Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 2871-2883
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2021.25379.2998

Integration of deep learning model and feature selection for multi-label classification

Hossein Ebrahimi[a], Kambiz Majidzadeh[a,*], Farhad Soleimanian Gharehchopogh[a]

[a]*Department of IT and Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Multi-label data classification differs from traditional single-label data classification, in which each input sample participated with just one class tag. As a result of the presence of multiple class tags, the learning process is affected, and single-label classification can no longer be used. Methods for changing this problem have been developed. By using these methods, one can run the usual classifier classes on the data. Multi-label classification algorithms are used in a variety of fields, including text classification and semantic image annotation. A novel multi-label classification method based on deep learning and feature selection is presented in this paper with specific meta-label-specific features. The results of experiments on different multi-label datasets demonstrate that the proposed method is more efficient than previous methods.

Keywords: Machine Learning, Classification, Multi-Label, Meta-Label-Specific Features, Deep Learning.

1. Introduction

One of the most important topics in machine learning is classification. The goal of classification is to develop a computational model that can correctly classify new unlabeled samples using a set of labelled samples. A well-established paradigm of machine learning is single-label classification. The model predicts accurately and quickly in a wide array of fields. Single-label classification involves learning from a set of samples which are associated with a single label, such as binary classification and multi-class classification. Multi-label classification (MLC) assigns a set of relevant labels to an instance simultaneously, unlike traditional classification. Recent developments and applications of

*Corresponding author
Email addresses: H.seyedebrahimi@iaurmia.ac.ir (Hossein Ebrahimi), K.majidzadeh@iaurmia.ac.ir (Kambiz Majidzadeh), Bonab.farhad@gmail.com (Farhad Soleimanian Gharehchopogh)

Received: October 2021 *Accepted:* December 2021

MLC include diagnostics, music analysis, emotion detection, and image/video annotation. Each of these unseen instances must be labeled. Among many molecular functions that can be tied to one gene sequence in bioinformatics, for instance. In-text categorization, a new article can cover multiple aspects of an event, thus being assigned to a set of multiple topics [31, 18, 28].

MLC tasks can be solved using either problem transformation or algorithm adaptation. In the former, the MLC task is converted into a single-label classification task, called label ranking (LR) [25]. In the latter, traditional machine learning algorithms are adapted to handle multi-label data sets (MLDs). In general, Binary Relevance (BR), Label Powerset (LP) and Classifier Chains (CC) are the three most common methods for transforming problems. This method converts a multi-label problem into an independent binary problem. Following that, a traditional classifier is used to process each binary problem [20, 1, 17, 13]. By identifying each sets of labels uniquely, the LP creates a multi-class dataset from the original MLD. It is then used to train a regular classification algorithm, which predicts classes and returns subsets of labels. Many multi-label ensemble-based methods are based on both BR and LP. By considering the label correlation task, CC overcomes the BR limitations [22].

Algorithms are adapted to accommodate MLC-specific changes as part of the algorithm adaptation process. An MLD is implemented by revisiting the single-label classification process. Multilabel k Nearest Neighbors (MLKNN), multiclass multilabel perceptron (MMP), and Ranking Support Vector Machine (Rank-SVM) have all been proposed as adaptations of traditional classifiers [40, 23].

MLC suffers from the imbalanced nature of the data space, where the samples are not evenly distributed, and the labels are not uniformly positioned. It is not effective for coping with the imbalance problem in an MLC to apply problem transformation and adaptation techniques [14, 35, 11]. The imbalanced nature of datasets presents a significant challenge for many real-world applications, such as fraud detection, risk management, and medical diagnosis. A diagnostic problem involving a disease that is rare as compared to healthy people within a population, for instance, requires the task to look for people with diseases. A classification model that could properly label rare patterns is hence an effective one. With single-label classification, imbalanced class distributions have been extensively studied using commonly used approaches, such as resampling methods. As a result of imbalances between labels and label-sets, the existing strategies cannot directly address an MLC's imbalanced problem. Having more labels complicates the imbalance problem [30].

Three important criteria for classifying and comparing multi-label classifiers are presented in the following due to their importance and widespread use [33].

Firstly, how well the learning algorithm deals with multi-label data sets is the first criterion. On multi-label data, supervised-based learning methods can be divided into two general categories. The first group of problems is called transitional problems, while the second group is called adaptation methods [38].

Transitional learners map the problem to a single label. On the other hand, the second category includes methods that can directly apply multi-label data [8, 27, 9].

In addition, the classification based on the output generated by multi-label learners is important. The output of a multi-label learner is either a classification model or a label rating model. For each test sample, a model that performs multi-label classification can specify relevant and unrelated labels. In label rating, however, all available tags are ranked for every sample.

An integrated Deep learning and selection method is proposed in this paper for the classification of multi-label datasets. In this method, the features are presented as graph models, and the selected feature sets are then selected using a deep learning-based model in order to perform the multi-label classification.

In the reminder of this paper, in section 2 the related works are reviewed, in section 3 the details

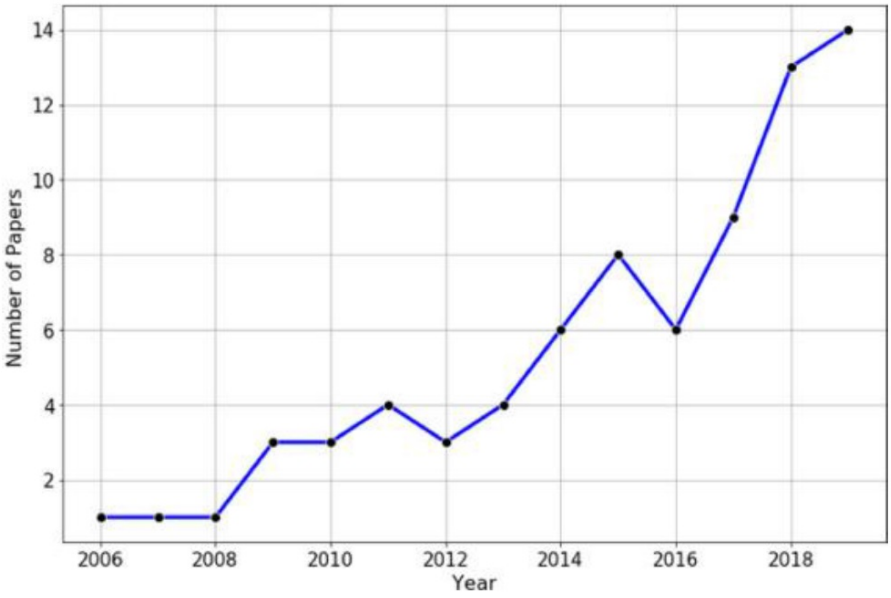

Figure 1: Publishing trends for Imbalanced Multi-label Classification.

of proposed method are explained, then the performance of the proposed method is evaluated in section 4, and finally the overall of this paper is concluded.

2. Related Works

A Multi-Label Classification (MLC) is a method of arranging data by categorizing it by more than one label at once. A wide range of application domains has contributed to the growth of MLC in recent years. In multi-label datasets, the problem of class imbalance has, however, become a characteristic of the data set, where the samples and their corresponding labels are not distributed randomly over the space. In Fig. 1, the number of imbalanced multi-label publications is presented from 2006-2019. In comparison to other periods, the number of publications has grown steadily between 2012 and 2015, and between 2016 and 2019. A decrease in the number of publications occurred in 2016 in comparison to 2015, followed by an increase in subsequent years. Over the past few years, imbalanced MLC has been the subject of many new publications. Research has focused extensively on imbalanced MLC, suggesting that the topic continues to be of value to researchers. The research literature on multi-classification will be reviewed in this section [2, 21, 41].

2.1. Transfer models

In this category, methods attempt to break down the multi-tag data set into single tags. Otherwise, you need to break down the original multi-tag data set. The data set is then classified using single-tag classifiers. As a result, each classifier is combined to produce a multi-tag classifier.

It was suggested in [2] that multi-labeled datasets can be converted to single labeled datasets by using six different techniques:

- Pick the most relevant tag set for each instance based on its index.
- To select a label for each sample, find the one with the lowest index among its associated tag sets.
- Make copies of the samples so that each copy is assigned only one sample tag.

- We assign weights to every pair (sample, label) as before
- Randomly selecting one of the samples
- If a sample has more than one tag, delete it.

It is a disadvantage of these methods that they destroy much of the data in an initial multi-tag data set. Some tags are lost in the samples. Because of this, the learning algorithm will not incorporate all the information from the initial data set. This model is clearly less efficient than one created from the entire initial training dataset. [21] proposes that a new technique, "K-way Tree based eXtreme Multi-Label Classifier (KTXMLC)" is useful for maintaining correlations between features using feature-label representation technique and node partitioning with clustering algorithm. Furthermore, in [41], a new re-weighting algorithm for multi-label classification problems is introduced. Using multi-label classification, dual aggregated networks are proposed in [36]. An approach was developed to identify discriminant multiscale information about different target objects from image data by aggregating both feature and classifier levels. For the development of data-driven label clustering, a new label dependence criterion was presented in [34].

A methodology described in [32] provides four common tricks associated with data analysis. Multi-labeled datasets are each divided into single-labeled datasets by each of these programs. These methods include: One Versus Rest (OVR), One Versus One (OVO), One By One (OBO), and Label Powerset (LP). In addition, a deep learning-based approach combined with label-attention and domain-specific pre-training is proposed for the classification of multi-label legal documents [24]. An algorithm for multilabel feature selection based on optimization is proposed in [6]. The paper uses an improved NSGA III algorithm and two archives to enhance the diversity and convergence of NSGA III.

2.2. Adaptation models

Data analysis and algorithm development are combined in a method called adaptation. For the first category of algorithms, a multi-label classifier is produced that can consider all instances and all classes in the training dataset at once. Second, a single-label classifier can be improved, and multi-label datasets can be subdivided into subsets. The second method has produced and designed several efficient and effective multi-label classification algorithms.

Methods currently used in adaptation are based on dependence rule learners, decision trees, sample-based methods, neural networks, and SVM classifiers.

MMAC, which is a technique presented in [26], creates classification rules by using a set of dependencies explored in that paper. After that, it removes the instances that follow these rules, and repeats the rule search on the remaining data. As long as, at least one rule is followed, all samples will be considered. Compared to other methods, this method is suited to training samples, as its flexibility on new data is reduced.

3. Proposed Method

To improve classification accuracy for multi-label data, a deep learning-based approach is presented in this section. Graph-based feature selection is also used in this method to reduce data dimensions.

There are many attributes associated with data. In the case of data mining applications, many of these features are irrelevant or redundant. These unrelated and redundant features negatively affect the machine learning algorithm and increase computational complexity. The reduction of data

size is therefore a fundamental task of data mining and machine learning. Models with reduced features are more generalizable than original models. For a classification task with n dimensions and C classes, a minimum of 10 X n X C is required. Whenever it is not practicable to provide this much training data, reducing the features reduces the amount of training data needed. Consequently, the classification algorithm's performance improves [3, 10, 29].

Managing large data dimensions is harder than managing small data dimensions or performing computations and analyses. In light of this, dimensionality is a crucial part of knowledge discovery. Despite the opportunities they create, multidimensional data platforms present many computational challenges. Large data sets have the disadvantage that most of the time all the features are not important to finding the knowledge hidden within. Dimensionality reduction is therefore important in many fields.

There are two main methods for reducing dimensions:
Methods that map a multidimensional space into a smaller one are known as feature-based extraction methods. When they combine the values of existing attributes, they create fewer attributes that contain all (or most) of the information in the original attributes. In general, there are two categories of methods: linear and nonlinear. Multidimensional spaces can be mapped into smaller spaces using feature extraction methods. A linear method and a nonlinear method belong to two different categories. The simplest and easiest linear methods are those that seek out a sub-public space. Nevertheless, nonlinear methods seek to find subliminal spaces, which are extremely difficult to analyze.
Methods based on feature selection: These methods reduce data size by selecting a subset of the primary features. When analyzing data, reduction spaces, or classification are favored over main spaces. Feature selection is a common solution to the problem of reducing dimensions. By removing the irrelevant and redundant attributes, a subset of the primary attributes is selected. There is a limitless space of possible subsets in the search space for identifying the most appropriate features.

In this paper, a deep learning-based feature selection method is used to reduce the initial dimension of a dataset. Figure 2 illustrates the general flowchart of the proposed method.

To apply the graph-based methods, the solution space of the feature selection must be demonstrated by a weighted graph. To this end, in the first step, the initial features are shown with a graph Graph $=< G, E >$, where $G = \{G_1, G_2, ..., G_n\}$ is a set of original features in which each feature shows a node in the graph, $E = \{(G_i, G_j) : G_i, G_j \in G\}$ denotes the set of edges of the graph, and w_{ij} indicates the similarity between two features G_i and G_j that are linked by the edge (G_i, G_j). This study employs the Pearson similarity criteria [19] to compute the similarity value between different features. This similarity between the two features G_i and G_j is calculated as below:

$$w_{ij} = \left| \frac{\sum_p (x_i - \overline{x_i})(x_j - \overline{x_j})}{\sqrt{\sum_p (x_i - \overline{x_i})^2}\sqrt{\sum_p (x_j - \overline{x_j})^2}} \right| \tag{3.1}$$

Where x_i and x_j denote the vectors of features G_i and G_j, respectively. Variables $\overline{x_i}$ and $\overline{x_j}$ indicate the average of vectors x_i and x_j, over p samples. Greater similarity between the two features causes the Pearson criterion between the two features to be closer to one, and reciprocally the dissimilarity of the two features will cause the Pearson criterion of the two features to be closer to 0.

The initial features are grouped in several clusters in the second step of the proposed feature selection. Based on their similarity, feature clustering is designed to divide the primary features into several groups. Accordingly, each cluster has many features in common, and a different cluster has fewer features in common. Prior to implementing the clustering algorithm, the number of clusters is generally determined [12]. This means, most of these methods require the user to specify the clusters

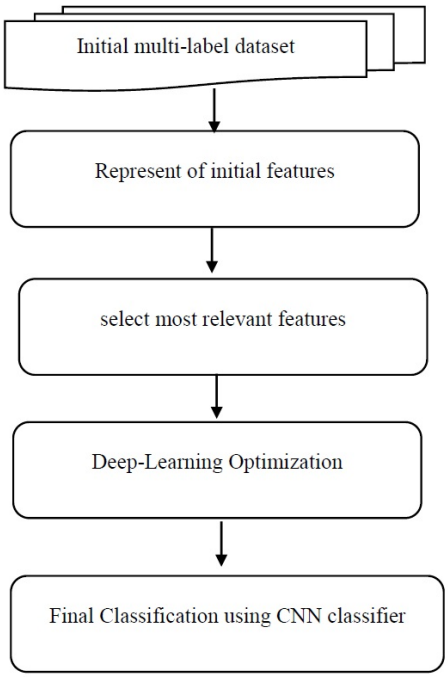

Figure 2: Overall flowchart of the proposed method

in parameter k. A number of clusters can only be determined by trial and error as a function of the initial characteristics of the work, not by formulae. Therefore, we used a community detection algorithm to cluster the data in this paper, called Louvain [4]. It maximizes a modularity function to detect communities in a graph. Identification of communities in large networks is simple, efficient, and easily implemented. Due to its computational complexity of $O(n \log n * n)$, which can be used to detect communities in very large networks within short computing times, the algorithm can be used to detect communities in very large networks. In two steps, the method detects communities in a network. The first step involves assigning each node to a community, in order to increase the specificity of the network. The second step involves merging the previously defined communities to create a brand new network. The process iterates until networks are significantly improved in terms of modularity. Two advantages are associated with this method. As a first advantage, its steps are intuitive and easy to follow; as a second advantage, the algorithm is extremely fast.

The third stage involves selecting the appropriate features from each cluster. Here, using the concept of term variance, the proposed method is intended to find the optimal feature subset. This step involves selecting a number of features from each cluster that are well suited to represent all the characteristics of that cluster. When features from each cluster are clustered and the most effective features are selected, the selected features cover all of the features that matter. Utilizing the variance criterion, the most important features are selected from each. This is one of the minimally complex and highly efficient term variance (TV) criteria. By analyzing the term variation criterion, we determine the strength of the attribute as below:

$$TV_i = \frac{1}{\mid G \mid} \sum_{j=1}^{|G|} (Attr(j,i) - \overline{Attr(i)})^2 \tag{3.2}$$

where, $Attr(j,i)$ indicated attribute j of the nodes i in dataset and $\mid G \mid$ shows the number of all features in dataset.

The final classification step follows the feature selection step. Multi-label classification is handled at this stage by using the deep learning algorithm. In the construction industry, deep learning has become a topic of interest over the past few years. Deep learning has been gaining popularity in many fields for data recognition, processing, and decision-making. Modern records management technologies provide large amounts of information requiring manual parsing and processing, a lengthy, inefficient, and error-prone process. By automating these tasks, variances would be able to be detected faster and respond to negative impacts quickly. In a daily construction cycle, deep learning has proven to be a highly effective tool for identifying objects or tracking processes that would vastly improve daily operation. One of the advantages of deep learning is its ability to create predictive models without pre-defining relationships. Despite its hefty size, the construction industry still faces a number of challenges. Project success is traditionally determined by factors such as time, cost, and quality. A project's status can be determined by tracking metrics during its development. Construction projects can, however, face delays or exceed their budget. There are many variables and quantifiable metrics that determine the success of a project, but uncertainties can delay and damage projects. We could greatly benefit from automatic anomaly detection onsite. The monitoring of structures and infrastructure post-construction is crucial for the prompt detection of damage. The purpose of this paper is to show how deep learning can help solve many of the problems that modern construction projects face. An effective infrastructure management and project control solution can be delivered using such a solution.

Over the past few years, the number of studies on deep learning in construction has grown exponentially and applications have spread to many areas of construction since their introduction. The use of deep learning in these applications includes tracking and monitoring construction operations, equipment usage, worker productivity, and uncovering ways to improve project performance and efficiency. In addition, deep learning can be used to detect unsafe work and monitor workers. In addition, deep learning can be used to automatically identify construction workers and pair them with their respective qualifications so that they are working within the scope of their expertise.

Computers can learn from past experiences through deep learning, a subset of machine learning. It imitates biological neural networks by using artificial neural networks and other machine learning algorithms. By combining these layers, we may extract features, transform them, and analyze patterns using supervised or unsupervised learning. Similar to the brain, DL categorizes and labels the data it receives to understand it. Despite the fact that some machine learning algorithms might not be able to differentiate and learn multiple complexity levels from a variety of data sets, this technique can. Also, it can handle unstructured and unlabeled data without supervision. As there are many layers in it, it is regarded as 'deep'. An DL model typically consists of three layers: input (received data), hidden (extracts patterns), and output (produces the results). As one-layer outputs, the next layer receives it as an input.

Deep learning has drastically enhanced many industries due to its ability to automate tasks that were previously manual and time-consuming, which is the reason for its increasing popularity in this digital age. Natural language processing, speech recognition, image processing, and video and voice recognition are some areas where deep learning techniques are applied.

After selecting an appropriate subset of features, these features are used as the input of the classification model. Accurate features selection, as well as the use of efficient prediction model to analyze data, will lead to higher classification. A variety of machine learning algorithms for classification are proposed in the previous works. The machine learning classification algorithm such as such as support vector machine, artificial neural network, deep learning, fuzzy system, and ensemble learning models are used for the multi-label classification.

In this paper Convolutional Neural Network (CNN) is utilized for final classification. The Convo-

lutional Neural Network is one of the most widely used methods for deep learning. Based on living creatures' natural perception mechanisms, these images were created. These feed-forward networks consist of multiple convolutional, pooling, and fully connected layers and require large datasets to train [3, 10, 29, 19]. In the convolutional layer, input characteristics are identified and studied. They can identify objects such as shapes and objects such as edges and lines using this method. In the pooling layer, the number of features fed into it is minimized by acting as a funnel between two convolutional layers [36]. In each layer the role is to deliver an output regarding the classification of messages (depending on how complex the system needs to be). Information is summarized in the fully connected layers usually at the end. The CNN is used to identify features in an image by incorporating values from its pixels. The strength of this method is extracting features and constructing representations, while its weakness is maintaining parameter tuning. In addition to the standard CNN method, there are variations that incorporate regions-based CNN (or R-CNN), fast R-CNN, and faster R-CNN [12, 4, 7].

Through a differentiable function, a CNN architecture transforms the input volume into an output volume (e.g. holding class scores). Layers are commonly used in a few different ways. Further information is provided below.

3.1. Convolutional layer

A CNN's core building block is the convolutional layer. In this layer, the parameters are learned filters (or kernels) that have a small receptive field, but extend to the depth of the input volume. For each filter, the dot product between the entries and the input volume is computed during the forward pass to produce a 2-dimensional activation map. By learning filters like this, the network can detect certain types of features when they occur at specific positions in the input [39].

3.2. Local connectivity

As a result, it is not possible to connect all neurons in a previous volume with high-dimensional inputs such as images since such a network architecture lacks spatial sensitivity. The convolutional network exploits spatially local correlation by forcing sparse connectivity among neurons in adjacent layers: each neuron is only connected to a small region of the input volume.

This connectivity is measured by a hyperparameter called the receptive field of the neuron. During a given input volume's depth, the total number of connections is always local (across width and height). With such an architecture, the learned filters react strongly to an input pattern that is spatially local [15].

3.3. Spatial arrangement

The depth, stride, and padding size of the convolutional layer determine the output volume of the layer.

Input volume depth determines how many neurons connect to a particular region within an input volume. Different features in the input cause these neurons to fire. With the raw image taken as input, the first convolutional layer may activate different neurons in response to oriented edges or blobs of color.

Height and width columns are assigned based on the stride. If the stride is 1, then we move the filters one pixel at a time. This leads to heavily overlapping receptive fields between the columns, and to large output volumes. For any integer $S > 0$ a stride S means that the filter is translated S units at a time per output. In practice, $S > 3$ is rare. Greater stride means smaller overlaps between receptive fields and smaller volumetric output dimensions.

Occasionally, padding the input volume with zeros (or other values) on the boundary of the input is convenient. A third hyperparameter controls the size of this padding. By padding the output volume, the spatial size of the output can be controlled. A "same" padding is a way to retain the spatial size of an input volume, especially when it is desirable [16, 5, 37].

The size of the output volume depends on the size of the input volume W, the kernel field size K of the convolutional layer neurons, the stride S, and the amount of zero padding P on the border. The number of neurons that "fit" in a given volume is then:

$$\frac{W - K + 2P}{S} + 1$$

In all other cases, the strides are incorrect, and the neurons cannot be tiled to fit across the input volume in a symmetric manner. If the stride is $S = 1$ and the zero padding is set to $P = (K - 1)/2$, the input volume and output volume will have the same spatial size. Using every neuron from the previous layer is not necessary in every case. In the design of neural networks, the padding may be used only for part of the calculation.

4. Experimental Results

A comparison of proposed multi-label classification methods is presented in this section. In this context, one of the new methods in this field is compared to the proposed one.

MATLAB programming language is used in this article to implement feature selection methods. Also, all the tests in this article were performed on a system with a 2.3 GHz Corei3 processor and 2 GB of internal memory (RAM). In the rest of this section, the data sets used to identify accident-affected areas, as well as practical results, are described, respectively.

We conducted experiments on a variety of real-world multi-label datasets, including Emotions, Scene, Yeast and Genbase datasets.

In the reminder of this section, we introduce the classification evaluation criteria that we use to evaluate the performance of the proposed systems.

The three criteria that are mainly used to evaluate the quality of performance of algorithms for detecting damaged areas are Sensitivity, Specificity, and Classification Rate, are used.

Execution time is also a criterion used in this article to compare different methods. This criterion is used to assess the computational complexity of different methods. It is clear that an automatic classifier will be more efficient and less computationally complex the longer it takes to identify damaged areas. Data sets are randomly divided into training and experimental data during experiments on the proposed method. A data set consisting of 70% educational data and 30% experimental data will be used for this purpose. The method of classification was also applied ten times in all experiments. The method of classification was compared using an average of ten different implementations after identifying the educational and experimental sets.

In Tables 1,2,3,4, the proposed deep learning-based method, is compared with Multi-Label Classification with weighted classifier [31]. In this table, Sensitivity, Specificity, and Classification rates are evaluated in the proposed method and the base paper method. As the results of these experiments show, in all cases, the proposed method is more accurate than the method compared.

Also, Figure 3 shows the average accuracy of the proposed method on different datasets. Also, Figure 3 shows the average accuracy of the proposed method on different datasets. As can be seen in this figure, the proposed method has a higher average accuracy than the base paper method.

Moreover, Table 5 also shows the execution time for the proposed method and Multi-label classification with weighted classifier, respectively. As the results of this experiment show, the proposed method has a shorter execution time.

Table 1: Comparison of the proposed method with Multi-label classification with weighted classifier on Emotions dataset

	Multi-label classification with weighted classifier			Proposed method		
	Best	Worst	Average	Best	Worst	Average
Sensitivity (%)	84.72	81.48	83.73	86.21	82.76	85.79
Specificity (%)	86.78	83.94	85.99	87.12	83.38	86.77
Classification (%)	85.51	81.31	84.43	87.39	84.54	85.67

Table 2: Comparison of the proposed method with Multi-label classification with weighted classifier on Scene dataset

	Multi-label classification with weighted classifier			Proposed method		
	Best	Worst	Average	Best	Worst	Average
Sensitivity (%)	82.63	81.65	83.48	87.82	80.38	86.54
Specificity (%)	87.71	83.43	84.22	87.47	85.82	84.71
Classification (%)	86.18	82.33	84.01	87.28	83.62	84.54

Table 3: Comparison of the proposed method with Multi-label classification with weighted classifier on Yeast dataset

	Multi-label classification with weighted classifier			Proposed method		
	Best	Worst	Average	Best	Worst	Average
Sensitivity (%)	84.61	81.41	83.22	87.28	83.91	84.17
Specificity (%)	86.89	82.04	84.21	87.86	83.81	84.43
Classification (%)	85.09	80.28	83.08	86.99	84.58	85.61

Table 4: Comparison of the proposed method with Multi-label classification with weighted classifier on Genbase dataset

	Multi-label classification with weighted classifier			Proposed method		
	Best	Worst	Average	Best	Worst	Average
Sensitivity (%)	84.71	80.42	83.72	86.81	82.97	86.02
Specificity (%)	85.39	82.31	84.70	88.59	84.87	86.72
Classification (%)	84.42	80.87	82.82	86.02	83.42	84.63

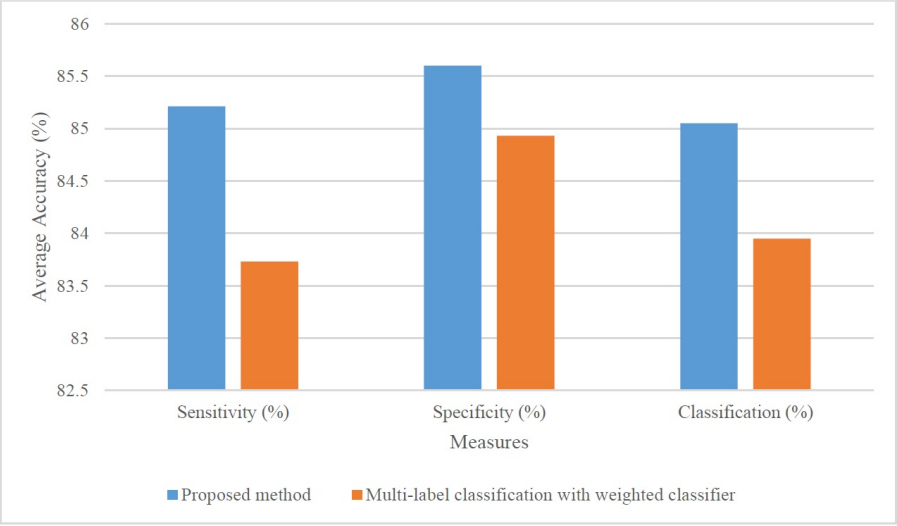

Figure 3: Average accuracy of the proposed method on different datasets and average accuracy of the proposed method on different datasets

Table 5: Comparison of expectation time in proposed method and Multi-label classification with weighted classifier

Execution Time	Multi-label classification with weighted classifier	Proposed method
1	119	86
2	118	81
3	112	80
4	115	84
5	118	90
6	125	76
7	113	81
8	108	80
9	118	91
10	121	79
Average	114.7	84.6

5. Conclusion

Using large amounts of data, data mining aims to find patterns and rules hidden in the data and to make sense of it. It can be argued that any data set can provide valuable information, and the key point is that any amount of data can contain something valuable, and we can access that by analyzing the data. As a form of data analysis, classification enables us to build models to describe data or construct directional mirrors for data. In multi-label classification, each data point is tagged with one or more tags. These tags are referred to as related tags. multi-label learners are designed to map multi-label data to group of related tags. As a data mining learning algorithm, multi-label classifiers are a popular learning pattern. Graph-based feature selection and deep learning are combined in this paper to improve the accuracy of multi-label classification. According to the numerical results, the proposed method has a higher accuracy than previous methods, as well as a lower computational complexity.

References

[1] Z. Al-Makhadmeh and A. Tolba, *Utilizing IoT wearable medical device for heart disease prediction using higher order Boltzmann model: A classification approach*, Measurement 147 (2019) 106815.

[2] M.R. Boutell, J. Luo, X. Shen and C.M. Brown, *Learning multi-label scene classification*, Pattern Recogn. 37(9) (2004) 1757–1771.

[3] H. Chao, X. Fang, J. Zhang, F. Homayounieh, C.D. Arru, S.R. Digumarthy, R. Babaei, H.K. Mobin, I. Mohseni, L. Saba, A. Carriero, Z. Falaschi, A. Pasche, G. Wang, M.K. Kalra and P. Yan, *Integrative analysis for COVID-19 patient outcome prediction*, Med. Image Anal. 67 (2021) 101844.

[4] H. Chen, W. Cai, F. Wu and Q. Liu, *Vehicle-mounted far-infrared pedestrian detection using multi-object tracking*, Infrared Phys. Tech. 115 (2021) 103697.

[5] J. Djolonga, J. Yung, M. Tschannen, R. Romijnders, L. Beyer, A. Kolesnikov, J. Puigcerver, M. Minderer, A. D'Amour, D. Moldovan and S. Gelly, *On robustness and transferability of convolutional neural networks*, Proc. IEEE/CVF Conf. Computer Vision and Pattern Recogn. (2021) 16458–16468.

[6] H. Dong, J. Sun, X. Sun and R. Ding, *A many-objective feature selection for multi-label classification*, Knowledge-Based Syst. 208 (2020) 106456.

[7] W.A. Ezat, M.M. Dessouky and N.A. Ismail, *Evaluation of deep learning YOLOv3 algorithm for object detection and classification*, Menoufia J. Elect. Eng. Res. 30(1) (2021) 52–57.

[8] A. Hashemi, M.B. Dowlatshahi and H. Nezamabadi-Pour, *MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality*, Expert Syst. Appl. 142 (2020) 113024.

[9] J. Hu, Y. Li, W. Gao and P. Zhang, *Robust multi-label feature selection with dual-graph regularization*, Knowledge-Based Syst. 203 (2020) 106126.
[10] B. Jeong, N. Ko, C. Son and J. Yoon, *Trademark-based framework to uncover business diversification opportunities: Application of deep link prediction and competitive intelligence analysis*, Comput. Indust. 124 (2021) 103356.
[11] Z. Ji, B. Cui, H. Li, Y.-G. Jiang, T. Xiang, T. Hospedales and Y. Fu, *Deep ranking for image zero-shot multi-label classification*, IEEE Trans. Image Proces. 29 (2020) 6549–6560.
[12] L. Kalake, W. Wan and L. Hou, *Analysis based on recent deep learning approaches applied in real-time multi-object tracking: A review*, IEEE Access 9 (2021) 32650–32671.
[13] C. Khammassi and S. Krichen, *A GA-LR wrapper approach for feature selection in network intrusion detection*, Comput. Secur. 70 (2017) 255–277.
[14] S. Khandagale, H. Xiao and R. Babbar, *Bonsai: diverse and shallow trees for extreme multi-label classification*, Machine Learn. 109(11) (2020) 2099–2119.
[15] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj and D.J. Inman, *1D convolutional neural networks and applications: A survey*, Mech. Syst. Signal Sroces. 151 (2021) 107398.
[16] Z. Li, F. Liu, W. Yang, S. Peng and J. Zhou, *A survey of convolutional neural networks: analysis, applications, and prospects*, IEEE Trans. Neural Networks Learn. Syst. (2021) 1–21.
[17] M. Mafarja and S. Mirjalili, *Whale optimization approaches for wrapper feature selection*, Appl. Soft Comput. 62 (2018) 441–453.
[18] F. Morais-Rodrigues, R. Silverio, R.K. Kato, D.L.N. Rodrigues, J. Valdez-Baez, V. Fonseca, E.J. San, L.G.R. Gomes, R.G. dos Santos, M.V.C. Viana, J.d.C. Ferraz Dutra, M.T.D. Parise, D. Parise, F.F. Campos, S.J. de Souza, J.M. Ortega, D. Barh, P. Ghosh and M.A. dos Santos, *Analysis of the microarray gene expression for breast cancer progression after the application modified logistic regression*, Gene 726 (2020) 144168.
[19] A.A. Ojugo and O. Nwankwo, *Spectral-cluster solution for credit-card fraud detection using a genetic algorithm trained modular deep learning neural network*, J. Inf. Visual. 2(1) (2021) 15–24.
[20] D. Paul, A. Jain, S. Saha and J. Mathew, *Multi-objective PSO based online feature selection for multi-label classification*, Knowledge-Based Syst. 222 (2021) 106966.
[21] P. Prajapati and A. Thakkar, *Performance improvement of extreme multi-label classification using K-way tree construction with parallel clustering algorithm*, J. King Saud Univ. Comput. Info. Sci., (2021).
[22] I. Rejer and M. Twardochleb, *Gamers' involvement detection from EEG data with cGAAM – A method for feature selection for clustering*, Expert Syst. Appl. 101 (2018) 196–204.
[23] M.S. Salman, Y. Du, D. Lin, Z. Fu, A. Fedorov, E. Damaraju, J. Sui, J. Chen, A.R. Mayer, S. Posse, D.H. Mathalon, J.M. Ford, T.V. Erp and V.D. Calhoun, *Group ICA for identifying biomarkers in schizophrenia: 'Adaptive' networks via spatially constrained ICA show more sensitivity to group differences than spatio-temporal regression*, NeuroImage: Clinic. 22 (2019) 101747.
[24] D. Song, A. Vold, K. Madan and F. Schilder, *Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training*, Inf. Syst. (2021) 101718.
[25] Q.Z. Song, L. Zhao, X.K. Luo and X.C. Dou, *Using deep learning for classification of lung nodules on computed tomography images*, J. Healthcare Engin. 2017 (2017) Article ID 8314740.
[26] F.A. Thabtah, P. Cowling, Y. Peng, R. Rastogi, K. Morik, M. Bramer and X. Wu, *MMAC: A new multi-class, multi-label associative classification approach*, Proc. Fourth IEEE Int. Conf. Data Min. 2004 (2004) 217–224.
[27] W. Wang, Q.Y. Dai, F. Li, Y. Xiong and D.-Q. Wei, *MLCDForest: multi-label classification with deep forest in disease prediction for long non-coding RNAs*, Brief. Bioinf. 22(3) (2020).
[28] R. Wang, S. Kwong, X. Wang and Y. Jia, *Active k-labelsets ensemble for multi-label classification*, Pattern Recogn. 109 (2021) 107583.
[29] D. Wang, M. Li, N. Ben-Shlomo, C.E. Corrales, Y. Cheng, T. Zhang and J. Jayender, *A novel dual-network architecture for mixed-supervised medical image segmentation*, Comput. Med. Imag. Graph. 89 (2021) 101841.
[30] T. Wu, Q. Huang, Z. Liu, Y. Wang and D. Lin, *Distribution-balanced loss for multi-label classification in long-tailed datasets*, A. Vedaldi, H. Bischof, T. Brox and J.M. Frahm, Computer Vision – ECCV 2020. ECCV 2020, Lecture Notes in Computer Science, Springer, 12349 (2020).
[31] Y. Xia, K. Chen and Y. Yang, *Multi-label classification with weighted classifier selection and stacked ensemble*, Info. Sci. 557 (2021) 421–442.
[32] J. Xu, *An extended one-versus-rest support vector machine for multi-label classification*, Neurocomput. 74(17) (2011) 3114–3124.
[33] M. Yang, W. Zhao, L. Chen, Q. Qu, Z. Zhao and Y. Shen, *Investigating the transferring capability of capsule networks for text classification*, Neural Networks 118 (2019) 247–261.

[34] X.H. Yap and M. Raymer, *Multi-label classification and label dependence in in silico toxicity prediction*, Toxicol. Vitro 74 (2021) 105157.
[35] R. You, Z. Guo, L. Cui, X. Long, Y. Bao and S. Wen, *Cross-modality attention with semantic graph embedding for multi-label classification*, Proc. AAAI Conf. Artificial Intell. (2020).
[36] D. Yun, J. Ryu and J. Lim, *Dual aggregated feature pyramid network for multi label classification*, Pattern Recogn. Lett. 144 (2021) 75–81.
[37] R. Zeleznik, B. Foldyna, P. Eslami, J. Weiss, I. Alexander, J. Taron, C. Parmar, R.M. Alvi, D. Banerji, M. Uno and Y. Kikuchi, *Deep convolutional neural networks to predict cardiovascular risk from computed tomography*, Nature Commun. 12(1) (2021) 1–9.
[38] Y. Zhang, Y. Wang, X.-Y. Liu, S. Mi and M.-L. Zhang, *Large-scale multi-label classification using unknown streaming images*, Pattern Recogn.99 (2020) 107100.
[39] W. Zhao, F. Chen, H. Huang, D. Li and W. Cheng, *A new steel defect detection algorithm based on deep learning*, Comput. Intell. Neurosci. (2021).
[40] X. Zheng, W. Zhu, C. Tang and M. Wang, *Gene selection for microarray data classification via adaptive hypergraph embedded dictionary learning*, Gene 706 (2019) 188–200.
[41] Y. Zhong, B. Du and C. Xu, *Learning to reweight examples in multi-label classification*, Neural Networks 142 (2021) 428–436.