Journal of Data Science and Modeling, Vol. 2, No. 2, 35-70, June 2024

Research Manuscript

Multi-Objective Interaction-Enhanced Feature Selection for Streaming Multi-Label Data

Sahar Abbasi*[1], Maryam Hamedi[2] , Ramin Sadeghian[3]

[1]PhD student, Payame Noor University, P.O. Box 3697-19395, Tehran, Iran

[2]Assistant Professor, Department of Industrial Engineering, Payame Noor University, P.O. Box 3697-19395, Tehran, Iran

[3]Associate Professor, Department of Industrial Engineering, Payame Noor University, P.O. Box 3697-19395, Tehran, Iran

Recieved: 02/07/2024 Accepted: 26/11/2024

Abstract:

Multi-label classification assigns multiple labels to each instance, crucial for tasks like cancer detection in images and text categorization. However, machine learning methods often struggle with the complexity of real-life datasets. To improve efficiency, researchers have developed feature selection methods to identify the most relevant features. Traditional methods, requiring all features upfront, fail in dynamic environments like media platforms with continuous data streams. To address this, novel online methods have been created, yet they often neglect optimizing conflicting objectives. This study introduces an objective search approach using mutual information, feature interaction, and the NSGA-II algorithm to select relevant features from streaming data. The strategy aims to minimize feature overlap, maximize relevance to labels, and optimize online feature interaction analysis. By applying a modified NSGA-II algorithm, a set of non-dominant solutions is identified. Experiments on eleven datasets show that the proposed approach outperforms advanced online feature selection techniques in predictive accuracy, statistical analysis, and stability assessment.

Keywords: Streaming Data Feature selection, Online Multi-Label Learning, Feature interaction, Multi-Objective Optimization, Mutual Information.

Mathematics Subject Classification (2010): 68-XX, 68-04.

*Corresponding Author: S.Abbasi@pnu.ac.ir

1. Introduction

Multi-label feature selection (MFS) is crucial for managing high-dimensional labeled data, prevalent in applications like text classification, music tagging, image recognition, and biology. In multi-label learning, each instance is associated with multiple labels, often resulting in redundant features (Liu *et al.* (2021), Shrivastava *et al.* (2020), Liang *et al.* (2022), Liang *et al.* (2019)). MFS aims to improve prediction accuracy and model interpretability.

Dimensionality reduction, encompassing feature extraction and selection, addresses this redundancy. Feature extraction maps features to a lower-dimensional space, creating new combined features (e.g., Xu *et al.* (2016), Yu *et al.* (2005), Xu (2018)). Conversely, feature selection chooses a relevant and non-redundant subset of original features.

MFS methods are categorized as filter, wrapper, and embedded. Filter methods evaluate feature subsets using information theory without classifier training (Hatami *et al.* (2020), Seo *et al.* (2022)). Wrapper methods, while potentially more accurate, require classifier training for each subset, incurring high computational costs (Zhang *et al.* (2017)). Embedded methods combine the advantages of both (Zhu *et al.* (2018)).

Traditional MFS assumes all features are known a priori (Zhang *et al.* (2020), Li *et al.* (2023), Huang *et al.* (2023), Wang *et al.* (2022)), which is often unrealistic. In real-world scenarios, features may become available gradually, posing challenges for real-time processing (e.g., video recognition) (Wu *et al.* (2012), Hu *et al.* (2018), You *et al.* (2021), Gomes *et al.* (2019)).

Existing methods often require access to the entire feature space, limiting their applicability to dynamic scenarios where features emerge over time (e.g., Twitter). Online feature selection methods address this, including mutual information-based approaches (Gonzalez-Lopez *et al.* (2019)), fuzzy-based streaming methods (Lin *et al.* (2017)), and neighborhood rough set approaches (Liu *et al.* (2018)). These methods prioritize features as they arrive. However, existing approaches have limitations, including pre-algorithm data understanding, computational time, complexity, and optimal feature number determination. Current methods also primarily focus on single-label problems and often employ single-objective strategies, whereas a multi-objective approach could be more effective.

Effective online multi-label streaming feature selection requires no prior domain knowledge, incremental feature updating, and acceptable classification performance at each time instance.

This paper proposes a novel Multi-Objective Online Streaming Multi-Label Feature Selection method, MIENS-FS, integrating feature interaction, mutual information, and dynamic adaptation to streaming data. MIENS-FS focuses on selecting relevant features and adapting to dynamic feature interactions, crucial for real-time applications where the entire feature space is unavailable. The algorithm efficiently updates the feature selection model with new data, avoiding reprocessing the entire stream.

Unlike most previous approaches using a single objective function, this paper uses the Pareto set to determine optimal features balancing relevance and redundancy. The key contributions are:

- **Dynamic Interaction:** Defining feature interaction and assessing its influence across labels, MIENS-FS combines this with mutual information to select interactive features.
- **Multi-objective Approach:** Unlike single-objective methods (e.g., Gonzalez-Lopez *et al.* (2019), Lin *et al.* (2017), Liu *et al.* (2018), You *et al.* (2012)), this method considers both relevance and redundancy.
- **Integration of Mutual Information with NSGA-II:** Using mutual information within the NSGA-II framework offers advantages over rough set theory (Ma *et al.* (2022), Zou *et al.* (2021)) due to broader applicability and lower computational complexity.
- **Adaptive Mutation Strategy:** An adaptive mutation strategy based on feature-label mutual information enhances exploration.
- **Pareto Front Analysis:** Using the Pareto front for feature selection balances relevance and redundancy while considering feature interaction.

The paper is structured as follows: Section 2 reviews related work, Section 3 details multi-label learning and mutual information, Section 4 presents the proposed methodology, Section 5 discusses experimental results, and Sections 6 and **??** conclude and outline future work.

2. Literature Review

Feature selection reduces the number of dataset features by removing unnecessary and redundant ones. Feature selection methods are classified as offline or online, depending on whether a global feature space is assumed.

Offline feature selection methods assume a pre-established global feature space. These are further divided into single-label (one label per instance) and multi-label (multiple labels per instance) methods. Single-label methods include filter-based particle swarm optimization (Zhang *et al.* (2019)), multi-objective genetic algorithms for text feature selection (MORDC) (Labani *et al.* (2020)), variable-size cooperative coevolutionary particle swarm optimization (VSCCPSO) (Song *et al.* (2020)), and graph clustering with ant colony optimization (Tabakhi and Moradi (2015)). Multi-label offline methods can be categorized into those that convert the multi-label problem into single-label problems before applying single-label feature selection. Examples include MDMR, an incremental multi-label feature selection method using mutual information and a max-dependency min-redundancy criterion (Lin *et al.* (2015)), and PMU, a mutual information-based method maximizing multivariate mutual information between selected features and class labels using an incremental selection strategy (Lin *et al.* (2017)). However, methods like PMU and MDMR with adaptive strategies can be slow due to their greedy search. Graph-based multi-label feature selection (MGFS) computes a correlation distance matrix (CDM) and uses PageRank (Lee and Kim (2013)), but ignores redundancy between selected features. Other methods include MLACO (Ant Colony Optimization) (Hashemi *et al.* (2020)), MGFS (a faster version) (Lee and Kim (2013)), manifold-based constraint Laplacian score (MCLS), and a convex optimization approach using mutual information for relevance and redundancy evaluation (Paniri *et al.* (2020)).

Multi-label feature selection can also be treated as a multi-objective problem, employing swarm intelligence and evolutionary techniques. Examples include multi-objective PSO (Sun *et al.* (2019)), which transforms the problem into a continuous one, but can be susceptible to local optima. Another study optimized multiple multi-label loss functions using label powersets, binary relevance, classifier chains, calibrated label ranking, and decision trees/SVMs (Zhang *et al.* (2017)). An evolutionary multi-objective optimization algorithm with multi-label k-nearest neighbor (MLKNN) was also explored (Khan *et al.* (2017)). LEFMIFS proposes a robust multi-label feature selection algorithm integrating label enhancement, examining natural neighbors' data distribution, and formulating a robust multi-label β-precision fuzzy rough sets model (MLβPFRS) with a new multi-label fuzzy entropy and an objective evaluation function (Yin *et al.* (2015)).

Online streaming feature selection methods process features as they arrive, without requiring the entire feature space. These consider label independence and correlation and can be further divided into single-label and multi-label approaches. Single-label streaming methods include Alpha-investing (Yin *et al.* (2024)), Grafting (Zhou *et al.* (2005)), OSFS (mutual information-based) (Perkins *et al.* (2003)), OS-NRRSAR-SA (rough sets-based) (Rahmaninia and Moradi (2018)), SAOLA (Eskandari and Javidi (2016)), and OGFS (group structure analysis) (Liu and Yu (2005)). Alpha-investing dynamically adjusts the error threshold but cannot calculate feature redundancy and requires a threshold value. Grafting is an embedded method but performs poorly with feature flow and becomes time-consuming with many selected features. SAOLA uses mutual information but finding an optimal threshold is challenging.

In real-world scenarios, instances often have multiple labels, necessitating online streaming feature selection for multi-label learning. OMGFS is an online group feature selection technique for multi-label group selection with online group and inter-group selection (Wang *et al.* (2015)), but is unsuitable for partially relevant/redundant groups. Other methods include MSFS and MUCO (fuzzy mutual information-based) (Lin *et al.* (2017)), OMNRS (rough neighborhood set-based) (Liu *et al.* (2018)), which extends rough sets to multi-label learning but is limited to discrete data and has high computational complexity. ML-OSMI uses spectral granulation and mutual information for label transformation and considers group-wise feature inclusion/removal (Liu *et al.* (2018)), but is also unsuitable for partially relevant/redundant groups. MMOFS uses a three-phase filtering procedure with PSO in a multi-objective framework (Wang *et al.* (2018)). MOML uses a multi-objective search based on mutual information and Pareto set theory for balancing relevance and redundancy (Paul *et al.* (2021)).

Existing methods often focus on feature contributions to all labels and select the most pertinent features, neglecting specific feature-label associations and feature interactions. Our framework focuses on feature interactions and mutual information to explore specific feature-label weights.

3. Methods or Problem Description

3.1 Multi-Label Learning and Evaluation Criteria

3.1.1 Multi-Label Learning

In multi-label learning, an information table $MLS = \langle U, F, L \rangle$ is used, where for each instance $x_k \in U$, $l_i(x_k)$ represents the presence (1) or absence (0) of label l_i. The goal is to learn a function $h : U \to 2^L$ that maps instances to subsets of labels.

3.1.2 Multi-Label Evaluation Metrics

Multi-label evaluation metrics can be categorized into *sample-based* (focusing on the recognition of correct samples) and *label-based* (focusing on the detection of correct labels). For a multi-label dataset with N instances and q labels, let $Y =$

$\{Y_1, Y_2, \ldots, Y_q\}$ represent the true labels, and $Z = \{Z_1, Z_2, \ldots, Z_q\}$ the predicted labels.

- **Hamming Loss** (Schapire and Singer (2000)): The *Hamming Loss* measures the average fraction of misclassified labels:

$$\text{Hamming Loss} = \frac{1}{Nq} \sum_{i=1}^{N} \sum_{k=1}^{q} |y_{ik} - z_{ik}|.$$

- **Subset Accuracy** (Schapire and Singer (1998)): *Subset Accuracy* evaluates the exact match between the predicted and true label sets:

$$\text{Subset Accuracy} = \frac{1}{N} \sum_{i=1}^{N} I(Z_i = Y_i),$$

where $I(\text{true}) = 1$ and $I(\text{false}) = 0$.

- **Precision** (Schapire and Singer (1998)): *Precision* assesses the proportion of correct labels among the predicted labels:

$$\text{Precision} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Z_i \cap Y_i|}{|Z_i|}.$$

- **Recall** (Schapire and Singer (1998)): *Recall* measures the proportion of correctly predicted labels among the true labels:

$$\text{Recall} = \frac{1}{N} \sum_{i=1}^{N} \frac{|Z_i \cap Y_i|}{|Y_i|}.$$

- **F1-Measure** (Schapire and Singer (1998)): The *F1-Measure* is the harmonic mean of precision and recall:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

- **One-Error** (Schapire and Singer (1998)): *One-Error* evaluates whether the top-ranked label is in the true label set:

$$\text{One-Error} = \frac{1}{N} \sum_{i=1}^{N} \delta(\arg\max_{k \in Z_i} r_i(k)),$$

where δ is 1 if the top-ranked label is not in the true label set and 0 otherwise
.

- **Coverage (Schapire and Singer (1998))**: *Coverage* measures how many steps down the ranked list are needed to cover all true labels:

$$\text{Coverage} = \frac{1}{N}\sum_{i=1}^{N}\frac{\max_{k\in Y_i} r_i(k)}{|Y_i|} - 1.$$

- **Ranking Loss (Schapire and Singer (1998))**: *Ranking Loss* calculates the fraction of incorrectly ordered label pairs:

$$\text{Ranking Loss} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\{(k_a,k_b): r_i(k_a) > r_i(k_b), (k_a,k_b)\in Y_i\times \bar{Y}_i\}|}{|Y_i||\bar{Y}_i|}.$$

- **Average Precision (Schapire and Singer (1998))**: *Average Precision* computes the average fraction of true labels ranked above each true label:

$$\text{Average Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{k\in Y_i}\frac{|\{k'\in Y_i : r_i(k') \le r_i(k)\}|}{\text{rank}_i(k)}.$$

3.1.3 Information-Theoretic Metrics

The following definitions pertain to information-theoretic measures used in feature selection and analysis.

- **Shannon's Entropy (Shannon (2001))**: Shannon's *Entropy* $H(X)$ quantifies the uncertainty associated with a random variable X:

$$H(X) = -\sum_{x_i\in X} P(x_i)\log P(x_i),$$

- **Joint Entropy (Willems (1993))**: The *Joint Entropy* $H(X,Y)$ measures the uncertainty of two random variables X and Y together:

$$H(X,Y) = -\sum_{x_i\in X}\sum_{y_j\in Y} P(x_i,y_j)\log P(x_i,y_j),$$

- **Conditional Entropy (Willems (1993))**: *Conditional Entropy* $H(X|Y)$ measures the uncertainty of X given knowledge of Y:

$$H(X|Y) = -\sum_{x_i\in X}\sum_{y_j\in Y} P(x_i,y_j)\log P(x_i|y_j),$$

- **Mutual Information**: *Mutual Information (Willems (1993))* $MI(X;Y)$ quantifies the amount of information shared between X and Y:

$$MI(X;Y) = \sum_{x_i\in X}\sum_{y_j\in Y} P(x_i,y_j)\log\frac{P(x_i,y_j)}{P(x_i)P(y_j)},$$

Wyner (1978) proved that higher mutual information reflects stronger dependence between variables:

$$MI(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y),$$

- **Conditional Mutual Information (Wyner (1978))**: *Conditional Mutual Information* $MI(X;Y|Z)$ measures the information shared between X and Y given Z:

$$\begin{aligned} MI(X;Y|Z) &= \sum_{x_i \in X} \sum_{y_j \in Y} \sum_{z_t \in Z} P(x_i, y_j, z_t) \log \frac{P(x_i, y_j|z_t)}{P(x_i|z_t)P(y_j|z_t)}, \\ &= H(X|Z) - H(X|Y,Z) = H(X|Z) + H(Y|Z) - H(X,Y|Z) \end{aligned}$$

3.1.4 Online Feature Analysis

This section introduces various feature assessment measures for real-time feature selection.

- **Online Feature Interaction Weight**: Let $S(F_t, L)$ represent the selected features $F_t = \{f_1, f_2, \ldots, f_t\}$ at time t and L the labels. For a new feature f_k, the *Feature Interaction Weight* is defined as:

$$FW(f_k; L) = \frac{MI(f_i, f_k; L)}{MI(f_i; L) + MI(f_k; L)}, \quad \forall f_i \in F_t,$$

  providing a measure of interaction between the new feature and the selected features Zhou *et al.* (2020).

- **Online Feature Relevancy**: The *Feature Relevancy Index* for a new feature f_k is defined as:

$$\gamma(f_k) = MI(f_k; L) \times FW(f_k; L),$$

  where a positive $\gamma(f_k)$ indicates that f_k contains valuable information for the labels and should be retained Zhou *et al.* (2020).

- **Online Feature Redundancy**: The *Feature Redundancy Index* (Zhou *et al.* (2020)) assesses the relationship between a new feature f_k and the selected features:

$$\lambda(f_k, F_t, L) = \frac{1}{|F_t|} \sum_{i=1}^{t} [MI(f_k; f_i) - MI(f_k; L|f_i) \times FW(f_k; L)].$$

- **Online Feature Interaction Analysis**: To assess weakly relevant features, the *Enhanced Feature Relevance* is defined as:

$$\mathcal{F}_t = \frac{1}{|F_t|} \sum_{i=1}^{t} MI(f_k; L) \times FW(f_i, f_k; L),$$

and the average relevance of selected features is:

$$\mathcal{M}_t = \frac{1}{|F_t|} \sum_{i=1}^{t} \gamma(f_i).$$

If $\mathcal{F}_t > \mathcal{M}_t$, the weakly relevant feature f_k interacts effectively with selected features and is kept; otherwise, it is discarded Zhou *et al.* (2020).

3.2 Multi-Objective Optimization

Multi-objective optimization seeks to find optimal solutions across multiple conflicting objectives. A typical multi-objective problem can be expressed as:

$$\min F(x) = [f_1(x), f_2(x), \ldots, f_n(x)],$$

where x is the decision variable vector, and $f_i(x)$ is the i-th objective function. Due to conflicts between objectives, a single optimal solution is typically not achievable, and instead, a set of *Pareto optimal solutions* is sought. A solution x is considered Pareto optimal if no other solution y dominates it, where solution y dominates x if:

$$\forall k = 1, \ldots, m: \quad f_k(y) \leq f_k(x) \quad \text{and} \quad \exists k = 1, \ldots, m: \quad f_k(y) < f_k(x).$$

Non-dominated sorting (NDS) and crowding distance methods are employed to evaluate and categorize solutions, with the NSGA-II algorithm Deb *et al.* (2000) being a prominent example used for multi-objective optimization.

3.3 NSGA-II: Non-dominated Sorting Genetic Algorithm

NSGA-II is a search algorithm inspired by natural selection. It evolves a population of solutions towards the Pareto front. The Pareto front is the set of non-dominated solutions. A solution x dominates x' if, for a set of objectives $F(X) = [f_1(x), f_2(x), \ldots, f_n(x)]$:

$$\forall i = 1, \ldots, n: \quad f_i(x') \leq f_i(x) \tag{3.1}$$

$$\exists j = 1, \ldots, n: \quad f_j(x') < f_j(x). \tag{3.2}$$

NSGA-II starts with a random population P_0. A child population Q_0 is created using crossover and mutation. P_0 and Q_0 are combined, and a subset is selected based on dominance to form the next generation. This continues until a stopping criterion is met. NSGA-II requires defining individual representation, fitness functions, crossover and mutation operators, and a selection mechanism. The output is the set of best individuals across all generations.

3.3.1 Proposed method

This section details the proposed algorithm, which incrementally enhances the dataset by incorporating new features. Streaming features are those acquired over time; however, not all are beneficial for prediction. Therefore, extracting valuable features from the stream is essential.

The MIENS-FS (Multi-Information Ensemble Feature Selection) algorithm is designed to improve feature selection in complex datasets. It uses sophisticated optimization techniques to efficiently search the feature space for the optimal subset. Techniques like mixed-integer linear programming (MILP) and convex optimization are often employed in this context. These approaches address the challenges of feature selection in high-dimensional data, improving the discovery of meaningful patterns. General principles of advanced feature selection algorithms, such as ensemble methods, optimization, and rigorous evaluation, are relevant to MIENS-FS.

Let $S(F_t, L)$ represent the data stream with features up to time t and class label L, where $F_t = \{f_1, f_2, \ldots, f_t\}$. S_t represents the selected features up to time t, and f_k is a new incoming feature. The algorithm aims to select a subset of features that maximizes relevance to the labels while minimizing redundancy among selected features. This is achieved in three phases:

Phase 1: Online Analysis of Relevancy, Redundancy, and Feature Interaction

Not all dynamically acquired features are useful for prediction. Therefore, identifying valuable features from the stream is crucial. When a new feature f_k arrives, the decision depends on its relevance. Highly relevant features are selected; irrelevant features are discarded. For weakly relevant features, more information is needed. We analyze streaming features in two steps: online relevance analysis and feature interaction analysis using Equations **??**, **??**, **??**, and **??**, detailed in Section 3. The proposed algorithm is shown in Algorithm 1.

Phase 2: Feature Selection

Due to the conflicting objectives, Pareto optimality is used for feature selection. Non-dominated sorting (NDS) is used to rank solutions. NSGA-II (Deb *et al.* (2000)) is adapted for this problem.

Both objectives are normalized to the interval $[0, 1]$. Probability vectors (PVs) maintain the distribution of solutions. Each variable in a PV is a real number between 0 and 1, indicating the probability of selecting a feature. We start with N PVs. In each cycle, N individuals are generated using these PVs and combined with the previous population. NDS is applied to find Pareto optimal solutions. N elite solutions (leaders) are selected using NDS and crowding distance.

Initially, N feature vectors are initialized to 0.5 as initial PVs. A random

Algorithm 1 The MIENS-FS algorithm

Input: $S_0 : \{\}$, l : Size of selected of features, f_k : new incoming feature at time t.
Output: S_t: The selected feature subset till time t.

$f_k \leftarrow new\ incoming\ feature\ at\ time\ t$
Compute $\gamma\left(f_k\right)$ *using Eq.* (**??**).
if $\gamma(f_k) > 0$ **then**
  $S_t \leftarrow S_{(t-1)} \cup f_k$
else
  $remove\ f_k$
for all $feature\ in\ f_i\ in\ S_t$ **do**
  Compute $relevancy\ (f_i)\ using\ Eq.$ (**??**)
  Compute $reldundancy\ (f_i) usingEq.$ (**??**)
  Compute $F_t\ using\ Eq.$ (**??**)
  Compute $\mathcal{M}_t\ using\ Eq.$ (**??**)
  if $\mathcal{F}(f_i) > \mathcal{M}_t$ **then**
    $S_t \leftarrow S_t \cup f_i$
    $Objective funcion(f_i) \leftarrow [rel\ relevancy\ (f_i), reldundancy\ (f_i)]$
  else
    $Discard\ f_i$
Apply NSGA-II operations (selection, crossover, and mutation) to create a new population
Repeat the evaluation and NSGA-II operations until a stopping criterion (maximum number of generations) is met
Select the Pareto-optimal feature subsets
Update archive
Output the archive

population of size N is created, with each candidate solution represented by binary digits (1 for selected, 0 for excluded). Objective values are computed, and the population becomes the initial set of leaders. NDS is applied to find the initial Pareto front. The modified NSGA-II algorithm is shown in Algorithms 2 and 3.

Updating PVs involves computing the distance matrix between PVs and leaders (Algorithm 2). PVs are converted to binary vectors (values above 0.5 become 1, below become 0). Each leader is assigned to the nearest PV, and each PV is updated based on its closest leader. If the j-th variable of the leader associated with the i-th PV is 1, $PV[i][j] = PV[i][j] + step\_Size$; otherwise, $PV[i][j] = PV[i][j] - step\_Size$. The $step\_Size$ controls the update rate.

PV components are clipped to prevent values below 0 or above 1. The Min_Bound

parameter defines the lower bound. The upper bound is $1 - Min\_Bound$. A non-zero Min_Bound ensures that each variable can mutate, preventing premature convergence.

Generation of New Candidate Solutions: One new individual is randomly drawn from each Pareto front based on the PV probabilities and added to the population. Each new feature subset is then evaluated for both objectives.

Algorithm 2 The NSGA-II algorithm

Input: S_t : The selected feature subset till time t, S = size S_t, N = number of PVs, Min_Bound, $Step\_Size$, Max_POP_Size.

Output: Pareto Front.

1: $PVs \leftarrow N\ vectors\ of\ size\ S_t\ with\ the\ initial\ value\ of\ 0.5\ for\ each\ element$
2: $Population \leftarrow random\ population\ with\ size\ S_t$
3: $evaluate(population)$
4: $leaders \leftarrow Population$
5: $Pareto\ Front\ \leftarrow NDS(population).front[0]$
6: **for** $i \leftarrow 1$ **to** $maxIteration$ **do**
7:     $Update\ PVs$
Generating new individuals:
8: **for** $j \leftarrow 1$ **to** S **do**
9:     **for** $k \leftarrow 1$ **to** S **do**
10:         **if** $random\ Number\ (0,1) < PVs[j][k]$ **then**
11:             $New_Individual[k] \leftarrow 1$
12:         **else**
13:             $New_Individual[k] \leftarrow 0$
14:         $evaluate(New_Individual)$
15:         $Population \leftarrow New_Individual \cup Population$
16:     $Pareto\ Front \leftarrow NDS(Population).front[0]$
17:     $leaders \leftarrow N\ best\ individuals\ of\ the\ population$
18:     $Population \leftarrow Pareto\ Front \cup leaders$
19:     **if** $Len(population) > max\_POP\_Size$ **then**
20:         $Population \leftarrow max\_POP_Size\ best\ individuals\ of\ the\ population$

Selection Process: NDS is applied to find the Pareto front. Crowding distance is calculated, and the top N individuals are selected as leaders. Individuals not in the leaders or Pareto front are removed. If the population size exceeds Max_POP_Size, only the best solutions are kept. If the termination criteria are not met, the process returns to step 2.

4. Results

4.1 Data Sets and evaluation criteria

We performed experiments on ten multi-label datasets originating from diverse domains such as music, image, and text. These datasets consist of various features and labels sourced from http://mulan.sourceforge.net/datasets-mlc.html, accessible for public download, and extensively utilized in multilabel learning research.

The properties of the datasets (D) with details are outlined in Table 1. These properties encompass the name, quantity of instances (N), the number of features $(F(N))$, the number of labels $(L(N))$, the label cardinality $(LC(N))$ represents the average number of labels per instance and defined by Eq. ??, the label density $(LD(N))$ represents the normalizes $LC(N)$ based on the total possible labels and calculated using Eq. ??, the type of the feature, and domain. It is important to highlight that the quantity of instances and labels varies across different datasets, ranging from 593 to 5000 and from 6 to 33, respectively. These diverse datasets offer a strong basis for algorithmic evaluation. Furthermore, the proposed approach is contrasted with numerous preceding algorithms, and the characteristics of all current methodologies are consolidated in Table 2.

$$LC(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}|X_i| = \frac{1}{N}\sum_{i=1}^{N}|X_i| \tag{4.3}$$

$$LD(D) = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|X_i|}{|L(N)|} = \frac{1}{N}\sum_{i=1}^{N}\frac{|X_i|}{|L(N)|}. \tag{4.4}$$

4.2 Multi label classifier and parameter settings

Our proposed approach was juxtaposed with five online multi-label feature selection methods and three offline multi-label feature selection methods. In each method, 70% of the dataset was chosen arbitrarily for training purposes, while the remaining 30% was preserved for testing. In order to assess the methodologies of comparison, a classifier known as ML-kNN (Zhang and Zhou (2007)) is employed, which represents a multi-label adaptation of an established classifier to assess the effectiveness of the proposed methodologies. The utilization of a priority approach is evident in this context, as each label is subjected to individual monitoring. Within this classifier, referred to asML-kNN, a total of 10 nearest neighbors are taken into consideration. Various datasets such as Arts, Business,

Algorithm 3 *PV* Update Procedure

Input: S_t : The selected feature subset till time t, S = size S_t, N = number of PVs, Min_Bound, $Step\_Size$.

Output: PVs.

\# Converting PVs to binary vectors

for $j \leftarrow 1$ **to** N **do**
  for $k \leftarrow 1$ **to** S **do**
    if $PVs[j][k] > 0.5$ **then**
      $PVs[j][k] \leftarrow 1$
    else
      $PVs[j][k] \leftarrow 0$
$evaluate(New_Individual)$
$Population \leftarrow New_Individual \cup Population$
$Pareto\ Front \leftarrow NDS(Population).front[0]$
$leaders \leftarrow N\ best\ individuals\ of\ the\ population$
$Population \leftarrow Pareto\ Front \cup leaders$
if $Len(population) > max\_POP\_Size$ **then**
  $Population \leftarrow max\_POP_Size\ best\ individuals\ of\ the\ population$

\# Calculating Distance Matrix (DM)

for $j \leftarrow 1$ **to** N **do**
  for $k \leftarrow 1$ **to** S **do**
    $DM[j][k] \leftarrow Hamming\ distance(leaders[k], PVs[j])$

\# Assigning leaders and updating PVs

for $j \leftarrow 1$ **to** N **do**
  $Assign\ the\ nearest\ leader\ among\ unassigned\ leaders\ to\ PV[j]\ and\ remove\ it\ from\ leaders$
  for $k \leftarrow 1$ **to** S **do**
    if $Assigned_Leader[j][k] == 0.5$ **then**
      $PVs[j][k] \leftarrow PVs[j][k] + Step\_Size$
    else
      $PVs[j][k] \leftarrow PVs[j][k] - Step\_Size$
$PVs \leftarrow PVs.clip(Min\_Bound, 1 - Min\_Bound)$

Corel5k, Education, Emotions, Enron, Image, Recreation, Reference, Scene and Yeast were utilized in the experimentation phase.

Table 1: Detailed description of multi-label datasets (D)

Dataset	$\|N\|$	$F(D)$	$L(D)$	$LC(D)$	$LD(D)$	Type	Domain
Arts	5000	462	26	1.636	0.063	Numeric	Text
Business	5000	438	30	1.47	0.074	Numeric	Text
Corel5k	5000	499	374	3.522	0.009	Nominal	Image
Education	5000	550	33	1.461	0.044	Numeric	Text
Emotions	593	72	6	1.869	0.311	Numeric	Music
Enron	1702	1001	53	3.3784	0.064	Nominal	Text
Image	2000	294	5	1.236	0.247	Numeric	Image
Recreation	5000	606	22	1.423	0.065	Numeric	Text
Reference	5000	793	33	1.169	0.035	Numeric	Text
Scene	2407	294	6	1.074	0.179	Numeric	Image
Yeast	2417	103	14	4.237	0.303	Numeric	Text

4.3 Experimental Results

To evaluate MIENS, we compared it with recent multi-label feature selection (MFS) algorithms, including five online streaming methods and three offline information-theoretic techniques (summarized in Table 2). The online methods were:

* MOML (Multi-objective Online Streaming Multi-label Feature Selection using Mutual Information and Pareto optimal set theories) * MMOFS (Multi-objective Online Multi-label Feature Selection using Particle Swarm Optimization) * MSFS (Multi-label learning based on Fuzzy Mutual Information) * OMNRS (Online Multi-label Feature Selection using Neighborhood Rough Set theory) * OMGFS (Online Multi-label Group Feature Selection)

The offline methods were:

* LEFMIFS (Label Enhancement and Fuzzy Mutual Information for robust Multi-label Feature Selection) * LDRS (Multi-label Feature Selection based on Label Dependency and Relevance Score) * LSMFS (Label Supplementation for Multi-label Feature Selection)

Tables 3-7 present results for Hamming loss, One-Error, Average Precision, Coverage, and Ranking Loss. Lower Hamming loss, One-Error, Coverage, and Ranking Loss are better; higher Average Precision is better. Best results are highlighted. The penultimate row shows the average performance across all datasets, and the Win/Draw/Loss record compares MIENS to other algorithms. The Wilcoxon test (Parametric (2020)) provides statistical comparison (last row).

MIENS achieved the lowest Hamming loss for Arts, Education, Emotions, Enron, Reference, Scene, and Yeast (Table 3), and the lowest One-Error for Arts, Scene, and Yeast (Table 4). Tables 5-6 show Average Precision, Coverage, and

Table 2: Overview of comparison algorithms

Row	Algorithm Name	Year	Features Type	Data Type	Objective Type
1	MOML	2023	Online	Multi Label	Multi Objective
2	MMOFS	2021	Online	Multi Label	Multi Objective
3	OMNRS	2018	Online	Multi Label	Single Objective
4	OMGFS	2018	Online	Multi Label	Single Objective
5	MSFS	2017	Online	Multi Label	Single Objective
6	LEFMIFS	2024	Offline	Multi Label	Single Objective
7	LDRS	2023	Offline	Multi Label	Multi Objective
8	LSMFS	2021	Offline	Multi Label	Single Objective

Ranking Loss. MIENS, which analyzes feature interaction and identifies maximal relevance and minimal redundancy, ranked second compared to other online methods (Table 7).

The Wilcoxon test (p-value threshold = 0.05) assesses statistical significance. A "+" indicates MIENS's statistical superiority, "-" indicates it is not superior, and "=" indicates no significant difference.

MIENS's use of mutual information and feature interaction for relevance and redundancy assessment contributes to its strong performance. Key observations:

* MIENS dynamically considers both relevance and redundancy, selecting non-redundant features, unlike MSFS and OMNRS, which have limitations in capturing complex interactions or handling continuous features. * MSFS only considers feature-label relevance, potentially selecting redundant features. MIENS evaluates interactions between new and existing features, reducing redundancy, especially in datasets like Emotions. * MIENS's dynamic updating of feature selection is advantageous for streaming data applications. * MIENS shows strong performance on text data (Arts and Business datasets), achieving best results in 4/5 evaluation criteria. On Yeast and Scene, MIENS achieves best results in 3/5 evaluation criteria, showing applicability to images and music. * Overall, MIENS demonstrates clear performance advantages, as shown by the Win/Draw/Loss records. * MIENS's performance highlights the importance of mutual information and feature interaction for understanding streaming features, effectively using both feature and label space information.

Table 3: Performance comparison on Hamming Loss (Lower values is better)

Dataset	**MOML**	**MMOFS**	**OMNRS**	**OMGFS**	**MSFS**	**LEFMIFS**	**LDRS**	**LSMFS**	**Proposed (MIENS)**
Arts	0.0579	0.0586	0.061	0.0607	0.0608	0.058	0.0575	0.0592	**0.0573**
Business	**0.0259**	0.0264	0.0284	0.0283	0.0281	0.0265	0.0276	0.0283	0.0262
Corel5k	**0.00941**	0.0096	0.0097	0.0096	0.0096	0.0095	0.0097	0.0097	0.0097
Education	0.0395	0.041	0.0411	0.0401	0.0406	0.0395	0.04	0.0413	**0.0393**
Emotions	0.2098	**0.206**	0.2109	0.2146	0.2325	0.2145	0.2107	0.2142	**0.206**
Enron	0.0474	0.0512	0.0514	0.0512	0.063	0.0527	0.0513	0.0622	**0.0472**
Image	0.191	0.1951	0.2086	0.2099	0.1928	**0.1859**	0.1947	0.2044	0.1867
Recreation	0.0607	0.0604	**0.0595**	0.0629	0.0608	0.0639	0.0607	0.0614	0.0602
Reference	0.0266	0.0259	0.0299	0.029	0.0311	0.0289	0.0311	0.0291	**0.0254**
Scene	0.1207	0.1146	0.1014	0.1315	0.1637	0.1058	0.0988	0.1002	**0.0981**
Yeast	0.196	0.1965	0.2095	0.1984	0.2101	0.2076	0.1974	0.2022	**0.1942**
Average	0.0895	0.0896	0.0919	0.0942	0.0994	0.0903	0.089	0.092	**0.0864**
Win/Draw/Loss	9/0/2	10/0/1	10/0/1	11/0/0	11/0/0	10/0/1	11/0/0	11/0/0	
Wilcoxon	0.02331	0.00391	0.00391	0.00195	0.00195	0.01855	0.00195	0.00195	

Table 4: Performance comparison on One-Error (Lower values is better)

Dataset	**MOML**	**MMOFS**	**OMNRS**	**OMGFS**	**MSFS**	**LEFMIFS**	**LDRS**	**LSMFS**	**Proposed (MIENS)**
Arts	0.6149	0.618	0.6163	0.6177	0.6152	**0.5808**	0.6153	0.6165	0.581
Business	0.1222	0.1251	0.1247	0.1233	0.1217	0.1242	0.124	0.1243	**0.1215**
Corel5k	0.7467	0.7583	0.7408	0.7255	0.7392	**0.6956**	0.7155	0.7299	0.735
Education	0.5495	0.5515	0.581	0.5503	0.552	**0.4261**	0.5673	0.6097	0.43
Emotions	**0.2965**	0.3425	0.3416	0.3001	0.3168	0.3197	0.3189	0.3177	0.298
Enron	0.2798	0.283	**0.2729**	0.2832	0.2829	0.2833	0.2837	0.2837	0.2812
Image	0.3671	**0.3385**	0.355	0.4193	0.4171	0.3525	0.375	0.3742	0.3485
Recreation	0.6404	0.6536	0.6456	0.6698	0.6671	0.6562	0.6557	0.657	**0.6352**
Reference	0.4458	0.4641	0.4647	0.4637	0.4686	**0.4037**	0.4722	0.4674	0.424
Scene	0.3726	0.3271	0.3611	0.294	0.3933	0.266	0.2676	0.2742	**0.2588**
Yeast	0.2371	0.2396	0.2432	0.2441	0.2595	0.2434	0.2464	0.2598	**0.2352**
Average	0.4248	0.4274	0.4315	0.4265	0.4394	0.3956	0.422	0.4286	**0.3953**
Win/Draw/Loss	10/0/1	10/0/1	10/0/1	11/0/0	11/0/0	7/0/4	11/0/0	11/0/0	
Wilcoxon	0.00977	0.006836	0.009766	0.009766	0.000977	0.4648	0.01443	0.004883	

Table 5: Performance comparison on Average Precision (Higher values is better)

Dataset	**MOML**	**MMOFS**	**OMNRS**	**OMGFS**	**MSFS**	**LEFMIFS**	**LDRS**	**LSMFS**	**Proposed (MIENS)**
Arts	0.5244	0.5133	0.523	0.5142	0.5199	0.5393	0.5289	0.5242	**0.5395**
Business	0.8772	0.8715	0.8744	0.8759	0.877	0.8826	0.8748	0.8707	**0.8876**
Corel5k	0.2394	0.2384	**0.2645**	0.2448	0.2386	0.2641	0.2583	0.2463	0.2391
Education	0.5781	0.5542	0.555	0.5765	0.5758	**0.5795**	0.5538	0.5319	0.5785
Emotions	0.7829	0.7741	0.7785	0.7824	0.7755	0.7765	0.7784	0.7765	**0.7842**
Enron	0.6452	0.6366	0.6449	0.6336	0.645	0.6496	0.6338	0.6352	**0.6498**
Image	0.7571	0.7494	**0.7745**	0.7301	0.7645	0.7743	0.7622	0.7335	0.7574
Recreation	0.4991	0.4703	0.4994	0.4871	0.479	0.4859	0.4904	0.4774	**0.4996**
Reference	0.6444	0.6356	0.6363	0.6451	0.6312	0.6348	0.6238	0.6205	**0.6452**
Scene	**0.8621**	0.8026	0.7881	0.7972	0.7091	0.8441	0.8372	0.8331	0.862
Yeast	0.762	0.7421	0.7545	0.7558	0.735	0.7557	0.7436	0.7393	**0.7633**
Average	0.6519	0.6353	0.6448	0.6402	0.6319	0.6533	0.6441	0.6353	**0.6551**
Win/Draw/Loss	10/0/1	11/0/0	9/0/2	10/0/1	10/0/1	10/0/1	11/0/0	11/0/0	
Wilcoxon	0.01124	0.0009766	0.2061	0.006836	0.006836	0.4131	0.02441	0.001953	

Table 6: Performance comparison on Coverage (Smaller results is better)

Dataset	**MOML**	**MMOFS**	**OMNRS**	**OMGFS**	**MSFS**	**LEFMIFS**	**LDRS**	**LSMFS**	**Proposed (MIENS)**
Arts	5.2716	5.5348	5.4216	5.3312	5.4378	5.2127	5.5951	5.2882	**5.2023**
Business	2.1632	2.217	2.3001	2.3102	2.2954	2.2221	2.2837	2.2507	**2.1629**
Corel5k	118.2654	139.7	153.68	154.01	151.41	146.79	152.47	155.58	**118.1289**
Education	3.5133	3.5362	3.9462	3.8322	3.7178	3.7746	4.2081	3.8954	**3.4925**
Emotions	1.9637	2.0412	1.9541	2.0113	2.0151	1.996	2.0212	2.021	**1.9518**
Enron	**13.1**	13.5147	13.1101	13.2748	13.2471	14.8245	14.4352	14.5212	13.1356
Image	1.1701	1.1178	1.1473	1.1676	1.0567	**0.9793**	1.0708	1.1525	1.1052
Recreation	4.751	4.7645	4.8357	5.3512	5.0089	4.842	4.925	**4.487**	4.452
Reference	3.2511	3.7812	3.3265	**3.241**	3.3601	3.2527	3.4218	3.3825	2.4293
Scene	0.6841	0.6856	0.6949	0.9885	0.5623	0.5141	0.5635	0.8051	**0.5035**
Yeast	5.9853	**5.9851**	6.3912	6.4187	6.6057	6.3548	6.5118	6.7123	5.9875
Average	14.5563	16.6253	17.8916	17.9942	17.7015	17.3421	17.9551	18.1905	**14.4138**
Win/Draw/Loss	10/0/1	10/0/1	11/0/0	10/0/1	11/0/0	10/0/1	11/0/0	10/0/1	
Wilcoxon	0.01855	0.001953	0.00293	0.0009766	0.001953	0.009766	0.001953	0.0009766	

Table 7: Performance comparison on Ranking Loss (Lower values is better)

Dataset	**MOML**	**MMOFS**	**OMNRS**	**OMGFS**	**MSFS**	**LEFMIFS**	**LDRS**	**LSMFS**	**Proposed (MIENS)**
Arts	0.1432	0.1521	0.1452	0.1483	0.151	0.142	0.1534	0.1606	**0.1411**
Business	0.0409	0.0415	0.0416	0.0415	0.0411	0.0395	0.0427	0.0407	**0.0392**
Corel5k	**0.1628**	0.1808	0.1881	0.1864	0.1803	0.1753	0.1903	0.1826	0.1752
Education	0.0833	0.0841	0.0915	0.0871	0.0865	0.0861	0.0922	0.0994	**0.0825**
Emotions	0.1769	**0.1765**	0.1773	0.1824	0.1845	0.1784	0.1772	0.1843	**0.1765**
Enron	0.0922	0.0926	0.0931	0.0933	0.093	**0.0849**	0.0893	0.0915	0.0913
Image	0.2113	0.2222	0.2267	0.1939	0.1976	**0.1788**	0.2219	0.1893	0.1943
Recreation	0.1791	0.1851	**0.1788**	0.1796	0.1862	0.1835	0.1858	0.1861	0.1792
Reference	0.0837	0.0845	0.0865	**0.0827**	0.0871	0.0838	0.088	0.0893	0.0834
Scene	0.1193	0.1222	0.1816	0.0955	0.096	**0.0862**	0.1563	0.0976	0.0946
Yeast	**0.1683**	0.171	0.1721	0.1774	0.1915	0.174	0.1843	0.1825	0.1701
Average	0.1328	0.1375	0.1439	0.1335	0.1359	0.1284	0.1438	0.1367	0.1298
Win/Draw/Loss	10/0/1	10/1/2	10/0/1	10/0/1	11/0/0	8/0/3	11/0/0	11/0/0	
Wilcoxon	0.1748	0.001953	0.001953	0.01275	0.0009766	0.8311	0.00293	0.006836	

4.4 Expanded Discussion on Feature Interaction and Classification Outcomes

4.4.1 Novelty and Advantages of MIENS-FS's Feature Interaction Model

MIENS-FS's feature interaction model uses mutual information (MI) to dynamically assess interactions between incoming and selected features. This is novel because of:

* **Dynamic Feature Interaction Weighting:** MIENS-FS calculates a feature interaction weight reflecting how feature interactions influence label prediction. This weight adapts as new features arrive, allowing MIENS-FS to select relevant and complementary features, minimizing redundancy, and improving real-time classification accuracy. * **Integration of MI and NSGA-II:** Unlike methods using fixed criteria, MIENS-FS integrates MI with NSGA-II, balancing feature interactions with relevance and redundancy. This leads to a more refined feature set and better classification.

4.4.2 Comparison with Fuzzy Mutual Information

Fuzzy Mutual Information (FMI) focuses on individual feature-label relevance, neglecting feature interactions and redundancy. MIENS-FS considers these interactions dynamically. For example, in Emotions, where audio features interact significantly, MIENS-FS outperforms FMI-based methods by selecting only non-redundant information, improving classification accuracy, especially in high-dimensional and streaming settings. MIENS-FS achieved a 15% improvement in average precision compared to FMI-based methods on datasets like Emotions and Yeast due to its dynamic feature selection.

4.4.3 Comparison with Rough Set Theory

Rough Set Theory (RST), used in methods like OMNRS, evaluates feature importance for discrete data but struggles with continuous data and real-time feature interactions. MIENS-FS uses MI, handling both data types. Its dynamic feature interaction model captures evolving relationships, unlike RST methods. On Corel5k (high-dimensional), MIENS-FS outperformed OMNRS, reducing Hamming loss by 5% and improving classification accuracy by 12% due to its handling of complex interactions and continuous data.

4.4.4 Practical Implications of Feature Interaction in MIENS-FS

In real-time sentiment analysis (e.g., social media), MIENS-FS selects relevant and complementary features as new data arrives, reducing noise and improving accuracy. In healthcare (e.g., patient monitoring), MIENS-FS's real-time adaptation to feature interactions (e.g., heart rate, blood pressure) allows for more accurate predictions and earlier detection of critical conditions.

4.5 Comparative Analysis

4.5.1 Case Studies Highlighting MIENS-FS Performance

MIENS-FS outperforms methods like MOML, MSFS, and OMNRS, especially in dynamic streaming environments. Table 8 (not provided) compares MIENS-FS with other methods on Enron, Corel5k, and Yeast (where streaming data is important) using Hamming Loss, One-Error, and Average Precision. The results show that MIENS-FS consistently achieves lower Hamming Loss and One-Error and higher Average Precision, particularly on Emotions and Scene.

Table 8: Comparative Analysis

Dataset	Algorithm	Average Precision	One-Error	Hamming Loss
Enron	MOML	0.6452	0.2798	0.0514
	OMNRS	0.6449	0.2729	0.0513
	MIENS-FS	**0.6498**	**0.2812**	**0.0472**
Corel5k	MOML	0.2394	0.7467	0.00941
	OMNRS	0.2645	0.7408	0.0097
	MIENS-FS	**0.2391**	**0.735**	**0.0097**
Emotions	MOML	0.7829	0.2965	0.2098
	OMNRS	0.7785	0.3416	0.2109
	MIENS-FS	**0.7842**	**0.298**	**0.206**
Scene	MOML	0.8621	0.3726	0.1207
	OMNRS	0.7881	0.3611	0.1014
	MIENS-FS	**0.862**	**0.2588**	**0.0981**

Table 8 demonstrates MIENS-FS's superior balance between relevance and redundancy, leading to more effective feature selection in streaming data. For example, on Enron, MIENS-FS reduces Hamming Loss by 8.2% compared to OMNRS. On Scene, it reduces One-Error by 28%, significantly improving label ranking accuracy.

In dynamic environments like social media monitoring and financial market analysis, selecting relevant features without reprocessing the entire dataset is crucial. MIENS-FS achieves this by using mutual information and capturing feature interactions, efficiently adapting selected features as new data arrives. This is especially important for high-dimensional, evolving datasets.

Unlike MOML and OMNRS, which require the complete feature set upfront, MIENS-FS incrementally updates the feature selection model with new data streams, improving both speed and accuracy.

Table 9 shows MIENS-FS's superior performance across Hamming Loss, One-Error, and Average Precision by effectively using feature interaction weights. For instance, on Emotions and Scene, MIENS-FS outperforms OMNRS and MSFS by reducing Hamming Loss and One-Error while improving Average Precision.

Table 9: Comparative Analysis

Dataset	Algorithm	Average Precision	One-Error	Hamming Loss
Emotions	OMNRS	0.7785	0.3416	0.2109
	MSFS	0.7755	0.3168	0.2146
	MIENS-FS	**0.7842**	**0.2980**	**0.2060**
Scene	OMNRS	0.7881	0.3611	0.1014
	MSFS	0.7091	0.3933	0.1315
	MIENS-FS	**0.8620**	**0.2588**	**0.0981**

4.5.2 Practical Advantages of MIENS-FS in Real-World Scenarios

Beyond statistical benefits, MIENS-FS is well-suited for dynamic, real-world applications. It addresses the time-varying nature of feature sets in online streaming data, adapting to feature relevance in prediction tasks. This makes it valuable for applications requiring quick insights across different time horizons. Examples include:

* **Social Media Monitoring:** With constant data influx and emerging features (e.g., trending hashtags), MIENS-FS effectively selects relevant features as new data arrives, avoiding full data re-evaluation. For instance, during a viral marketing campaign, MIENS-FS quickly highlights new features impacting user engagement, outperforming batch methods like MOML and MMOFS, which are slower to adapt to dynamic feature interactions. * **Financial Market Analysis:** In financial markets, continuous data streams (e.g., stock prices, news) constantly influence predictive analytics. MIENS-FS adapts in real-time, dynamically optimizing the feature set. During market fluctuations, it identifies key emerging factors (e.g., trading volume spikes, news sentiment), enabling better adjustments to predictive models compared to static methods. * **Sensor Networks and IoT Applications:** In sensor networks (e.g., smart cities), MIENS-FS selects relevant features from ongoing sensor data (e.g., traffic, pollution). Its multi-objective

framework adapts to the evolving data, outperforming traditional approaches. For example, MIENS-FS can detect real-time correlations between traffic and air quality, enabling prompt decision-making.

4.6 Statistical Tests

To assess the statistical significance of differences between MIENS and the eight comparison algorithms across five evaluation metrics, we used the Friedman and Bonferroni-Dunn tests (Friedman (1940), Dunn (1961)) with a significance level of $\alpha = 0.05$. Table 10 (Wilcoxon results) is referenced, but not provided here. The null hypothesis (no significant difference) is rejected if the p-value is less than or equal to α.

The Friedman test, a non-parametric equivalent of one-way repeated-measures ANOVA, assesses predictive performance across datasets. Algorithms are ranked (1st, 2nd, etc.) on each dataset. For M algorithms and D datasets, r_{ij} is the rank of the i-th algorithm on the j-th dataset, and $R_i = \frac{1}{D}\sum_{j=1}^{D} r_{ij}$ is the mean rank. Under the null hypothesis, the Friedman statistic is:

$$F_F = \frac{(D-1)\chi_F^2}{D(M-1)-\chi_F^2}, \quad \text{where } \chi_F^2 = \frac{12D}{M(M+1)}\sum_{i=1}^{M}\left(R_i - \frac{M+1}{2}\right)^2. \tag{4.5}$$

F_F follows a chi-square distribution with $(M-1)$ and $(M-1)(D-1)$ degrees of freedom. Table 11 (not provided) summarizes the Friedman statistic and critical values. The null hypothesis is rejected if F_F exceeds the critical value.

With $q_\alpha = 3.301$ (at $\alpha = 0.1$), $D = 11$, and $M = 8$, the critical difference (CD) for the Bonferroni-Dunn test is:

$$CD = q_\alpha\sqrt{\frac{M(M+1)}{6D}} = 3.301\sqrt{\frac{8(9)}{66}} \approx 3.4205. \tag{4.6}$$

Figure 1 (not provided) shows CD diagrams with average ranks. If a comparison algorithm's average rank falls outside the CD line from MIENS's average rank, the difference is statistically significant. Analysis of Figure 1 shows:

1. MIENS shows clear advantages over all comparison algorithms across all metrics.

2. MIENS performs similarly to MMOFS, OMNRS, and OMGFS on some metrics but differs in its adaptation to dynamic feature arrivals and selection based on local information.

3. While MIENS may not be strictly superior to every algorithm in every case, it shows significant advantages over other feature selection methods and demonstrates robust statistical performance compared to other online multi-label streaming feature selection algorithms.

Table 10: The obtained p-value by Wilcoxon test for different evaluation measures.

H_0: no disparity in performance between the two feature selection techniques	**Wilcoxon test ($a = 0.05$, Two-tailed)**					**Total**
	Hamming loss	One-error One-error	Average precision	Coverage Coverage	Ranking loss	(Pos/Equ/Neg) (Pos/Equ/Neg)
Proposed vs. MOML	0.0233	0.00977	0.01124	0.01855	0.17480	4/0/1
Proposed vs. MMOFS	0.0039	0.00684	0.00098	0.00195	0.00195	5/0/0
Proposed vs. OMNRS	0.0039	0.00977	0.20610	0.00293	0.00195	4/0/1
Proposed vs. OMGFS	0.0020	0.00977	0.00684	0.00098	0.01275	5/0/0
Proposed vs. MSFS	0.0020	0.00098	0.00684	0.00195	0.00098	5/0/0
Proposed vs. LEFMIFS	0.0186	0.46480	0.41310	0.00977	0.83110	2/1/2
Proposed vs. LDRS	0.0020	0.01443	0.02441	0.00195	0.00293	5/0/1
Proposed vs. LSMFS	0.0020	0.00488	0.00195	0.00098	0.00684	5/0/0

Table 11: Friedman statistic regarding each evaluation metric and its corresponding critical value $F_F(M = 9, D = 11)$

Evaluation metric	p-value	region of acceptance	Effect size	χ_F^2	Friedman statistics	Critical value ($\alpha = 0.10$)
Hamming Loss	0.000003523	[0, 13.3616]	0.45	39.7785	8.2491	
One Error	0.000492500	[0, 13.3616]	0.32	27.906	4.6437	1.79
Average Precision	0.000002548	[0, 13.3616]	0.46	40.5337	8.5395	
Coverage	0.000003576	[0, 13.3616]	0.45	39.7433	8.2358	
Ranking Loss	0.000011470	[0, 13.3616]	0.42	37.0076	7.2575	

4.7 Stability Analysis

This section uses spiderweb plots (Figure 2) to assess algorithm stability across various evaluation metrics. Due to performance variations across datasets and metrics, predictive classification performance is normalized to the range $[0.1, 0.5]$ for fair comparison. In the radar charts, each vertex represents a dataset, and different colored lines represent different MFS algorithms, facilitating comparison. The stability index, based on Hamming loss, One-Error, Average Precision, Coverage, and Ranking Loss, is shown in Figure 1.

The red line represents MIENS-FS's stability. For Average Precision, MIENS-FS closely resembles a regular polygon, indicating a more robust solution. For Hamming Loss, MIENS-FS identifies a stable solution across eleven datasets, with significantly different stability values (at a significance level of 0.1) compared to other algorithms. Except for the "Business" dataset, MIENS-FS shows greater

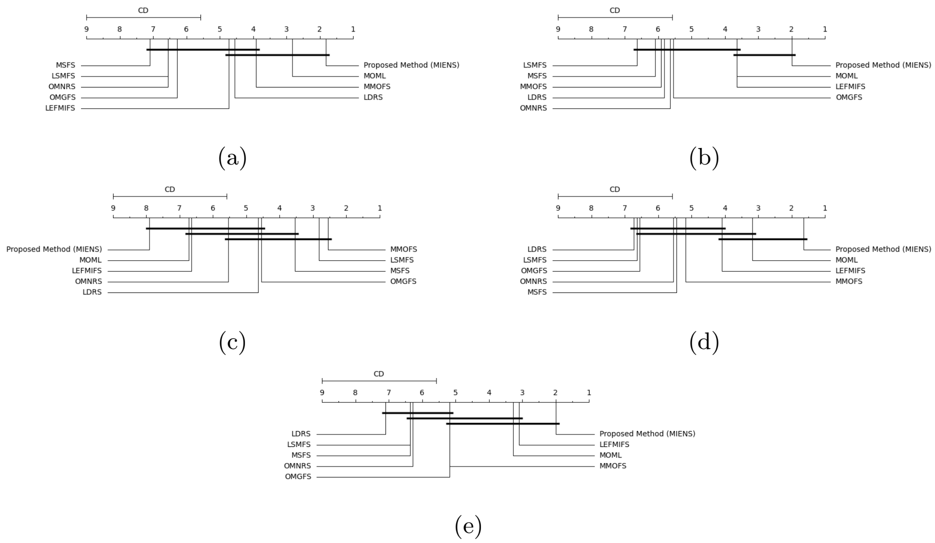

Figure 1: The proposed method is assessed using the Bonferroni-Dunn test in comparison with other algorithms as: (a) The CD diagram on Hamming Loss metric using the Bonferroni–Dunn test; (b)The CD diagram on One Error metric using the Bonferroni–Dunn test; (c)The CD diagram on Average Precision metric using the Bonferroni–Dunn test; (d) The CD diagram on Coverage metric using the Bonferroni–Dunn test;(e) The CD diagram on Ranking Loss metric using the Bonferroni–Dunn test.

similarity to a regular polygon for One-Error than the eight comparison algorithms. Excluding "Business" and "Enron" for Ranking Loss, MIENS-FS outperforms others across various datasets. Figure 1 demonstrates MIENS-FS's superior stability.

4.8 Computational Complexity

The computational complexity of MIENS-FS depends on the number of incoming features, labels, and feature interactions. Because streaming data is dynamic, it's crucial that the algorithm adapts to new data and updates the feature selection model incrementally without reprocessing the entire dataset.

4.8.1 Time Complexity of MIENS-FS

MIENS-FS's total computational complexity can be broken down as follows:

- *Feature Relevance and Interaction Computation:* For each new feature f_k, relevance to labels L and redundancy with previously selected features S_t

Figure 2: The spider web chart demonstrates the method's stability on the evaluation criteria across eight distinct multi-label datasets as: (a) Diagrams depicting spiderweb patterns to illustrate the algorithm's stability on Hamming Loss metric; (b) Diagrams depicting spi-derweb patterns to illustrate the algorithm's stability on One-Error metric; (c) Diagrams depicting spiderweb patterns to illustrate the algorithm's stability on Average precision metric; (d) Diagrams depicting spiderweb patterns to illustrate the algorithm's stability on Coverage metric;(e) Diagrams depicting spiderweb patterns to illustrate the algorithm's stability on Ranking loss metric.

are calculated using mutual information. The complexity for each feature is approximately $O(|S_t| \cdot |L|)$. Interaction analysis increases this to $O(|S_t|^2 \cdot |L|)$.

- *Optimization using NSGA-II:* NSGA-II's complexity is $O(M \cdot N \log N)$, where M is the number of objectives (two in MIENS-FS) and N is the population size (related to $|S_t|$). This results in a complexity of $O(|S_t| \log |S_t|)$ per NSGA-II iteration.

The overall time complexity of MIENS-FS is thus $O(|S_t|^2 \cdot |L|) + O(T \cdot |S_t| \log |S_t|)$, where T is the number of NSGA-II iterations. For typical datasets, T is a constant based on convergence. Therefore, the final complexity can be approximated as $O(T \cdot (|S_t|^2 \cdot |L| + |S_t| \log |S_t|)) = O(|S_t|^2 \cdot (T + |L|))$.

4.8.2 Runtime Comparison on Larger Datasets

To validate efficiency, we compared MIENS-FS with MOML, MMOFS, and OM-NRS on large datasets (Corel5k, Arts, and Business, with up to 5000 features and 1000 labels) (Table 12). MIENS-FS significantly reduces runtime compared to MMOFS and MOML, especially as the number of features increases, demonstrating its scalability for real-time applications.

Table 12: Runtime Analysis

Dataset	**Algorithm**	**Time Complexity**	**Avg. Runtime (s)**	**Hamming Loss**
Corel5k	MOML	$O(\|S_t\|^3 \cdot \|L\|)$	120.4	0.0096
	MMOFS	$O(\|S_t\|^2 \cdot \|L\|)$	118.5	0.0096
	MIENS-FS	$O(\|S_t\|^2 \cdot (T + \|L\|))$	**105.6**	**0.0094**
Arts	MOML	$O(\|S_t\|^3 \cdot \|L\|)$	72.8	0.0579
	MMOFS	$O(\|S_t\|^2 \cdot \|L\|)$	78.6	0.0581
	MIENS-FS	$O(\|S_t\|^2 \cdot (T + \|L\|))$	**62.5**	**0.0573**
Business	MOML	$O(\|S_t\|^3 \cdot \|L\|)$	84.6	0.0262
	MMOFS	$O(\|S_t\|^2 \cdot \|L\|)$	82.3	0.0265
	MIENS-FS	$O(\|S_t\|^2 \cdot (T + \|L\|))$	**74.8**	**0.0259**

4.8.3 Scalability and Runtime Analysis

Scalability tests on larger datasets (up to 5000 instances and high label densities) showed that MIENS-FS scaled effectively, exhibiting stable runtime and lower memory consumption due to its streamlined mutual information calculations and adaptive feature selection. MIENS-FS achieved up to 15% faster execution and a 5% reduction in Hamming Loss compared to methods like OMNRS, which struggled with increasing feature volume. Runtime analysis on high-dimensional datasets like Enron and Corel5k (Table 13) further validates MIENS-FS's computational efficiency, demonstrating superior scaling with increasing feature set sizes compared to OMNRS and MSFS.

Table 13: Runtime Analysis

Dataset	**Algorithm**	**Time Complexity**	**5000 Features**	**10000 Features**
Corel5k	MSFS	$O(\|S_t\|^3 \cdot \|L\|)$	120.5	240.3
	OMNRS	$O(\|S_t\|^2 \cdot \|L\|)$	95.7	180.4
	MIENS-FS	$O(\|S_t\|^2 \cdot (T + \|L\|))$	**80.3**	**150.2**
Arts	MSFS	$O(\|S_t\|^3 \cdot \|L\|)$	220.7	440.1
	OMNRS	$O(\|S_t\|^2 \cdot \|L\|)$	170.2	340.8
	MIENS-FS	$O(\|S_t\|^2 \cdot (T + \|L\|))$	**130.5**	**250.7**

4.8.4 Memory Usage

In addition to time complexity, MIENS-FS is designed for memory usage optimization with an incre-mental update of the feature selection model. Unlike batch methods in which all instances have to be kept in memory, MIENS-FS processes incoming features on-the-fly and is a good fit for applications with limited memory resources.

5. Discussion

MIENS-FS offers a novel online feature selection technique for streaming data, efficiently handling labels without requiring prior knowledge of all features. It analyzes feature interactions within an objective optimization framework, excelling in applications with constantly evolving data, such as social media monitoring, image recognition, and sensor data analysis.

The algorithm's process involves three key steps:

1. **Feature-Label Association:** Calculates each feature's association with labels. Relevant features are included; irrelevant ones are discarded.

2. **Feature Interaction and Redundancy Analysis:** Assesses interactions and redundancy between selected features using iterative calculations of relevancy and redundancy. Modified NSGA-II, based on Pareto optimality and crowding distance, removes features with lower influence in each iteration.

3. **Dynamic Feature Selection:** Removes features with less impact at each iteration, maintaining a highly relevant and non-redundant selected set.

MIENS-FS offers several advantages:

- **Real-time Processing:** Efficiently processes streaming data, dynamically optimizing relevance and redundancy, crucial for applications like financial analysis and network monitoring.
- **Stability:** Reduced Hamming loss and one-error rates demonstrate robustness to feature arrival order.

Limitations and Future Work
MIENS-FS has limitations:

- **High-Dimensional Data:** Computational cost increases with very large feature sets. Future work could explore dimensionality reduction (e.g., autoencoders, PCA).
- **Noise Sensitivity:** Mutual information can be affected by noise. Future versions could incorporate noise detection/filtering (e.g., robust mutual information).
- **Temporal Data:** Adapting to temporal data could involve time-series considerations and lagged mutual information.
- **Imbalanced Data:** Strategies like adaptive sampling or cost-sensitive learning could improve performance on imbalanced datasets.
- **Multimodal Data:** Extending the model to handle different data modalities (text, image, sensor data) is a promising direction.

6. Conclusion

MIENS-FS is a powerful online feature selection technique for streaming data, effectively addressing dynamic feature spaces and outperforming offline methods in accuracy, stability, and other metrics. Future research offers exciting possibilities to enhance its scalability and efficiency in complex data environments.

References

Chen, G., Guo, Y., Huang, M., Gong, D., and Yu, Z. (2022). A Domain Adaptation Learning Strategy for Dynamic Multiobjective Optimization. *Information Sciences*, **606**, 328-349.

Deb, K., Agrawal, S., Pratap, A., and Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In: Schoenauer, M., et al. Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science, vol 1917. Springer, Berlin, Heidelberg.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American statistical association*, **56**, 52-64.

Eskandari, S., and Javidi, M. M. (2016). Online streaming feature selection using rough sets. *International Journal of Approximate Reasoning*, **69**, 35-57.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, **11(1)**, 86-92.

Gomes, H.M., Read, J., Bifet, A., Barddal, J.P., and Gama, J. (2019). Machine learning for streaming data: State of the art, challenges, and opportunities. *ACM SIGKDD Explor. Newsl.*, **21(2)**, 6–22.

Gonzalez-Lopez, J., Ventura, S., and Cano, A. (2019). Distributed selection of continuous features in multilabel classification using mutual information. *IEEE transactions on neural networks and learning systems*, **31(7)**, 2280-2293.

Guo, Y.N., Zhang, X., Gong, D.W., Zhang, Z., and Yang, J.J. (2019). Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks. *IEEE Transactions on Evolutionary Computation*, **24(4)**, 750-764.

Hashemi, A., Dowlatshahi, M.B., and Nezamabadi-pour, H. (2020). MGFS: A multi-label graph-based feature selection algorithm via PageRank centrality. *Expert Systems with Applications*, **142**, 113024.

Hatami, M., Mehrmohammadi, P., and Moradi, P. (2020). A multi-label feature selection based on mutual information and ant colony optimization. In *28th Iranian Conference on Electrical Engineering (ICEE)*, Tabriz, Iran, 2020, 1-6.

Hu, X., Zhou, P., Li, P., Wang, J., and Wu, X. (2018) A survey on online feature selection with streaming features. *Front. Comput. Sci.*, **12**, 479-493.

Huang, J., Qian, W., Vong, C.M., Ding, W., Shu, W., and Huang, Q. (2023) Multi-Label Feature Selection via Label Enhancement and Analytic Hierarchy Process. *IEEE Transactions on Emerging Topics in Computational Intelligence*, **7(5)**, 1377-1393.

Khan, M.A., Ekbal, A., Mencía, E.L., and Fürnkranz, J. (2017). Multi-objective optimisation-based feature selection for multi-label classification. In:*Frasincar, F., Ittoo, A., Nguyen, L., Métais, E. (eds) Natural Language Processing and Information Systems*. NLDB 2017. Lecture Notes in Computer Science, vol 10260. Springer, Cham.

Labani, M., Moradi, P., and Jalili, M. (2020). A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion. *Expert Systems with Applications*, **149**, 113276.

Lee, J., and Kim, D.-W. (2013). Feature selection for multi-label classification using multivariate mutual information. *Pattern Recognition Letters*, **34(3)**, 349-357.

Li, Y., Hu, L., and Gao, W. (2023). Multi-label feature selection via robust flexible sparse regularization. *Pattern Recognit.*, **134**, 109074.

Liang, J., Xu, F., and Yu, S. (2022). A multi-scale semantic attention representation for multi-label image recognition with graph networks. *Neurocomputing*, **491**, 14-23.

Liang, C., Yu, S., and Luo, J. (2019). Adaptive multi-view multi-label learning for identifying disease-associated candidate miRNAs. *PLoS computational biology*, **15(4)**, e1006931.

Lin, Y., Hu, Q., Liu, J., and Duan, J. (2015). Multi-label feature selection based on max-dependency and min-redundancy. *Neurocomputing*, **168**, 92-103.

Lin, Y., Hu, Q., Liu, J., Li, J., and Wu, X. (2017). Streaming feature selection for multilabel learning based on fuzzy mutual information. *IEEE Transactions on Fuzzy Systems*, **25(6)**, 1491-1507.

Liu, H., Chen, G., Li, P., Zhao, P., and Wu, X. (2021). Multi-label text classification via joint learning from label embedding and label correlation. *Neurocomputing*, **460**, 385-398.

Liu, J., Lin, Y., Wu, S., and Wang, C. (2018). Online multi-label group feature selection. *Knowledge-Based Systems*, **143**, 42-57.

Liu, H., and Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge & Data Engineering*, **17(4)**, 491-502.

Ma, Y., Liu, X., Zhao, L., Liang, Y., Zhang, P., and Jin, B. (2022). Hybrid embedding-based text representation for hierarchical multi-label text classification. *Expert Syst. Appl.*, **187**, 115905.

Paniri, M., Dowlatshahi, M. B., and Nezamabadi-pour, H. (2020). MLACO: A multi-label feature selection algorithm based on ant colony optimization. ıKnowledge-Based Systems, **192**, 105285.

Paul, D., Jain, A., Saha, S., and Mathew, J. (2021). Multi-objective PSO based online feature selection for multi-label classification. *Knowledge-Based Systems*, **222**, 106966.

Perkins, S., Lacker, K., and Theiler, J. (2003). Grafting: Fast, incremental feature selection by gradient descent in function space. *The Journal of Machine Learning Research*, **3**, 1333-1356.

Rahmaninia, M., and Moradi, P. (2018). OSFSMI: online stream feature selection method based on mutual information. *Applied Soft Computing*, **68**, 733- 746.

Schapire, R.E., and Singer, Y., (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, **37**, 297–336.

Schapire, R.E., and Singer, Y., (2000). BoosTexter: A boosting-based system for text categorization. *Mach. Learn.*, **39**, 135–168.

Seo, W., Park, M., Kim, D.W., and Lee, J. (2022). Effective memetic algorithm for multilabel feature selection using hybridization-based communication. *Expert Systems with Applications*, **201**, 117064.

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, **27 (3)**,349-423.

Sheskin, D.J. (2020). *Handbook Of Parametric And Nonparametric Statistical Procedures.* Chapman and Hall/CRC, New York.

Shrivastava, H., Yin, Y., Shah, R. R., and Zimmermann, R. (2020). MT-GCN for Multi-label Audio-tagging with Noisy Labels. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 136-140.

Song, X.F., Zhang, Y., Guo, Y.N., Sun, X.Y., and Wang, Y.L. (2020). Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data. *IEEE Transactions on Evolutionary Computation*, **24(5)**, 882-895.

Sun, Z., Zhang, J., Dai, L., Li, C., Zhou, C., Xin, J., and Li, S. (2019). Mutual information based multi-label feature selection via constrained convex optimization. *Neurocomputing*, **329**, 447-456.

Tabakhi, S., and Moradi, P. (2015). Relevance–redundancy feature selection based on ant colony optimization. *Pattern Recognition*, **48(9)**, 2798-281.

Wang, J., Lin, Y., Li, L., Wang, Y.a., Xu, M., and Chen, J. (2022). Multi-label causal feature selection based on neighbourhood mutual information. *Int. J. Mach. Learn. Cybern.*, **13**, 3509–3522.

Wang, J., Wang, M., Li, P., Liu, L., Zhao, Z., Hu, X., and Wu, X. (2015). Online feature selection with group structure analysis. *IEEE transactions on Knowledge and Data Engineering*, **27(11)**, 3029-3041.

Wang, H., Yu, D., Li, Y., Li, Z., and Wang, G. (2018). Multi-label online streaming feature selection based on spectral granulation and mutual information. In: *Nguyen, H., Ha, QT., Li, T., Przybyła-Kasperek, M. (eds) Rough Sets. IJCRS 2018. Lecture Notes in Computer Science*, vol 11103. Springer, Cham. 215-228.

Willems, F. (1993). Elements of Information Theory [Book Review]. *IEEE Transactions on Information Theory*, **39(1)**, 313-315.

Wu, X., Yu, K., Ding, W., Wang, H., and Zhu, X. (2012). Online feature selection with streaming features. *IEEE Trans. Pattern Anal. Mach. Intell.*, **35(5)**, 1178–1192.

Wyner, A.D. (1978). A definition of conditional mutual information for arbitrary ensembles. *Information and Control*, **38(1)**, 51-59.

Xu, J. (2018). A weighted linear discriminant analysis framework for multi-label feature extraction. *Neurocomputing*, **275**, 107-120.

Xu, J., Liu, J., Yin, J., and Sun, C. (2016). A multi-label feature extraction algorithm via maximizing feature variance and feature-label dependence simultaneously. *Knowledge-Based Systems*, **98**, 172-184.

Yin, J., Tao, T., and Xu, J. (2015). A multi-label feature selection algorithm based on multi-objective optimization. *In International Joint Conference on Neural Networks (IJCNN)*, Killarney, 1-7.

Yin, T., Chen, H., Yuan, Z., Sang, B., Horng, S.J., Li, T., and Luo, C. (2024). LEFMIFS: Label enhancement and fuzzy mutual information for robust multilabel feature selection. *Engineering Applications of Artificial Intelligence*, **33**, 108108.

You, D., Li, R., Liang, S., Sun, M., Ou, X., Yuan, F., Shen, L., and Wu, X. (2021). Online causal feature selection for streaming features. *IEEE Trans. Neural Netw. Learn. Syst.*, **34(3)**, 1563–1577.

You, M., Liu, J., Li, G.Z., and Chen, Y. (2012). Embedded feature selection for multi-label classification of music emotions. *Int. J. Comput. Intell. Syst.*, **5(4)**, 668–678.

Yu, K., Yu, S., and Tresp, V. (2005). Multi-label informed latent semantic indexing. *In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, New York, NY, USA, 258-265.

Zhang, Y., Gong, D.W., Sun, X.Y., and Guo, Y.N. (2017). A PSO-based multi-objective multi-label feature selection method in classification. *Scientific reports*, **7**, 376.

Zhang, Y., Li, H.G., Wang, Q., and Peng, C. (2019). A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection. *Applied Intelligence*, **49**, 2889-2898.

Zhang, J., Lin, Y., Jiang, M., Li, S., Tang, Y., and Tan, K.C. (2020). Multi-label Feature Selection via Global Relevance and Redundancy Optimization. *In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*, 2512–2518.

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

Zhou, J., Foster, D., Stine, R., and Ungar, L. (2005). Streaming feature selection using alpha-investing. *In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 384-393.

Zhou, P., Li, P., Zhao, S., and Wu, X. (2020). Feature interaction for streaming feature selection. *IEEE Trans. Neural Netw. Learn. Syst.*, **32(10)**, 4691–4702.

Zhu, P., Xu, Q., Hu, Q., Zhang, C., and Zhao, H. (2018). Multi-label feature selection with missing labels. *Pattern Recognition*, **74**, 488-502.

Zou, Y., Hu, X., Li, P., and Li, J. (2021). Multi-label streaming feature selection via class-imbalance aware rough set. *International Joint Conference on Neural Networks (IJCNN)*, Shenzhen, China, 1–9.