Current Applied Science and Technology Vol. 23 No. 1 (January-February 2023)

Research article

Improving Multi-label Classification Using Feature Reconstruction Methods

Worawith Sangkatip[1] and Phatthanaphong Chomphuwiset[2]*

[1]Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand
[2]Polar Lab, Department of Computer Science, Faculty of Informatics, Mahasarakham University, Mahasarakham, Thailand

Received: 8 December 2021, Revised: 17 March 2022, Accepted: 28 June 2022

DOI: 10.55003/cast.2022.01.23.013

Abstract

Keywords

multi-label classification;

multi-label feature transformation;

feature engineering;

high density data;

feature reconstruction

Multi-label classification (MLC) is a supervised classification method that allows for a data instance with more than one class label (or target). Solving MLC is still a challenging task. MLC can potentially generate complex decision boundaries as the method is a non-mutual exclusive classification method. Recently, many techniques have been proposed to cope with the complexity of MLC problems, such as the Problem transform method (PTM), the Adaptation method (AM), and the Ensemble method (EM). These techniques can generally produce good results with certain datasets. However, they have poor classification performance when the number of possible class-labels is larger, even if the dataset is well-presented (high density). The aim of this work was to solve the MLC problems by performing a feature reconstruction process on the original data features. The proposed feature reconstruction method generates a set of compact features from the original data instances. AutoEncoder is deployed to learn and encode the features of the data (as the constructed feature steps) before they are classified by learning algorithms (or classifiers). We conducted experiments using different multi-label classifiers based on and around PTM, AM, and EM, on the set of the standard dataset. The results from the experiments demonstrated that the proposed feature reconstruction technique provides promising classification results, especially with high-density data.

*Corresponding author: E-mail: Phatthanaphong.c@msu.ac.th

1. Introduction

Multi-label classification (MLC) is a supervised classification method that essentially takes input instances and classifies them into a set of target values (labels) simultaneously [1, 2]. In general, the search space of the MLC problem is large compared with that of multi-class classification (MCC) and grows exponentially when the number of possible labels increases [3]. In addition, MLC is a non-mutually exclusive classifier. Therefore, MLC can produce complex decision boundaries. In addition, the number of data instances used in training processes can affect the performance of the classification. Inadequate data instances, compared to the number of class labels, can produce poor classification results [4]. MLC problems can be solved by transforming the problems into a set of single multi-class classifications. This transformation approach, which has been applied and is applicable to various MLC problems, is known as the problem transformation method (PTM) [5, 6]. MLC is converted to an n-class problem, where n is the number of the class labels extracted from the set of the multi-class labels. In addition to PTM, the Adaptive Method (AM), which involves applying the available classification technique (for multi-class problems), has also been implemented to solve various MLC problems.

Over the past decades, aplications of multi-label learning for solving MLC problems has gained more attention in the research [7]. Initially, Tsoumakas and Katakis [8] compiled and summarized the solutions of MLC into two categories, i.e. (i) adaptation method and (ii) problem transformation method. Madjarov *et al.* [9] and Sangkatip and Phuboon-Ob [10] presented an expanded experiment to compare the performance of different types of classification algorithms for MLC. Their study derived and experimented with three classification-based algorithm groups: PTM, AM, and Ensemble Methods (EM). In the PTM, the Binary Relevance (BR) method and the Label Lower Powerset (LP) method were implemented, which transformed the MLC problem into basis problem subsets of binary-classification problems. Then, an aggregation strategy was applied to obtain the final label set. In the AM, Decision Tree algorithms were applied to carry out the classification of multi-label data [11]. The C4.5 algorithm was one of the common algorithms deployed and was known as ML-C4.5 [12]. The K-Nearest Neighbors algorithm was also applied to MLC problems. The technique considers a set of neighbor data instances to derive the actual label set of a given data instance. This technique is known as ML-*k*NN [13]. Apart from those examples, Neural Network-based methods that have been used effectively to compile the MLC problem have also been reported in the literature [14]. In the EM, MLC was broken down into smaller problems. Then, each sub-problem was handled separately before they were reassembled to produce the final classification results using, for instance, voting schemes [15-17].

Several past studies have attempted to improve the efficiency of MLC by reducing the size of the data instances. The LIFT Method was introduced. The LIFT used a *k*-means clustering algorithm to group the positive and negative instances of each label in the data [18]. Then, the characteristics of the data were extracted through the distance measurement between the data instances and the cluster centers of each label. Subsequently, the relationship between the labels was established by creating additional attributes of the data [19]. Huang *et al.* [20] proposed a technique to learn the dispersion of label attributes, including common attributes. They applied double-label correlation to differentiate labels for each category. Multi-label classifiers are built on low-dimensional visualizations with the learned attributes. From that perspective, recent research into MLC has gained more attention in developing a future engineering method to improve the data features, assisting in the classification processes [21, 22]. Feature engineering can be divided into several categories, including feature transformation, feature generation, feature selection, and feature reconstruction [21]. Deep learning approaches have also been active in recent years. Feature extraction and generation are some of the applicable techniques that have been implemented to improve the quality of data features. Using Convolutional Neural Networks (CNNs) for feature extraction and generation, CNNs map the input data space to another data representation based on training data instances [23]. Dimensionality reduction is one of the techniques used to transform data features. There are two categories of dimensionality reduction methods. One is feature selection (FS) and the other is feature transformation (FT). Feature selection keeps only useful features and dismisses others while feature transformation constructs a new but smaller number of features out of the original ones [24]. One current FT method can be applied by implementing, for example, deep learning algorithms [25], and unsupervised network algorithms, which learn to encode data to extract the relationships of the data. Cheng *et al.* [26] used a deep learning technique to build and extract relationships between attributes and labels in a multi-label classification. Feature reconstruction, as a transformation process, can be considered a tool to generate a set of new feature sets (based on the original data features). The reconstructed features are anticipated to be compact and descriptive, which can be used in the classification process. This work applies the AutoEncoder approach to learn insight into the data features and construct more meaningful active features.

2. Materials and Methods

2.1 Multi-label classification

The task of MLC can be viewed as an instantiating of the structure output prediction paradigm. The goal is to define a set of labels for each data instance. Let X be a space of data instances comprising n data instances $\boldsymbol{x}$, i.e. $\forall \boldsymbol{x} \in X, \boldsymbol{x} = \{x_1, \dots x_d\}$ (where d is the number of instance features) a set of d-dimensional features divided from x , and a set p a possible label space $Y = \{\boldsymbol{y_1}, \dots \boldsymbol{y_p}\}$, i.e. $\boldsymbol{y} = \{y_1, .. y_m\}$ where $y = \{0,1\}$ and m denotes the dimension of the labels $\boldsymbol{y}$ associated with $\mathbf{x}$, as demonstrated in Table 1.

Table 1. Representation of data instances

	$\boldsymbol{X}$				$\boldsymbol{Y}$
$\boldsymbol{x}_1$	x_{11}	x_{12}	$\cdots$	x_{1d}	$\boldsymbol{y}_1 = \{y_{11}, \dots, y_{1m}\}$
$\boldsymbol{x}_2$	x_{21}	x_{22}	$\cdots$	x_{2d}	$\boldsymbol{y}_2 = \{y_{21}, \dots, y_{2m}\}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\cdots$	$\vdots$
$\boldsymbol{x}_n$	x_{n1}	x_{n2}	$\vdots$	x_{nd}	$\boldsymbol{y}_n = \{y_{n1}, \dots, y_{nm}\}$

We denote the quantity L as a loss value for learning models. Therefore, MLC was aimed at finding h such that:

$$\min_{L} h: X \to 2^{Y} \qquad (1)$$

The MLC methods are separated into two categories: problem transformation and algorithm adaptation. Problem transformation methods approach the problem of MLC by transforming the multi-label dataset into one or multiple datasets. These datasets are then approached with simpler, single-target machine learning methods and build into one or multiple single-target models. At prediction time, it is required that all built models are invoked to generate the prediction for the test example. Algorithm adaptation methods include some adaptation of the training and prediction phases of the single target methods in order to handle multiple labels simultaneously. For example, trees change the heuristic used when creating the splits, and Support Vector Machines (SVMs) employ additional threshold techniques. The adaptations provide a mechanism to handle the dependency between the labels directly. Their grouping is based on the underlying paradigm being adapted. The literature recognizes five defined groups of algorithm adaptation methods according to the performed adaptation: trees, neural networks, support vector machines, instance-based and probabilistic. There are additional methods that utilize various approaches from other domains, e.g., genetic programming, but they lack a common ground to unite them and are classified as unspecified methods.

2.1.1 Transformation-based classifiers

A transformation-based classifier (TBC) transforms an MLC into a simpler classification problem, which can be potentially solved by single-label multi-class classification. The classification essentially provides possible values for the transformed class-labels, which are the set of distinct unique subsets of the label in the original data instance [27]. A number of techniques were proposed. Label Powerset generally generates a new set of single class labels. Given a data instance x with a corresponding label $\boldsymbol{y} = \{1,0,0,1,1\}$ in the original MLC problem, the Label Powerset will generally transform the data instance into a new label $y_{1,4,5}$ which can deliberately be used with available multi-class classifier techniques. An example of the problem transformation results is demonstrated in Table 2 (b).

In addition to Label Powerset, the Binary Relevance (BR) method is one of TBC methods that are commonly used to solve problems. BR breaks down a MLC problem into distinct single-label binary classification problems, one for each of the m labels in the set $\boldsymbol{y} = \{y_1, \ldots, y_m\}$ [28]. In the learning process, the original multi-label training dataset is transformed to m datasets, and each of them is associated with a binary class-label obtained from the original y. After the multi-label data has been transformed, a set of q binary classifiers $H_j(x), j = 1..m$ is constructed using the new m training dataset. The BR generates a set of m classifiers as follows:

$$H = \{M_{y_j}(\boldsymbol{x}, \boldsymbol{y}_j) \rightarrow y' \in \{0,1\} | y_j \in y, j = 1, \ldots, m\} \tag{2}$$

where M denotes a set of train models (classifiers) and y' designates predicted label set.

Tabel 2. Example of Label Powerset multi-label transformation. Multi-label data instances (a) are transformed to a multi-class classification problem (b). The transformed problem becomes 3-class problem, i.e. $y = \{y_1, y_{1,2}, y_{1,4,5}\}$.

X	Y
$\boldsymbol{x}_1$	$\boldsymbol{y}_1 = \{1,0,0,1,1\}$
$\boldsymbol{x}_2$	$\boldsymbol{y}_2 = \{1,0,0,1,1\}$
$\boldsymbol{x}_3$	$\boldsymbol{y}_3 = \{1,0,0,0,0\}$
$\boldsymbol{x}_4$	$\boldsymbol{y}_4 = \{1,1,0,0,0\}$
$\boldsymbol{x}_5$	$\boldsymbol{y}_5 = \{1,0,0,0,0\}$

(a) MLC

X	Y
$\boldsymbol{x}_1$	$\boldsymbol{y}_1 = \{y_{1,4,5}\}$
$\boldsymbol{x}_2$	$\boldsymbol{y}_2 = \{y_{1,4,5}\}$
$\boldsymbol{x}_3$	$\boldsymbol{y}_3 = \{y_1\}$
$\boldsymbol{x}_4$	$\boldsymbol{y}_4 = \{y_{1,2}\}$
$\boldsymbol{x}_5$	$\boldsymbol{y}_5 = \{y_1\}$

(b) TBC

2.1.2 Adaptation-based classifiers

Multi-label k Nearest Neighbor (ML-kNN) is introduced by Zhang and Zhou [13]. ML-kNN determines a label set of an instance (x) of unknown label $(y(x) \subseteq y)$ by utilizing the Maximum A Posteriori (MAP) method to predict the label set of x. Given an unknown label set x, the ML-kNN examines k neighbors of x (based-on a distance metric) and counts the number of neighbor (s) belonging to each class y_i.

$$P(y_i|s) = \frac{P(s|y_i)P(y_i)}{P(s)}. \tag{3}$$

For each label y_i, ML-kNN generates a h_i classifier to predict the final label set:

$$h_i = \begin{cases} 1 & P(y_i = 1|s) > P(y_i = 0|s) \\ 0 & otherwise. \end{cases} \tag{4}$$

Support Vector Machine (SVM) has also been applied in Adaptation-based classifiers (ABC) to solve MLC [29]. Conventionally, SVM is introduced to cope with binary classification problems. However, in multi-label classification, a ranking version SVM was proposed (RANK-SVM) [30].

2.1.3 Ensemble-based classifiers

Ensemble-based classifier (EBC) transform an MLC problem into a set of smaller problems p (ensemble on subset of the problems) [8, 31, 32]. Each of the problems is solved separately as a subset classifier. The results of the subset classifiers are aggregated (assembled) to produce a final decision of the classification. Random k-Labelsets (RAkEL) are one of EBCs [8, 33]. The technique builds a random subset of the original labels to learn a single-label classifier (binary) for the prediction of each element in the powerset of the subset. To illustrate the basis of the basic idea of the RAkEL, consider Table 3, which shows four random subsets ($M_j, j = 1, \dots, 2^k$ and k is a size of feature subset obtained from y) of the MLC problems, for $k = 2$. For each subset problem, k binary classifications are performed. Then, the final decision is aggregated by a voting mechanism.

Table 3. Example of problems subsets in RAkEL technique (k=2) applied to data presented in Table.1 The final decision is made by thresholding (using a pre-determined value, e.g. $\tau = 0.5$) the average number of votes (AV_i) of each label dimension (y_i). The prediction for $y_i$is 1 when $AV_i > \tau$ and 0 otherwise.

Model	y_1	y_2	y_3	y_4	y_5
M_1 (1,5)	1	-	-	-	1
M_2 (2,4)	-	0	-	1	-
M_3 (2,3)	-	1	1	-	-
M_4 (4,5)	-	-	-	1	0
avg-votes	1/1	1/2	1/1	2/2	1/2
prediction	1	0	1	1	0

2.2 Methods

Solving multi-label classification is essentially a challenging task. Many methods (proposed to solve MLC problems) usually involve a design of algorithms that cope with the problem and yet produce promising classification results. This work focuses on the feature engineer-based method, where the features of data instances are explored and transformed into a compact form used in a subsequent classification process. Therefore, in this section, we provide details of the datasets used in this study and the proposed method.

2.2.1 Dataset

The Multi-label datasets (MLD) used in this work was collected from the Mulan datasets website [34]. There are 8 standard datasets comprising different data domains, as demonstrated in Table 4. The number of feature dimensions (d) in the dataset is varied. In addition, each dataset is associated with a different number of class labels. For example, the yeast dataset has 2417 data instances (the biggest dataset) with 103 feature dimensions and 14 labels. The Cardinality of the dataset denotes the variation of each class labels in the dataset. The Density of the dataset explains the variation of the class labels with respect to the number of labels in the dataset.

Table 4. Multi-label datasets

Datasets	Domain	Instances	Features	Labels	Cardinality	Density
birds	audio	645	260	19	1.014	0.053
enron	text	1702	1001	53	3.378	0.064
emotions	music	593	72	6	1.869	0.311
medical	text	978	1449	45	1.245	0.028
yeast	biology	2417	103	14	4.237	0.303
scene	image	2407	294	6	1.074	0.179
cal500	music	502	68	174	26.044	0.15
foodtruck	recommend	407	21	12	2.29	0.191

2.2.2 Feature reconstruction using AutoEncoder

The work proposes a technique that transforms a set of features (Feature Transform: FT) of given data instances into a compacted feature space (X'). To achieve this, we introduce a transformation function $t: \boldsymbol{x} \rightarrow \boldsymbol{x}', \boldsymbol{x}' = \{x'_1, \ldots, x'_k\}$ where k is a dimension of the transformed features and $k \ll d$. We adopt an AutoEncoder technique as the transformation function (t) that encodes the input data features. The proposed technique compresses the input data instances with an encoder module using an AutoEncoder technique (EN) [35]. In addition, we introduce an extension mechanism that incooperates the Target-label into the AutoENcoder network (TEN) during the network training process. The extension process can potentially compact the original features and maintain the context of the transformed features with respect to the original data labels. Two processes are carried out to classify the data instances using the proposed feature reconstruction method, i.e., feature reconstruction and multi-label classification, as illustrated in Figure 1.

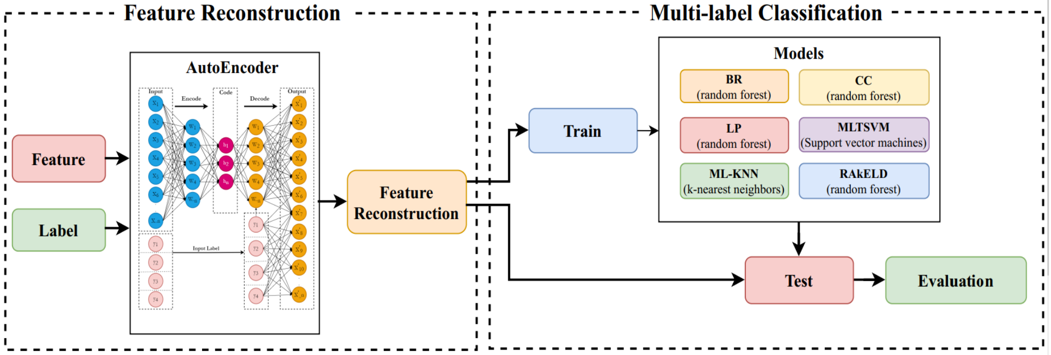
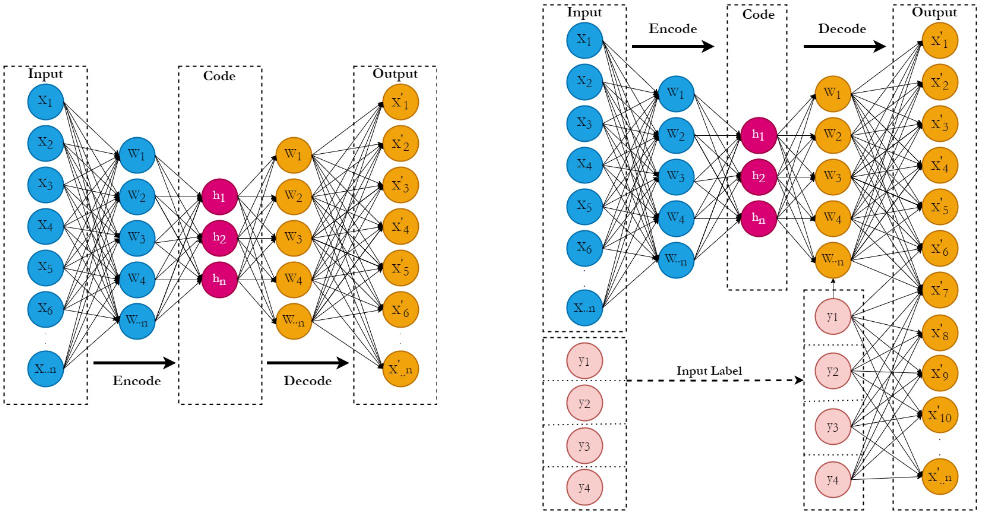


Figure 1. The overall process of feature reconstruction for solving MLC

1) Feature reconstruction

AutoEncoder technique is applied in this work to encode the input data instances as the main procedure for compacting the original features of the data ($\boldsymbol{x}$). AutoEncoder is a Neural Network (NN) that can be used to learn and derive the representation of data. The network is broken down into two main modules, i.e., the encoder and decoder module, as illustrated in Figuer 2(a).



(a) AutoEncoder for EN

(b) AutoEncoder for TEN

Figure 2. AutoEncoder architecture(EN) and TEN

The encoder module encodes the input data instance, while the decoder attempts to decode the encoded data. During the training process, the decoder actively tries to decode the encoded data to be identical to the original data representation (data features). This process can be performed through an optimization approach, aiming to minimize a certain criterion. In this work, we denote L as a loss that measures the difference between the input instance (x) and the decoded data (x'). To obtain both solid encoder and decoder module, we define a loss function as follows:

$$\min_{W,W\prime,b,b\prime} L(\boldsymbol{x},\boldsymbol{x}') = ||x - \sigma(W'(\sigma(W\boldsymbol{x}+b))+b')||^2 \tag{5}$$

This loss function is minimized with respect to the network parameters (W, W', b, b') and σ denotes activation functions where $W$and W' are the network weights and b' is the network bias [33]. After the training process, the encoder module will be used to reconstruct compact features for the subsequent classification procedure.

The constructed features (using the encoder module from the trained decoder) can not potentially be applicable to represent the data features. Therefore in this work, we integrate the class labels (**y**) as a set of the augmented node to the input data instances, and we define this as the TEN method. Then, the output layer (associated with the decoder module) is compiled to generate $|x| +$

$|y|$ output nodes, as illustrated in Figure 2(b). The optimization can subsequently be performed using the loss function below:

$$\min_{W,W\prime,b,b\prime} L(\boldsymbol{x},\boldsymbol{x}') = ||\boldsymbol{x}^\frown \boldsymbol{y} - \sigma(W'(\sigma(W\boldsymbol{x}+b))+b')||^2 \qquad (6)$$

$\boldsymbol{x}^\frown \boldsymbol{y}$ denotes the concatenated vectors between an instance $\boldsymbol{x}$ and its associated label $\boldsymbol{y}$. To generate a discriminative feature from the network (W,b) , we reconstruc the features using a reconstruction, $\tau(.)$, from the input original feature (x) as follows:

$$\tau(\boldsymbol{x}) = \sigma(W\boldsymbol{x}+b). \qquad (7)$$

2) Multi-label classification

The previous section provides the details of the feature reconstruction used in this research work. The reconstructed features $(\tau(\boldsymbol{x}))$ are then fed into the classification process $f: \tau(\boldsymbol{x}) \rightarrow \boldsymbol{y}'$, where $f(.)$ is a mapping function or a classifier. This work uses various classification techniques to classify the original data $\boldsymbol{x}$ and the reconstructed $\tau(\boldsymbol{x})$. The details of the classification settings and experiments will be explained in the next section.

2.2.3 Experiment setup and evaluation metrics

The previous section explained the proposed method for constructing a new feature subset. The input data instances are fed into the encoder module (EN and TEN) to generate compact features. Then, the classification is carried out. This section provides the details of the experiment conducted to evaluate the performance of the proposed method.

To evaluate the performance of the proposed feature construct method, we used six multi-label classification methods to classify datasets through instance transformations. These classification techniques were used to examine the effectiveness of the proposed method when experimenting with various common MLC classification techniques, i.e., PTM, Adaptation method, and Ensemble technique. Binary relevance (BR) and Classifier Chains (CC) [36] were used in the experiments. These two classification techniques are based around the problem transformation method. In addition to the PTM, the Label Powerset (LP) was also implemented in this work as the technique is the fundamental method used for MLC problems. The Adaptation method, i.e. MLTSVM [37] and ML-kNN, were also utilized. Finally, the disjoint RAkEL (RAkELd) method [38], where the subsets of labels are non-overlapping, and which is an Ensemble-based technique for MLC, was used in the experiments.

For each dataset, the dataset was divided into training and test sets. The training data was used to train the AutoEncoder. We separated 60% of the data instances from the dataset to construct the training process. The other 40% of the data instances were used to test the performance of the classification performance (by all six classifiers).

In the experiment, we utilized Scikit-multilearn as a primary tool to conduct various experiments [39]. We chose ten common evaluation metrics for MLC [40]. These evaluation metrics covered both example-based metrics and label-based metrics, namely Precision, Recall, F1, Macro Precision, Macro Recall, Macro F1, Micro Precision, Micro Recall, Micro F1, and Hamming Loss. For the sake of representation simplicity, these evaluation metrics were denoted as Precision, Recall, F1, Macro P, Macro R, Macro F1, Micro P, Micro R, Micro F1, and H Loss. Precision and Recall are defined as the average proportion between the number of correctly predicted labels. The measurement metrics are defined in equations 8-17.

For each classifier, true positives (tp_j), true negatives (tn_j), false positives (fp_j), and false negatives (fn_j) obtained (based on the metrics) are calculated for each label $y: j = 1 \ldots m$,. Macro F_1 is essentially the harmonic mean obtained from Precision and Recall based on an average of each label y_j, and an average over all labels. In addition, Micro F_1 is the harmonic mean of Micro derived from Precision and Micro Recall in the above definition.

$$\text{Precision} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Y'_i|}{|Y'_i|} \tag{8}$$

$$\text{Recall} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Y'_i|}{|Y_i|} \tag{9}$$

$$\text{F1} = \frac{1}{n}\sum_{i=1}^{n}\frac{|Y_i \cap Y'_i|}{|Y_i| + |Y'_i|} \tag{10}$$

$$\text{Hamming Loss} = \frac{1}{|N| \cdot |L|}\sum_{i=1}^{|N|}\sum_{j=1}^{|L|} xor(y_{ij}, y'_{ij}) \tag{11}$$

$$\text{Micro Precision } (MiP) = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fp_j} \tag{12}$$

$$\text{Micro Recall (MiR)} = \frac{\sum_{j=1}^{m} tp_j}{\sum_{j=1}^{m} tp_j + \sum_{j=1}^{m} fn_j} \tag{13}$$

$$\text{Micro F1} = \frac{2 \times MiR \times MiP}{MiR + Mip} \tag{14}$$

$$\text{Macro Precision} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fp_j} \tag{15}$$

$$\text{Macro Recall} = \frac{1}{m}\sum_{j=1}^{m}\frac{tp_j}{tp_j + fn_j} \tag{16}$$

$$\text{Macro F1} = \frac{1}{m}\sum_{j=1}^{m}\frac{2 \times R_j \times P_j}{R_j + P_j} \tag{17}$$

3. Results and Discussion

After training the AutoEncoder, we conducted two separate experiments. The first experiment was aimed to examine the efficiency of the feature construction of the two techniques, i.e., EN and TEN.

In addition, we experimented on each dataset separately (eight datasets). The experimental results are listed in Tables 5-12 which demonstrate the results from the experiments carried out on the eight different datasets using EN and TEN for the feature reconstruction. From the experimental results, we can observe that the construction method TEN outperforms EN for almost all of the datasets for all measurement metrics. The TEN results were better or the same outcomes for all datasets and classifiers. Consider the Yeast and Emotions datasets (Tables 5 and 6); TEN produced better results than EN for all different classifiers and evaluation matrices. Based on the data description (shown in Table 4), the Yeast and Emotions datasets were the only two datasets with a high-density value (>0.3). The density practically measures the dispersion of the data. With the MLC dataset, the density signifies the distribution of the data labels. High density accounts for low label dispersion, well presented. Compared to other datasets, e.g., the Birds and Medical dataset, EN and TEN provide marginally the same results for some classifiers. Using TEN for the Yeast dataset, the best performance (measured by $F1$) was 78.0%, obtained from the BR technique. And, the best performance resulting from the Emotions dataset using TEN was 62.0%.

To explore the sensitivity of the proposed technique, we did an experiment using TEN as the feature reconstruction method and compared it to the Native (original) data features. We experimented using the same set of six classifiers with Yeast and Emotion datasets, and the results are demonstrated in Tables 13-14 which present the performance of the proposed TEN to construct a new feature set and the results of the classification obtained from the Native data features were compared. The Yeast and Emotions datasets were the only two datasets used with a high-density value (>0.3) (shown in Table 4). Density practically measures the dispersion of the data. With the MLC dataset, the density signifies the distribution of the data labels. High density accounts for low label dispersion. It can beobserved that the proposed TEN was superior to the Native data features when they were classified by the six MLC techniques (p=0.0001). Therefore, TEN tended to work well with the high-density dataset (well-presented data) for MLC problems. In addition, the visual representation of the performance comparison is illustrated in Figures 3 and 4.

Figure 3 shows the results of the classification of the proposed technique (TEN) and the Native data features when the size of the reconstructed was varied from m' =10 to m' =100. It can be observed that TEN gave better results than the Native features, even if the dimensions of the reconstructed feature were small (m'=10). Figure 4 depicts a comparative representation of different evaluation metrics.

4. Conclusions

In this study, we proposed a technique for improving the performance of multi-label classification (MLC) with a feature reconstruction method. In the proposed feature reconstruction, we applied the AutoEncoder technique that intentionally encoded the input data instance to generate a compact feature representation of them. We implemented two of the construction procedures. AutoEncoder alone (EN) was built to encode the feature subsets of the data instances. AutoEncoder with Target class (TEN) was constructed to derive a compact set of data instances and maintain the contextual insights of the dataset, conveying the class-label representation. To evaluate the performance of the proposed method (TEN), we collected 8-standard datasets, which were acquired from different domains and different data settings. We conducted the experiments by applying six classifiers, which were derived from three different MLC techniques (PTM, AM, and EM). The experiments were separated into two folds. The first experiment explored the effectiveness of the TEN and EN in the feature reconstruction process. In comparison, the second experiment was constructed to measure the proposed technique’s performance (TEN) compared with the original data feature used in MLC.

Table 5. Comparative results for Yeast dataset

Metric	BR		CC		LP		MLTSVM		ML-*k*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.69±0.03	**0.85±0.03**	0.66±0.02	**0.83±0.02**	0.61±0.02	**0.79±0.03**	0.63±0.02	**0.83±0.03**	0.59±0.02	**0.78±0.03**	0.66±0.02	**0.82±0.02**
Recall↑	0.52±0.01	**0.76±0.00**	0.52±0.01	**0.76±0.02**	0.56±0.02	**0.74±0.01**	0.59±0.02	**0.77±0.01**	0.51±0.01	**0.75±0.01**	0.56±0.02	**0.76±0.00**
F1↑	0.56±0.02	**0.78±0.01**	0.55±0.01	**0.77±0.02**	0.56±0.02	**0.74±0.02**	0.59±0.02	**0.78±0.02**	0.52±0.01	**0.74±0.01**	0.58±0.02	**0.77±0.01**
Macro P↑	0.42±0.03	**0.75±0.04**	0.39±0.01	**0.73±0.04**	0.40±0.02	**0.66±0.03**	0.34±0.02	**0.71±0.02**	0.37±0.01	**0.65±0.04**	0.42±0.06	**0.72±0.04**
Macro R↑	0.29±0.00	**0.54±0.01**	0.30±0.01	**0.53±0.02**	0.35±0.01	**0.54±0.01**	0.34±0.01	**0.54±0.01**	0.32±0.00	**0.59±0.02**	0.33±0.01	**0.54±0.01**
Macro F1↑	0.31±0.01	**0.59±0.01**	0.31±0.00	**0.58±0.02**	0.35±0.01	**0.57±0.01**	0.32±0.01	**0.58±0.01**	0.34±0.01	**0.61±0.02**	0.33±0.02	**0.58±0.01**
Micro P↑	0.70±0.03	**0.87±0.03**	0.67±0.03	**0.85±0.02**	0.62±0.02	**0.79±0.02**	0.63±0.03	**0.84±0.03**	0.59±0.02	**0.78±0.03**	0.66±0.02	**0.83±0.02**
Micro R↑	0.52±0.01	**0.75±0.01**	0.52±0.01	**0.74±0.02**	0.56±0.02	**0.73±0.01**	0.58±0.02	**0.75±0.01**	0.51±0.01	**0.74±0.01**	0.56±0.01	**0.75±0.01**
Micro F1↑	0.59±0.02	**0.80±0.01**	0.58±0.01	**0.79±0.01**	0.58±0.02	**0.76±0.01**	0.61±0.02	**0.79±0.01**	0.55±0.01	**0.76±0.01**	0.60±0.02	**0.79±0.01**
H Loss↓	0.21±0.01	**0.11±0.00**	0.22±0.00	**0.12±0.01**	0.24±0.01	**0.14±0.00**	0.23±0.01	**0.12±0.01**	0.25±0.01	**0.14±0.01**	0.22±0.01	**0.12±0.00**

Table 6. Comparative results for Emotions dataset

Metric	BR		CC		LP		MLTSVM		ML-*k*NN		RAkELd	
	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN	EN	TEN
Precision↑	0.36±0.07	**0.56±0.01**	0.38±0.05	**0.60±0.05**	0.40±0.05	**0.57±0.03**	0.36±0.06	**0.60±0.04**	0.32±0.05	**0.63±0.03**	0.39±0.03	**0.55±0.04**
Recall↑	0.31±0.04	**0.52±0.05**	0.35±0.06	**0.57±0.05**	0.42±0.04	**0.56±0.06**	0.47±0.08	**0.68±0.08**	0.27±0.08	**0.63±0.07**	0.38±0.08	**0.56±0.04**
F1↑	0.31±0.04	**0.51±0.02**	0.34±0.04	**0.55±0.01**	0.39±0.03	**0.54±0.03**	0.39±0.05	**0.62±0.04**	0.27±0.06	**0.60±0.04**	0.36±0.05	**0.53±0.03**
Macro P↑	0.40±0.09	**0.62±0.02**	0.36±0.08	**0.62±0.04**	0.40±0.04	**0.57±0.05**	0.15±0.03	**0.59±0.04**	0.32±0.07	**0.63±0.03**	0.39±0.06	**0.60±0.04**
Macro R↑	0.29±0.04	**0.51±0.06**	0.33±0.07	**0.58±0.07**	0.39±0.04	**0.58±0.08**	0.43±0.08	**0.70±0.07**	0.25±0.08	**0.62±0.08**	0.36±0.07	**0.57±0.05**
Macro F1↑	0.32±0.05	**0.54±0.03**	0.33±0.07	**0.57±0.03**	0.39±0.04	**0.55±0.03**	0.22±0.04	**0.63±0.03**	0.27±0.07	**0.62±0.05**	0.35±0.06	**0.57±0.01**
Micro P↑	0.42±0.07	**0.64±0.03**	0.38±0.05	**0.65±0.04**	0.40±0.05	**0.57±0.03**	0.36±0.06	**0.60±0.04**	0.35±0.05	**0.65±0.02**	0.39±0.04	**0.60±0.03**
Micro R↑	0.31±0.04	**0.53±0.05**	0.36±0.05	**0.58±0.06**	0.42±0.03	**0.58±0.07**	0.47±0.10	**0.70±0.08**	0.28±0.08	**0.64±0.07**	0.37±0.07	**0.58±0.04**
Micro F1↑	0.36±0.04	**0.58±0.02**	0.37±0.05	**0.61±0.02**	0.41±0.03	**0.57±0.03**	0.40±0.06	**0.64±0.04**	0.31±0.07	**0.64±0.04**	0.38±0.05	**0.59±0.02**
H Loss↓	0.37±0.03	**0.25±0.02**	0.41±0.03	**0.24±0.02**	0.40±0.02	**0.28±0.02**	0.46±0.04	**0.25±0.02**	0.40±0.04	**0.24±0.04**	0.40±0.03	**0.26±0.02**

[1] ↑ indicates the higher value, the better, and ↓ the lower, the better

Table 7. Comparative results for Scene dataset

Metric	BR		CC		LP		MLTSVM		ML-*k*NN		RAkELd	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.29±0.18	**0.31±0.20**	0.29±0.18	**0.31±0.20**	0.36±0.21	**0.38±0.23**	0.41±0.27	**0.42±0.28**	0.36±0.21	**0.37±0.22**	0.32±0.19	0.32±0.18
Recall↑	0.28±0.18	**0.30±0.20**	0.27±0.18	**0.29±0.20**	0.33±0.21	**0.35±0.22**	0.38±0.26	**0.39±0.27**	0.37±0.22	0.37±0.24	0.30±0.19	0.30±0.18
F1↑	0.28±0.18	**0.30±0.20**	0.28±0.18	**0.30±0.20**	0.34±0.21	**0.36±0.23**	0.39±0.26	**0.40±0.27**	0.36±0.21	0.36±0.22	0.30±0.19	**0.31±0.18**
Macro P↑	0.29±0.08	**0.30±0.09**	0.29±0.08	**0.30±0.11**	0.26±0.09	**0.28±0.09**	**0.28±0.10**	0.25±0.11	0.24±0.10	**0.25±0.10**	0.27±0.10	0.27±0.10
Macro R↑	0.22±0.08	0.22±0.07	0.21±0.07	**0.23±0.07**	0.27±0.11	**0.28±0.09**	0.30±0.12	0.30±0.12	0.28±0.09	0.28±0.09	0.25±0.09	0.25±0.09
Macro F1↑	0.20±0.07	**0.21±0.07**	0.20±0.06	**0.21±0.08**	0.21±0.06	**0.23±0.08**	0.24±0.09	0.24±0.09	0.22±0.07	**0.23±0.08**	0.21±0.06	0.21±0.07
Micro P↑	0.47±0.25	**0.49±0.25**	0.46±0.25	**0.48±0.26**	0.36±0.21	**0.38±0.23**	0.41±0.27	**0.42±0.28**	0.39±0.21	0.39±0.23	**0.43±0.23**	0.41±0.25
Micro R↑	0.28±0.18	**0.30±0.20**	0.28±0.17	**0.29±0.19**	0.33±0.19	**0.35±0.21**	0.38±0.24	**0.39±0.26**	0.38±0.22	0.38±0.23	0.30±0.18	0.31±0.18
Micro F1↑	0.35±0.21	0.37±0.22	0.34±0.20	**0.36±0.22**	0.35±0.20	**0.37±0.22**	0.39±0.26	**0.40±0.27**	0.38±0.21	0.38±0.23	0.35±0.20	0.35±0.21
H Loss↓	0.18±0.05	0.18±0.05	0.19±0.05	**0.18±0.06**	0.23±0.07	**0.22±0.08**	0.21±0.09	0.21±0.09	0.22±0.07	0.22±0.08	**0.20±0.06**	0.21±0.07

Table 8. Comparative results for Medical dataset

Metric	BR		CC		LP		MLTSVM		ML-*k*NN		RAkELd	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.24±0.06	**0.28±0.04**	0.27±0.05	**0.28±0.04**	**0.58±0.08**	0.56±0.08	0.38±0.06	0.38±0.07	0.43±0.05	**0.45±0.05**	0.25±0.03	**0.27±0.04**
Recall↑	0.21±0.06	**0.23±0.04**	0.23±0.05	0.23±0.04	0.52±0.07	0.52±0.07	0.31±0.04	0.31±0.05	0.41±0.04	**0.42±0.05**	0.22±0.03	0.22±0.04
F1↑	0.22±0.06	**0.24±0.04**	0.24±0.05	0.24±0.04	**0.54±0.07**	0.53±0.07	0.33±0.05	0.33±0.06	0.41±0.04	**0.43±0.05**	0.23±0.03	**0.24±0.04**
Macro P↑	0.08±0.03	**0.09±0.02**	0.09±0.03	0.09±0.02	**0.13±0.02**	0.10±0.01	0.03±0.01	0.03±0.01	**0.09±0.02**	0.08±0.01	**0.09±0.02**	0.08±0.02
Macro R↑	0.04±0.01	0.04±0.01	0.04±0.01	0.04±0.01	**0.12±0.01**	0.11±0.01	0.04±0.00	0.04±0.00	**0.10±0.02**	0.09±0.01	0.04±0.01	0.04±0.01
Macro F1↑	0.05±0.01	0.05±0.01	0.05±0.01	0.05±0.01	**0.11±0.02**	0.10±0.01	0.03±0.00	0.03±0.00	**0.09±0.02**	0.08±0.01	0.05±0.01	0.05±0.01
Micro P↑	0.79±0.10	**0.83±0.05**	**0.85±0.07**	0.81±0.06	**0.59±0.07**	0.58±0.08	0.38±0.06	0.38±0.07	0.57±0.09	0.57±0.07	**0.81±0.05**	0.78±0.08
Micro R↑	0.21±0.05	**0.24±0.04**	0.23±0.04	**0.24±0.04**	0.51±0.06	0.51±0.07	**0.31±0.04**	0.30±0.05	0.41±0.04	**0.42±0.04**	0.24±0.02	0.24±0.03
Micro F1↑	0.33±0.07	**0.37±0.05**	**0.37±0.05**	0.36±0.04	**0.55±0.06**	0.54±0.07	0.34±0.05	0.34±0.06	0.48±0.05	0.48±0.05	**0.37±0.03**	0.36±0.05
H Loss↓	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.02±0.00	0.03±0.00	0.03±0.00	0.03±0.00	**0.02±0.00**	0.02±0.00	0.02±0.00

[1] ↑ indicates the higher value, the better, and ↓ the lower, the better

Table 9. Comparative results for Cal500 dataset

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.56±0.00	**0.57±0.01**	0.57±0.01	**0.58±0.01**	**0.35±0.01**	0.34±0.01	0.35±0.01	0.35±0.01	0.44±0.02	**0.46±0.01**	0.53±0.01	**0.54±0.01**
Recall↑	0.26±0.01	**0.27±0.01**	0.25±0.02	**0.27±0.00**	0.35±0.01	0.35±0.01	0.35±0.01	0.35±0.01	0.31±0.01	**0.32±0.00**	0.27±0.01	**0.28±0.01**
F1↑	0.35±0.01	**0.36±0.01**	0.34±0.02	**0.35±0.01**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.34±0.01	0.36±0.01	**0.37±0.00**	0.34±0.01	**0.36±0.01**
Macro P↑	0.14±0.01	**0.16±0.02**	0.13±0.01	**0.16±0.02**	**0.17±0.01**	0.16±0.01	0.16±0.01	**0.17±0.01**	0.15±0.02	**0.17±0.00**	0.14±0.01	**0.17±0.01**
Macro R↑	0.08±0.00	0.08±0.01	0.08±0.01	**0.09±0.01**	**0.17±0.01**	0.16±0.01	0.16±0.01	**0.17±0.01**	0.12±0.01	**0.13±0.00**	0.09±0.00	**0.10±0.00**
Macro F1↑	0.09±0.00	**0.10±0.01**	0.08±0.01	**0.10±0.01**	0.16±0.01	0.16±0.01	0.15±0.01	**0.16±0.01**	0.13±0.01	**0.14±0.00**	0.10±0.00	**0.11±0.00**
Micro P↑	0.55±0.01	**0.56±0.01**	0.55±0.01	**0.56±0.02**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.34±0.01	0.44±0.02	**0.45±0.01**	0.52±0.01	**0.53±0.01**
Micro R↑	0.26±0.01	**0.27±0.01**	0.24±0.02	**0.27±0.01**	**0.35±0.01**	0.34±0.01	0.35±0.01	0.35±0.01	0.31±0.01	**0.32±0.00**	0.26±0.01	**0.28±0.01**
Micro F1↑	0.35±0.01	0.36±0.01	0.34±0.02	**0.36±0.01**	**0.35±0.01**	0.34±0.01	0.34±0.01	0.35±0.01	0.36±0.01	**0.38±0.00**	0.35±0.01	**0.37±0.01**
H Loss↓	0.14±0.00	0.14±0.00	0.14±0.00	0.14±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.20±0.00	0.16±0.00	0.16±0.00	0.15±0.00	0.15±0.00

Table 10. Comparative results for Birds dataset

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.09±0.03	0.09±0.03	0.09±0.04	0.09±0.04	0.10±0.04	0.10±0.05	**0.05±0.02**	0.04±0.01	**0.09±0.04**	0.08±0.03	0.10±0.05	**0.11±0.06**
Recall↑	0.07±0.02	0.06±0.02	0.06±0.03	**0.07±0.02**	**0.11±0.03**	0.10±0.03	0.03±0.01	0.02±0.01	0.05±0.01	0.05±0.01	0.08±0.04	0.08±0.04
F1↑	0.07±0.02	0.06±0.02	0.06±0.03	**0.07±0.03**	0.10±0.03	0.10±0.04	0.03±0.02	0.03±0.01	0.06±0.02	0.06±0.02	0.08±0.03	**0.09±0.04**
Macro P↑	**0.15±0.04**	0.12±0.04	0.13±0.04	**0.14±0.05**	0.14±0.03	0.14±0.05	**0.10±0.07**	0.07±0.04	**0.09±0.02**	0.08±0.02	0.12±0.04	**0.16±0.09**
Macro R↑	**0.08±0.03**	0.06±0.01	0.07±0.03	**0.08±0.03**	**0.13±0.04**	0.12±0.04	**0.05±0.03**	0.03±0.01	0.06±0.02	0.06±0.02	0.09±0.03	0.09±0.03
Macro F1↑	**0.09±0.02**	0.07±0.01	0.08±0.03	**0.09±0.03**	0.12±0.03	0.12±0.04	**0.06±0.04**	0.03±0.01	0.07±0.01	0.07±0.01	0.09±0.03	**0.10±0.04**
Micro P↑	0.34±0.08	0.34±0.10	0.30±0.08	**0.36±0.10**	0.23±0.05	**0.24±0.05**	**0.31±0.10**	0.26±0.05	**0.31±0.09**	0.27±0.05	0.30±0.07	**0.31±0.13**
Micro R↑	**0.13±0.05**	0.11±0.05	0.12±0.07	**0.13±0.06**	0.20±0.06	0.20±0.07	**0.07±0.03**	0.05±0.01	0.11±0.03	0.11±0.04	0.15±0.07	0.15±0.06
Micro F1↑	**0.19±0.06**	0.17±0.05	0.16±0.07	**0.19±0.07**	0.21±0.05	0.21±0.06	**0.11±0.04**	0.09±0.02	0.15±0.04	0.15±0.04	0.19±0.07	**0.20±0.08**
H Loss↓	0.06±0.01	0.06±0.01	0.06±0.01	0.06±0.01	0.08±0.01	0.08±0.01	0.06±0.01	0.06±0.01	0.06±0.00	0.07±0.01	0.07±0.01	0.07±0.00

[1] ↑ indicates the higher value, the better, and ↓ the lower, the better

Table 11. Comparative results for Enron dataset

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.66±0.03	**0.68±0.03**	0.66±0.03	**0.68±0.03**	0.54±0.01	**0.56±0.01**	0.52±0.01	0.52±0.01	0.58±0.02	**0.59±0.02**	0.66±0.03	**0.67±0.03**
Recall↑	0.43±0.01	**0.45±0.01**	0.44±0.01	**0.46±0.01**	0.48±0.01	**0.50±0.02**	0.45±0.01	0.45±0.01	**0.46±0.02**	0.45±0.02	**0.46±0.01**	0.45±0.01
F1↑	0.49±0.01	**0.51±0.02**	0.50±0.02	**0.51±0.01**	0.49±0.01	**0.51±0.01**	0.46±0.01	0.46±0.01	0.48±0.02	**0.49±0.02**	**0.52±0.01**	0.51±0.01
Macro P↑	0.20±0.02	**0.21±0.02**	0.20±0.02	0.20±0.02	0.21±0.01	**0.22±0.01**	**0.09±0.01**	0.08±0.01	0.19±0.01	0.19±0.01	0.21±0.02	**0.22±0.02**
Macro R↑	0.10±0.01	0.10±0.01	0.10±0.01	0.10±0.01	0.13±0.01	0.13±0.02	0.08±0.00	0.08±0.00	**0.13±0.01**	0.12±0.01	0.11±0.01	0.11±0.01
Macro F1↑	0.12±0.01	0.12±0.01	0.12±0.01	0.12±0.01	0.14±0.01	**0.15±0.02**	0.07±0.00	0.07±0.00	0.14±0.01	0.14±0.01	0.12±0.01	0.12±0.01
Micro P↑	0.71±0.01	0.71±0.02	0.70±0.02	**0.71±0.02**	0.56±0.01	0.56±0.01	0.54±0.01	0.54±0.01	0.59±0.01	**0.61±0.01**	0.69±0.02	**0.71±0.02**
Micro R↑	0.42±0.01	**0.43±0.01**	0.43±0.01	**0.44±0.01**	0.44±0.01	**0.46±0.02**	0.40±0.01	0.40±0.01	0.44±0.02	0.44±0.02	**0.45±0.01**	0.43±0.02
Micro F1↑	0.53±0.01	**0.54±0.01**	0.53±0.02	**0.54±0.01**	0.49±0.01	**0.50±0.01**	0.46±0.01	0.46±0.01	0.51±0.01	0.51±0.01	0.54±0.01	0.54±0.02
H Loss↓	0.05±0.00	0.05±0.00	0.05±0.00	0.05±0.00	0.06±0.00	0.06±0.00	0.06±0.00	0.06±0.00	0.05±0.00	0.05±0.00	0.05±0.00	0.05±0.00

Table 12. Comparative results for Foodtruck dataset

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	EN	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**	**EN**	**TEN**
Precision↑	0.57±0.08	**0.59±0.11**	0.55±0.10	**0.63±0.10**	0.54±0.12	**0.67±0.12**	0.67±0.14	0.67±0.14	0.45±0.09	**0.52±0.08**	0.59±0.11	**0.63±0.12**
Recall↑	0.44±0.09	0.44±0.05	0.39±0.09	**0.41±0.06**	0.38±0.08	**0.43±0.07**	0.41±0.09	0.41±0.09	0.43±0.04	**0.47±0.05**	0.43±0.08	**0.46±0.08**
F1↑	0.44±0.07	**0.45±0.06**	0.40±0.08	**0.45±0.07**	0.39±0.08	**0.48±0.08**	0.47±0.10	0.47±0.10	0.39±0.05	**0.43±0.06**	0.44±0.07	**0.48±0.09**
Macro P↑	0.19±0.04	**0.20±0.03**	0.15±0.03	**0.19±0.03**	0.16±0.02	**0.19±0.03**	0.06±0.01	0.06±0.01	0.16±0.01	**0.19±0.03**	0.16±0.05	**0.19±0.06**
Macro R↑	0.13±0.01	**0.14±0.01**	0.11±0.01	**0.12±0.01**	**0.13±0.02**	0.12±0.02	0.08±0.00	0.08±0.00	0.14±0.02	**0.18±0.02**	0.12±0.01	**0.14±0.02**
Macro F1↑	0.14±0.02	0.14±0.01	0.11±0.02	**0.12±0.02**	0.13±0.02	0.13±0.03	0.07±0.01	0.07±0.01	0.14±0.01	**0.17±0.02**	0.12±0.02	**0.14±0.03**
Micro P↑	0.60±0.10	**0.65±0.10**	0.59±0.10	**0.67±0.12**	0.49±0.08	**0.65±0.11**	0.67±0.14	0.67±0.14	0.48±0.04	**0.51±0.04**	0.57±0.09	**0.66±0.13**
Micro R↑	0.34±0.05	**0.36±0.05**	0.31±0.04	**0.33±0.06**	0.31±0.05	**0.33±0.06**	0.30±0.07	0.30±0.07	0.35±0.04	**0.40±0.05**	0.34±0.04	**0.36±0.07**
Micro F1↑	0.43±0.06	**0.46±0.07**	0.40±0.05	**0.44±0.08**	0.38±0.06	**0.44±0.08**	0.42±0.09	0.42±0.09	0.40±0.04	**0.44±0.04**	0.43±0.05	**0.46±0.09**
H Loss↓	0.17±0.03	**0.16±0.02**	0.17±0.03	**0.16±0.03**	0.19±0.03	**0.16±0.03**	0.16±0.03	0.16±0.03	0.19±0.02	0.19±0.02	0.17±0.02	**0.16±0.03**

[1] ↑ indicates the higher value, the better, and ↓ the lower, the better

Table 13. Comparative results for Yeast dataset between Native and TEN

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	Native	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**
Precision↑	0.73±0.03	**0.85±0.03**	0.73±0.03	**0.83±0.02**	0.66±0.02	**0.79±0.03**	0.67±0.02	**0.83±0.03**	0.64±0.02	**0.78±0.03**	0.70±0.03	**0.82±0.02**
Recall↑	0.55±0.01	**0.76±0.00**	0.57±0.02	**0.76±0.02**	0.62±0.02	**0.74±0.01**	0.63±0.01	**0.77±0.01**	0.60±0.02	**0.75±0.01**	0.61±0.04	**0.76±0.00**
F1↑	0.60±0.01	**0.78±0.01**	0.61±0.02	**0.77±0.02**	0.62±0.02	**0.74±0.02**	0.63±0.02	**0.78±0.02**	0.59±0.02	**0.74±0.01**	0.62±0.02	**0.77±0.01**
Macro P↑	0.57±0.05	**0.75±0.04**	0.56±0.07	**0.73±0.04**	0.49±0.06	**0.66±0.03**	0.50±0.04	**0.71±0.02**	0.47±0.02	**0.65±0.04**	0.51±0.03	**0.72±0.04**
Macro R↑	0.32±0.00	**0.54±0.01**	0.34±0.01	**0.53±0.02**	0.39±0.01	**0.54±0.01**	0.38±0.00	**0.54±0.01**	0.40±0.01	**0.59±0.02**	0.36±0.03	**0.54±0.01**
Macro F1↑	0.35±0.00	**0.59±0.01**	0.37±0.01	**0.58±0.02**	0.40±0.01	**0.57±0.01**	0.37±0.00	**0.58±0.01**	0.42±0.01	**0.61±0.02**	0.37±0.01	**0.58±0.01**
Micro P↑	0.74±0.03	**0.87±0.03**	0.73±0.02	**0.85±0.02**	0.67±0.02	**0.79±0.02**	0.68±0.02	**0.84±0.03**	0.65±0.02	**0.78±0.03**	0.70±0.03	**0.83±0.02**
Micro R↑	0.55±0.01	**0.75±0.01**	0.57±0.01	**0.74±0.02**	0.61±0.02	**0.73±0.01**	0.62±0.01	**0.75±0.01**	0.59±0.01	**0.74±0.01**	0.60±0.03	**0.75±0.01**
Micro F1↑	0.63±0.01	**0.80±0.01**	0.64±0.01	**0.79±0.01**	0.64±0.02	**0.76±0.01**	0.65±0.01	**0.79±0.01**	0.62±0.01	**0.76±0.01**	0.65±0.02	**0.79±0.01**
H Loss↓	0.19±0.01	**0.11±0.00**	0.19±0.01	**0.12±0.01**	0.21±0.01	**0.14±0.00**	0.20±0.01	**0.12±0.01**	0.22±0.01	**0.14±0.01**	0.20±0.01	**0.12±0.00**

Table 14. Comparative results for Emotions dataset between Native and TEN

Metric	**BR**		**CC**		**LP**		**MLTSVM**		**ML-*k*NN**		**RAkELd**	
	Native	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**	**Native**	**TEN**
Precision↑	0.56±0.07	**0.56±0.01**	0.56±0.05	**0.60±0.05**	0.57±0.05	**0.57±0.03**	0.31±0.06	**0.60±0.04**	0.42±0.05	**0.63±0.03**	0.55±0.03	**0.55±0.04**
Recall↑	0.51±0.07	**0.52±0.05**	0.51±0.06	**0.57±0.05**	0.50±0.04	**0.56±0.06**	0.36±0.08	**0.68±0.08**	0.37±0.08	**0.63±0.07**	0.53±0.08	**0.56±0.04**
F1↑	0.50±0.04	**0.51±0.02**	0.50±0.04	**0.55±0.01**	0.53±0.03	**0.54±0.03**	0.31±0.05	**0.62±0.04**	0.37±0.06	**0.60±0.04**	0.51±0.05	**0.53±0.03**
Macro P↑	0.61±0.09	**0.62±0.02**	0.59±0.08	**0.62±0.04**	0.56±0.04	**0.57±0.05**	0.11±0.03	**0.59±0.04**	0.42±0.07	**0.63±0.03**	0.57±0.06	**0.60±0.04**
Macro R↑	0.51±0.07	**0.51±0.06**	0.57±0.07	**0.58±0.07**	0.57±0.04	**0.58±0.08**	0.28±0.08	**0.70±0.07**	0.35±0.08	**0.62±0.08**	0.52±0.07	**0.57±0.05**
Macro F1↑	0.53±0.05	**0.54±0.03**	0.50±0.07	**0.57±0.03**	0.55±0.04	**0.55±0.03**	0.15±0.04	**0.63±0.03**	0.37±0.07	**0.62±0.05**	0.52±0.06	**0.57±0.01**
Micro P↑	0.54±0.07	**0.64±0.03**	0.52±0.05	**0.65±0.04**	0.56±0.05	**0.57±0.03**	0.32±0.06	**0.60±0.04**	0.45±0.05	**0.65±0.02**	0.60±0.04	**0.60±0.03**
Micro R↑	0.45±0.04	**0.53±0.05**	0.50±0.05	**0.58±0.06**	0.50±0.03	**0.58±0.07**	0.30±0.10	**0.70±0.08**	0.38±0.08	**0.64±0.07**	0.53±0.07	**0.58±0.04**
Micro F1↑	0.48±0.04	**0.58±0.02**	0.55±0.05	**0.61±0.02**	0.56±0.03	**0.57±0.03**	0.31±0.06	**0.64±0.04**	0.41±0.07	**0.64±0.04**	0.56±0.05	**0.59±0.02**
H Loss↓	0.29±0.03	**0.25±0.02**	0.29±0.03	**0.24±0.02**	0.29±0.02	**0.28±0.02**	0.40±0.04	**0.25±0.02**	0.30±0.04	**0.24±0.04**	0.29±0.03	**0.26±0.02**

[1] ↑ indicates the higher value, the better, and ↓ the lower, the better

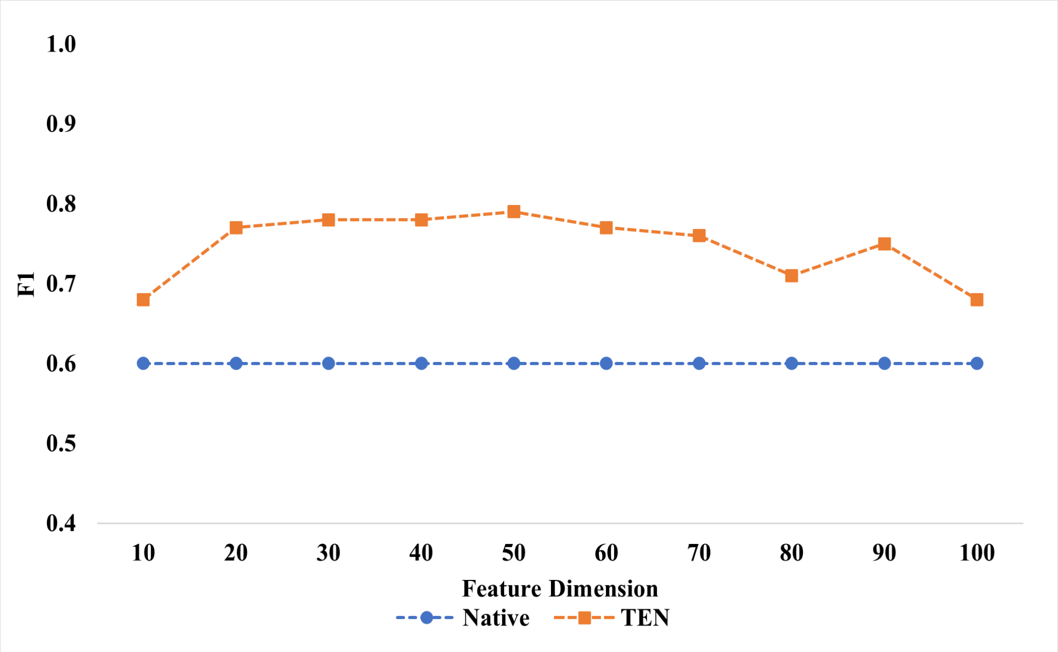
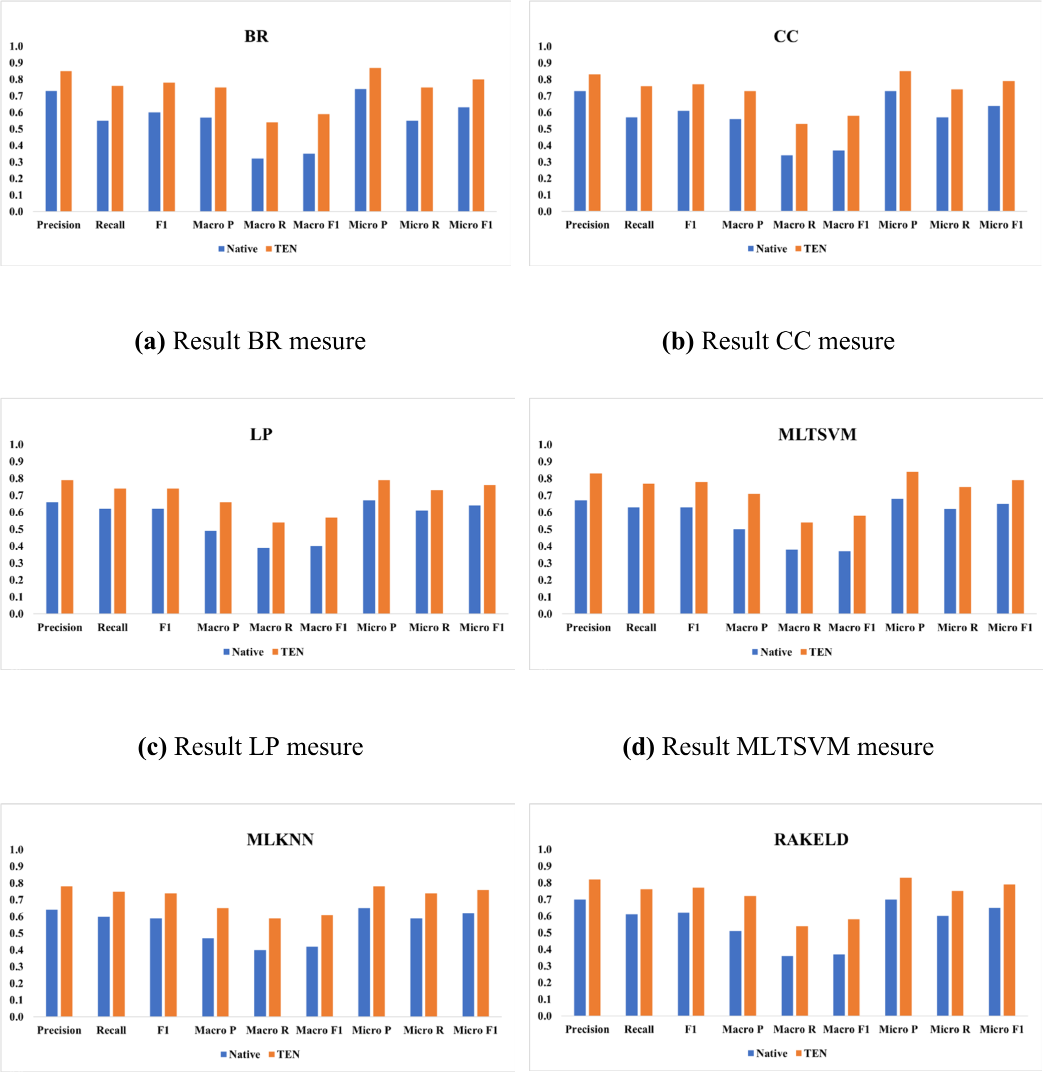


Figure 3. The results were obtained from the proposed feature reconstruction method and the Native data feature. In the reconstruction method, the number of feature dimensions is varied, from 10 to 100.

The experimental results deliberately delineated the performance of the proposed technique. For all data sets, TEN essentially provided promising results, which were better than EN. TEN worked well with the Yeast and Emotions datasets and gave better results for all the MLC algorithms and the measurement metrics. The Yeast and Emotion were the only two datasets with high density. The density of the dataset in MLC indicates the wellness of presentation of the class labels. Therefore, TEN worked well with the high-density dataset (well-presented data) for MLC problems. In addition, the results obtained from the second experiment on the Yeast and Emotions datasets showed that the reconstruction technique was superior to the Native data features (without feature transformation processes). In general, feature reconstruction can produce different sizes of compact features. Therefore, we varied the sizes of the reconstructed features to observe the sensitivity of the technique. The results indicated that TEN gave better results than the Native features for all MLC problems and measurement metrics.

In future work, we will investigate how to further improve methods of dealing with classification problems, and especially datasets with low density. From the classification perspective, we anticipate exploring ways to improve the classification process as well.



(a) Result BR mesure

(b) Result CC mesure

(c) Result LP mesure

(d) Result MLTSVM mesure

(e) Result ML-*k*NN mesure

(f) Result RAkELd mesure

Figure 4. Comparative results for Yeast dataset between Native features and TEN

5. Acknowledgements

This research project was financially supported by Mahasarakham University. The authors would like to thank our advisors, lecturers, technicians, and respondents for their full support.

References

[1] Chandran, S.A. and Panicker, J.R., 2017. An efficient multi-label classification system using ensemble of classifiers. *Proceeding of International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Kerala, India, July 6-7, 2017, pp. 1133-1136.

[2] Prajapati, P. and Thakkar, A., 2021. Performance improvement of extreme multi-label classification using K-way tree construction with parallel clustering algorithm. *Journal of King Saud University - Computer and Information Sciences*, DOI: 10.1016/j.jksuci.2021.02.014.

[3] Bogatinovski, J., Todorovski, L., Dzeroski, S. and Kocev, D., 2021. *Comprehensive Comparative Study of Multi-label Classification Methods*. [online] Available at: https://arxiv.org/pdf/2102.07113.pdf.

[4] Alazaidah, R. and Ahmad, F.K., 2016. Trending challenges in multi label classification. *Journal of Advanced Computer Science and Applications*, 7, DOI: 10.14569/IJACSA.2016.071017.

[5] Pushpa, M. and Karpagavalli, S., 2017. Multi-label classification: Problem transformation methods in Tamil Phoneme classification. *Journal of Procedia Computer Science*, 115, 572-579.

[6] Alluwaici, M., Junoh, A.K. and Alazaidah, R., 2020. New problem transformation method based on the local positive pairwise dependencies among labels. *Journal of Information and Knowledge Management,* 19(01), DOI: 10.1142/S0219649220400171.

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

[8] Tsoumakas, G. and Katakis, I., 2007. Multi-label classification: An overview. *Journal of Data Warehousing and Mining*, 3(3), 1-13.

[9] Madjarov, G., Kocev, D., Gjorgjevikj, D. and Džeroski, S., 2012. An extensive experimental comparison of methods for multi-label learning. *Journal of Pattern Recognition*, 45(9), 3084-3104.

[10] Sangkatip, W. and Phuboon-Ob, J., 2020. Non-communicable diseases classification using multi-label learning techniques. *Proceeding of the 5th International Conference on Information Technology (InCIT)*, Chonburi, Thailand, October 21-22, 2020, pp. 17-21.

[11] Sousa, R. and Gama, J., 2016. Online multi-label classification with adaptive model rules. *Proceedings of 17th Conference of the Spanish Association for Artificial Intelligence*, Salamanca, Spain, September 14-16, 2016, pp. 58-67.

[12] García, S.M., Mantas, C., Castellano, F. and Abellán, J., 2019. Ensemble of classifier chains and Credal C4.5 for solving multi-label classification. *Journal of Progress in Artificial Intelligence*, 8, DOI: 10.1007/s13748-018-00171-x.

[13] Zhang, M.L. and Zhou, Z.H., 2007. ML-KNN: A lazy learning approach to multi-label learning. *Journal of Pattern Recognition*, 40(7), 2038-2048.

[14] Zhang, M.L. and Zhou, Z.H., 2006. Multilabel neural networks with applications to functional genomics and text categorization. *Journal of IEEE Transactions on Knowledge and Data Engineering*, 18(10), 1338-1351.

[15] Read, J., Pfahringer, B. and Holmes, G., 2008. Multi-label classification using ensembles of pruned sets. *Proceedings of the IEEE International Conference on Data Mining*, Pisa, Italy, December 15-19, 2008, pp. 995-1000.

[16] Jin, W., Hong, W., Cuiping, X., Weihua, O., Qiaosong, C. and Xin, D., 2017. Ensembles of classifier chains for multi-label classification based on Spark. *Journal of University of Science and Technology of China*, 47(4), 350-357.

[17] Tsoumakas, G., Katakis, I. and Vlahavas I., 2011. Random k-labelsets for multilabel classification. *Journal of IEEE Transactions on Knowledge and Data Engineering,* 23(7), 1079-1089.

[18] Zhang, M.-L., 2011. Lift: Multi label learning with label-specific features. *Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37, 1609-1614.

[19] Gao, W., Hu, J., Li, Y. and Zhang, P., 2020. Feature redundancy based on interaction information for multi-label feature selection. *Journal of IEEE Access*, 8, 146050-146064.

[20] Huang, J., Li, G. and Wu, X., 2018. Joint feature selection and classification for multilabel learning. *Journal of IEEE Transactions on Cybernetics*, 48, 1-14.

[21] Guozhu, D. and Huan, L., 2018. *Feature Engineering for Machine Learning and Data Analytics.* New York: CRC Press.

[22] Hafeez, G., Khan, I., Jan, S., Shah, I.A., Khan, F.A. and Derhab, A., 2021. A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid. *Journal of Applied Energy*, 299, 117178.

[23] Emmert-Streib, F., Yang, Z., Feng, H., Tripathi, S. and Dehmer, M., 2020. An introductory review of deep learning for prediction models with big data. *Journal of Frontiers in Artificial Intelligence*, 3, DOI: 10.3389/frai.2020.00004.

[24] Deng, Z., Wang, S. and Chung, F.L., 2013. A minimax probabilistic approach to feature transformation for multi-class data. *Journal of Applied Soft Computing*, 13(1), 116-127.

[25] Patterson, J. and Gibson, A., 2017. *Deep Learning*. California: O'Reilly Media.

[26] Cheng, Y., Zhao, D., Wang, Y. and Pei, G., 2019. Multi-label learning with kernel extreme learning machine autoencoder. *Journal of Knowledge-Based Systems*, 178, 1-10.

[27] Read, J., Puurula, A. and Bifet, A., 2015. Multi-label classification with meta-labels. *Proceedings of the IEEE International Conference on Data Mining*, Shenzhen, China, December 14-17, 2015, pp. 941-946.

[28] Cherman, E., Monard, M.-C. and Metz, J., 2011. Multi-label problem transformation methods: a case study. *CLEI Electronic Journal*, 14, DOI: 10.19153/cleiej.14.1.4.

[29] Cortes, C. and Vapnik, V., 1995. Support-vector networks. *Journal of Machine Learning*, 20(3), 273-297.

[30] Elisseeff, A. and Weston, J., 2001. Kernel methods for multi-labelled classification and categorical regression problems. *Advances in Neural Information Processing Systems*, 14, 681-687.

[31] Gibaja, E., Moyano, J. and Ventura, S., 2016. An ensemble-based approach for multi-view multi-label classification. *Journal of Progress in Artificial Intelligence*, 5, DOI: 10.1007/s 13748-016-0098-9.

[32] Rokach, L., Schclar, A. and Itach, E., 2013. Ensemble methods for multi-label classification. *Journal of Expert Systems with Applications*, 41, DOI: 10.1016/j.eswa.2014.06.015.

[33] Kimura, K., Kudo, M., Sun, L. and Koujaku, S., 2016. Fast random k-labELsets for large-scale multi-label classification. *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR)*, Cancun, Mexico, December 4-8, 2016, pp. 438-443.

[34] Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J. and Vlahavas, I., 2011. MULAN: A Java library for multi-label learning. *Journal of Machine Learning Research*, 12, 2411-2414.

[35] Liou, C.-Y., Cheng, W.-C., Liou, J.-W. and Liou, D.-R., 2014. Autoencoder for words. *Journal of Neurocomputing*, 139, 84-96.

[36] Read, J., Pfahringer, B., Holmes, G. and Frank, E., 2011. Classifier chains for multi-label classification. *Journal of Machine Learning*, 85(3), 333-359.

[37] Chen, W.-J., Shao, Y.-H., Li, C.-N. and Deng, N.-Y., 2016. MLTSVM: a novel twin support vector machine to multi-label learning. *Journal of Pattern Recognition*, 52, 61-74.

[38] Tsoumakas, G., Katakis, I. and Vlahavas, I., 2011. Random k-labelsets for multilabel Classification. *Journal of IEEE Transactions on Knowledge and Data Engineering*, 23(7), 1079-1089.

[39] Szymański, P. and Kajdanowicz, T., 2019. Scikit-multilearn: a scikit-based Python environment for performing multi-label classification. *Journal of Machine Learning Research,* 20, 209-230.

[40] Wu, X.-Z. and Zhou, Z.-H., 2017. A unified view of multi-label performance measures. *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, August 6-11, 2017, pp. 3780-3788.