label classification results using classifier chain algorithm

Evaluation metrics	CC NB	SVM	k-NN	J48
Accuracy↑	0.777	0.86	0.818	0.841
Hamming loss↓	0.187	0.106	0.133	0.115
One error↓	0.047	0.035	0.06	0.045
Micro-F1↑	0.806	0.88	0.836	0.865
Macro-F1↑	0.836	0.878	0.86	0.882
Avg. Precision↑	0.652	0.581	0.608	0.595

Table 4. Multi-label classification results using label powerset algorithm

Evaluation metrics	LP NB	SVM	k-NN	J48
Accuracy↑	0.797	0.86	0.817	0.829
Hamming loss↓	0.163	0.103	0.134	0.125
One error↓	0.034	0.039	0.06	0.051
Micro-F1↑	0.827	0.88	0.837	0.854
Macro-F1↑	0.855	0.898	0.859	0.873
Avg. Precision↑	0.614	0.583	0.606	0.606

In Table 2, implementation with multi-label BR method showed varying results across the baseline SLC algorithms. SVM classifier achieved the best results in most of the metrics evaluated, while decision trees (J48) algorithm followed closely. The NB algorithm had the least results across the evaluation metrics. This could be due to the nature of the experimental dataset since most learning algorithms are sensitive to data. In addition, the combination of the binary relevance MLC method with the learning algorithms could have a significant influence on the classification performance.

Assessing the performance of the CC multi-label classification method likewise showed competitive results. It could be seen that SVM classifier again achieved the best results across all evaluation metrics used except for avg. precision where the naïve Bayes algorithm displaced the classifier to top position achieving 65.2% avg. precision value. Furthermore, NB classification algorithm had the least results with the CC method closely similar to the binary relevance method. Since classifier chain MLC algorithm takes into consideration labels correlation, this had improvement over the binary relevance method. Consistently, the combination of CC and the baseline SLC algorithms performed better across the performance measures.

Table 4 reports the classification performance with LP multi-label classification algorithm. From the table, SVM classification model consistently proved to be an efficient and powerful learning algorithm. The baseline classifier had the overall highest scores of 86% accuracy, 88% micro-F1, 89.8% macro-F1, and 10.3% hamming loss. In terms of error rate and avg. precision, naïve Bayes classifier had better results of 0.034 and 61.4% respectively. As previously established, the nature of experimental datasets as well as the MLC methods applied on the learning algorithms may significantly influence the classification performance.

In general, analysis of the classification performance of BR, CC, and LP multi-label classification methods showed competitive results with the baseline classifiers: SVM, NB, k-NN, and J48 learning algorithms. This is due to the fact that every classifier has its strength and weakness. It is difficult to conclude on one ultimate best classifier. However, the SVM classification algorithm proved to be a consistent and efficient classifier. It achieved with BR, CC and LP multi-label classification methods, the best accuracy, micro-F1, and macro-F1 results of 86%, 88%, 89.8% respectively. Followed closely is the decision tree (J48) with the second highest accuracy, micro-F1, and macro-F1 results of 84.1%, 86.6%, 88.2% respectively. Consequently, SVM classifier performed best with the MLC methods achieving the best lowest hamming loss (0.103), while naïve Bayes classifier had the lowest error rate (0.034) and highest avg. precision value of 65.2%.

4. CONCLUSION

This research study is based on the application of multi-label classification methods in Quranic text (verses) labeling problem. In the experimental work, three MLC algorithms: BR, CC, and LP were implemented with four traditional single-label classifiers: NB, SVM, k-NN, and J48. The implementation followed the PT strategy, where the standard SLC algorithms functioned as the baseline classifiers. The classification performance was validated exhaustively using six standard evaluation metrics often employed in MLC problems. Consistently, the SVM classifier in combination with the MLC methods achieved the top ranked position. The SLC algorithm achieved the overall best results across the performance metrics. We could infer from the classification results that SVM learning algorithm is very efficient with relatively large dataset. In the future works, we looking forward to exploring and implementing MLC techniques to other related text classification problems. Also, the study will focus on the development of a complete English Quran dataset that could be standardized for machine learning tasks.

ACKNOWLEDGEMENTS

The authors would like to thank the Ministry of Higher Education, Malaysia for supporting this research under Fundamental Research Grant Scheme Vot K213 (FRGS/1/2019/ICT02/UTHM/02/2) and Universiti Tun Hussein Onn Malaysia for Multidisciplinary Research, Vot H511.

REFERENCES

[1] A. O Adeleke, N. A. Samsudin, A. Mustapha, and N. M. Nawi, “Comparative Analysis of Text Classification Algorithms for Automated Labelling of Quranic Verses,” *Int. J. on Advance Science, Engineering and Info. Tech,* vol. 7, no. 4, pp. 1419-1427, 2017, doi: 10.18517/ijaseit.7.4.2198.

[2] A. Adeleke, N. Samsudin, A. Mustapha, and S. A. Khalid, “Automating Quranic Verses Labeling Using Machine Learning Approach” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 16, no. 2, pp. 925-931, 2019, doi: 10.11591/ijeecs.v16.i2.pp925-931.

[3] J. Hartmaan, J. Huppertz, C. Schamp, and M. Heitmann, “Comparing automated text classification method,” *Int J. of Research in Marketing,* vol. 36, no. 1, pp. 20-38, 2019, doi: 10.1016/j.ijresmar.2018.09.009.

[4] N. K. Mishra and P. K. Singh, “FS-MLC: Feature Selection for Multi-label classification using clustering in feature space,” *Information Processing & Management,* vol. 57, mo. 4, 2020, doi: 10.1016/j.ipm.2020.102240.

[5] S. M. Garcia, C. J. Mantas, J. G. Castellano, and J. Abellan, “Non-parametric predictive inference for solving multi-label classification,” *Applied Soft Computing,* vol. 88, 2020, doi: 10.1016/j.asoc.2019.106011.

[6] M. Bogaert, J. Lootens, D. V. Poel, and M. Ballings, “Evaluating Multi-label classifiers and recommender systems in the financial service sector,” *European J. of Operational Research,* vol. 279, no. 2, pp. 620-634, 2019, doi: 10.1016/j.ejor.2019.05.037.

[7] Y. Zhang, Y. Wang, X.-Y. Liu, S. Mi, and M.-L. Zhang, “Large-scale multi-label classification using unknown streaming images,” *Pattern Recognition,* vol. 99, 2020, doi: 10.1016/j.patcog.2019.107100.

[8] L. A. C.-Diego, N. Bessis, and I. Korkontzelos, “Classifying emotions in stack overflow and JIRA using a multi-label approach,” *Knowledge-Based Systems,* vol. 195, 2020, doi: 10.1016/j.knosys.2020.105633.

[9] S. Khan and A. R. Baig, “Ant colony optimization based hierarchical multi-label classification algorithm,” *Applied Soft Computing,* vol. 55, pp. 462-479, 2017, doi: 10.1016/j.asoc.2017.02.021.

[10] R. B. Pereira, A. Plastino, B. Zadrozny, and L. H. C. Merschmann, “Correlation analysis of performance measures for multi-label classification,” *Information Processing and Management,* vol. 54, no. 3, pp. 359-369, 2018, doi: 10.1016/j.ipm.2018.01.002.

[11] T. Gong, B. Liu, Q. Chu, and N. Yu, “Using multi-label classification to improve object detection,” *Neurocomputing,* vol. 370, pp. 174-185, 2019, doi: 10.1016/j.neucom.2019.08.089.

[12] V. Kumar, A. K. Pujari, V. Padmanabhan, and V. R. Kagita, “Group preserving label embedding for multi-label classification,” *Pattern Recognition,* vol. 90, pp. 23-34, 2019, doi: 10.1016/j.patcog.2019.01.009.

[13] A. Akbarnejad and M.S. Baghshah, “A Probabilistic multi-label classifier with missing and noisy labels handling capability,” *Pattern Recognition Letters,* vol. 89, pp. 18-24, 2017, doi: 10.1016/j.patrec.2017.01.022.

[14] H. Cevikalp, B. Benligiray, and O. N. Gerek, “Semi-supervised robust deep neural networks for multi-label image classification,” *Pattern Recognition,* vol. 100, 2019, doi: 10.1016/j.patcog.2019.107164.

[15] F. Gargiulo, S. Silvestri, M. Ciampi, and G. D. Pictro, “Deep neural network for hierarchical extreme multi-label text classification,” *Applied Soft Computing,* vol. 79, pp. 125-138, 2019, doi: 10.1016/j.asoc.2019.03.041.

[16] J. Cao, S. Wang, R. Wang, X. Zhang, and S. Kwong, “Content-oriented image quality assessment with multi-label SVM classifier,” *Signal Processing: Image Communication,* vol. 78, pp. 388-397, 2019, doi: 10.1016/j.image.2019.07.018.

[17] B. Buddhahai, W. Wongseree, and P. Rakkwamsuk, “A non-intrusive load monitoring system using multi-label classification approach,” *Sustainable Cities and Society,* vol. 39, pp. 621-630, 2018, doi: 10.1016/j.scs.2018.02.002.

[18] R. Sousa and J. Gama, “Multi-label classification from high-speed data streams with adaptive model rules and random rules,” *Progress in Artificial Intelligent,* vol. 7, pp. 177-187, 2018, doi: 10.1007.

[19] R. Cerri, M. P. Basgalupp, R. C. Barros, and A. C. P. L. F. Carvalho, "Inducing Hierarchical Multi-label classification rules with Genetic Algorithms," *Applied Soft Computing*, vol. 77, pp. 584-604, 2019, doi: 10.1016/j.asoc.2019.01.017.
[20] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757-1771, 2004, doi: 10.1016/j.patcog.2004.03.009.
[21] G. Tsoumakas and K. Ioannis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 3, no. 3, pp. 1-13, 2007, doi: 10.4018/jdwm.2007070101.
[22] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, pp. 333-359, 2011, doi: 10.1007/s10994-011-5256-5.
[23] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, vol. 5782, pp. 254-269, 2009, doi: 10.1007/978-3-642-04174-7_17.
[24] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music by emotion," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 1, pp. 1-9, 2011, doi: 10.1186/1687-4722-2011-426793.
[25] A. Perez, R. Basagoiti, R. A. Cortez, F. Larrinaga, E. Barrasa, and A. Urrutia, "A case study on the use of machine learning techniques for supporting technology watch," *Data & Knowledge Engineering*, vol. 117, pp. 239-251, 2018, doi: 10.1016/j.datak.2018.08.001.
[26] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec," *Information Sciences*, vol. 477, pp. 15-29, 2019, doi: 10.1016/j.ins.2018.10.006
[27] M. Tan, J. Zhou, Z. Peng, J. Yu, and F. Tang, "Fine-grained image classification with factorized deep user click feature," *Information Processing & Management*, vol. 57, no. 3, 2020, doi: 10.1016/j.ipm.2019.102186Get.
[28] N. K. Mishra and P. K. Singh, "Feature construction and smote-based imbalance handling for multi-label learning," *Information Sciences*, vol. 563, pp. 342-357, 2021, doi: 10.1016/j.ins.2021.03.001.
[29] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011, pp. 145-158, doi: 10.1007/978-3-642-23808-6_10.
[30] A. A. Gonzalez, J. F. D.-Pastor, J. J. Rodriguez, and C. G. Osorio, "Study of data transformation techniques for adapting single-label prototype selection algorithms to multi-label learning," *Expert Systems with Applications*, vol. 109, pp. 114-130, 2018, doi: 10.1016/j.eswa.2018.05.017.
[31] G. Nan, Q. Li, R. Don, and J. Liu, "Local positive and negative correlation-based k-labelsets for multi-label classification," *Neurocomputing*, vol. 318, pp. 90-101, 2018, doi: 10.1016/j.neucom.2018.08.035.
[32] S. Vluymans, C. Cornelis, F. Herrera, and Y. Saeys, "Multi-label classification using a fuzzy rough neighborhood consensus," *Information Sciences*, vol. 433-434, pp. 96-114, 2018, doi: 10.1016/j.ins.2017.12.034.
[33] Z. Zhang and C. Sun, "Multi-site structural damage identification using a multi-label classification scheme of machine learning," *Measurement*, vol. 154, 2020, doi: 10.1016/j.measurement.2020.107473.
[34] A. Law and A. Ghosh, "Multi-label classification using a cascade of stacked autoencoder and extreme learning machines," *Neurocomputing*, vol. 358, pp. 222-234, 2019, doi: 10.1016/j.neucom.2019.05.051.