symmetry

MDPI

Article

Unified Graph-Based Missing Label Propagation Method for Multilabel Text Classification

Adil Yaseen Taha, Sabrina Tiun *, Abdul Hadi Abd Rahman, Masri Ayob and Ali Sabah Abdulameer

Center for Artificial Intelligence Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia; adil.yaseen89@gmail.com and P89062@siswa.ukm.edu.my (A.Y.T.); abdulhadi@ukm.edu.my (A.H.A.R.); masri@ukm.edu.my (M.A.); alisabahphd@gmail.com and P88931@siswa.ukm.edu.my (A.S.A.)
* Correspondence: sabrinatiun@ukm.edu.my

Abstract: In multilabel classification, each sample can be allocated to multiple class labels at the same time. However, one of the prominent problems of multilabel classification is missing labels (incomplete labels) in multilabel text. The multilabel classification performance is reduced significantly with the presence of missing labels. In order to address the incomplete or missing label problem, this study proposes two methods: an aggregated feature and label graph-based missing label handling method (GB-AS), and a unified graph-based missing label propagation method (UG-MLP). GB-AS is used to obtain an initial label matrix based on the similarity of both document levels: feature-based weighting representation and label-based weighting representation. On the other hand, UG-MLP is introduced to construct a mixed graph that combines GB-AS and label correlations into a single groundwork. A high-order label correlation is learned from the incomplete training data and applied to supplement the missing label matrix, which guides the creation of multilabel classification models. The combination of the mixed graphs by UG-MLP is aimed to obtain the benefits of both graphs to increase the classification performance. To evaluate UG-MLP, the metrics of precision, recall and F-measure were used on three benchmark datasets, namely, the Reuters-21578, Bibtex and Enron datasets. The experimental results show that UG-MLP outperformed GB-AS as well as other state-of-the-art approaches. Therefore, we can infer from the findings that by plotting a unified graph based on joining aggregated feature and label weightings together with the label correlation, the performance of multilabel classification can be improved.

Keywords: text mining; multilabel classification; label propagation; missing labels; label correlations; feature correlations

Citation: Taha, A.Y.; Tiun, S.; Rahman, A.H.A.; Ayob, M.; Abdulameer, A.S. Unified Graph-Based Missing Label Propagation Method for Multilabel Text Classification. *Symmetry* **2022**, *14*, 286. https://doi.org/10.3390/sym14020286

Academic Editor: Jeng-Shyang Pan

Received: 18 December 2021
Accepted: 14 January 2022
Published: 31 January 2022

1. Introduction

In multilabel learning, each label is connected with one or more labels simultaneously. The main key difference between multilabel and single-label learning is that the labels in multilabel learning are correlated. Therefore, the multilabel learning task is slightly more difficult to resolve. In machine learning and data mining, multilabel learning is a task that suffers from the curse of high dimensionality. There are various intricacies in real-world multilabel datasets that reduce classifiers' performance. The following are open problems: high dimensionality, feature and label correlations and missing labels in multilabel classification [1]. This paper intended to focus on the problem of multilabel learning with missing labels or incomplete labels. Given training cases that have an incomplete or partial collection of these labels (i.e., some of their labels are missing), the suggested approach in this research seeks to label each test item with multiple labels. Handling high dimensionality and feature correlations in multilabel learning may not effectively work if it does not consider the missing label problem (incomplete and noisy label space). Most contemporary approaches treat this problem as a supervised weak-label learning problem, assuming that there are enough partially labeled examples available

[2–4]. Collecting or annotating such instances, on the other hand, is costly and time consuming. In multilabel learning, usually, the label sets of objects sharing the same cluster are strongly connected, whereas label sets of other clusters are loosely correlated [5].

Most of the existing multilabel learning methods assume a full dataset is given, and each instance in it is attached to a full form of the label set. It is hard to believe that multilabel datasets are full, and it is incorrect to do so (all possible class combinations have training instances), mainly when the size of their base classes is large. In addition, it is also hard to believe that all examples are assigned to complete and correct label sets. Several factors contribute to the difficulty of gathering high-quality data: Firstly, the missing label problem occurs because of the existence of synonyms and ambiguities among distinct classes, causing annotators to select only a portion of the labels with similar meanings [3,6]. The missing label problem is also caused by the subjectivity of manual labeling by tag providers, and the large size of the category vocabulary in some datasets makes it difficult for labelers to annotate each instance [3]. Furthermore, incomplete labeling problems caused by the huge number of instances and possible assigned labels result in a high cost in labor and time. In other words, the performance of multilabel learning algorithms is influenced by label incompleteness. Based on this, an effective multilabel text classification method should handle missing labels and label correlations. Thus, the following are the key contributions of this paper.

1. Proposing the graph-based aggregated similarity weighting of features and labels method (GB-AS) to predict missing labels in multilabel text classification;
2. Further improving GB-AS by combining it with label correlations to create a unified graph method called UG-MLP.

This paper is organized into six sections: It starts with the introduction of the paper. Section 2 introduces related work on multilabel learning with missing labels, while Section 3 presents GB-AS. Section 4 presents UG-MLP. Section 5 introduces the implementation of GB-AS and UG-MLP in multilabel text classification. The experiments are highlighted in Section 6. Discussions are introduced in Section 7. Lastly, the last section (Section 8) winds up this paper.

2. Related Work

Many researchers have noticed that fully supervised information for multilabel learning is difficult to acquire. There are some works that focused directly on solving the problem of multilabel learning with missing labels [4,6–18], which has also been called learning with missing label assignments or incomplete labels.

Hashemi et al. [15] presented the MGFS method, which is a graph-based multilabel feature selection. The correlation distance matrix (CDM) is formed using their proposed method, which estimates the correlation distance between characteristics and each class label. The Euclidean distance is then used with the CDM to create a complete weighted feature label network with nodes representing features. Lastly, the relevance of graph nodes is determined via the weighted PageRank algorithm. LSML, a new technique for learning label-specific features for multilabel classification with missing labels, was presented by Huang et al. [4]. First, by learning high-order label correlations, a new supplemental label matrix is created from the partial label matrix. Then, for each class label, a label-specific data representation is learned, and the multilabel learner is built concurrently using the learned high-order label correlations. Sun et al. [18] presented the cost-sensitive label ranking approach with low-rank and sparse constraints, called CORALS, to enhance missing labels and remove noisy labels at the same time, the relevance ordering of all possible labels on each instance, including both missing and noisy labels, is optimized by reducing a cost-sensitive ranking loss. Zhu et al. [12] suggested a novel multilabel feature selection with missing labels to address missing labels, multilabel learning and feature selection simultaneously. Multilabel feature selection, on the other hand, solely characterizes pairwise label associations by generating a graph at the instance level.

Meanwhile, He et al. [10] proposed a novel multilabel classification method with label correlations, missing labels and feature selection, named MLMF. MLMF allows for combined learning of multilabel classification and label correlations as well as joint learning of independent binary classifiers. Wu et al. [8] suggested a novel methodology based on a unified network of label dependencies to solve the challenge of multilabel learning with missing labels. To convey label information from provided labels to missing labels, a uniform network of label dependencies is established using a mixed dependency graph.

Guan and Li [7] presented a novel Bayesian model with label regularization and label confidence constraints, named BM-LRC, to handle the difficulties of incomplete labels in multilabel text classification and exploit label correlations with two label constraints. The label manifold regularization might aid in the handling of incomplete labels. The label confidence constraints, contrarily, can prevent overestimation of negative labels induced by regularizing labels, resulting in a safer inference. Ibrahim et al. [9] proposed a weighted loss function to account for the confidence in each label/sample pair that can easily be incorporated to adjust a pre-trained model on missing labels or incomplete labels in multilabel text dataset problems. Pal et al. [11] presented an attention-based graph neural network (AGNET) model for capturing the attentive correlation structure between labels. A feature matrix and a correlation matrix are used by the graph attention network to capture and examine the fundamental dependencies between the labels and to build classifiers for the assignment. The generated classifiers are used on sentence feature vectors obtained from the text feature extraction network to enable end-to-end training. Label imputation in training sets was performed by Ma and Chow using their two-level label recovery approach [13]. This approach recovers the label matrix by using an instance-wise semantic relational graph and a label-wise semantic relational graph. These two graphs show that two-level semantic relationships may be reliably captured. In addition, a label-specific feature selection method was proposed for performing label prediction in testing sets. Wang et al. [14] presented new principles of multilabel information entropy and multilabel correlative information used to identify the unnecessary features, feature independence and feature interaction. In a multilabel text dataset, feature interaction is used to choose more valuable characteristics that could otherwise be overlooked due to the inadequate label space.

Song et al. [17] proposed a label mask multilabel text classification model (LM-MTC), which is inspired by the idea of cloze questions in language models. LM-MTC is able to capture implicit relationships among labels through the powerful ability of pre-trained language models. To create a label-based masked language model (MLM), they assigned a separate token to each conceivable label and randomly masked the token with a certain probability. However, six multilabel datasets, including the Reuters-21578 text dataset, were used to test the proposed method. The proposed method was compared against eleven other methods including the following: binary relevance (BR), classifier chains (CC), CNN, CNN-RNN, hierarchical attention network (HAN), HAN + label graph (LG), BERT, BERT + MLM, MEGNET and label-wise (LW). On all datasets, the proposed method outperformed the other methods in terms of the F-measure. Li and Yang [18] proposed a dependence maximization-based label embedding approach for obtaining the latent space, where the label and feature information can both be included at the same time. In addition, instead of using the encoding method, the low-rank factorization model on the label matrix is used to leverage label correlations. The Hilbert–Schmidt independence criterion increases the reliance between the feature space and label space in order to improve predictability. For multilabel text classification, a CNN integrated with a capsule network was presented by Yan S et al. [19]. To extract information relating to classification outcomes in high-dimensional features, they utilized a capsule network instead of a pool layer in the CNN. In addition, they explored joining a recurrent neural network (RNN) with a convolutional neural network (CNN) to describe the frequency and space properties of a capsule network to complete categorization. Nevertheless, two multilabel datasets, including the Reuters-21578 text dataset, were used to evaluate the proposed

method. The results demonstrated that the proposed method outperformed Conv, Conv-Cap and Rec-Conv. In terms of the F-measure, the proposed method achieved a higher result on all the datasets.

Class labels typically have connections with one another in multilabel learning. Previous research has shown that solving the problem of missing labels can considerably increase multilabel learning performance, both theoretically and practically [7–11]. However, when class labels from the training data are missing, the label correlation directly acquired from the incomplete label matrix may be erroneous, greatly affecting the performance of multilabel classifiers. In the meantime, previous approaches to multilabel learning with missing labels primarily use a similar representation of the data consisting of all the features in the discrimination of all the class labels [4,16,18]. As previously described, this common strategy may be suboptimal. Therefore, the difficult problem of multilabel learning with missing labels is how to learn accurate label correlations from the incomplete label data and use them to guide the development of classification models.

In this paper, we propose GB-AS to predict the missing labels and then further improve it by combining it with label correlations to create a unified graph method, called UG-MLP, to improve the performance of multilabel learning.

3. Graph-Based Aggregated Similarity Weighting of Features and Labels Method (GB-AS)

This section presents the proposed graph-based aggregated similarity weighting of features and labels method, GB-AS, to aid in handling missing labels in multilabel text classification. To solve missing label problems, GB-AS recovers the underlying label matrix (Y) via transferring the label information from the feature space and label distribution of the nearest neighbor to the label space. Nearest neighbor instances often share similar features and label information, which means that it is highly likely that these instances have the same set of labels. Prior to describing the GB-AS method, the following presents an instance (document) in a weighted graph representation.

3.1. Instance-Level Feature/Label Graph Construction

A weighted graph of instances is constructed based on an assumption: an edge connecting two instances carries a notion of similarity. Thus, if instances are connected, they share similar features and label information, which means that it is highly likely that these instances have the same set of labels. The graph shown in Figure 1 demonstrates the similarity between different documents, e.g., between documents D1 and D3, there is a similarity of 70%. Therefore, this information can be used to predict missing labels.

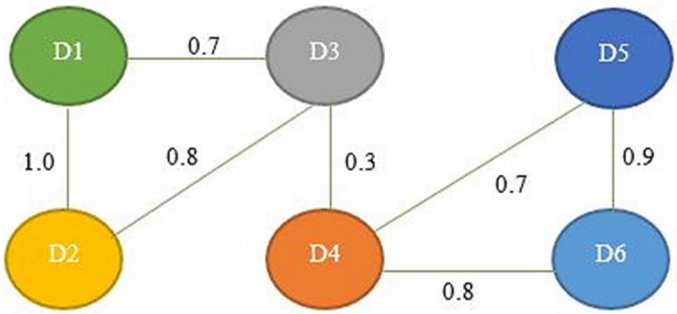

Figure 1. Example of a weighted graph of instances based on feature similarity and label similarity.

Multilabel text data can be represented using either a feature space representation or a label space representation as follows:

Feature space representation D_f:

$$D_f = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} f_{11} & \vdots & f_{1m} \\ f_{21} & \vdots & f_{2m} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ f_{N1} & \vdots & f_{Nm} \end{bmatrix}$$

where X_i is an instance (text or document), and f_{ij} is the value of feature j in document i.

Label space representation D_l:

$$D_l = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} l_{11} & \vdots & l_{1z} \\ l_{21} & \vdots & l_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ l_{N1} & \vdots & l_{Nz} \end{bmatrix}$$

where X_i is an instance (text or document), and l_{ij} is the value of feature j in document i.

Based on the representation of the instance-level feature and label space above, the handling of the missing label problem is based on the following considerations:

- Graph-based missing label handling with feature similarity weighting (GB-FS);
- Graph-based missing label handling with label similarity weighting (GB-LS).

With the assumption that, by combining information of both the feature and the label, predicting missing label can be more accurate, we propose the graph-based missing label handling that aggregates both document-level feature-based weighting and label-based weighting into one aggregated similarity weighting, GB-AS. The following Figure 2 illustrates the construction of the GB-AS algorithm, where the output of GB-AS is an input to predict missing labels.

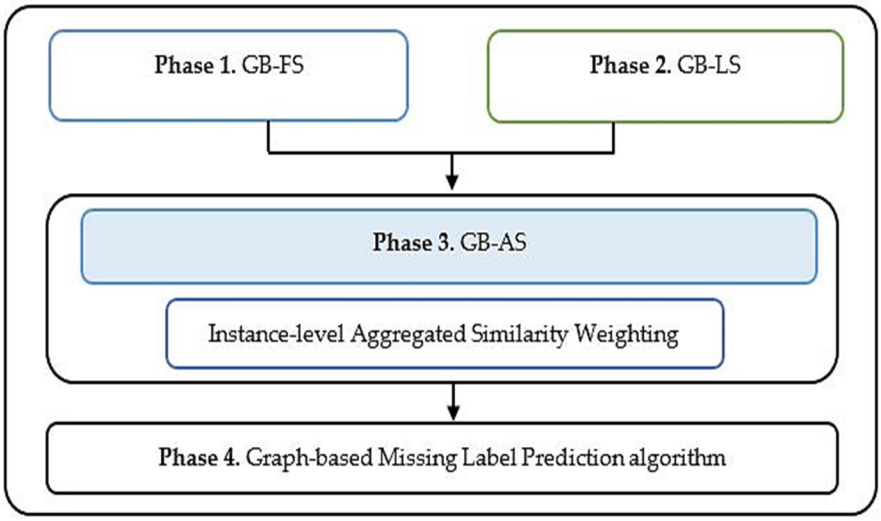

Figure 2. Graph-based aggregated similarity weighting of features and labels to predict missing labels.

The following subsection describes the four phases of the graph-based aggregated similarity weighting of features and labels to predict missing labels shown in Figure 2.

3.1.1. Phase 1. Instance-Level Feature Space Similarity Weighting

As mentioned above, the feature space representation represents each instance (document) as a vector of feature values as follows:

$$D = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} f_{11} & \vdots & f_{1m} \\ f_{21} & \vdots & f_{2m,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ f_{N1} & \vdots & f_{Nm} \end{bmatrix}$$

The instance feature space similarity weighting is obtained as follows:

1. Given the feature space representation matrix, each instance is modeled as a vector of feature values:

$$xi = \{f_{i1} \quad \vdots \quad f_{im}\}$$

$$xj = \{f_{11} \quad \vdots \quad f_{1m}\}$$

where both xi and xj are documents.

2. To compute the pairwise similarity values between two documents where each document is represented by a numerical feature vector, the cosine similarity is used to measure the similarity. Cosine similarity is one of the most well-known similarity measures which is applied to text documents [20,21] based on the following Equation (1):

$$\cos(x_i, x_j) = \frac{|x_i| * |x_j|}{\sqrt{x_i^2 * x_j^2}} \tag{1}$$

3. The feature space is built using the k-highest similarity neighborhoods to maintain the intrinsic local correlation information. To ensure feature space representation validity in recovering label structures, the weight matrix W of the feature space similarity is defined as in the following Equation (2) [2,22]:

$$W^{fs} = w_{ij}^{fs} = \frac{exp^{(\cos(x_i, x_j))}}{\sum_q exp^{(\cos(x_i, x_q))}} \; if \; xj \in N_k(xi) \tag{2}$$

where $W \in R^{n \times n}$ and N_k (xi) denote the k-highest similarity neighborhoods of the $i-th$ instance measured by $\cos(x_i, x_j)$; exp is the natural logarithm; $\sum_q exp^{(\cos(x_i, x_q))}$ is the summation of cosine similarities of the $i-th$ instance with all its k-highest similarity neighborhoods. The summation is used as a normalization method to make sure that the similarities are between 0 and 1.

3.1.2. Phase 2. Instance-Level Label Space Similarity Weighting

The label space representation represents each instance (document), *X*, as a vector of its assigned labels, l, where the assigned labels are assigned by a human as in training data or by a multilabel classifier for testing data values.

$$D = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} l_{11} & \vdots & l_{1z} \\ l_{21} & \vdots & l_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ l_{N1} & \vdots & l_{Nz} \end{bmatrix}$$

The instance label similarity matrix is obtained as follows:

1. Given the label space representation matrix, each instance is modeled as a vector of assigned label values:

$$xi = \{l_{i1} \ \vdots \ l_{im}\}$$
$$xj = \{l_{j1} \ \vdots \ l_{jm}\}$$

where both xi and xj are documents, and l_{i1} can be 1 if label l_1 is assigned to document xi, 0 if l_1 is not assigned to document xi or −1 if l_i is unknown or missing for document xi.

2. The hamming-based similarity is often used for the measurement of label similarity relationships in multilabel datasets, and it demonstrates how close the label sets of x and y instances are [23–25]. R (x_i, x_j) is the complement of the normalized Hamming distance between the label sets of two elements. It is defined as follows (see Equation (3)):

$$\mathrm{R}(x_i, x_j) = 1 - \frac{|L_{xi} \Delta L_{xj}|}{z} \quad (3)$$

$L_{xi}\Delta L_{xj}$ is the number of labels that have different assignments with xi and xj, Δ is the symmetric difference between its two arguments, $|\cdot|$ is the cardinality of the resulting set and z is the number of labels in the label set.

3. The weighted matrix of the label space, W^{ls}, is constructed as follows (see Equation (4)):

$$W^{ls} = w_{ij}^{ls} = \frac{\mathrm{R}(x_i, x_j)}{\sum_q \mathrm{R}(x_i, x_q)} \ if \ xj \in N_k(xi) \quad (4)$$

where $W \in \mathrm{R}^{n \times n}$ and N_k (xi) denote the k-highest similarity neighborhoods of the $i-th$ instance measured by the Hamming-based similarity, and $\sum_q \mathrm{R}(x_i, x_q)$ is the summation of the Hamming-based similarities of the $i-$ th instance with all its k-highest similarity neighborhoods. The summation is used as a normalization method to make sure that all similarities are between 0 and 1.

3.1.3. Phase 3. Instance-Level Aggregated Similarity Weighting

In the previous steps (Phase I and Phase II), two document-level weighting matrices are obtained. The first one is the document-level feature-based weighting matrix in which the similarities between documents are estimated based on the weighting of their shared features (see Phase I). The second one is the document-level label-based weighting matrix in which the similarities between documents are calculated based on their shared labels (see Phase II).

The similarity weighting of GB-AS is the document-level aggregated similarity weighting of both document-level feature-based weighting and document-level label-based weighting. The following is the function of the aggregated similarity weighting (see Equation (5)):

$$w_{ag}^{d} = \alpha \, w^{ls} + \beta \, w^{fs} \quad (5)$$

where α and β are the weight numbers decided by the characteristics of the label space similarity matrix (see Equation (2)) and feature space similarity matrix (see Equation (4)). Several experiments are conducted to find the best values of α and β.

3.1.4. Phase 4: GB-AS

As shown in Algorithm 1, step 1 is used to build a graph using Equation (5), and the following describes, in more detail, the steps of the algorithm (see Algorithm 1):

Step 1: A graph $G = \{V, E, W\}$ is constructed that consists of nodes V which represent documents and edges E which reflect the similarity between the vertices.

Step 2: The transition matrix or weighting matrix T is obtained using either the document-level feature-based weighting matrix w^{fs}, the document-level label-based weighting matrix w^{fs} or the aggregated similarity weighting matrix w^{ag}. The weight T_{ij} between the vertex of document d_i and the vertex of document d_j represents the transition probability of information between these two vertices.

Step 3: Next, the algorithm initializes the label matrix Y. For the label matrix $Y=\begin{bmatrix} l_{11} & \vdots & l_{1z} \\ l_{21} & \vdots & l_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ l_{N1} & \vdots & l_{Nz} \end{bmatrix}$, $l_{i,k}$ is 1 if document x_i is labeled as class K, 0 if document x_i is not labeled as class K or $\frac{1}{2}$ if the label is missing.

Step 4: First iterative step in missing label prediction: In this step, the new label matrix is calculated using Equation (6) [26]:

$$Y^{t+1} \leftarrow \varphi\, T\, Y^t + (1-\varphi)\, Y \quad (6)$$

where φ is used to control the percentage of label values which are obtained from neighbors in each iteration. If φ approaches 0, it means label values are obtained from the neighbors gradually (iteration after iteration or step by step); if it approaches 1, it means that the node's label is decided by its neighbors completely.

Step 5: Second iterative step in label matrix normalization: This step fixes the label matrix by changing back the values of assigned and unassigned labels to their initial state. This means that for any $l_{i,k}$ which has a value of 1 or 0 in the original label matrix and whose value changed in the previous iteration, it returns to its original value of 1 or 0. Only missing labels that were originally $\frac{1}{2}$ in the original label matrix whose values are kept change gradually in the iterative process. At the end of the iteration process, for each missing label (assigned in the original matrix), if its final value approaches 1, the missing label is assigned a 1; if its value approaches 0, the missing label is assigned a 0.

Algorithm 1: GB-AS

Input:
D = [$\boldsymbol{X_1}$, $\boldsymbol{X_2}$,…., $\boldsymbol{X_N}$], a dataset of N examples
Y: The Initial label matrix for N instances
k: The number of nearest neighbors used in Equations (2) and (4)
α: Parameter used in Equation (6)
Output: Y labeled matrix
//Build Graph
Step 1: W ← build a graph () using either Equations (2), (4) or (5)
Step 2: T ← obtain_ Transition matrix (W)
Step 3: initialize the label matrix

$$\boldsymbol{Y^0} \leftarrow \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{l_{11}} & \vdots & \boldsymbol{l_{1z}} \\ \boldsymbol{l_{21}} & \vdots & \boldsymbol{l_{2z,}} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \boldsymbol{l_{N1}} & \vdots & \boldsymbol{l_{Nz}} \end{bmatrix}$$

//label propagation
While Y not convergence do
step 4: $\boldsymbol{Y^{t+1}} \leftarrow \boldsymbol{\varphi\, T\, Y^t} + (\boldsymbol{1-\varphi})\ \boldsymbol{Y^t}$
step 5: $\boldsymbol{Y^{t+1}} \leftarrow$ Normalize ($\boldsymbol{Y^{t+1}}$)
End while

End

Once GB-AS has been constructed (as in Figure 1), GB-AS can be used as an input in UG-MLP. The following describes, in detail, how GB-AS is used in UG-MLP, as well as how UG-MLP predicts missing labels.

4. Unified Graph-Based Missing Label Propagation (UG-MLP)

Unlike single-label text classification methods, multilabel text classification assigns more than one label to each instance, and these labels are frequently related. Existing methods for handling missing labels in multilabel learning are built based on the assumption that missing label information of an instance can be propagated from its k-nearest neighbors [3,13]. Some of these methods use the first-order label correlation exploitation strategy which ignores label correlations [2–4].

Different from most of the existing algorithms [2,4,13], in multilabel classification with missing labels, this work defines the missing label recovery as a function of both (1) information propagated from its k-nearest neighbors based on the feature space-based similarity and label distribution-based similarity to the label space, and (2) label correlation and label-specific feature learning, which are combined into a single framework. A high-order label correlation is learned from the incomplete training data and applied to augment the missing label matrix and guide the construction of multilabel classification models.

To address the missing label problem, this work proposes a merged model of missing label handling by assembling a mixed graph, as shown in Figure 3. The model jointly incorporates (1) the instance-level feature space-based similarity and the label distribution-based similarity, GB-AS, and (2) accurate label correlations. The following describes the unified graph-based model for missing label handling based on nearest neighbor feature and label similarity and label correlation methods:

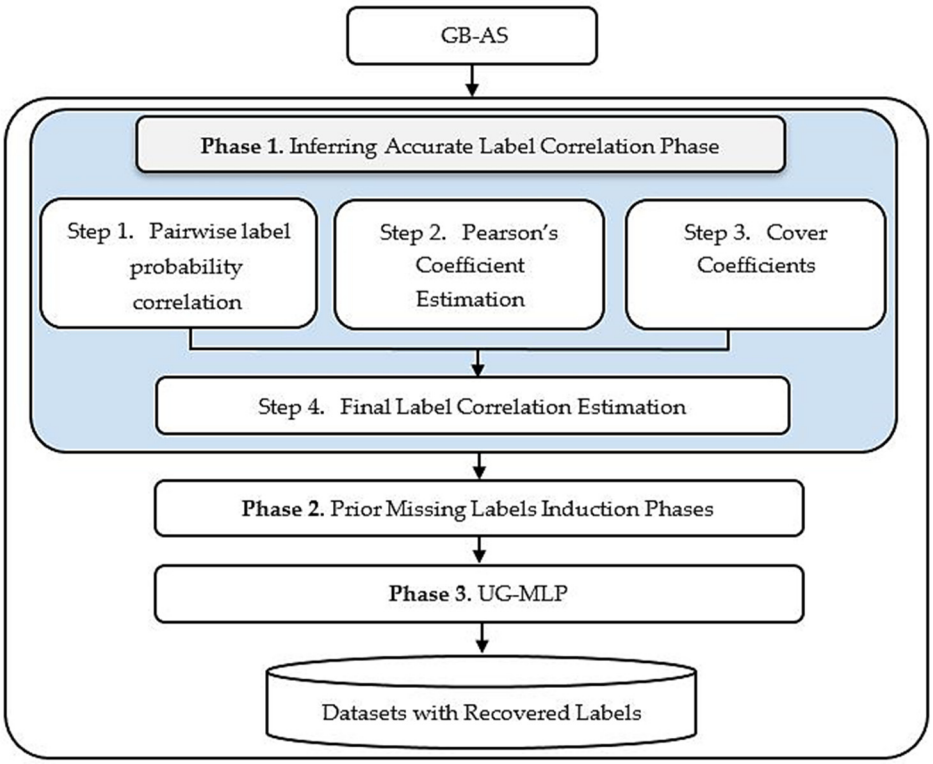

Figure 3. UG-MLP method based on GB-AS and accurate label correlations.

The input of the UG-MLP method shown in Figure 3 is the datasets with missing labels, and its output is datasets with recovered labels. In addition, Figure 3 illustrates the overall architecture of graph-based integrated information missing label propagation and handling methods in multilabel text classification. The following describes, in detail, each of the phases mentioned in Figure 3.

4.1. Phase 1. Inferring Accurate Label Correlation Phase

It is essential and crucial to use the label correlation to handle the missing labels in multilabel text classification, and in multilabel text classification in general. In real-world multilabel classification, labels can have strong interdependencies, and some of them may even be missing. Based on this, if two or more labels have strong interdependencies, i.e., they frequently co-occur in many cases, in the new case, this strong interdependency information can be used to predict if one of them is missing. In this phase, the label correlation matrix is obtained using the following steps:

Step 1: Pairwise label probability correlation estimation: A pairwise label probability correlation from the dataset is obtained by calculating the probability of pairwise labels [4]. The pairwise label probability correlation is defined as the conditional probability of a label given another label, as shown in Equation (7):

$$p(l_i|l_j) = \frac{T(l_i, l_j) + s}{T(l_j) + 2s} \; T(l_j) \neq 0 \tag{7}$$

where l_i, l_j are the two labels from the label set, $T(l_j)$ is the number of document instances with the label l_j and $T(l_i, l_j)$ represents the number of document instances that simultaneously have both labels l_i and l_j. $s > 0$ is the smoothness parameter. It is important to note that $p(l_i|l_j)$ label correlations are not asymmetric.

Step 2: Pearson's coefficient estimation: In this step, the label correlation asymmetry from the dataset is obtained by calculating Pearson's correlation coefficient [27]. The input to the algorithm is a $z \times n$ Boolean matrix, E, whose rows are indexed by the set of labels, $L = \{l_1, , \ldots, l_m\}$, and columns by the elements of a set of text instances, $X = \{r_1, \ldots, r_n\}$. If label l_i is assigned to document x_u, $E_{i,u} = 1$. The input matrix is obtained from the label space representation. The following matrix is an example of a Boolean matrix, E, as shown in Table 1.

Table 1. Example of a Boolean matrix, E.

		x1	x2	x3	x4	x5	x6
E=	l1	1	0	1	0	0	0
	l2	0	0	0	1	1	0
	l3	0	0	1	1	1	0
	l4	1	0	0	0	1	0
	l5	0	1	1	1	0	0
	l6	0	0	1	1	0	1
	l7	1	1	0	0	1	1

Given the label matrix E, the label correlation is defined as the correlation coefficient, r (also known as Pearson's coefficient), as shown in Equation (8) [27]:

$$r(l_i, l_j) = \frac{\sum_{k=1}^{n} (l_{i,k} - \overline{l_i})(l_{j,k} - \overline{l_j})}{n\sigma^{l_i}\sigma^{l_j}} \; T(l_j) \neq 0 \tag{8}$$

where n is the size of the dataset, $l_{i,k}$ and $l_{j,k}$ are the values of l_i and l_j assigned to instance documents $x_k, \overline{l_i}$ and $\overline{l_j}$ are the averages of values of l_i and l_j, respectively, and $\sigma^{l_i}\sigma^{l_j}$ are the respective standard deviations of l_i and l_j.

Step 3: Cover coefficients: As in step 2, the input to the algorithm is a $z \times n$ Boolean matrix, E. The entries of C denote pairwise cover coefficient values among the labels. The

cover coefficient measure between two labels l_i and l_j is the probability that a text x_u labeled by l_i is also labeled by l_j. Informally, the cover coefficient of a label with respect to another denotes the extent to which the assignment profile of the first label is covered by that of the second one [28].

Let ϖ_i be the reciprocal of the sum of the entries in the ith row and δ_u the reciprocal of the sum of the entries in the uth row column of the E matrix. The cover coefficient Γ_{ij} between labels l_i and l_j is obtained using the following formula in Equation (9):

$$\Gamma_{ij} = \varpi_i \sum_{i=1}^{n} \left(E_{ik} \times \delta_u \times E_{kj}\right) \quad (9)$$

Step 4: Final label correlation estimation: In this work, the final label correlation, LC, is calculated using the three labels to label relation measures, namely, pairwise label probability correlation, $p(l_i|l_j)$ (see step 1), Pearson's coefficient, $r(l_i, l_j)$ (see step 2), and cover coefficient correlation coefficient, Γ_{ij} (see step 3), based on the following aggregated label correlation function (see Equation (10)):

$$LC_{ij} = \lambda_1 p(l_i|l_j) + \lambda_2 \, \mathrm{r}(\mathrm{l_i}, \mathrm{l_j}) + \lambda_3 \, \Gamma_{ij} \quad (10)$$

where $\lambda_1 = \lambda_2 = \lambda_3$, and $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

Finally, a final label correlation matrix LC is obtained:

$$LC = \begin{bmatrix} Lc_{11} & \vdots & Lc_{1z} \\ Lc_{21} & \vdots & Lc_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ Lc_{z1} & \vdots & Lc_{zz} \end{bmatrix}$$

This is used in Equation (11) described in phase 2 to estimate the prior induction of these missing labels, $\breve{Y}(i,j)$, where $\breve{Y}(i,j) = Lc_{i1}Y_{1,j} + Lc_{i2}Y_{2,j} + \cdots . + Lc_{in}Y_{1,n}$.

4.2. Phase 2. Prior Missing Label Induction Phase

This phase uses the output from phase 1, which is the label correlation matrix, to give a prior estimation of the missing labels in the label matrix Y.

Based on the problem definition, the label matrix Y of multilabel text classification with missing labels is $Y = \begin{bmatrix} l_{11} & \vdots & l_{1z} \\ l_{21} & \vdots & l_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ l_{N1} & \vdots & l_{Nz} \end{bmatrix}$, where $Y_{ij} = l_{ij} \in \left\{1, 0, \frac{1}{2}\right\}$. If l_{ij} is 1, it means document x_i belongs to class l_j; if l_{ij} is 0, it means document x_i does not belong to class l_j; if l_{ij} is $\frac{1}{2}$, it means the label is missing, and that it is unknown if document x_i belongs to class l_j or not.

In this phase, a prior induction of these missing labels in the label matrix Y can be carried out by estimating the likelihood of a missing label l_j for document x_i based on Equation (11) [18]:

$$\breve{Y}(i,j) = \begin{cases} \sum_{k=1}^{n} LC_{i,k} \, Y_{k,j} \; If \; Y_{i,j} = \frac{1}{2} \\ \quad \ldots \\ Y_{i,j} \; otherwise \end{cases} \quad (11)$$

By making sure $\breve{Y}(i,j) \in [0,1]$, $\breve{Y}(i,j)$ is normalized as $\breve{Y}(i,j) = \breve{Y}(i,j) / \sum_{k=1}^{n} \breve{Y}(i,k)$, Equation (11) estimates the missing label of an incompletely annotated document using the already known labels of the document and the label correlation, where $\breve{Y}(i,j) = Lc_{i1}Y_{1,j} + Lc_{i2}Y_{2,j} + \cdots . + Lc_{in}Y_{1,n}$. For example, if the label is missing but it has large correlations with the labels already labeled for the *i*th document, then it may be assigned a

large value. The final output of this phase is a new label matrix $\check{Y}$ with a prior prediction of missing values.

$$Y = \begin{bmatrix} l_{11} & \vdots & l_{1z} \\ l_{21} & \vdots & l_{2z,} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ l_{N1} & \vdots & l_{Nz} \end{bmatrix} \Longrightarrow \check{Y} = \begin{bmatrix} \widetilde{l_{11}} & \vdots & \widetilde{l_{1z}} \\ \widetilde{l_{21}} & \vdots & \widetilde{l_{2z,}} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \widetilde{l_{N1}} & \vdots & \widetilde{l_{Nz}} \end{bmatrix}$$

The final output of this phase is a new label matrix $\check{Y}$ with a prior prediction of missing values. It is used in the missing label prediction iterative step, Equation (12).

4.3. Phase 3. UG-MLP Phase

In this phase, UG-MLP is implemented. UG-MLP addresses the missing label problem by jointly incorporating (i) the instance-level feature space-based similarity and label distribution-based similarity and (ii) accurate label correlations. The following describes, in more detail, the steps of the UG-MLP algorithm (see Algorithm 2):

Step 1: Missing label prediction iterative step: In this step, the new label matrix is calculated using the following Equation (12) [10]:

$$\check{Y}^{t+1} \leftarrow \varphi T \check{Y}^{t} + (1-\varphi)\tilde{Y}^{t} \tag{12}$$

where φ is used to control the percentage of label values which are obtained from neighbors in each iteration. If α approaches 0, it means label values are obtained from neighbors iteratively (slowly). If α approaches 1, it means that the node's label is decided by its neighbors completely.

Step 2: Second iterative step in label matrix normalization: This step fixes the label matrix by changing back the values of assigned and unassigned labels to their initial state. This means that for any $l_{i,k}$ which has a value of 1 or 0 in the original label matrix and whose value changed in the previous iteration, it returns back to its original value of 1 or 0. Only missing labels that were originally $\frac{1}{2}$ in the original label matrix whose values were kept change gradually in the iterative process. At the end of the iteration process, for each missing label (assigned in the original matrix), if its final value approaches 1, it is assigned a 1; if its value approaches 0, it is assigned a 0.

Algorithm 2: UG-MLP

Input:
D = [$\boldsymbol{X_1}$, $\boldsymbol{X_2}$,…., $\boldsymbol{X_N}$], a dataset of N examples
Y: The Initial label matrix for N instances
k: The number of nearest neighbors used in Equations (2) and (4)
α: Parameter used in Equation (6)
Output: Y labeled matrix
//Build Graph phase
W ← build a graph () using either Equations (2), (4) or (5)
T ← obtain_ Transition_matrix (W)
//initialize the label matrix

$$\boldsymbol{Y^0} \leftarrow \boldsymbol{Y} = \begin{bmatrix} \boldsymbol{l_{11}} & \vdots & \boldsymbol{l_{1z}} \\ \boldsymbol{l_{21}} & \vdots & \boldsymbol{l_{2z,}} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \boldsymbol{l_{N1}} & \vdots & \boldsymbol{l_{Nz}} \end{bmatrix}$$

Phase 1//Inferring Accurate Label Correlation Phase

For each label $\boldsymbol{l_i}$ do

For each label $\boldsymbol{l_j}$ do

p($\boldsymbol{l_i} \mid \boldsymbol{l_j}$) = Pairwise_probability_correlation($\boldsymbol{l_i}$, $\boldsymbol{l_j}$)

r($\boldsymbol{l_i}, \boldsymbol{l_j}$) = Pearson_correlation($\boldsymbol{l_i}, \boldsymbol{l_j}$)

$\boldsymbol{\Gamma_{ij}}$ = cover _Coefficient_correlation ($\boldsymbol{l_i}, \boldsymbol{l_j}$)

$\boldsymbol{LC_{ij}} = \boldsymbol{\lambda_1}$ p($\boldsymbol{l_i} \mid \boldsymbol{l_j}$)+ $\boldsymbol{\lambda_2}$ r($\boldsymbol{l_i}, \boldsymbol{l_j}$)+ $\boldsymbol{\lambda_3}$ $\boldsymbol{\Gamma_{ij}}$ using Equation(10)

End For

End for

Phase 2//Prior Missing Labels Induction Phases

For each document $\boldsymbol{X_i}$ do

For each label $\boldsymbol{Y_{i,j}} \in$ labels of $\boldsymbol{X_i}$ and $\boldsymbol{Y_{i,j}} \in$Y do

IF($\boldsymbol{Y_{i,j}}$ = 0.5)

$\breve{\boldsymbol{Y}}(i,j)$ = Estimate_Prior_Info($\boldsymbol{Y_{i,j}}$)

Eles

$$\boldsymbol{Y_{i,j}} = \boldsymbol{Y_{i,j}}$$

End if

End For

$$\breve{\boldsymbol{Y}} = \begin{bmatrix} \widetilde{\boldsymbol{l_{11}}} & \vdots & \widetilde{\boldsymbol{l_{1z}}} \\ \widetilde{\boldsymbol{l_{21}}} & \vdots & \widetilde{\boldsymbol{l_{2z,}}} \\ \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots \\ \widetilde{\boldsymbol{l_{N1}}} & \vdots & \widetilde{\boldsymbol{l_{Nz}}} \end{bmatrix}$$

Phase 3//Graph-based Missing Label Propagation Phase

While Y not convergence do

$\breve{\boldsymbol{Y}}^{\boldsymbol{t+1}} \leftarrow \varphi \ \boldsymbol{T}\, \breve{\boldsymbol{Y}}^{\boldsymbol{t}} + (1-\varphi)\, \widetilde{\boldsymbol{Y}}^{\boldsymbol{t}}$ using Equation (12)

$\breve{\boldsymbol{Y}}^{\boldsymbol{t+1}} \leftarrow$ Normalize ($\breve{\boldsymbol{Y}}^{\boldsymbol{t+1}}$)

End while

End

5. Implementation of GB-AS and UG-MLP in Multilabel Text Classification

In order to prove whether the proposed GB-AS and UG-MLP methods are effective in solving multilabel classification with missing or incomplete labels, we implemented both GB-AS and UG-MLP in multilabel text classification. The following Figure 4 illustrates the overall architecture of all the stages in implementing the GB-AS and UG-MLP methods, which include the following: (1) pre-processing phase, missing label handling or missing label recovery phase, (2) ensemble feature selection phase, (3) multilabel text classification phase, (4) evaluation phase.

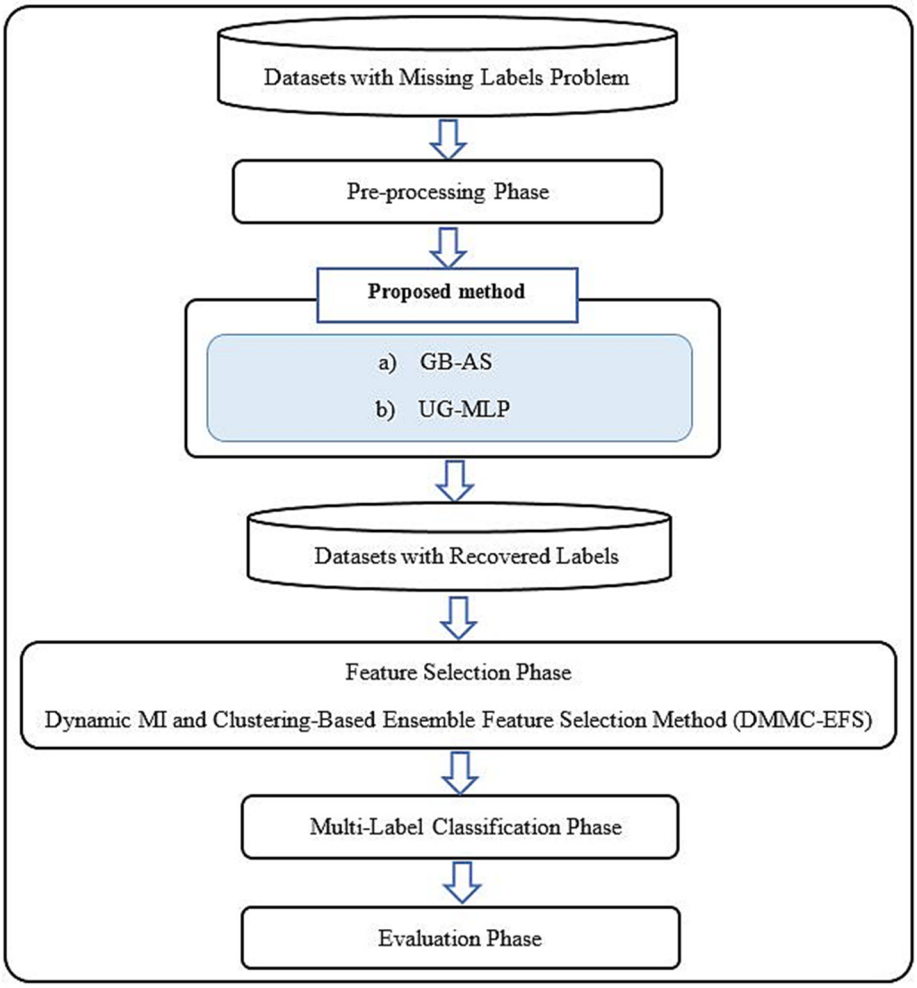

Figure 4. The overall architecture of multilabel text classification with missing label handling (GB-AS and/or UG-MLP).

5.1. Pre-Processing

Pre-processing is a necessary step before implementing machine learning techniques. It includes four steps, namely, (1) tokenization, (2) normalization, (3) stop word removal and (4) stemming. To begin, tokenization attempts to convert a document's text into a machine learning-friendly structure. The tokenization method entails converting a text into discrete fragments separated by a space or a specific indicator, with each unit matching a single word. Next, the normalization stage seeks to clean the data by removing noise and undesirable data such as special characters. After that, the stop word task is used to eliminate superfluous words such as conjunctions, pronouns and prepositions. Finally, stemming is the process of determining the root or stem of a word. Stemming is a crucial step for dealing with high-dimensional and sparse data, especially with multilabel text data classification, because it isolates the word's root form from its inflectional or derivational variants.

5.2. Multilabel Two-Layer MI and Clustering-Based Ensemble Feature Selection Method (DMMC-EFS)

Feature selection (FS) methods improve the performance of text classification tasks in terms of their learning speed and efficacy. FS methods also decrease the number of data dimensions and eliminate data that are useless, unnecessary or noisy. In order to reduce the dimensionality and to improve the classification performance, this work used a dynamic two-layer MI and clustering-based ensemble feature selection (DMMC-EFS)

method suggested by [29] to select features with strong class discrimination ability. The DMMC-EFS method considers the (1) dynamic global weight of features, (2) heterogeneous ensemble and (3) maximum dependency and relevancy and minimum redundancy of features. This method aims to overcome the high dimensionality of multilabel datasets and achieve a superior multilabel text classification performance, through Equation (13) and Equation (14):

$$SFWij = \frac{(fw(i,j) * ASWj * ai)}{ASW_i} \quad (13)$$

$$DSWj = \frac{DSWj}{sumSFj} \quad (14)$$

5.3. Multilabel Classification Model

For evaluation, the AdaBoost.MH multilabel learning model was used in this work. The AdaBoost.MH model was selected because it is a state-of-the-art multilabel classification algorithm that is frequently utilized in multilabel text classification studies [29,30]. AdaBoost.MH iteratively builds several weak classifiers before grouping them into a final classifier that can estimate multiple labels for a given occurrence. Boosting algorithms, such as the AdaBoost adaptive booster, transform a weak classifier into a strong one through integration and training. The AdaBoost algorithm can change the weight distribution of training data and consistently select the best weak classifier from the sample weight distribution. The AdaBoost algorithm can adaptively alter the weight distribution of training data and consistently select the best weak classifier from the sample weight distribution to integrate all weak classifiers and vote by a specific weight to produce a classifier model. The AdaBoost.MH algorithm is a multilabel variant of the AdaBoost algorithm [31].

6. Experiments

6.1. Multilabel Text Dataset

Three datasets were used in this study: Bibtex, Enron and Reuters-21578, which are described in Table 2. They are publicly available datasets for multilabel text classification problems. The values of cardinality, instances, labels, attributes and average imbalance ratio per label (avgIR) are displayed. The cardinality is used to measure the average number of classes concerning each instance. As for the density, it is calculated by dividing the cardinality by the total sum of labels. These datasets can be downloaded from the Mulan website [32].

Table 2. Summary description of the multilabel text classification datasets.

Dataset	Instances	Attributes	Classes	Cardinality	Density	Diversity	avgIR
Bibtex	7395	1836	159	2.402	0.015	0.386	12.498
Enron	1702	1001	53	3.378	0.064	0.442	73.953
Reuters-21578	6000	500	103	1.462	0.014	0.135	54.081

6.2. Evaluation Metric and Experiment Setup

The results of the experiment on multilabel classification were measured using the following three evaluation metrics: precision, recall and F-measure, using Equations (15)–(17), respectively. In this domain, these evaluation metrics are well known for drawing comparisons [29,33–35].

$$\text{M\_PRECISION} = \sum_{i=1}^{d} \frac{\text{TP}_\text{i}}{\text{PT}_\text{i} + \text{FP}_\text{i}} \tag{15}$$

$$\text{M\_RECALL} = \sum_{i=1}^{d} \frac{\text{TP}_\text{i}}{\text{PT}_\text{i} + \text{FP}_\text{i}} \tag{16}$$

$$\text{M}_{\text{F}\beta} = \sum_{i=1}^{d} \frac{(\beta^2 + 1)\text{Pr} \times \text{Re}}{\beta^2 \text{Pr} + \text{Re}} \tag{17}$$

To verify the effectiveness of UG-MLP as well as GB-AS, this study used the Reuters-21578, Bibtex and Enron datasets. Experiments were conducted with the three multilabel datasets under various missing percentages. Thus, the percentage of missing labels was set as 10%, 20%, 30%, 40% and 50%, as suggested in [4]. In particular, when there is a 0% missing proportion, this indicates that the label matrix is full. The label structure is degraded to a larger extent as the missing percentage rises.

In the experiments, all the datasets were set up to have percentages of missing labels of 10%, 20%, 30%, 40% and 50%. The multilabel classification followed the diagram shown in Figure 3 and was used to classify the datasets. To see the effectiveness of GB-AS and UG-MLP separately, we set up three types of multilabel classifications. First, as a baseline, both GB-AS and UG-MLP were not applied. Thus, there was no label recovery mechanism used in the multilabel classification. In other words, the baseline multilabel classification just implements the feature selection DMMC-EFS, and thus we named the baseline as DMMC-EFS. The second approach applied only GB-AS to recover labels and DMMC-EFS as the FS. This was to assess the effectiveness of only using aggregated feature and label information. The third approach applied UG-MLP to recover labels and DMMC-EFS as the FS. The third approach is a label recovery mechanism that not only has aggregated feature and label information (as in GB-AS) but also autocorrelation labels. In other words, UG-MLP should hypothetically be able to classify the dataset better than the other approaches despite how high the percentage of missing labels is.

6.3. Evaluation Metric and Experiment Setup

The results obtained (F-measure) for DMMC-EFS after label recovery with one of the four missing label handling methods are shown in Table 2 and Figure 4. Based on the results of this experiment, almost the same observations were made: The incompleteness of class labels significantly influences the performance of multilabel classifiers, and these approaches to modeling missing labels offer a better performance than DMMC-EFS in most cases. Since DMMC-EFS does not deal with missing labels, its performance degrades rapidly as the missing rate rises. The performance of missing label handling methods, GB-AS and UG-MLP, declines relatively slow with the increase in the missing rate. As expected, the UG-MLP approach outperforms the other methods is due to the recovery of the missing labels by exploiting label correlations. Additionally, the results of two state-of-the-art methods (i.e., LM-MTC [17] and Rec-Conv-Cap [19]) that were reviewed in Section 2 are also compared in Table 2. These results are discussed in the next section.

7. Discussion

From the obtained results (see Tables 3–5) regarding the evaluation and comparison of missing label handling methods, i.e., GB-AS and UG-MLP, on all the datasets, the following important observations were made: The missing class label problem influences the performance of multilabel text classification. The performance of good multilabel classifiers which achieve high results with complete label datasets such as the multilabel classifier with DMMC-EFS decreases rapidly as the missing rate increases. Therefore, to preserve their good performance, a multilabel text classification model should have a missing label handling method. Meanwhile, the performance of multilabel learning with DMMC-

EFS, which handles missing label problems using missing label handling methods, i.e., GB-AS and UG-MLP, declines somewhat relative slowly with the increase in the missing rate. It can be noticed that the results for the Bibtex dataset are better than those for the Reuters-21578 dataset. In addition, comparing the proposed methods against LM-MTC [17] and Rec-Conv-Cap [19] using the Reuters-21578 dataset, it can be seen that, on average, the proposed methods achieve a better performance. This difference in performance could be because the other methods do not utilize unified graphs. It is also worth noting that the performance of some of the algorithms used for comparison (especially DMMC-EFS) declines quickly for the datasets with a high class imbalance ratio (see Table 2). The problem of missing labels might be affected directly by the balance of the classes. On the other hand, solving the missing label problem may also cause class imbalance as they are related. In other words, if a dataset initially has a high class imbalance ratio, it will possibly become worse after solving the missing label problem. Working on such an issue is promising; hence, it can be considered as future work.

Table 3. Performance (F-measure) of the GB-AS and UG-MLP methods on the Reuters-21578 dataset.

	Label Missing Rate				
	10%	**20%**	**30%**	**40%**	**50%**
DMMC-EFS (baseline)	80.76	75.74	72.22	66.31	60.38
LM-MTC [17]	82.1	79.35	77.37	72.6	69.90
Rec-Conv-Cap [19]	85.3	80.02	78.8	75.32	73.2
GB-AS + DMMC-EFS	86.21	82.42	81.71	79.95	77.92
UG-MLP + DMMC-EFS	87.34	86.31	84.38	82.37	80.63

Table 4. Performance (F-measure) of the GB-AS and UG-MLP methods on the Bibtex dataset.

	Label Missing Rate				
	10%	**20%**	**30%**	**40%**	**50%**
DMMC-EFS (baseline)	82.14	78.2	75.58	69.99	64.86
GB-AS + DMMC-EFS	85.31	83.45	82.89	81.4	79.63
UG-MLP + DMMC-EFS	89.85	87.88	85.13	83.9	82.14

Table 5. Performance (F-measure) of the GB-AS and UG-MLP methods on the Enron dataset.

	Label Missing Rate				
	10%	**20%**	**30%**	**40%**	**50%**
DMMC-EFS (baseline)	83.43	79.22	76.09	70.99	65.04
GB-AS + DMMC-EFS	86.93	86.18	83.73	82.13	80.58
UG-MLP + DMMC-EFS	90.22	88.54	86.81	85.35	84.07

The superb performance of the proposed UG-MLP against GB-AS indicates the effectiveness of unifying label space recovery with nearest neighbor similarity learning and the superiority of label space recovery by exploiting sparse high-order label correlations. The superb performance of the proposed UG-MLP also indicates the effectiveness of the proposed method of solving multilabel learning with missing labels. From the line graphs in Figures 5–7, the results obtained using the proposed UG-MLP outperform those obtained using GB-AS on all datasets. The proposed UG-MLP behaves in a similar way in all datasets regarding its performance with the increase in the missing rate.

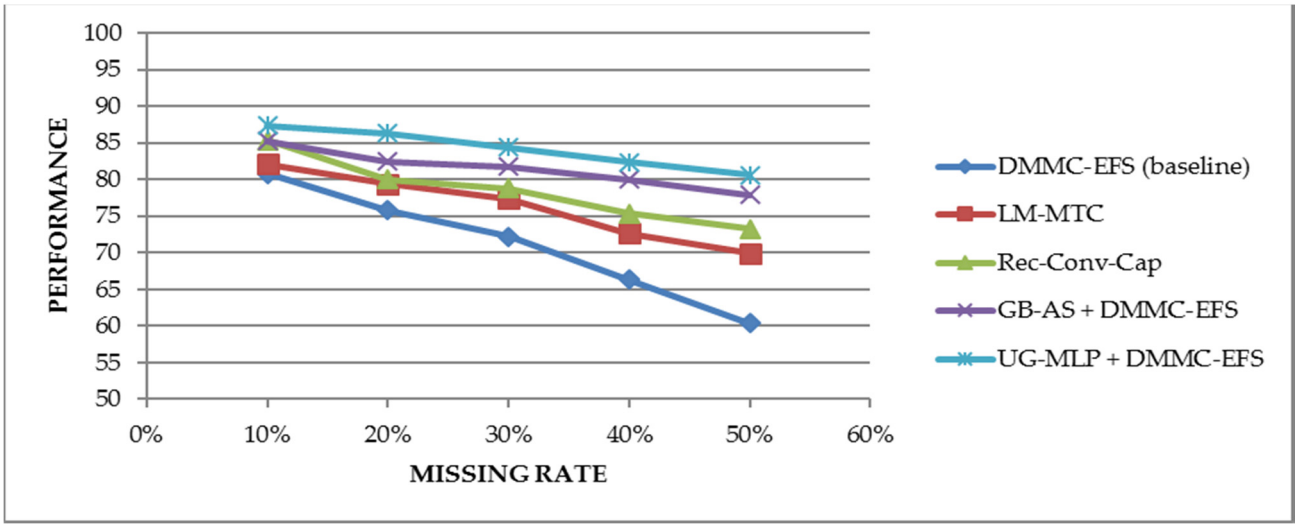

Figure 5. Performance (F-measure) of the GB-AS and UG-MLP methods on the Reuters-21578 dataset.

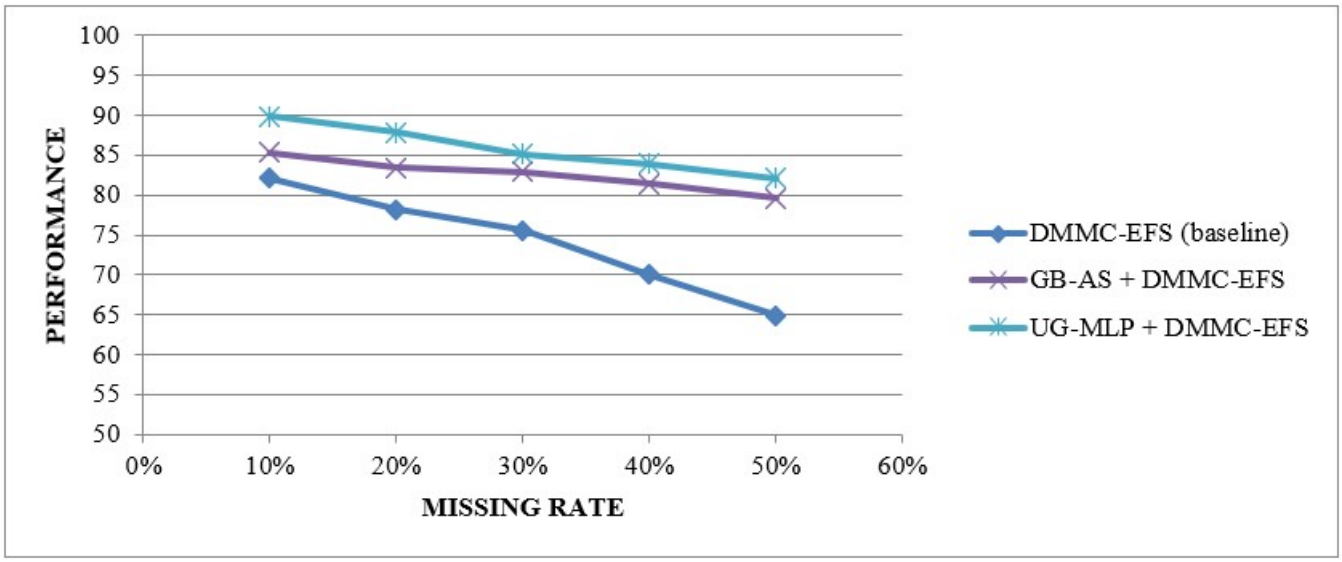

Figure 6. Best performance (F-measure) of the GB-AS and UG-MLP methods on the Bibtex dataset.

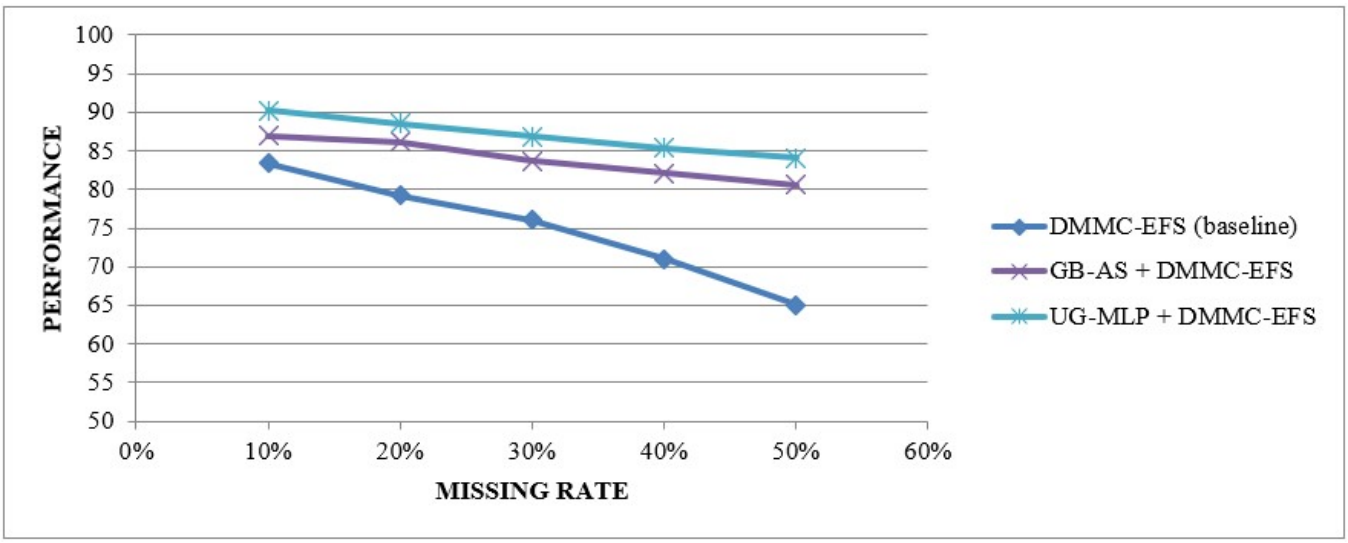

Figure 7. Best performance (F-measure) of the GB-AS and UG-MLP methods on the Enron dataset.

The solutions provided in this work might be useful in a variety of applications as they show a performance increase compared to the baseline. Organizations that manage text files such as those in health may utilize the solutions provided in this study.

8. Conclusions

This paper presented a scalable multilabel text classification method to handle missing label and label correlation problems of multilabel datasets. This paper designed several missing label prediction methods for multilabel feature selection. First, this paper introduced the GB-AS method for multilabel text classification. Then, this paper proposed a new method, UG-MLP, for multilabel text classification that considers unifying label space recovery with nearest neighbor similarity learning and the superiority of label space recovery by exploiting sparse high-order label correlations. Based on the obtained results, the performance of the missing label prediction method UG-MLP indicates its effectiveness in solving multilabel learning with missing labels. In light of this, the reduction in the missing labels has a direct impact on the performance of text classification in the multilabel domain problem. However, the computational complexity of the proposed methods is higher than that of the baseline methods as the running time has increased notably. This may be considered as a limitation, and investigating it is suggested as future work.

Author Contributions: Conceptualization, A.Y.T. and S.T.; methodology, A.Y.T.; software, A.Y.T.; validation, A.Y.T. and S.T.; formal analysis, A.Y.T.; investigation, A.Y.T., S.T., and A.S.A.; resources, A.Y.T., S.T., and A.H.A.R.; data curation, A.Y.T. and S.T.; writing—original draft preparation, A.Y.T.; writing—review and editing, S.T.; visualization, A.Y.T.; supervision, S.T., A.H.A.R., and M.A.; project administration, S.T.; funding acquisition, S.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Higher Education Malaysia under research code: FRGS/1/2020/ICT02/UKM/02/1.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

BERT + MLM	BERT + Masked Language Model
BM-LRC	Bayesian Model with Label Regularization and Label Confidence constraints
CC	Classifier Chains
CDM	Correlation Distance Matrix
CNN	Convolutional Neural Network
CNN-RNN	Convolutional Neural Network-Recurrent Neural Network
MLMF	Missing Labels and Multilabel Feature Selection
Conv	Convolutional Networks
Conv-Cap	Convolutional Capsule Network
CORALS	Cost-Sensitive Label Ranking Approach with Low-Rank and Sparse Constraints
DMMC-EFS	Dynamic Two-Layer MI and Clustering-Based Ensemble Feature Selection
GB-AS	Graph-Based Aggregated Similarity Weighting of Features and Labels Method
GB-FS	Graph-Based Missing Label Handling with Feature Similarity Weighting
GB-LS	Graph-Based Missing Label Handling with Label Similarity Weighting
HAN + LG	Hierarchical Attention Network + Label Graph
HAN	Hierarchical Attention Network
LM-MTC	Label Mask Multilabel Text Classification
LSML	Label-Specific Features for Multilabel Classification with Missing Labels
LW	Label-Wise
MEGNET	Multilabel Text Classification using Attention-Based Graph Neural Network
MGFS	Graph-Based Multilabel Feature Selection

MLM	Masked Language Model
MLMF	Missing Labels and Multilabel Feature Selection
Rec-Conv	Recurrent Convolutional Network

References

1. Braytee, A. Robust Classification of High Dimensional Unbalanced Single and Multi-Label Datasets. Ph.D. Thesis. University of Technology, Sydney, Australia, February 2018.
2. Xu, T.; Zhao, L. A Structure-Induced Framework for Multi-Label Feature Selection with Highly Incomplete Labels. *IEEE Access* **2020**, *8*, 71219–71230. https://doi.org/10.1109/access.2020.2987922.
3. Tan, Q.; Yu, Y.; Yu, G.; Wang, J. Semi-supervised multi-label classification using incomplete label information. *Neurocomputing* **2017**, *260*, 192–202. https://doi.org/10.1016/j.neucom.2017.04.033.
4. Huang, J.; Qin, F.; Zheng, X.; Cheng, Z.; Yuan, Z.; Zhang, W.; Huang, Q. Improving multi-label classification with missing labels by learning label-specific features. *Inf. Sci.* **2019**, *492*, 124–146. https://doi.org/10.1016/j.ins.2019.04.021.
5. Ma, J.; Tian, Z.; Zhang, H.; Chow, T.W. Multi-Label Low-dimensional Embedding with Missing Labels. *Knowledge-Based Syst.* **2017**, *137*, 65–82. https://doi.org/10.1016/j.knosys.2017.09.005.
6. Ma, J.; Chow, T.W. Topic-based algorithm for multilabel learning with missing labels. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 2138–2152.
7. Guan, Y.; Li, X. Multilabel Text Classification with Incomplete Labels: A Safe Generative Model with Label Manifold Regularization and Confidence Constraint. *IEEE MultiMedia* **2020**, *27*, 38–47.
8. Wu, B.; Jia, F.; Liu, W.; Ghanem, B.; Lyu, S. Multi-label learning with missing labels using mixed dependency graphs. *Int. J. Comput. Vis.* **2018**, *126*, 875–896.
9. Ibrahim, K.M.; Epure, E.V.; Peeters, G.; Richard, G. Confidence-based Weighted Loss for Multi-label Classification with Missing Labels. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 8–11 June 2020; pp. 291–295.
10. He, Z.-F.; Yang, M.; Gao, Y.; Liu, H.-D.; Yin, Y. Joint multi-label classification and label correlations with missing labels and feature selection. *Knowl.-Based Syst.* **2019**, *163*, 145–158. https://doi.org/10.1016/j.knosys.2018.08.018.
11. Pal, A.; Selvakumar, M.; Sankarasubbu, M. MAGNET: Multi-Label Text Classification using Attention-based Graph Neural Network. In Proceedings of the 12th International Conference on Agents and Artificial Intelligence, Valletta, Malta, 22–24 February 2020; pp. 494–505. https://doi.org/10.5220/0008940304940505.
12. Zhu, P.; Xu, Q.; Hu, Q.; Zhang, C.; Zhao, H. Multi-label feature selection with missing labels. *Pattern Recognit.* **2018**, *74*, 488–502. https://doi.org/10.1016/j.patcog.2017.09.036.
13. Ma, J.; Chow, T.W. Label-specific feature selection and two-level label recovery for multi-label classification with missing labels. *Neural Netw.* **2019**, *118*, 110–126.
14. Wang, C.; Lin, Y.; Liu, J. Feature selection for multi-label learning with missing labels. *Appl. Intell.* **2019**, *49*, 3027–3042. https://doi.org/10.1007/s10489-019-01431-6.
15. Hashemi, A.; Dowlatshahi, M.B.; Nezamabadi-pour, H. MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality. *Expert Syst. Appl.* **2020**, *142*, 113024.
16. Zhao, F.; Guo, Y. Semi-Supervised Multi-Label Learning with Incomplete Labels. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 27 July 2015.
17. Song, R.; Chen, X.; Liu, Z.; An, H.; Zhang, Z.; Wang, X.; Xu, H. Label Mask for Multi-Label Text Classification. *arXiv* **2021**, arXiv:2106.10076.
18. Li, Y.; Yang, Y. Label Embedding for Multi-label Classification Via Dependence Maximization. *Neural Process. Lett.* **2020**, *52*, 1651–1674. https://doi.org/10.1007/s11063-020-10331-7.
19. Yan, S. Enhancing Deep Learning-Based Multi-label Text Classification with Capsule Network. In *Journal of Physics: Conference Series*; IOP Publishing: Hangzhou, China, 2020; Volume 1621, p. 012037. https://doi.org/10.1088/1742-6596/1621/1/012037.
20. Nguyen, D.T.; Chen, L.; Chan, C.K. Clustering with Multiviewpoint-Based Similarity Measure. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 988–1001. https://doi.org/10.1109/tkde.2011.86.
21. Baeza-Yates, R.; Ribeiro-Neto, B. *Modern Information Retrieval*; ACM Press: New York, NY, USA, 15 May 1999.
22. Fallahpour, A.; Wong, K.Y.; Rajoo, S.; Fathollahi-Fard, A.M.; Antucheviciene, J.; Nayeri, S. An integrated approach for a sustainable supplier selection based on Industry 4.0 concept. *Environ. Sci. Pollut. Res.* **2021**, 1–19. https://doi.org/10.1007/s11356-021-17445-y.
23. Vluymans, S.; Cornelis, C.; Herrera, F.; Saeys, Y. Multi-label classification using a fuzzy rough neighborhood consensus. *Inf. Sci.* **2018**, *433*, 96–114. https://doi.org/10.1016/j.ins.2017.12.034.
24. Wang, W.; Tian, G.; Zhang, T.; Jabarullah, N.H.; Li, F.; Fathollahi-Fard, A.M.; Wang, D.; Li, Z. Scheme selection of design for disas-sembly (DFD) based on sustainability: A novel hybrid of interval 2-tuple linguistic intuitionistic fuzzy numbers and regret theory. *J. Clean. Prod.* **2021**, *281*, 124724.
25. Fallahpour, A.; Nayeri, S.; Sheikhalishahi, M.; Wong, K.Y.; Tian, G.; Fathollahi-Fard, A.M. A hyper-hybrid fuzzy decision-making framework for the sustainable-resilient supplier selection problem: A case study of Malaysian Palm oil industry. *Environ. Sci. Pollut. Res.* **2021**, 1–21. https://doi.org/10.1007/s11356-021-12491-y.

26. Liu, B.; Li, Y.; Xu, Z. Manifold regularized matrix completion for multi-label learning with ADMM. *Neural Netw.* **2018**, *101*, 57–67. https://doi.org/10.1016/j.neunet.2018.01.011.
27. Manna, S.; Pati, S.K. Missing Value Imputation Using Correlation Coefficient. In *Computational Intelligence in Pattern Recognition*; Springer: Singapore, 2020; pp. 551–558.
28. Mudiyanselage, D.L. Multi-Label Classification Using Higher-Order Label Clusters. Ph.D. Thesis, University of Nebraska at Omaha, United States. December, 2018.
29. Taha, A.Y.; Tiun, S.; Abd Rahman, A.H.; Ayob, M.; Sabah, A. A Dynamic Two-Layers MI and Clustering-based Ensemble Feature Selection for Multi-Labels Text Classification. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. https://doi.org/10.14569/ijacsa.2020.0110764.
30. Pant, P.; Sabitha, A.S.; Choudhury, T.; Dhingra, P. Multi-label Classification Trending Challenges and Approaches. In *Emerging Trends in Expert Applications and Security*; Springer: Singapore, 2019; pp. 433–444.
31. Al-Salemi, B.; Ayob, M.; Noah, S.A.M. Feature ranking for enhancing boosting-based multi-label text categorization. *Expert Syst. Appl.* **2018**, *113*, 531–543. https://doi.org/10.1016/j.eswa.2018.07.024.
32. Tsoumakas, G.; Katakis, I.; Vlahavas, I. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*; Springer: Boston, MA, USA, 2009; pp. 667–685.
33. Adel, A.; Omar, N.; Albared, M.; Al-Shabi, A. Feature selection method based on statistics of compound words for arabic text classification. *Int. Arab J. Inf. Technol.* **2019**, *16*, 178–185.
34. Taha, A.Y.; Tiun, S.; Abd Rahman, A.H.; Sabah, A. Multilabel Over-sampling and Under-sampling with Class Alignment for Imbalanced Multilabel Text Classification. *J. Inf. Commun. Technol.* **2021**, *20*, 423–456.
35. Taha, A.Y.; Tiun, S. Binary Relevance (BR) Method Classifier of Multi-Label Classification for Arabic Text. *J. Theor. Appl. Inf. Technol.* **2016**, *84*, 414–421.