Seo *et al. Journal of Big Data* (2024) 11:108
https://doi.org/10.1186/s40537-024-00961-2

Journal of Big Data

RESEARCH Open Access

Memetic multilabel feature selection using pruned refinement process

Wangduk Seo[1], Jaegyun Park[2], Sanghyuck Lee[3], A-Seong Moon[3], Dae-Won Kim[2*] and Jaesung Lee[1,3*]

*Correspondence:
dwkim@cau.ac.kr; curseor@cau.ac.kr

[1] AI/ML Innovation Research Center, Chung-Ang University, Heukseok-Dong, Dongjak-Gu, Seoul 06974, Republic of Korea
[2] School of Computer Science and Engineering, Chung-Ang University, Heukseok-Dong, Dongjak-Gu, Seoul 06974, Republic of Korea
[3] Department of Artificial Intelligence, Chung-Ang University, Heukseok-Dong, Dongjak-Gu, Seoul 06974, Republic of Korea

Abstract

With the growing complexity of data structures, which include high-dimensional and multilabel datasets, the significance of feature selection has become more emphasized. Multilabel feature selection endeavors to identify a subset of features that concurrently exhibit relevance across multiple labels. Owing to the impracticality of performing exhaustive searches to obtain the optimal feature subset, conventional approaches in multilabel feature selection often resort to a heuristic search process. In this context, memetic multilabel feature selection has received considerable attention because of its superior search capability; the fitness of the feature subset created by the stochastic search is further enhanced through a refinement process predicated on the employed multilabel feature filter. Thus, it is imperative to employ an effective refinement process that frequently succeeds in improving the target feature subset to maximize the benefits of hybridization. However, the refinement process in conventional memetic multilabel feature selection often overlooks potential biases in feature scores and compatibility issues between the multilabel feature filter and the subsequent learner. Consequently, conventional methods may not effectively identify the optimal feature subset in complex multilabel datasets. In this study, we propose a new memetic multilabel feature selection method that addresses these limitations by incorporating the pruning of features and labels into the refinement process. The effectiveness of the proposed method was demonstrated through experiments on 14 multilabel datasets.

Keywords: Feature selection, Multilabel classification, Label dependency

Introduction

With the rapid advancement of data analysis technology, big data has become a crucial asset in various fields, such as healthcare [1], finance [2], and cybersecurity [3]. Consequently, the complexity of those data has increased; they not only contain many patterns and features but also comprise multiple labels for each pattern [4]. Multilabel classification aims to assign specific patterns to multiple labels that may have beneficial dependencies for improving the classification accuracy [5]. Many real-world problems, such as image annotation [6], text categorization [7], protein function prediction [8], and music information retrieval [9], can be formulated as multilabel classification problems. We let $D = \{(x_1, Y_1), \ldots, (x_{|D|}, Y_{|D|}) | x_i \in \mathbb{R}^d, Y_i \in \mathcal{P}(L)\}$ be a training set, where x_i and $\mathcal{P}(L)$ denote the pattern and power set of the label set $L = \{l_1, \ldots, l_{|L|}\}$, respectively. For

instances where $x_u \notin D$, the optimal label combination $Y^* = \arg\max_{Y \in \mathcal{P}(L)} h(x_u, Y)$ is identified, wherein $h(\cdot, \cdot)$ is a function designed to assess relevance between a pattern and a label combination [5, 10, 11]. Because these labels frequently exhibit interdependencies, known as label dependency, wherein the presence of one label can affect the relevance or presence of another, $h(x_u, Y)$ can be computed more precisely by considering label dependency, thereby improving the classification accuracy [12].

The classification accuracy can be further improved by excluding noisy features through multilabel feature selection (MLFS) as it can prevent confusion in $h(\cdot, \cdot)$ calculations [13]. We let $S \subset F$ be a feature subset comprising the n most important features in a feature set, $F = \{f_1, \ldots, f_{|F|}\}$ $(n \ll |F|)$ [14]. Because labels can be interdependent, selecting features relevant to multiple labels simultaneously can improve S. Thus, exploiting those label dependencies among $2^{|L|}$ label combinations is preferable in MLFS. In practice, assessing all possible feature subsets is infeasible as the size of the search space becomes $2^{|F|}$. To overcome this challenge, conventional MLFS methods often employ a heuristic evolutionary search, which is particularly effective for navigating vast search spaces owing to its population-based search strategy [15–17]. Specifically, it involves generating candidate feature subsets and iteratively improving the fitness of these candidates through evaluations by a subsequent learner. Furthermore, recent advancements have significantly improved the search capabilities of classical evolutionary MLFS by integrating an effective refinement process [18, 19]. This process improves the candidates by replacing less important features with more important ones, which are determined through a scoring function, such as the mutual information between features and labels [20, 21]. In particular, to exploit the label dependencies, the score function often directly measures the relevance of each feature to L by summing up the conditional dependencies between the feature and label combinations.

The refinement process leverages the score function to prioritize the important features to include in S, thereby effectively eliminating the consideration of irrelevant features during the search process [22, 23]. However, score aggregation across all label combinations may overlook the relevance of each feature to specific label combinations with high uncertainty. Consequently, features that are highly relevant to these label combinations may be neglected during the refinement process, resulting in the exploration of feature subsets with lower fitness until the algorithm terminates. This is triggered by the inability to further reduce the uncertainty associated with the label combinations of high uncertainty. Moreover, the score function and subsequent learner are often incompatible, making feature subsets less effective for learners. As a result, these limitations can lead to select less effective feature subsets in multilabel datasets, thereby reducing the performance of the subsequent learner.

To address these limitations, this paper presents a simple yet effective memetic MLFS method that incorporates a pruning mechanism for both features and labels from the refinement process. From the perspective of labels, the proposed method first identifies a label combination Z with high uncertainty given by a candidate S and then prunes the remaining labels from the refinement process. Specifically, the proposed method sorts the labels in descending order of accuracy and includes labels with lower accuracy than the most significant difference in accuracy between two successive labels in Z. This causes the algorithm to focus on Z, thereby preventing a potential bias in the score

calculation by the employed filter. Next, the algorithm addresses the compatibility issue by identifying features only from $S' \neq S$ that yield a low uncertainty for Z, where S and S' are members of population P. Thus, the features in $F - S'$ are pruned, and only the features in S' that have been confirmed to yield a low uncertainty for Z are considered in the refinement process. To validate the effectiveness of the proposed method, we conducted empirical experiments and statistical tests on 14 multilabel datasets. All datasets and source codes are available at https://github.com/minercode625/PLCFS. The contributions of the proposed method are as follows:

- A novel memetic MLFS method is introduced, which includes a pruning mechanism for both features and labels during the refinement process.
- To address overlooked label dependencies and biases in score calculation, the proposed method identifies label combinations with high uncertainty and prunes the remaining labels from the refinement process.
- To resolve compatibility issues between the employed filter and the subsequent learner, the proposed method selectively prunes features that are irrelevant to the label combination with high uncertainty, as evaluated by the learner.
- The effectiveness of the proposed method is validated through empirical experiments on 14 multilabel datasets, demonstrating its superior performance compared to state-of-the-art MLFS methods.

Related works

MLFS methods can be classified into two main categories: filters and wrappers. Filter methods use the mathematical relationships between features and multiple labels to assess the importance of each feature. Subsequently, they select the top n features based on the importance scores of various criteria, such as information theory, for each feature. A mutual information-based label-distribution MLFS method was proposed by Qian et al. [24], which introduced label-distribution learning for MLFS for practical applications in which each instance can have a different relative significance to multiple labels. Additionally, a generalized entropy approximation for cardinality was proposed and applied for MLFS [25]. Moreover, label dependency was categorized into label independence, redundancy, and supplementation to identify features that provide considerable information regarding one label and others through categorization [26]. Furthermore, a new feature relevance term in the criterion was devised by weighting the relevance of each feature [27]. A fuzzy mutual information-based MLFS was proposed to combine label dependency and streaming labels [20]. The concepts of label gain and mutual aid were introduced to measure the internal influence of the label space by considering the label dependency [21]. However, these methods produce different results depending on the cardinality of the entropy calculation [25]. Additionally, their classification performance is limited owing to the absence of interaction with the subsequent learner.

Recent studies have employed various optimization-based approaches for MLFS, including manifold learning, graph-based feature selection, and metaheuristic optimization. For instance, one study projected original data onto a low-dimensional manifold space, preserving the essence of real labels while constructing a pseudo-label matrix [28].

In the manifold learning paradigm, sparse coefficients for MLFS were derived by using an objective function that integrates manifold regularization and dependence maximization [29]. Furthermore, the correlation between labels was assessed through iterative optimization to refine the label distribution [22]. To address the label dependency and imbalanced label distribution, a global and local label correlation-based MLFS was proposed, which employs a shared latent space [30]. By approaching MLFS as a bipartite graph-matching problem, the correlation between features and labels was quantified using edge weights to facilitate MLFS solution [31]. Additionally, metaheuristic search methods, such as ant colony optimization, have been considered, which conduct evolutionary searches based on feature redundancy and label relevance, thereby bypassing explicit learners [32]. This approach was further refined by incorporating temporal difference reinforcement learning to adjust the heuristic function dynamically [33]. Despite their efficacy, these methodologies may fail to capture label dependencies, necessitating extensive hyperparameter tuning and incurring extensive computational costs when constructing feature-label graphs. Recently, an effective pre-elimination strategy and statistically inspired crowding distance were proposed to enhance the search capability of the multimodal multi-objective genetic algorithm for MLFS [34].

Wrapper methods employ a subsequent learner, such as a multilabel naive Bayes classifier [11], to evaluate the fitness of candidate feature subsets [35]. Among the various approaches to wrapper methods, the population-based evolutionary search method has been demonstrated to be effective [36]. For instance, a genetic algorithm initializes each candidate as a binary string, depending on the selection of features [11]. Subsequently, the algorithm assesses the fitness of each candidate using a learner. Candidates for the next generation are selected based on their evaluated fitness, and new candidates are generated through random recombination. Meanwhile, several studies have exploited particle swarm optimization (PSO) to address multi-objective challenges in MLFS. These studies primarily focused on optimizing performance metrics and reducing the number of selected features [12]. For instance, an adaptive uniform mutation was introduced that controls the mutation probability over iterations, and a local search strategy based on differential learning was proposed. Furthermore, a decomposition-based PSO algorithm for MLFS was proposed to effectively handle the trade-off between multiple objectives [15]. Meanwhile, a simple yet effective MLFS method employing non-selection and selection operators was developed to filter unnecessary features while simultaneously analyzing the important ones efficiently [17]. A novel initialization strategy of the population was proposed to enhance the search capability of the genetic algorithm for MLFS [37]. The proposed initialization strategy was based on the mutual information between features and labels, calculating a probability distribution of the features for the initial population. These wrapper methods often perform better than the filter methods because they interact directly with subsequent learners [35]. However, these methods incur considerable computational costs to identify the optimal feature subset and exhibit unstable results over several runs owing to their randomness. To provide a clearer understanding of the differences between the filter and wrapper methods in MLFS, a comparative explanation is presented in Table 1. The table highlights the key differences between the two approaches in terms of the evaluation strategy, computational complexity, interaction with learning algorithms, and stability.

Table 1 Comparison of the filter and wrapper methods used for MLFS

Criteria	Filter methods	Wrapper methods
Evaluation strategy	Uses mathematical relations between features and multiple labels	Evaluate the fitness of candidate feature subsets using a specific learner
Computational complexity	Less complex due to independence from learners	High complexity due to iterative search and dependence on a learner
Interaction with learner	Limited interaction, which may lead to sub-optimal feature subsets for a specific learner	Direct interaction, usually resulting in better performance
Label dependency handling	Limited ability to handle label dependency	Better handling of label dependency when using multilabel learners
Stability	Less affected by randomness	May exhibit stability issues due to the randomness of search methods

Contemporary studies have adopted hybrid approaches that hybridize an evolutionary feature wrapper and a filter by directly adding or removing features from the candidates evaluated using the filter method. For instance, an improved genetic algorithm was combined with mutual information for the feature selection in gene expression data [38]. Moreover, the search capability of a binary ant-lion optimizer was improved using the rough set theory and conditional entropy [39]. To enhance the detection of the coronavirus disease 2019 using computed tomography images of the chest, a novel hybrid feature selection method was introduced that employs several genetic algorithms initialized through multiple filters [18]. An effective feature selection method that integrates beam search, genetic algorithm, and cuckoo search was proposed, specifically targeting heterogeneous multilabel datasets [23]. Because this method hybridizes more search strategies than conventional hybrid methods, the issue of compatibility between these strategies can become more critical. Furthermore, for multilabel text categorization, an information theory-based score function was integrated into a memetic MLFS without traditional problem transformation techniques [40]. In an effort to reduce the computational demands of the search process, an artificial immune optimization algorithm with a Fisher score has been proposed [19]. In the cybersecurity domain, a new approach for hybrid feature selection was proposed based on an extreme learning machine and a genetic algorithm for intrusion detection [41]. Despite these advancements, the classification efficacy of these methods remains constrained owing to the oversight of label dependency during the evaluation by a subsequent learner. To address this gap, the concept of label complementarity was introduced, marking a significant step toward an effective MLFS technique [42]. This method aims to refine the interactions among multiple subpopulations by leveraging the evaluation information to generate label combinations. Additionally, an enhanced communication strategy was proposed to prevent the generation of redundant solutions by incorporating a hybrid filter process [43]. Regarding the importance of label dependency, an effective method was proposed that combines a PSO algorithm with a sparse learning method to exploit the label dependency and avoid local optima during the search process [44]. However, despite recognizing the importance of label dependency, these methods do not fully leverage their potential to reduce the uncertainty associated with specific label combinations effectively.

Proposed method

Motivation

In the field of MLFS, it is well-known that effective exploitation of label dependency determines the success of MLFS. A set of features may simultaneously reduce the uncertainty of multiple labels. Hence, a final feature subset S, where $|S| = n$ ($n \ll |F|$), comprising those features, would lead to a superior classification accuracy of the subsequent learner. To identify the best feature subset, the algorithm must verify $2^{|F|}$ feature subsets. Because this task is infeasible in practice, the algorithm may employ a heuristic search strategy that generates candidate feature subsets and then improves the fitness of the incomplete solutions. In this paper, we considered a memetic search that enhances the stochastic global search capability of the evolutionary process using a greedy multilabel feature filter that considers label dependency. Figure 1 presents a schematic overview of the proposed method with the effective refinement of feature subsets.

During the evolutionary process, population P comprises a set of candidate feature subsets. According to the selected features, each feature subset will yield different discrimination power on each label; a set of labels whose uncertainty is still high may exist, resulting in lower accuracy for the subsequent learner. To improve fitness, the uncertainty of these labels should be reduced by modifying the feature members. In this case, conditioning by the target feature subset to be modified, including new features dependent on labels of high uncertainty, denoted by Z, will be a rational choice for the algorithm. Because this process can take too long owing to the stochastic nature of the evolutionary process, a filter method can be employed to identify features from $F \backslash S$ directly. This process, known as the refinement process, can improve the fitness of the feature subset and is the main advantage of memetic search over classical evolutionary search. However, conventional memetic MLFS suffer from two issues.

In conventional feature-filter methods, the merit or score of the new features to be included is calculated by summing up all the possible conditional dependencies between the features and label combinations. Thus, the algorithm attempts to reduce the uncertainty of all labels, not a specific label combination; i.e., the uncertainty of labels with low accuracy is implicitly reduced. Consequently, the score value can be biased to $Z^c = L \backslash Z$, particularly when $|Z| \ll |Z^c|$, neglecting the features that are highly dependent on Z. To circumvent this issue, the algorithm may prune Z^c from the refinement process. For this purpose, we consider a simple pruning process based on the accuracy value of each label. Given labels $\{l_1^s, l_2^s, \cdots, l_{|L|}^s\}$ sorted in descending order of accuracy, we let k be the index of the most significant difference in accuracy between two successive labels [45]. The algorithm then identifies labels with a lower accuracy than l_k^s as $Z = \{l_{k+1}^s, \cdots, l_{|L|}^s\}$, which requires $\mathcal{O}(|L| \log |L|)$ computation for sorting and $\mathcal{O}(|L|)$ to identify Z.

The next issue is the incompatibility between the multilabel feature filter employed for the refinement process and the employed learner, as features considered good from the viewpoint of the filter can be meaningless features from the viewpoint of the employed learner. It should be noted that employing the learner directly in the refinement process can be ineffective because it can lead to the exhaustive consumption of the limited computational costs, such as fitness function calls (FFCs), resulting in early algorithm

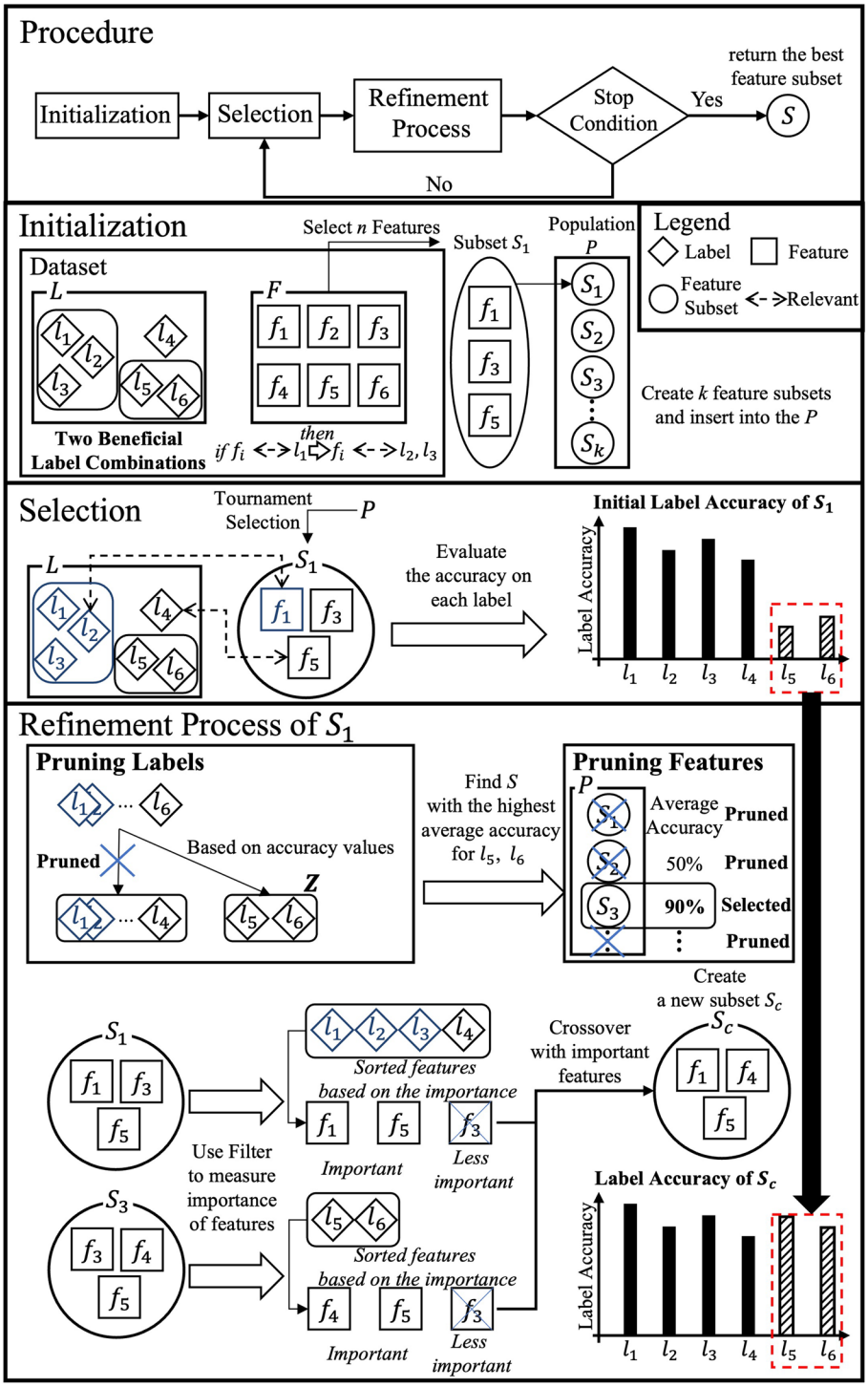


Fig. 1 Schematic overview of the proposed method comprising pruning and refinement processes

stops with rough feature subsets in the population P. To address this issue, we consider a strategy that exploits the feature subsets $S' \neq S$ in P whose contribution to specific label combinations from the viewpoint of the learner has already been verified through

a fitness evaluation. The proposed algorithm first exploits a feature subset, S', in P with the highest average accuracy for the labels in Z. The filter then ranks the features in S' based on their relevance to Z. Finally, the features with the highest relevance are introduced into S. This process can be viewed as a pruning process on features because the features in $F \setminus S'$ are ignored from the refinement process. It is noteworthy that any multilabel feature filter that ranks candidate features and considers label dependencies can be applied here.

Algorithm 1 Proposed Algorithm

Require: Dataset D, population size m, maximum FFCs v, maximum feature size n_{max}
Ensure: Best feature subset S^g
1: initialize a population P
2: evaluate feature subsets in P
3: $u \leftarrow m$, set consumed FFCs to m
4: **while** $u \leq v$ **do**
5: $N \leftarrow \emptyset$
6: $S \leftarrow tournament\_selection(P)$
7: $S',\ Z \leftarrow Pruning(S)$ ▷ Label Pruning
8: $N \leftarrow N \cup Crossover(S, S', Z)$ ▷ Crossover
9: $N \leftarrow N \cup Mutation(S, Z)$ ▷ Mutation
10: evaluate feature subsets in N
11: $u \leftarrow u + |N|$
12: $P \leftarrow P \cup N$
13: keep m best feature subsets in P
14: store the best feature subset in S^g
15: **end while**

Algorithm 2 *Pruning(S)*

Require: S
Ensure: S', Z
1: $L^s \leftarrow \{l_1^s, l_2^s, \cdots, l_{|L|}^s\}$ ▷ sort labels in descending order of accuracy
2: $k \leftarrow \arg\max_{1 \leq k \leq |L|} v_{l_i}^S - v_{l_{i+1}}^S$
3: $Z \leftarrow \{l_{k+1}^s, l_{k+2}^s, \cdots, l_{|L|}^s\}$
4: $S' \leftarrow \arg\max_{S' \in P-S} \frac{1}{|Z|} \sum_{l \in Z} r_l^S$

Algorithm

Table 2 summarizes the terms used in the proposed method, and Algorithm 1 presents its pseudocode. As the input parameters, the proposed method obtains the target multilabel dataset D, the maximum number of feature subsets m, and the maximum feature subset size n_{max}. The FFCs v, which is the maximum allowed number of evaluations performed by a subsequent learner, is used as the termination condition. The proposed method performs a search process until the available FFCs are exhausted. First, it initializes a population P of m feature subsets (Line 1). Thereafter, each feature subset randomly selects n_{max} features based on a uniform

Table 2 Notations used in the proposed method

Term	Meanings
D	Multilabel dataset
F	Feature set in $D, F = f_1, \cdots, f_{\|F\|}$
L	Label set in $D, L = l_1, \cdots, l_{\|L\|}$
Y	Label combination, $Y \in \mathcal{P}(L)$
Z	Label combination with high uncertainty, $Z \in \mathcal{P}(L)$
n_{max}	Number of features to be selected
P	Population of the feature subsets
m	Population size, representing the number of feature subsets
S	Feature subset of $F, \|S\| \leq n$
S^g	Optimal feature subset of the population
r_l^S	Accuracy for label l by learning S
u	Number of spent FFCs
v	Maximum number of FFCs allowed

distribution. The m initialized feature subsets are evaluated based on their consumption of m FFCs (Lines 2 and 3). Specifically, the classifier predicts a label combination for each pattern by learning based on each feature subset. Furthermore, the classifier measures the accuracy of each label r_l^S and the evaluation metric across the label. An empty population N is created to store the reproduced feature subsets (Line 5). After the evaluation, the tournament selection algorithm [46] selects a feature subset S to reproduce new feature subsets. The pruning process determines Z and S' corresponding to the selected S, as described in Algorithm 2 (Line 7). Thereafter, new feature subsets are created using the proposed crossover and mutation, as described in Algorithms 3 and 4 (Lines 8 and 9). The reproduced feature subsets are stored in N and are evaluated, and the number of spent FFCs u increases by $|N|$ (Line 10 and 11). Additionally, P includes the reproduced feature subsets, and the m feature subsets with the highest evaluation scores are retained in P (Lines 12 and 13). Finally, the best feature subset, S^g, in P is stored and obtained after the algorithm terminates (Line 14).

Algorithm 3 *Crossover*(S, S', Z)

Require: S, S', Z
Ensure: S^+
Require: $n \geq 0 \vee x \neq 0$
Ensure: $y = x^n$
1: $n \leftarrow \lceil n_{max} \cdot |Z|/|L| \rceil$ ▷ Determine the number of features selected from S'
2: $S^+ \leftarrow \emptyset$
3: $S^+ \leftarrow S^+ \cup Filter(S, Z^c, n_{max} - n)$
4: $S^+ \leftarrow S^+ \cup Filter(S', Z, n)$

Algorithm 4 *Mutation(S, Z)*

Require: S, Z
Ensure: S^+
1: $S^z \leftarrow \{f | num(f) = 0,\ where\ f \in F\}$ ▷ $num(f)$ is the count of f in P
2: **if** $S^z = \emptyset$ **then**
3: $n \leftarrow min(|S^z|, \lceil n_{max} \cdot |Z|/|L| \rceil)$
4: $S^+ \leftarrow \emptyset$
5: $S^+ \leftarrow S^+ \cup Filter(S, Z^c, n_{max} - n)$
6: $S^+ \leftarrow S^+ \cup Filter(S^z, Z, n)$
7: **else**
8: create S^+ by random initialization
9: **end if**

Algorithm 5 *Filter(S, L, n)*

Require: S, Y, the number of features to be selected n
Ensure: feature subset S^f sorted by Generalized Information-theoretic Criterion (GICS)
1: $S^f \leftarrow \emptyset$
2: **while** $|S^f| < n$ **do**
3: Include $f^+ \in \{S - S^f\}$ to S^f that maximize Equation (1)
4: **end while**

Algorithm 2 presents the pseudocode for the label pruning process, which determines Z and S' corresponding to the S selected via tournament selection. All labels in L are sorted in descending order of individual accuracy measured by the subsequent learner using the selected features in S (Line 1). The algorithm then calculates the differences in accuracy between two consecutive labels, and the location with the largest difference is stored as k (Line 2). Based on the k, the labels with lower accuracy $\{l_{k+1}, \cdots, l_{|L|}\}$ are set to Z (Line 3). The feature subset in P with the highest average accuracy in the determined Z is the feature subset, S' (Line 4).

Algorithm 3 presents the pseudocode of the proposed crossover operator. It obtains n features that are most relevant to Z from S' and merges them with the $n_{max} - n$ features from S that are most relevant to Z^c. Because the optimal size n is unknown, n is set according to S using the ratio of Z to the total number of labels. Specifically, the number of features n to be selected from S' is determined, where $n = \lceil n_{max} \cdot |Z|/|L| \rceil$ (Line 1). n denotes the ratio of the size of Z, $|Z|$, to that of all the labels, $|L|$. Thereafter, a new empty set S^+ is created (Line 2). The $n_{max} - n$ top-scoring features in S are selected using *Filter*, which scores each feature based on Z^c (Line 3). Similarly, n top-scoring features in S' are selected using *Filter*, based on Z (Line 4). All the selected features are added to the created S^+ (Lines 3 and 4).

Algorithm 4 presents the pseudocode of the proposed mutation operator. In the proposed method, certain features fail to be selected during the initialization of Algorithm 1 or can be removed from P when feature subsets are discarded. To avoid this problem for the features that may be highly relevant to Z, the proposed mutation operator continues providing these features to P. Specifically, features that are never selected or removed in P are added to S^z (Line 1). Subsequently, the number (n) of

features selected from S^z is determined as in Algorithm 3 (Line 3). A new empty set S^+ is then created (Line 4). If S^z is not empty, $n_{max} - n$ features are selected from S using *Filter*, which scores each feature in S based on its relevance to Z^c (Line 5). Additionally, the proposed mutation selects the n features most relevant to Z from S^z using *Filter* (Line 6). Thereafter, all the selected features are added to the created S^+ (Lines 5 and 6). However, if S^z is an empty set, S^+ is created through random initialization (Line 8).

Algorithm 5 describes the filter function *Filter*. The algorithm calculates the importance score of each feature for a specific feature subset S, label combination Y, and number n representing the size of the selected features. Subsequently, it selects the n highest-scoring features using the applied filter. Specifically, we employed a recent filter method called the generalized information-theoretic criterion (GICS) [25] for MLFS (Line 3). Equation 1 describes the criterion of GICS for calculating feature importance.

$$J(f^+, S, Y) = \sum_{l \in Y} H(f^+; l) - \frac{|Y|}{|S|} \sum_{f \in S} H(f^+; f) \tag{1}$$

Here, $H(X) = -\sum P(x) \log P(x)$ is the joint entropy of the involved variable set X, and $P(X)$ is the probability function. The computational complexity of the *Filter* is expressed as $\mathcal{O}(|F| \cdot |L| + |F| \cdot n)$. Thus, the overall computational complexity is $\mathcal{O}(v \cdot (|F| \cdot |L| + |F| \cdot n + C))$, where C denotes the complexity of the subsequent learner. Although the overall computational complexity of the proposed method primarily depends on the complexity of the *Filter*, it can vary based on the corresponding subsequent learner and the number of patterns and labels, similar to conventional wrapper methods.

Experimental results

Experimental settings

The effectiveness of the proposed method was evaluated using various multilabel datasets. Specifically, 14 widely-used multilabel benchmark datasets were selected to compare the proposed method against conventional methods and verify its performance [17, 31, 33, 39, 47]. The Emotions dataset [48] is one such dataset, comprising music data classified into six emotional clusters, eight rhythmic features, and 64 timbre features. The Enron and Llog dataset [49, 50], wherein each feature corresponds to the occurrence of a specific word, and each label represents the relevance of each text pattern to a specific subject, was generated using text-mining applications. The Genbase and Yeast datasets [51, 52] were created for the medical domain and include information regarding the functions of genes and proteins. The Medical dataset [53] was sampled from a large corpus of suicide letters obtained through natural language processing of free clinical text. The Scene dataset [54] comprises semantic indexes of still scenes, where each scene may comprise multiple objects. Moreover, the Tmc2007 dataset [55] contains text data of aviation safety reports that document problems which occurred during certain flights. The remaining seven datasets were obtained from the Yahoo dataset collection [56]. Specifically, the Yahoo dataset collection comprises 14 datasets, such as Business, Computers, and Medical, which are sourced

Table 3 Standard statistics of the multilabel datasets

Dataset	$\|W\|$	$\|F\|$	*Type*	$\|L\|$	*Card.*	*Den.*	*Distinct.*	*Domain*
Business	11,214	1096	Numeric	30	1.599	0.053	233	Text
Computers	12,444	34,096	Numeric	33	1.507	0.046	428	Text
Education	12,030	27,534	Numeric	33	1.463	0.044	511	Text
Emotions	593	72	Numeric	6	1.869	0.311	27	Music
Enron	1702	1001	Nominal	53	3.378	0.064	753	Text
Entertainment	12,730	32,001	Numeric	21	1.414	0.067	337	Text
Genbase	662	1185	Nominal	27	1.252	0.046	32	Biology
Health	9205	1530	Numeric	32	1.644	0.051	335	Text
Llog	1460	1004	Nominal	75	1.180	0.016	304	Text
Medical	978	1449	Nominal	45	1.245	0.028	94	Text
Recreation	12,828	30,324	Numeric	22	1.429	0.065	530	Text
Scene	2407	294	Numeric	6	1.074	0.179	15	Image
Tmc2007	28,596	981	Numeric	22	2.158	0.098	1341	Text
Yeast	2417	103	Numeric	14	4.237	0.303	198	Biology

from multilabel text categorization tasks on "yahoo.com". Table 3 lists the standard statistics of the 14 datasets used in the experiments, including the number of patterns $|W|$, number of features $|F|$, types of features, and number of labels $|L|$. If the feature type is numeric, the features were discretized using label-attribute interdependence maximization, which is a discretization method specialized for multilabel data [57]. The label cardinality *Card.* represents the average number of labels in each pattern and label density *Den.* denotes the label cardinality for the total number of labels. Additionally, *Distinct.* indicates the number of unique label subsets in L and *Domain* represents the applications related to each dataset.

Because the proposed method is based on a memetic algorithm, which is a hybrid of filter-based and wrapper-based methods, the comparison methods were selected from various types of evolutionary MLFS approaches, including filter-based, wrapper-based, and hybrid methods. Specifically, the proposed method was compared with four state-of-the-art evolutionary MLFS methods: MLACO [32], MLPSO [47], BMFS [31], and MMDE [34], as well as one hybrid method, HBALO [39]. The specific parameters for each method were set based on the values used in the original studies.

- MLACO: This method combines an ant colony optimization with a Markov decision process. The pheromone decay rate, learning rate, and discount rate were set to 0.2, 0.5, and 0.8, respectively.
- MLPSO: MLPSO employs a local learning strategy to improve the performance of PSO-based MLFS. The scale factor in the local learning strategy was set to a random value between [0.1, 0.9].
- BMFS: BMFS constructs a bipartite graph to represent the feature-label relationships based on the Hungarian algorithm.
- HBALO: HBALO hybridizes hill-climbing techniques with an ant-lion optimizer to identify relevant features quickly. The α and β parameters were set to 0.99 and 0.01, respectively.

- MMDE: MMDE is a multimodal optimization algorithm that combines multi-objective optimization with correlation-based feature selection. The number of quantiles, elimination percentage, and mutation rate were set to 24, 0.5, and 0.2, respectively.

To ensure fairness, the maximum number of allowable FFCs v, the population size m, and the number of selected features n_{max} were set to 300, 50, and 50, respectively. Multilabel naive Bayes (MLNB) [11] and a holdout cross-validation method were used to evaluate the quality of the feature subsets obtained using each method; 80% and 20% of each dataset were used as the training and test sets, respectively. Each experiment was repeated ten times, and the average values and standard deviations of the results were calculated.

The performance was evaluated using four evaluation metrics: multilabel accuracy, one-error, ranking loss, and multilabel coverage [58, 59]. We let $T = \{(w_i, \lambda_i)|1 \leq i \leq |T|\}$ be the given test set, where $\lambda_i \subseteq L$ denotes the correct label subset. For a given test sample w_i, the MLNB classifier should output a set of confidence values $0 \leq \Psi_{i,l} \leq 1$ for each label $l \in L$. Specifically, a series of functions $\{g_1, g_2, \ldots, g_{|L|}\}$ are induced from the training patterns. Next, each function g_k determines the class membership of l_k with respect to each pattern (i.e., $V_i = \{1_k | g_k(w_i) > \theta, 1 \leq k \leq |L|\}$, where θ is a predetermined threshold that was set to 0.5 in this study). The multilabel accuracy (*mlacc*) is defined as follows:

$$mlacc(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{|\lambda_i \cap V_i|}{|\lambda_i \cup V_i|}. \tag{2}$$

Furthermore, the one-error (*onerr*) is defined as

$$onerr(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} [\arg\max_{l_k \in L} g_k(w_i) \notin \lambda_i], \tag{3}$$

where $[\cdot]$ returns one if the proposition stated in the brackets is true and returns zero otherwise. Specifically, the one-error evaluates how many steps the top-ranked predicted label is not in the relevant label combination. For single-label classification problems, the one-error is identical to ordinary classification error [60].

The ranking loss (*rloss*) is defined as

$$rloss(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \frac{|\{(a, b)|a \in \lambda_i, b \in \overline{\lambda_i}, \psi_{i,a} \leq \psi_{i,b}\}|}{|\lambda_i||\overline{\lambda_i}|}, \tag{4}$$

where $\overline{\lambda_i}$ denotes the complementary set of λ_i. Therefore, the ranking loss measures the average fraction of (a, b) pairs with $\psi_{i,a} \leq \psi_{i,b}$ among all possible relevant and irrelevant label pairs.

Finally, the multilabel coverage (*mlcov*) is defined as

$$mlcov(T) = \frac{1}{|T|} \sum_{i=1}^{|T|} \max_{l \in \lambda_i} rank(l) - 1, \tag{5}$$

where $rank(\cdot)$ returns the rank of the corresponding relevant label $l \in \lambda_i$ according to $\psi_(i, l)$ in a non-increasing order, indicating that $\lambda_i \subseteq L$ represents the correct label subset. Thus, multilabel coverage measures the number of labels that must be marked positive for all relevant labels. Higher values of multilabel accuracy and lower values of the one-error, ranking loss, and multilabel coverage metrics indicate good classification performance.

After evaluating the performances of the methods on all the datasets, the performance of the proposed method was analyzed using a statistical tool to verify its potential. A paired t-test was conducted at 5% significance level to compare the performance of the proposed method with that of other MLFS methods for each dataset. The test procedure and its parameters are consistent with those employed in conventional statistical analyses of MLFS methods [42, 61]. The null hypothesis assumed that there was no difference in the distribution of performance values between the proposed method and the comparison methods for each dataset. The alternative hypothesis posited that the proposed method exhibits a different distribution of performance values compared to the comparison methods. If the null hypothesis was rejected, it was concluded that the proposed method demonstrated a statistically significant difference in performance compared to the comparison methods. A paired t-test was performed five times as the study employed five methods for the comparisons.

Comparison results

Tables 4, 5, 6, 7 present the experimental results of the proposed method and five conventional methods for the 14 multilabel datasets using four metrics: multilabel accuracy, one-error, ranking loss, and multilabel coverage. The best performances among the five comparison methods are indicated in bold. Additionally, the last row of each table includes the average rank of each metric for all the multilabel datasets. The resulting values are represented by their average values and corresponding standard deviations.

Tables 8, 9, 10, 11 present the pairwise comparison results of the paired t-tests at a 5% significance level for the four evaluation metrics. Each result is presented in the form of $a/b/c$, according to the number of times each method is statistically superior, similar, or inferior to the other methods, respectively. As each method was compared with five other methods, the integer values should be $a + b + c = 5$. As listed in Tables 4, 5, 6, 7, the proposed method outperformed the other MLFS methods on most multilabel datasets. Additionally, the proposed method demonstrated superior performance compared with the other methods for most datasets across most evaluation metrics. For multilabel accuracy, the proposed method achieved the best average rank (1.4) and exhibited superior performance in eight out of 14 datasets. For the one-error, the proposed method achieved the best average rank (2.1) and demonstrated superior performance in seven out of 14 datasets. Similar trends were observed for the ranking loss, and multilabel coverage metrics, where the proposed method achieved the best average ranks of 1.4, and 1.4, respectively, and demonstrated exceptional performance in the majority of the datasets. The experimental results presented in Tables 8, 9, 10, 11 indicate that the proposed method performed better than the other MLFS methods in a statistically significant manner. The proposed method consistently achieved a higher number of wins across all datasets and evaluation metrics, demonstrating a consistent performance across various

Table 4 Performance comparison for *multilabel accuracy* when $|S| = 50$, where the best-performing results for each dataset are highlighted in bold

Method	Business	Computers	Education	Emotions
Proposed	0.679 ± 0.010	0.415 ± 0.007	0.051 ± 0.006	**0.538 ± 0.043**
MLACO	0.655 ± 0.009	0.368 ± 0.007	0.058 ± 0.019	0.453 ± 0.051
MLPSO	0.677 ± 0.009	0.413 ± 0.009	0.025 ± 0.017	0.504 ± 0.032
HBALO	0.673 ± 0.011	0.354 ± 0.010	0.032 ± 0.011	0.487 ± 0.033
BMFS	**0.686 ± 0.008**	**0.417 ± 0.008**	0.001 ± 0.000	0.536 ± 0.041
MMDE	0.667 ± 0.010	0.396 ± 0.006	0.011 ± 0.001	0.476 ± 0.026
Method	**Enron**	**Entertainment**	**Genbase**	**Health**
Proposed	0.375 ± 0.015	0.150 ± 0.010	**0.938 ± 0.018**	**0.547 ± 0.012**
MLACO	0.283 ± 0.012	0.185 ± 0.022	0.118 ± 0.068	0.371 ± 0.039
MLPSO	0.303 ± 0.018	0.068 ± 0.047	0.217 ± 0.164	0.411 ± 0.039
HBALO	0.249 ± 0.016	0.148 ± 0.019	0.419 ± 0.042	0.448 ± 0.018
BMFS	**0.392 ± 0.016**	0.001 ± 0.000	0.055 ± 0.058	0.541 ± 0.011
MMDE	0.282 ± 0.014	0.016 ± 0.003	0.057 ± 0.045	0.343 ± 0.008
Method	**Llog**	**Medical**	**Recreation**	**Scene**
Proposed	**0.236 ± 0.013**	**0.640 ± 0.057**	0.083 ± 0.013	**0.577 ± 0.018**
MLACO	0.061 ± 0.006	0.006 ± 0.004	0.032 ± 0.010	0.440 ± 0.021
MLPSO	0.208 ± 0.080	0.196 ± 0.129	0.039 ± 0.023	0.541 ± 0.020
HBALO	0.231 ± 0.009	0.359 ± 0.074	0.067 ± 0.018	0.517 ± 0.018
BMFS	0.229 ± 0.016	0.001 ± 0.000	0.001 ± 0.000	0.489 ± 0.015
MMDE	0.178 ± 0.070	0.012 ± 0.008	0.014 ± 0.001	0.507 ± 0.014
Method	**Tmc2007**	**Yeast**	**Avg. Rank**	
Proposed	**0.451 ± 0.006**	**0.456 ± 0.015**	**1.4**	
MLACO	0.299 ± 0.028	0.430 ± 0.018	3.8	
MLPSO	0.328 ± 0.042	0.436 ± 0.016	3.7	
HBALO	0.368 ± 0.008	0.384 ± 0.015	3.9	
BMFS	0.441 ± 0.004	0.421 ± 0.008	3.6	
MMDE	0.293 ± 0.015	0.428 ± 0.023	4.6	

datasets. In contrast, the performances of the other MLFS methods were less consistent, exhibiting different results depending on the dataset and evaluation metric. These observations underscore the effectiveness of the proposed method in multilabel classification tasks compared with other methods.

Furthermore, the results of the box plot analysis are presented in Fig. 2, illustrating the performance of the proposed method and the comparison methods across four datasets for four evaluation metrics. The box plot graphically represents the distribution of performance for each method, with the horizontal axis displaying the conducted MLFS methods and the vertical axis indicating the performance value. The analysis clearly demonstrates that the proposed method consistently demonstrated superior performance compared to the other methods across all evaluation metrics.

Additionally, Fig. 3 illustrates the Bonferroni–Dunn post-hoc test critical distance (CD) diagram [61], revealing the relative performance of all methods where the calculated CD is 2.015. The horizontal axis represents the average rank of each method, with higher ranks on the left-hand side of each subfigure. If the average rank difference

Table 5 Performance comparison for *one-error* when $|S| = 50$, where the best-performing results for each dataset are highlighted in bold

Method	Business	Computers	Education	Emotions
Proposed	0.395 ± 0.418	**0.436 ± 0.008**	0.668 ± 0.012	**0.284 ± 0.052**
MLACO	0.393 ± 0.418	0.471 ± 0.007	0.692 ± 0.009	0.309 ± 0.056
MLPSO	0.393 ± 0.419	0.465 ± 0.009	0.683 ± 0.012	0.314 ± 0.036
HBALO	0.402 ± 0.412	0.495 ± 0.008	0.662 ± 0.015	0.310 ± 0.043
BMFS	**0.392 ± 0.419**	0.469 ± 0.008	0.692 ± 0.009	0.307 ± 0.041
MMDE	0.417 ± 0.402	0.469 ± 0.008	**0.652 ± 0.017**	0.379 ± 0.034
Method	**Enron**	**Entertainment**	**Genbase**	**Health**
Proposed	0.576 ± 0.340	0.609 ± 0.010	**0.714 ± 0.450**	0.744 ± 0.329
MLACO	0.597 ± 0.336	0.642 ± 0.010	0.752 ± 0.387	0.779 ± 0.284
MLPSO	0.611 ± 0.311	0.667 ± 0.039	0.861 ± 0.216	**0.446 ± 0.026**
HBALO	0.649 ± 0.308	0.620 ± 0.014	0.815 ± 0.343	0.999 ± 0.000
BMFS	0.572 ± 0.342	0.705 ± 0.007	0.920 ± 0.117	0.726 ± 0.353
MMDE	**0.465 ± 0.181**	**0.603 ± 0.011**	0.835 ± 0.208	0.778 ± 0.286
Method	**Llog**	**Medical**	**Recreation**	**Scene**
Proposed	0.996 ± 0.001	**0.741 ± 0.328**	0.710 ± 0.013	**0.271 ± 0.027**
MLACO	0.996 ± 0.001	0.880 ± 0.148	**0.696 ± 0.009**	0.380 ± 0.033
MLPSO	**0.887 ± 0.017**	0.797 ± 0.255	0.767 ± 0.023	0.326 ± 0.025
HBALO	0.996 ± 0.001	0.840 ± 0.247	0.705 ± 0.022	0.322 ± 0.019
BMFS	0.996 ± 0.001	0.888 ± 0.137	0.800 ± 0.005	0.304 ± 0.026
MMDE	0.996 ± 0.001	0.995 ± 0.000	0.742 ± 0.012	0.450 ± 0.026
Method	**Tmc2007**	**Yeast**	**Avg. Rank**	
Proposed	**0.317 ± 0.005**	**0.229 ± 0.020**	**2.1**	
MLACO	0.357 ± 0.012	0.239 ± 0.023	3.7	
MLPSO	0.435 ± 0.045	0.237 ± 0.021	3.4	
HBALO	0.358 ± 0.006	0.278 ± 0.027	4.0	
BMFS	0.331 ± 0.005	0.298 ± 0.015	3.8	
MMDE	0.447 ± 0.015	0.272 ± 0.018	4.0	

between the proposed method and each compared method is within the CD, it indicates that the proposed method is not significantly different from the corresponding method. The difference is within the CD when the proposed and compared methods are connected with a bold black line. Specifically, for the multilabel accuracy and ranking loss, the proposed method outperformed the other methods, indicating that the proposed method is statistically superior to the other methods, and for the multilabel coverage, the proposed method outperformed the other methods except for the BMFS method.

Finally, we conducted an additional experiment to compare the accuracy improvement for the label combination Z as the search progressed. Because the proposed method was designed to focus on label combinations with low accuracy, the proposed strategy works as intended, and the performance of those labels should improve. Figure 4 illustrates the average accuracies of the labels in Z during the search process on the six multilabel datasets conducted by each method as the number of spent FFCs increased. The averages for the low-accuracy labels were calculated using the proposed

Table 6 Performance comparison for *ranking loss* when $|S| = 50$, where the best-performing results for each dataset are highlighted in bold

Method	Business	Computers	Education	Emotions
Proposed	0.062 ± 0.025	**0.094 ± 0.003**	**0.108 ± 0.003**	**0.164 ± 0.023**
MLACO	0.063 ± 0.025	0.097 ± 0.003	0.111 ± 0.003	0.175 ± 0.025
MLPSO	0.062 ± 0.024	0.097 ± 0.003	0.110 ± 0.004	0.200 ± 0.029
HBALO	0.062 ± 0.026	0.102 ± 0.003	0.108 ± 0.003	0.170 ± 0.023
BMFS	**0.057 ± 0.026**	0.101 ± 0.007	0.111 ± 0.003	0.171 ± 0.023
MMDE	0.068 ± 0.025	0.100 ± 0.003	0.110 ± 0.005	0.227 ± 0.029
Method	**Enron**	**Entertainment**	**Genbase**	**Health**
Proposed	0.114 ± 0.011	**0.133 ± 0.004**	0.042 ± 0.031	**0.074 ± 0.015**
MLACO	0.141 ± 0.030	0.160 ± 0.004	**0.042 ± 0.024**	0.105 ± 0.032
MLPSO	0.131 ± 0.011	0.146 ± 0.013	0.134 ± 0.049	0.074 ± 0.011
HBALO	0.143 ± 0.010	0.136 ± 0.002	0.077 ± 0.027	0.098 ± 0.029
BMFS	**0.110 ± 0.012**	0.160 ± 0.001	0.167 ± 0.026	0.081 ± 0.028
MMDE	0.143 ± 0.007	0.139 ± 0.006	0.140 ± 0.041	0.106 ± 0.029
Method	**Llog**	**Medical**	**Recreation**	**Scene**
Proposed	**0.134 ± 0.023**	**0.081 ± 0.024**	0.197 ± 0.004	**0.093 ± 0.008**
MLACO	0.178 ± 0.025	0.099 ± 0.032	0.216 ± 0.003	0.140 ± 0.018
MLPSO	0.138 ± 0.014	0.138 ± 0.034	0.213 ± 0.005	0.116 ± 0.018
HBALO	0.172 ± 0.021	0.119 ± 0.032	**0.189 ± 0.004**	0.106 ± 0.012
BMFS	0.155 ± 0.024	0.166 ± 0.028	0.221 ± 0.002	0.105 ± 0.010
MMDE	0.172 ± 0.024	0.174 ± 0.025	0.203 ± 0.006	0.180 ± 0.014
Method	**Tmc2007**	**Yeast**	**Avg. Rank**	
Proposed	**0.078 ± 0.006**	**0.191 ± 0.009**	**1.4**	
MLACO	0.095 ± 0.005	0.201 ± 0.009	3.9	
MLPSO	0.123 ± 0.008	0.202 ± 0.013	3.4	
HBALO	0.118 ± 0.004	0.209 ± 0.008	3.4	
BMFS	0.080 ± 0.001	0.242 ± 0.007	3.9	
MMDE	0.145 ± 0.013	0.207 ± 0.007	5.0	

method, and all evolutionary algorithm-based comparison methods generated and evaluated the feature subsets. Specifically, the solid and dotted lines indicate the averages of the accuracy values for the lower 50% and 25% of the labels, respectively in the order of accuracy in L. The average of the lower 25% of the labels was always lower than that of the 50% of the labels; thus, the dotted line appears below the solid line. Figure 4 reveals that the proposed method achieved an incremental improvement in Z, regardless of the initialization condition. Moreover, it achieved a more significant improvement than the other methods. The average of the 25% lowest-accuracy labels of the proposed method was higher than that of the average of the 50% lowest-accuracy labels of the other methods for the Genbase, Medical, and Yeast datasets after the number of consumed FFCs reached 170. Therefore, it was confirmed that the proposed method is more effective because it focuses on reducing the uncertainty of Z by introducing its effective features.

Table 7 Performance comparison for *multilabel coverage* when $|S| = 50$, where the best-performing results for each dataset are highlighted in bold

Method	Business	Computers	Education	Emotions
Proposed	0.133 ± 0.024	**0.165 ± 0.003**	0.169 ± 0.003	**0.464 ± 0.019**
MLACO	0.133 ± 0.024	0.168 ± 0.003	0.171 ± 0.003	0.469 ± 0.019
MLPSO	0.133 ± 0.024	0.170 ± 0.003	0.171 ± 0.003	0.487 ± 0.025
HBALO	0.134 ± 0.023	0.172 ± 0.004	**0.168 ± 0.005**	0.499 ± 0.014
BMFS	**0.125 ± 0.025**	0.166 ± 0.005	0.171 ± 0.003	0.465 ± 0.015
MMDE	0.140 ± 0.024	0.171 ± 0.003	0.170 ± 0.005	0.509 ± 0.024
Method	**Enron**	**Entertainment**	**Genbase**	**Health**
Proposed	0.304 ± 0.009	**0.213 ± 0.003**	0.093 ± 0.030	**0.132 ± 0.018**
MLACO	0.337 ± 0.034	0.245 ± 0.005	**0.090 ± 0.022**	0.176 ± 0.029
MLPSO	0.336 ± 0.018	0.242 ± 0.003	0.243 ± 0.029	0.149 ± 0.017
HBALO	0.341 ± 0.016	0.223 ± 0.005	0.141 ± 0.030	0.168 ± 0.027
BMFS	**0.291 ± 0.014**	0.243 ± 0.003	0.219 ± 0.027	0.150 ± 0.025
MMDE	0.342 ± 0.011	0.222 ± 0.006	0.193 ± 0.043	0.177 ± 0.026
Method	**Llog**	**Medical**	**Recreation**	**Scene**
Proposed	**0.177 ± 0.019**	**0.116 ± 0.027**	0.280 ± 0.006	**0.261 ± 0.008**
MLACO	0.222 ± 0.026	0.135 ± 0.035	0.305 ± 0.005	0.301 ± 0.017
MLPSO	0.180 ± 0.022	0.192 ± 0.033	0.302 ± 0.003	0.300 ± 0.022
HBALO	0.216 ± 0.024	0.160 ± 0.033	**0.275 ± 0.006**	0.299 ± 0.008
BMFS	0.195 ± 0.024	0.204 ± 0.029	0.305 ± 0.003	0.270 ± 0.009
MMDE	0.214 ± 0.025	0.213 ± 0.026	0.285 ± 0.007	0.331 ± 0.011
Method	**Tmc2007**	**Yeast**	**Avg. Rank**	
Proposed	**0.209 ± 0.004**	**0.563 ± 0.010**	**1.4**	
MLACO	0.229 ± 0.006	0.566 ± 0.010	3.9	
MLPSO	0.274 ± 0.018	0.586 ± 0.008	3.9	
HBALO	0.261 ± 0.006	0.586 ± 0.012	3.8	
BMFS	0.211 ± 0.002	0.610 ± 0.010	3.4	
MMDE	0.297 ± 0.020	0.573 ± 0.010	4.7	

Table 8 Win/tie/loss results of the paired *t*-test for *multilabel accuracy*, where algorithms demonstrating superior performance are highlighted in bold

Dataset	Proposed	MLACO	MLPSO	HBALO	BMFS	MMDE
Business	2/2/1	1/3/1	1/3/1	1/2/2	5/0/0	0/0/5
Computers	3/2/0	0/2/3	3/2/0	0/1/4	3/2/0	1/1/3
Education	4/1/0	1/1/3	1/2/2	2/1/2	0/0/5	4/1/0
Emotions	4/1/0	3/0/2	2/0/3	0/1/4	4/1/0	0/1/4
Enron	4/0/1	1/2/2	2/1/2	0/0/5	5/0/0	1/1/3
Entertainment	2/2/1	2/2/1	1/0/4	2/2/1	0/0/5	5/0/0
Genbase	4/1/0	4/1/0	1/1/3	3/0/2	0/0/5	1/1/3
Health	4/1/0	0/0/5	2/0/3	3/0/2	4/1/0	1/0/4
Llog	1/4/0	1/4/0	1/4/0	1/4/0	1/4/0	0/0/5
Medical	4/1/0	2/3/0	2/1/2	3/1/1	0/0/5	1/0/4
Recreation	3/1/1	5/0/0	1/1/3	3/1/1	0/0/5	1/1/3
Scene	5/0/0	1/1/3	4/0/1	3/0/2	1/1/3	0/0/5
Tmc2007	5/0/0	3/0/2	0/1/4	2/0/3	4/0/1	0/1/4
Yeast	5/0/0	1/2/2	3/1/1	0/0/5	1/2/2	1/3/1
Total	**50/16/4**	25/21/24	24/17/29	23/13/34	28/11/31	16/10/44

Table 9 Win/tie/loss results of the paired *t*-test for *one-error*, where algorithms demonstrating superior performance are highlighted in bold

Dataset	Proposed	MLACO	MLPSO	HBALO	BMFS	MMDE
Business	2/1/2	3/2/0	3/2/0	1/0/4	2/3/0	0/0/5
Computers	5/0/0	1/2/2	2/2/1	0/0/5	1/3/1	1/3/1
Education	3/1/1	0/1/4	2/0/3	3/2/0	0/1/4	4/1/0
Emotions	5/0/0	1/3/1	1/3/1	1/3/1	1/3/1	0/0/5
Enron	3/2/0	1/2/2	1/2/2	0/1/4	3/2/0	0/5/0
Entertainment	3/2/0	1/1/3	1/1/3	3/1/1	0/0/5	4/1/0
Genbase	1/4/0	1/4/0	0/5/0	0/3/2	0/5/0	0/5/0
Health	1/3/1	1/2/2	5/0/0	0/0/5	3/1/1	1/2/2
Llog	0/4/1	0/4/1	5/0/0	0/4/1	0/4/1	0/4/1
Medical	5/0/0	2/1/2	4/0/1	0/3/2	1/1/3	0/1/4
Recreation	3/1/1	4/1/0	1/0/4	3/2/0	0/0/5	2/0/3
Scene	5/0/0	1/0/4	2/2/1	2/1/2	3/1/1	0/0/5
Tmc2007	5/0/0	2/1/2	0/1/4	2/1/2	4/0/1	0/1/4
Yeast	3/2/0	3/2/0	3/2/0	0/2/3	0/1/4	1/1/3
Total	**44/20/6**	21/26/23	30/20/20	15/23/32	18/25/27	13/24/33

Table 10 Win/tie/loss results of the paired *t*-test for *ranking loss*, where algorithms demonstrating superior performance are highlighted in bold

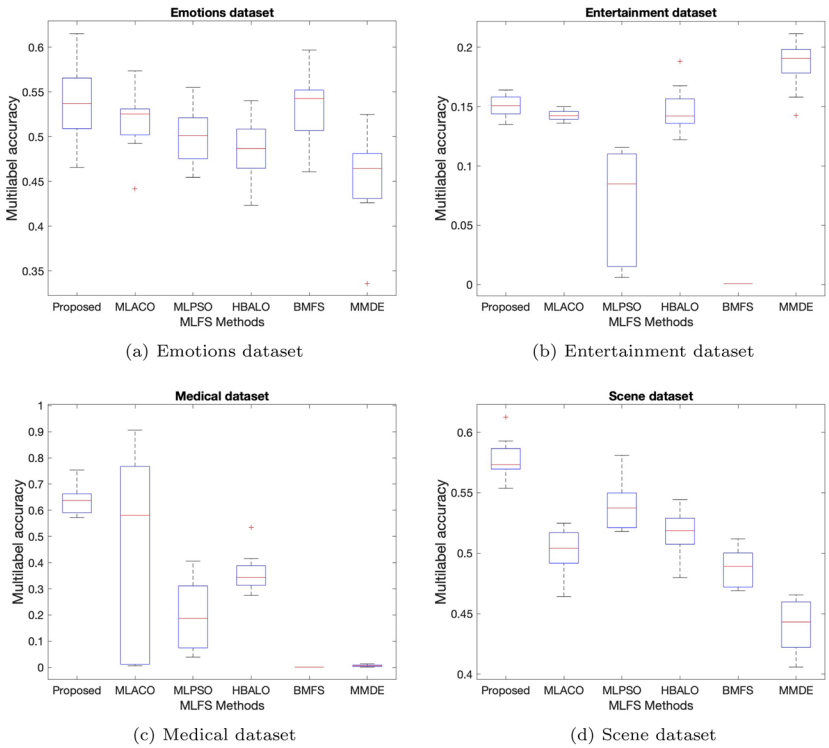
Dataset	Proposed	MLACO	MLPSO	HBALO	BMFS	MMDE
Business	1/3/1	1/3/1	1/3/1	1/3/1	5/0/0	0/0/5
Computers	5/0/0	2/2/1	2/2/1	0/2/3	0/4/1	0/2/3
Education	2/3/0	0/3/2	0/5/0	2/3/0	0/3/2	0/5/0
Emotions	5/0/0	2/2/1	0/1/4	2/2/1	2/2/1	0/1/4
Enron	4/1/0	0/3/2	2/1/2	0/2/3	4/1/0	0/2/3
Entertainment	5/0/0	0/1/4	2/1/2	3/1/1	0/1/4	2/2/1
Genbase	4/1/0	4/1/0	1/1/3	3/0/2	0/1/4	0/2/3
Health	3/2/0	0/1/4	3/2/0	2/0/3	3/2/0	0/1/4
Llog	4/1/0	0/1/4	3/2/0	0/2/3	3/1/1	1/1/3
Medical	4/1/0	2/3/0	2/1/2	3/1/1	0/1/4	0/1/4
Recreation	4/0/1	1/1/3	1/1/3	5/0/0	0/0/5	3/0/2
Scene	5/0/0	1/0/4	2/0/3	3/1/1	3/1/1	0/0/5
Tmc2007	4/1/0	3/0/2	1/1/3	1/1/3	4/1/0	0/0/5
Yeast	5/0/0	2/2/1	1/3/1	1/2/2	0/0/5	1/3/1
Total	**55/13/2**	18/23/29	21/24/25	26/20/24	24/18/28	7/20/43

In-depth analysis

This section analyzes the effectiveness of the proposed method by monitoring the performance changes based on the hyperparameters of the proposed method. In the proposed method, we employed an explicit pruning method to determine the label combinations Z, and $|Z|$ was varied according to the pruning results. However, a constant size of Z may yield better results owing to its simplicity. To validate this, we defined $p\% = \frac{|Z|}{|L|}$ as the percentage of the size of Z relative to the total number of labels. In the proposed method, the value p is dynamically determined through a

Table 11 Win/tie/loss results of the paired *t*-test for *multilabel coverage*, where algorithms demonstrating superior performance are highlighted in bold

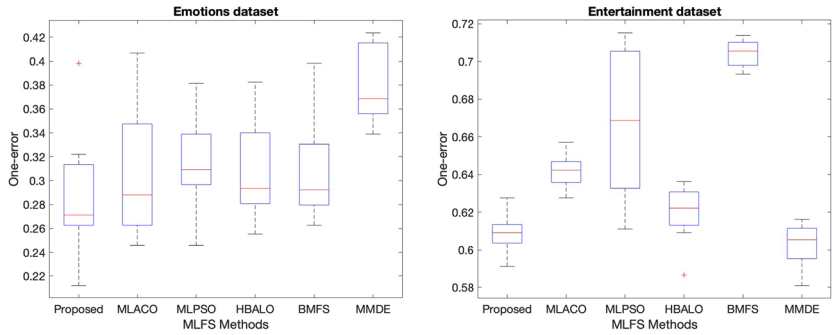
Dataset	Proposed	MLACO	MLPSO	HBALO	BMFS	MMDE
Business	1/3/1	1/3/1	1/3/1	1/3/1	5/0/0	0/0/5
Computers	4/1/0	3/1/1	0/3/2	0/2/3	2/3/0	0/2/3
Education	3/2/0	0/3/2	0/3/2	3/2/0	0/3/2	0/5/0
Emotions	3/2/0	3/2/0	1/1/3	0/2/3	3/2/0	0/1/4
Enron	4/0/1	0/3/2	0/3/2	0/3/2	5/0/0	0/3/2
Entertainment	5/0/0	0/2/3	0/2/3	3/1/1	0/2/3	3/1/1
Genbase	4/1/0	4/1/0	0/0/5	3/0/2	1/1/3	1/1/3
Health	4/1/0	0/1/4	2/3/0	2/1/2	3/1/1	0/1/4
Llog	4/1/0	0/1/4	3/2/0	0/2/3	3/1/1	1/1/3
Medical	4/1/0	3/2/0	1/1/3	3/1/1	0/1/4	0/2/3
Recreation	4/1/0	0/2/3	1/1/3	4/1/0	0/1/4	3/0/2
Scene	5/0/0	1/2/2	1/2/2	1/2/2	4/0/1	0/0/5
Tmc2007	4/1/0	3/0/2	1/1/3	1/1/3	4/1/0	0/0/5
Yeast	4/1/0	3/2/0	1/1/3	1/1/3	0/0/5	3/1/1
Total	**53/15/2**	21/25/24	12/26/32	22/22/26	30/16/24	11/18/41



(a) Emotions dataset (b) Entertainment dataset

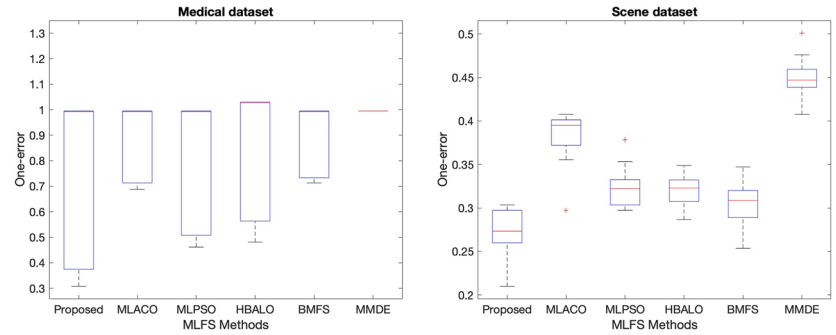
(c) Medical dataset (d) Scene dataset

Fig. 2 Box plot of four evaluation metrics for the proposed method and the compared methods



(a) Emotions dataset

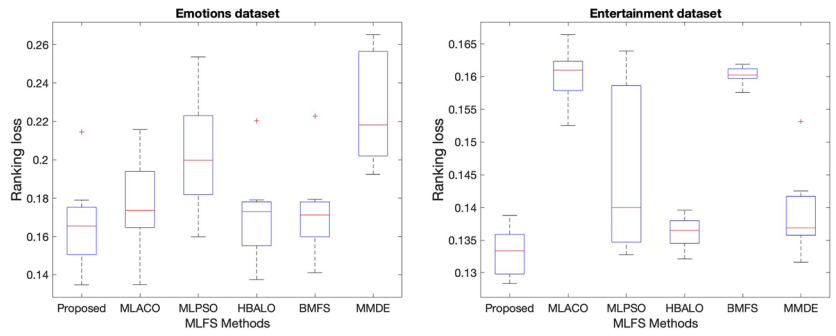
(b) Entertainment dataset



(c) Medical dataset

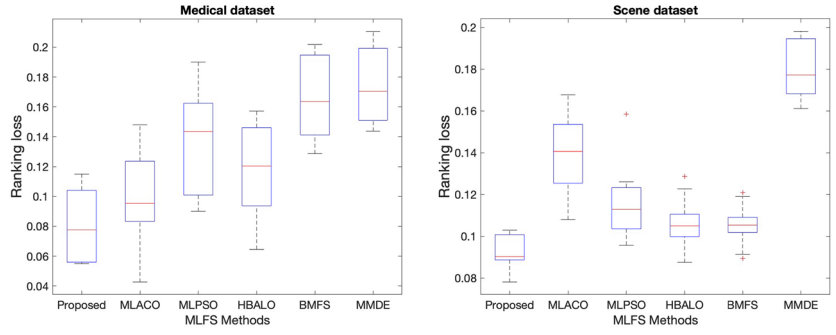
(d) Scene dataset

Fig. 2 continued



(a) Emotions dataset

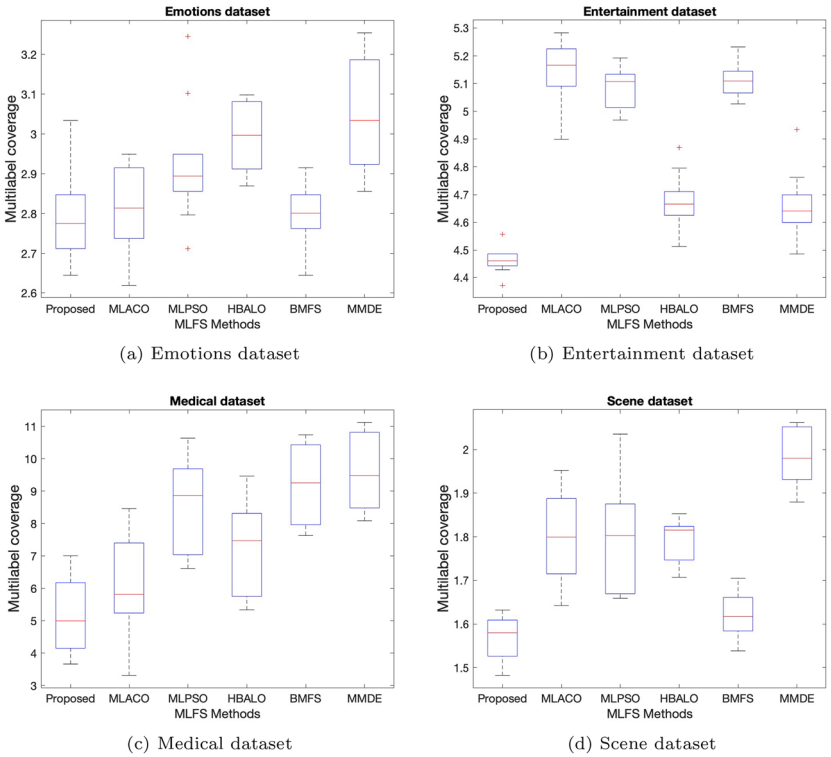
(b) Entertainment dataset



(c) Medical dataset

(d) Scene dataset

Fig. 2 continued



(a) Emotions dataset (b) Entertainment dataset

(c) Medical dataset (d) Scene dataset

Fig. 2 continued

pruning process according to the accuracy value of each label in Z. To confirm its effectiveness, we compared the performance of the proposed method with its counterpart, which employed a constant p value. Figure 5 shows the multilabel accuracies for the four multilabel datasets. The horizontal axis represents a constant p in Z, and the vertical axis indicates the multilabel accuracy performance. The dotted black line indicates the performance values of the multilabel accuracy at constant p values ranging from 0.1 to 0.5, and the solid red line indicates the performance of the proposed method, which dynamically determines p. For example, for $|L| = 100$, the proposed method using a constant $p = 0.1$ always selects the ten labels with the lowest accuracy for each label in Z. For all p values, the proposed method using the dynamic p achieved higher accuracy than that using a constant p, except for the Yeast dataset.

Next, regarding the success ratio of the guessing process, we conducted additional experiments by comparing the success ratios of improving the fitness of individuals in the population. Figure 6 shows the success ratios of the performance improvements after the reproduction process when a specific number of FFCs is spent during the experiments on the four multilabel datasets. The figure comprises two groups of bars; the first and second groups show the success ratios in terms of multilabel accuracy and average accuracy, respectively. We considered the average accuracy that can be obtained by averaging the accuracy values of the labels because our guessing process was based on the accuracy of each label. The success ratio was calculated by dividing the number of

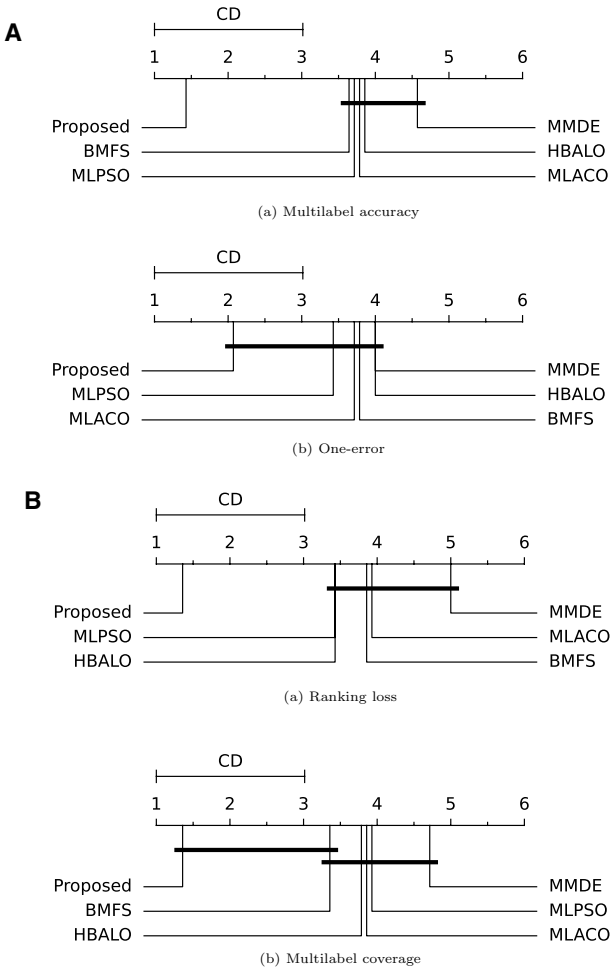


Fig. 3 Bonferroni–Dunn test results of the five comparison methods for four evaluation metrics

improvements by the total number of reproduction trials. Additionally, we assessed the success ratios when the number of consumed FFCs reached 300 (termination condition for the process) and 150 in the middle of the termination. The experimental results indicate that as the algorithm proceeded from 150 to 300 FFCs, the success ratio decreased because candidates in the population of 300 FFCs had good fitness values compared to those in the population of 150 FFCs. For the Genbase dataset, all bars were higher than 0.5, as indicated by the gray dotted line, meaning that the improvements occurred frequently until algorithm termination. For the Enron and Yeast datasets, the ratio was only lower than 0.5 for the multilabel accuracy when 300 FFCs were consumed. The results for the averaged accuracy of Z demonstrate that the created feature subsets were more likely to improve. Finally, for the Medical dataset, unlike the aforementioned results, the probability of improvements remained higher than 0.5 for the multilabel accuracy until the algorithm terminated. When the number of consumed FFCs was approximately 300, the average accuracy was less than 0.5. In conclusion, most of the experimental results showed that the success ratio was higher than 0.5, indicating that frequent performance improvements were observed.

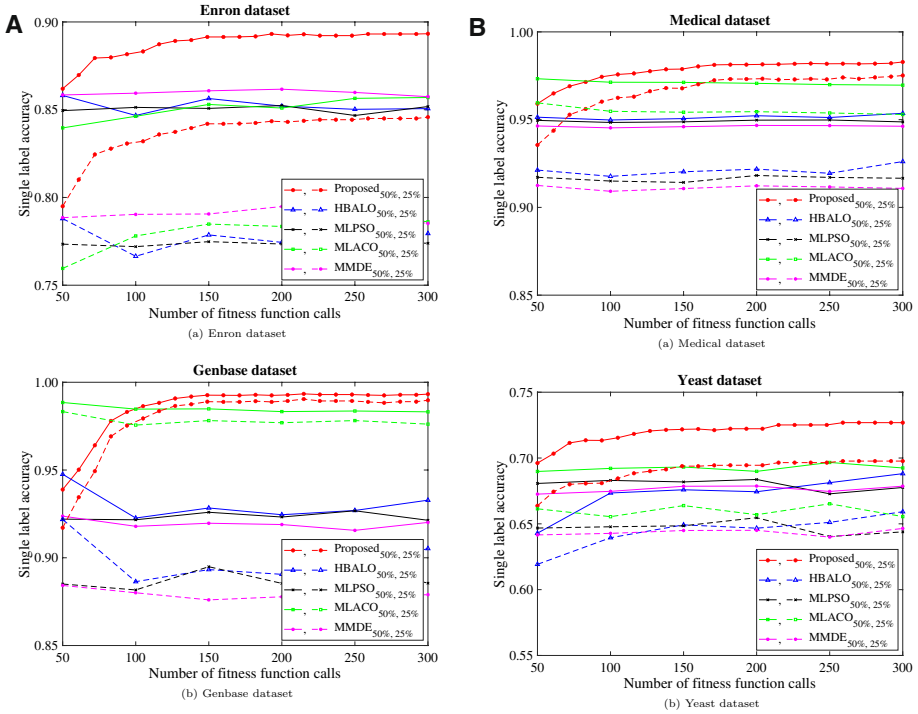


Fig. 4 Average accuracies for the lowest 50% and 25% labels for each FFC

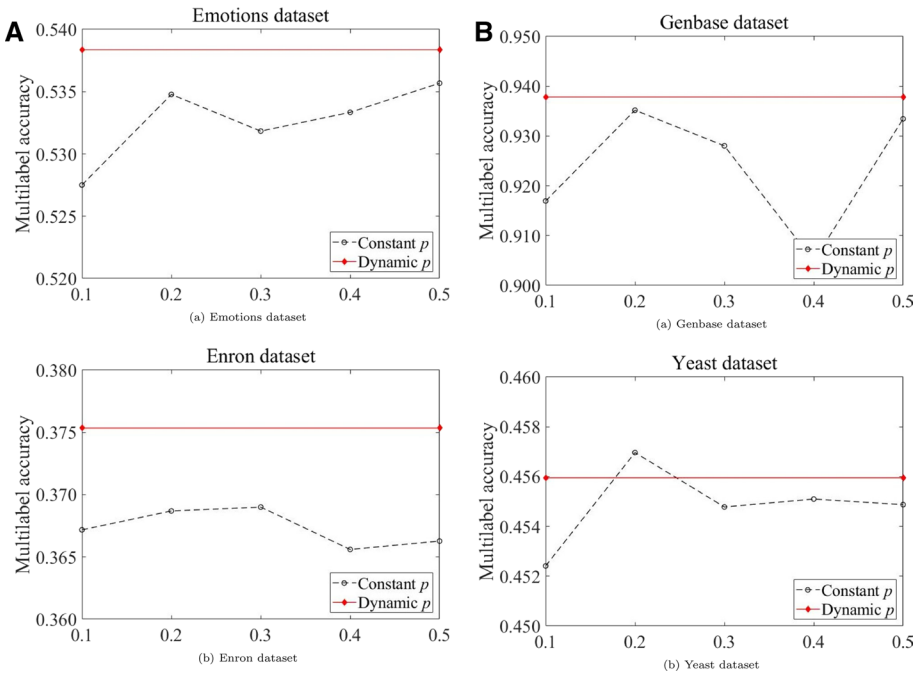


Fig. 5 Multilabel accuracies for the four datasets with varying *p*% and the proposed dynamic *p*%

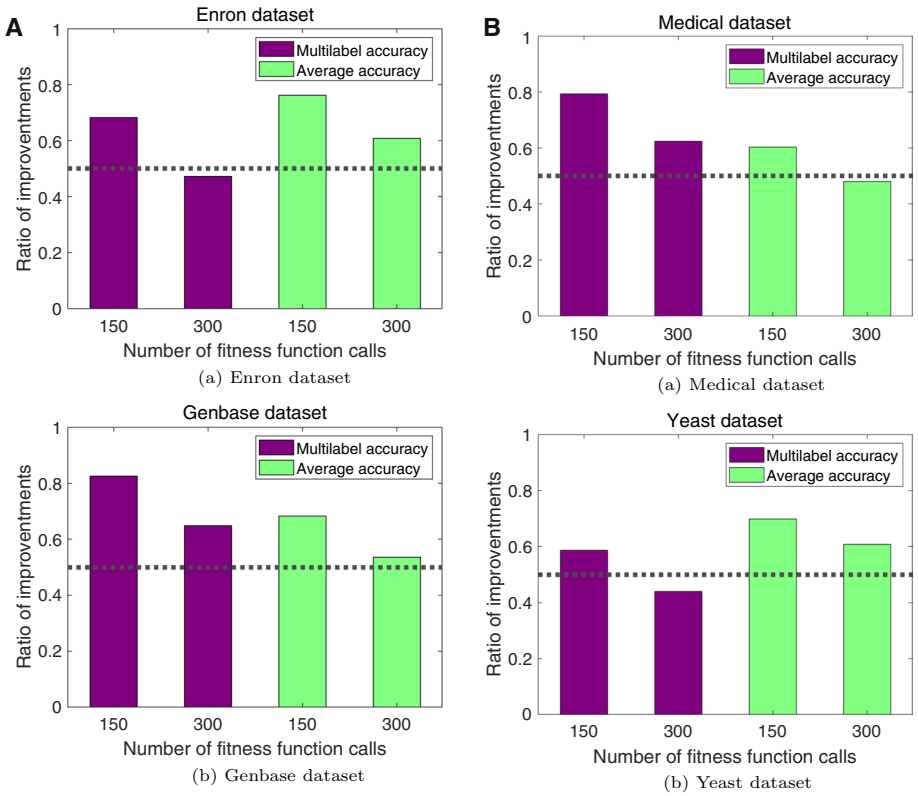


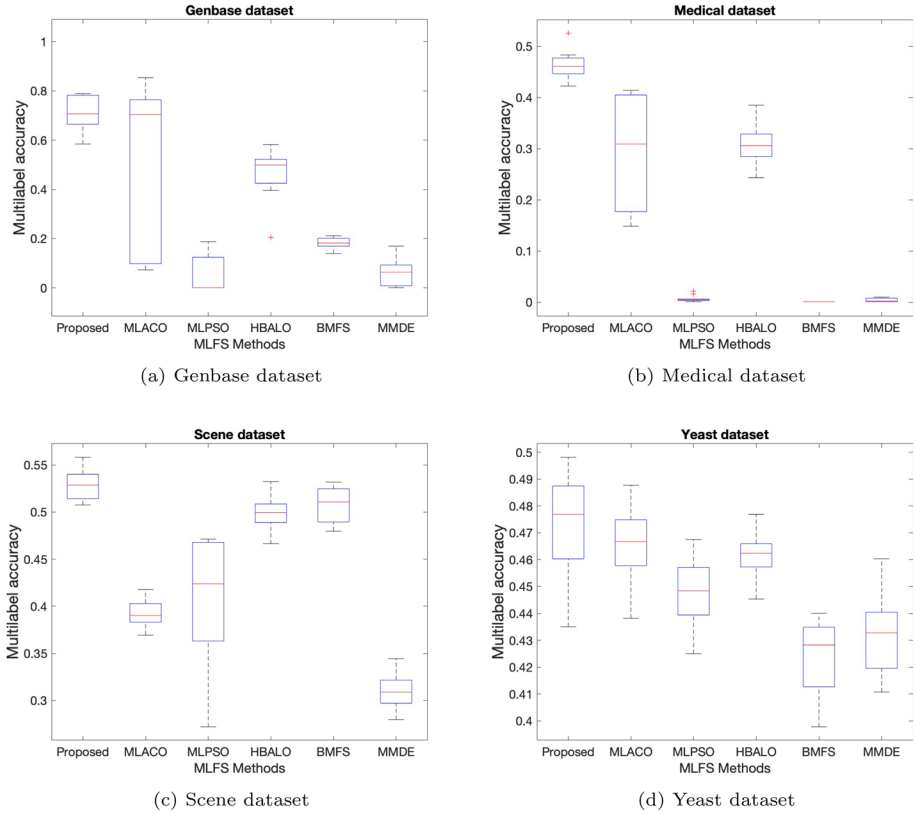
Fig. 6 Performance improvement ratios of multilabel and average accuracies of Z for feature subsets created from the selected feature subsets

Furthermore, regarding the number of selected features, we conducted additional experiments to analyze the performance of the proposed method when the maximum number of selected features (n_{max}) is set to 25. Figure 7 shows the box plots of the multilabel accuracy for the proposed method and the compared methods on the four datasets: the Genbase, Medical, Scene, and Yeast. Similar to Fig. 2, the proposed method consistently outperformed the other methods across all datasets, indicating the effectiveness of the proposed method.

Conclusions

In this study, we proposed an effective evolutionary multilabel feature selection. We introduced a novel method for effectively exploiting a label combination that could improve the performance of feature subsets. The experimental results and statistical tests showed that the proposed method significantly outperformed the five state-of-the-art feature selection methods on 14 multilabel datasets.

Future studies should aim to overcome the limitations of the proposed method. In each iteration, the feature subsets were changed by as much as the ratio of the size of pruned labels. However, the number of features to be changed can be adjusted mathematically depending on the redundancy between them. For instance, the redundancy can be measured by an information-theoretic score function and can be a weight



(a) Genbase dataset (b) Medical dataset

(c) Scene dataset (d) Yeast dataset

Fig. 7 Box plot of the multilabel accuracy for the proposed method and the compared methods when $n_{max} = 25$

parameter to adjust the ratio. When the redundancy is high, the ratio should be increased to improve the performance of the feature subsets. This method can further improve the success ratio of the proposed refinement process by allowing the replacement of redundant features with more effective ones.

Moreover, integrating the proposed method into the training process with a subsequent learner could enhance the performance of the method. For instance, both the proposed method and decision trees utilize pruning processes to reduce model complexity, yet the interaction between these two pruning processes is not well understood. Additionally, future studies should investigate various initialization methods to maximize the effectiveness of label relations in the initial population.

Author contributions

W.S., J.P., J.L. proposed the proposed hybrid MLFS. A.M. and S.L. obtained the datasets for research. Rich experience of D.K. and J.L. was instrumental in improving our work. J.P. and W.S. did the literature survey of the paper, and W.D. and J.L. wrote the main manuscript. All authors contributed to the editing and proofreading. All authors read and approved the final manuscript.

Funding

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-01341, Artificial Intelligence Graduate School Program (Chung-Ang University)) and the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. 2020R1A2C101357511).

Availibility of data and materials

The data that support the findings of this study are available in https://mulan.sourceforge.net/datasets-mlc.html.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no Conflict of interest.

Received: 23 March 2024 Accepted: 13 July 2024
Published online: 06 August 2024

References

1. Hancock JT, Wang H, Khoshgoftaar TM, Liang Q. Data reduction techniques for highly imbalanced medicare big data. J Big Data. 2024;11(1):8.
2. Kayikci S, Khoshgoftaar TM. Blockchain meets machine learning: a survey. J Big Data. 2024;11(1):9.
3. Devi AA, Babu ES. A lightweight multi-vector DDoS detection framework for IOT-enabled mobile health informatics systems using deep learning. Inf Sci. 2024;662: 120209.
4. Liu W, Wang H, Shen X, Tsang IW. The emerging trends of multi-label learning. IEEE Trans Pattern Anal Mach Intell. 2021;44(11):7955–74.
5. Zhang M-L, Zhou Z-H. ML-kNN: a lazy learning approach to multi-label learning. Pattern Recogn. 2007;40(7):2038–48.
6. Zhang Y, Wu J, Cai Z, Philip SY. Multi-view multi-label learning with sparse feature selection for image annotation. IEEE Trans Multimedia. 2020;22(11):2844–57.
7. Liu B, Liu X, Ren H, Qian J, Wang Y. Text multi-label learning method based on label-aware attention and semantic dependency. Multimed Tools Appl. 2022;81(5):7219–37.
8. Deng H, Ding M, Wang Y, Li W, Liu G, Tang Y. ACP-MLC: a two-level prediction engine for identification of anticancer peptides and multi-label classification of their functional types. Comput Biol Med. 2023;158: 106844.
9. Fan J, Nichols E, Tompkins D, Méndez AEM, Elizalde B, Pasquier P. Multi-label sound event retrieval using a deep learning-based siamese structure with a pairwise presence matrix. In: Proceedings—ICASSP IEEE international conference acoustics speech signal process. Barcelona, Spain. IEEE; 2020. p. 3482–6.
10. Kim H-C, Park J-H, Kim D-W, Lee J. Multilabel naïve bayes classification considering label dependence. Pattern Recognit Lett. 2020;136:279–85.
11. Zhang M-L, Peña JM, Robles V. Feature selection for multi-label naïve Bayes classification. Inf Sci. 2009;179(19):3218–29.
12. Zhang M-L, Zhang K. Multi-label learning by exploiting label dependency. In: Proceedings of the 16th ACM SIGKDD international conference Knowledge. Discovery Data Mining; 2010. p. 999–1008.
13. Lee J, Kim D-W. Mutual information-based multi-label feature selection using interaction information. Expert Syst Appl. 2015;42(4):2013–25.
14. Zhang P, Liu G, Gao W. Distinguishing two types of labels for multi-label feature selection. Pattern Recogn. 2019;95:72–82.
15. Demir K, Nguyen B, Xue B, Zhang M. Co-operative co-evolutionary many-objective embedded multi-label feature selection with decomposition-based PSO. In: Proceedings of the 2023 genetic and evolutionary computation conference, Lisbon, Portugal; 2023. p. 438–46.
16. Bidgoli AA, Ebrahimpour-Komleh H, Rahnamayan S. Reference-point-based multi-objective optimization algorithm with opposition-based voting scheme for multi-label feature selection. Inf Sci. 2021;547:1–17.
17. Ahadzadeh B, Abdar M, Safara F, Khosravi A, Menhaj MB, Suganthan PN. SFE: a simple, fast and efficient feature selection algorithm for high-dimensional data. IEEE Trans Evol Comput. 2023;27:1896–911.
18. Shaban WM, Rabie AH, Saleh AI, Abo-Elsoud M. A new covid-19 patients detection strategy (CPDS) based on hybrid feature selection and enhanced KNN classifier. Knowl Based Syst. 2020;205:106270.
19. Zhu Y, Li W, Li T. A hybrid artificial immune optimization for high-dimensional feature selection. Knowl Based Syst. 2023;260:110111.
20. Liu J, Lin Y, Ding W, Zhang H, Du J. Fuzzy mutual information-based multi-label feature selection with label dependency and streaming labels. IEEE Trans Fuzzy Syst. 2022;31:77–91.
21. Dai J, Huang W, Zhang C, Liu J. Multi-label feature selection by strongly relevant label gain and label mutual aid. Pattern Recogn. 2024;145: 109945.
22. Fan Y, Liu J, Tang J, Liu P, Lin Y, Du Y. Learning correlation information for multi-label feature selection. Pattern Recogn. 2024;145: 109899.
23. Priya RD, Sivaraj R, Anitha N, Devisurya V. Tri-staged feature selection in multi-class heterogeneous datasets using memetic algorithm and cuckoo search optimization. Expert Syst Appl. 2022;209: 118286.
24. Qian W, Huang J, Wang Y, Shu W. Mutual information-based label distribution feature selection for multi-label learning. Knowl Based Syst. 2020;195:105684.
25. Seo W, Kim D-W, Lee J. Generalized information-theoretic criterion for multi-label feature selection. IEEE Access. 2019;7:122854–63.
26. Zhang P, Liu G, Gao W, Song J. Multi-label feature selection considering label supplementation. Pattern Recogn. 2021;120: 108137.

27. Hu L, Gao L, Li Y, Zhang P, Gao W. Feature-specific mutual information variation for multi-label feature selection. Inf Sci. 2022;593:449–71.
28. Hu J, Li Y, Xu G, Gao W. Dynamic subspace dual-graph regularized multi-label feature selection. Neurocomputing. 2022;467:184–96.
29. Huang R, Wu Z. Multi-label feature selection via manifold regularization and dependence maximization. Pattern Recogn. 2021;120: 108149.
30. Faraji M, Seyedi SA, Tab FA, Mahmoodi R. Multi-label feature selection with global and local label correlation. Expert Syst App. 2024;246: 123198.
31. Hashemi A, Dowlatshahi MB, Nezamabadi-Pour H. A bipartite matching-based feature selection for multi-label learning. Int J Mach Learn Cybern. 2021;12(2):459–75.
32. Paniri M, Dowlatshahi MB, Nezamabadi-pour H. MLACO: a multi-label feature selection algorithm based on ant colony optimization. Knowl Based Syst. 2020;192:105285.
33. Paniri M, Dowlatshahi MB, Nezamabadi-pour H. Ant-TD: ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection. Swarm Evol Comput. 2021;64: 100892.
34. Hancer E, Xue B, Zhang M. A multimodal multi-objective evolutionary algorithm for filter feature selection in multi-label classification. IEEE Trans Artifi Intell. 2024. https://doi.org/10.1109/TAI.2024.3380590.
35. Dokeroglu T, Deniz A, Kiziloz HE. A comprehensive survey on recent metaheuristics for feature selection. Neurocomputing. 2022;494:269–96.
36. Xue B, Zhang M, Browne WN, Yao X. A survey on evolutionary computation approaches to feature selection. IEEE Trans Evol Comput. 2016;20(4):606–26.
37. Lim H, Kim D-W. MFC: initialization method for multi-label feature selection based on conditional mutual information. Neurocomputing. 2020;382:40–51.
38. Lu H, Chen J, Yan K, Jin Q, Xue Y, Gao Z. A hybrid feature selection algorithm for gene expression data classification. Neurocomputing. 2017;256:56–62.
39. Mafarja MM, Mirjalili S. Hybrid binary ant lion optimizer with rough set and approximate entropy reducts for feature selection. Soft Comput. 2019;23(15):6249–65.
40. Lee J, Yu I, Park J, Kim D-W. Memetic feature selection for multilabel text categorization using label frequency difference. Inf Sci. 2019;485:263–80.
41. Maseno EM, Wang Z. Hybrid wrapper feature selection method based on genetic algorithm and extreme learning machine for intrusion detection. J Big Data. 2024;11(1):24.
42. Park J, Park M-W, Kim D-W, Lee J. Multi-population genetic algorithm for multilabel feature selection based on label complementary communication. Entropy. 2020;22(8):876.
43. Seo W, Park M, Kim D-W, Lee J. Effective memetic algorithm for multilabel feature selection using hybridization-based communication. Expert Syst Appl. 2022;201: 117064.
44. Demir K, Nguyen BH, Xue B, Zhang M. Dual sparse structured subspaces and graph regularisation for particle swarm optimisation-based multi-label feature selection. IEEE Comput Intell Mag. 2024;19(1):36–50.
45. Chavent M. A monothetic clustering method. Pattern Recogn Lett. 1998;19(11):989–96.
46. Miller BL, Goldberg DE. Genetic algorithms, tournament selection, and the effects of noise. Complex Syst. 1995;9(3):193–212.
47. Zhang Y, Gong D-W, Sun X-Y, Guo Y-N. A PSO-based multi-objective multi-label feature selection method in classification. Sci Rep. 2017;7(376):1–12.
48. Trohidis K, Tsoumakas G, Kalliris G, Vlahavas IP, et al. Multi-label classification of music into emotions. ISMIR. 2008;8:325–30.
49. Klimt B, Yang Y. The enron corpus: A new dataset for email classification research. In: Proceedings of the European conference on machine learning, Pisa, Italy, Springer; 2004. p. 217–26.
50. Zhang M-L, Wu L. Lift: multi-label learning with label-specific features. IEEE Trans Pattern Anal Mach Intell. 2014;37(1):107–20.
51. Diplaris S, Tsoumakas G, Mitkas PA, Vlahavas I. Protein classification with multiple algorithms. In: Proceedings of the Panhellenic conference on informatics, Volos, Greece. Springer; 2005. p. 448–56. Springer
52. Elisseeff A, Weston J. A kernel method for multi-labelled classification. Adv Neural Inf Process Syst. 2001;14:681–7.
53. Pestian J, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Cohen KB, Duch W. A shared task involving multi-label classification of clinical free text. In: Proceedings of the biological, translational and clinical language processing; 2007. p. 97–104.
54. Boutell MR, Luo J, Shen X, Brown CM. Learning multi-label scene classification. Pattern Recogn. 2004;37(9):1757–71.
55. Srivastava AN, Zane-Ulman B. Discovering recurring anomalies in text reports regarding complex space systems. In: 2005 IEEE Aerospace conference: Big Sky, MT, USA. IEEE; 2005. p. 3853–62.
56. Ueda N, Saito K. Parametric mixture models for multi-labeled text. In: Advances in neural information processing system; 2003. p. 737–44.
57. Cano A, Luna JM, Gibaja EL, Ventura S. LAIM discretization for multi-label data. Inf Sci. 2016;330(1):370–84.
58. Nair-Benrekia N-Y, Kuntz P, Meyer F. Learning from multi-label data with interactivity constraints: an extensive experimental study. Expert Syst Appl. 2015;42(13):5723–36.
59. Zhang M-L, Zhou Z-H. A review on multi-label learning algorithms. IEEE Trans Knowl Data Eng. 2014;26(8):1819–37.
60. Sun Z, Zhang J, Dai L, Li C, Zhou C, Xin J, Li S. Mutual information based multi-label feature selection via constrained convex optimization. Neurocomputing. 2019;329:447–56.
61. Demsar J. Statistical comparisons of classifier over multiple data sets. J Mach Learn Res. 2006;7(1):1–30.

Publisher's Note