Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085049

ARTICLE

A Hybrid Genetic Algorithm with Information-Theoretic Local Search for Unsupervised Feature Selection

Seyeon Son[1] **and Hyunki Lim**[2,*]

[1]Division of Business Administration, Kyonggi University, Suwon, Republic of Korea
[2]Division of AI Computer Science and Engineering, Kyonggi University, Suwon, Republic of Korea

*Corresponding Author: Hyunki Lim. Email: hlim20@kyonggi.ac.kr

Received: 04 May 2026; Accepted: 31 July 2026; Published: 15 September 2026

ABSTRACT: Feature selection (FS) plays a crucial role in machine learning by reducing data dimensionality and improving learning efficiency. In many real-world scenarios, label information is unavailable, making unsupervised FS particularly important. While Genetic Algorithm (GA) offers a powerful global search mechanism for subset selection, it often suffers from premature convergence and struggles to refine solutions in complex search spaces. To address these limitations, we propose a hybrid GA that integrates an information-theoretic local search strategy for unsupervised FS. The proposed method integrates an information-theoretic local refinement procedure, consisting of DEL and ADD operations based on joint entropy, into a conventional GA framework. Unlike conventional evolutionary methods, our approach leverages information-theoretic measures not merely for evaluation, but as a guiding mechanism for fine-grained local exploration within the GA framework. By incorporating mutual information-based local refinement, the proposed method effectively overcomes the convergence bottlenecks of standard GAs, ensuring a more robust exploitation of feature dependencies. Experimental results on five datasets demonstrate that the proposed method consistently achieves higher clustering performance compared with conventional methods. These results imply that the proposed information-theoretic local refinement effectively mitigates the premature convergence problem of conventional GAs and improves search efficiency and solution quality compared to traditional heuristic and evolutionary approaches. It provides a promising framework for handling high-dimensional data in scenarios where label information is unavailable.

KEYWORDS: Unsupervised learning; feature selection; mutual information; genetic algorithm; particle swarm optimization

1 Introduction

Analyzing all features contained in high-dimensional datasets is a time-consuming and computationally expensive process. Thus, FS plays a crucial role in machine learning by improving model performance, reducing computational complexity, and enhancing interpretability. It can be divided into supervised learning with class labels and unsupervised learning without labels, but obtaining class labels from real datasets is sometimes expensive or impossible. In particular, unsupervised feature selection (UFS) methods have attracted significant attention because they deal with learning tasks without class labels, which are frequently encountered in real-world applications. UFS methods aim to identify a subset of informative and non-redundant features that preserve the intrinsic structure of the data while eliminating irrelevant or noisy features. Traditional UFS approaches can be broadly categorized into filter, wrapper, and embedded

methods [1,2]. Among these, wrapper-based UFS has demonstrated superior performance in many cases because it directly evaluates feature subsets by involving a learning algorithm in the selection process [3].

Despite their promising performance, wrapper-based UFS methods face significant challenges regarding computational complexity and search efficiency, particularly when applied to high-dimensional data. Since these methods require repeated executions of unsupervised learning algorithms, such as k-means clustering, to evaluate each candidate subset, the computational overhead becomes prohibitive as the feature dimensionality increases. To navigate the resulting NP-hard combinatorial search space, metaheuristic algorithms like the GA have been widely adopted [4,5]. GA is a population-based metaheuristic inspired by natural selection. It encodes candidate solutions as binary chromosomes and then evolves into optimal solutions through parent generation through roulette wheels, offspring generation through crossover, and mutation processes. Here, each bit indicates whether the corresponding feature is selected. However, these wrapper approaches often suffer from stagnation and slow convergence rates in large-scale spaces; they are prone to getting trapped in suboptimal regions because the stochastic nature of crossover and mutation operators lacks the precision needed for fine-grained local refinement [6]. Furthermore, standard GA mechanisms do not inherently account for the intricate dependencies among features, often failing to eliminate redundant variables during the evolutionary process. These limitations necessitate a more sophisticated search strategy that can enhance the convergence stability of GAs by incorporating deterministic local guidance. To address these issues, information-theoretic measurements provide a promising means of explicitly quantifying feature relevance and redundancy without relying on class labels. Shannon's entropy measures the uncertainty of a variable, while mutual information quantifies the shared information between two variables [7]. Features with high mutual information between them are considered redundant. Joint entropy further captures higher-order feature interactions, enabling more precise evaluation of feature subsets in unsupervised settings.

Motivated by the complementary strengths of GA-based global search and information-theoretic local guidance, we propose a novel UFS framework that integrates an information-theoretic local refinement strategy into a conventional GA. Our method incorporates a local refinement process that leverages mutual information and related measures to prioritize informative and mutually independent features. By combining the global exploration capability of GA with the fine-grained optimization offered by local refinement, the proposed approach achieves a better balance between search efficiency and selection accuracy. This integration not only accelerates convergence but also improves the quality of the selected feature subsets, leading to more robust UFS.

The principal contributions and advantages of the proposed framework are summarized as follows:

- Hybrid Search Strategy: We introduce a dual-layered search mechanism that integrates the global exploration of GA with a deterministic, information-theoretic local refinement. This hybrid approach effectively mitigates the exploration-exploitation trade-off, ensuring both broad search coverage and precise local optimization.
- Enhanced Convergence via Local Refinement: By incorporating mutual information-based guidance into the evolutionary process, our method overcomes the common GA pitfall of premature convergence. The local refinement accelerates the search toward high-quality regions of the feature space, significantly reducing the number of generations required to find optimal subsets.
- Information-Theoretic Criteria: The computational intractability of estimating full-order joint entropy in high-dimensional spaces motivates a different approach. We derive a formal objective function that decomposes complex joint dependencies into computable mutual information terms, providing a robust proxy for the total information content of a subset. This analytical formulation ensures that the genetic search is guided by a mathematically consistent and efficient local optimization criterion.

The remainder of this paper is organized as follows. Section 2 discusses the difference between supervised and unsupervised learning in FS and reviews related research. Section 3 specifically describes the proposed GA framework and the information-theoretic local refinement. Section 4 presents the experimental settings, comparative results, and analysis. The final section 5 summarizes the study and discusses future research directions.

2 Related Work

FS is an essential preprocessing step that identifies informative and non-redundant features from high-dimensional datasets. Depending on the availability of class label information, FS methods can be broadly divided into supervised and unsupervised approaches.

2.1 Supervised Method

Improved Ensemble Classifier with Genetic Algorithm (IECGA) [8] is a five-stage ensemble-based framework that integrates GA-based FS with heterogeneous supervised classifiers. It consists of a training layer encompassing FS, base classification, and ensemble classification, and a testing layer for prediction. A GA search method combined with a Correlation-based FS evaluator is employed to identify an optimal feature subset. The selected features are fed into three base classifiers—Random Forest, Support Vector Machine, and Naïve Bayes—which are subsequently combined through a soft voting ensemble technique. Similarly, Importance-Guided Particle Swarm Optimization (IGPSO) [9] is a hybrid framework that combines a two-stage MLP-based focal neural network with importance-guided binary PSO for large-scale FS. The focal neural network learns a feature importance vector using positive and negative sample training, which is then used to guide both population initialization and the particle update strategy. While both approaches demonstrate strong performance, they remain supervised methods that fundamentally depend on class label information, limiting their applicability to environments where labeled data is available.

2.2 Unsupervised Method

UFS can be broadly categorized into filter and wrapper methods [1,2,10].

2.2.1 Filter Method

Filter methods do not rely on a specific learning algorithm. Instead, they generate feature subsets by evaluating individual features or the relationships among features. To compute the importance of features, statistical or information-theoretic measures such as variance-covariance, linear correlation, entropy, and mutual information are used [11,12]. The Maximal Information Compression Index is a metric used to evaluate features based on their variance–covariance structure [13]. This approach aims to minimize information redundancy within the selected feature subset.

The dependency-based FS [14] uses statistical measures of dependence between features. If the correlation with other features is low, those features are evaluated as less dependent. Features with low dependence are considered unnecessary in the clustering process for label classification. For this reason, we select features with high dependence. The entropy-based filter method [15] proposes a distance-based similarity entropy. To measure the entropy of the feature, it is re-implemented as an exponential function instead of the usual logarithmic function. The Laplacian score combined with distance-based entropy measurements [16] presents a scheme to combine the Laplacian score with the distance entropy proposed by the entropy-based filter method [15]. The features are selected based on the distance entropy and the ranking generated by the

Laplacian score. Thus, information-theoretic approaches can use mutual information volume, entropy, etc. to evaluate the relevance and information redundancy of unlabeled features and generate feature subsets.

These filter methods have the advantage of low computational complexity, enabling very fast processing even for large-scale datasets. However, they usually rely on individual or feature evaluations to find optimal solutions, but do not explicitly optimize subsets, so the resulting subset of features may still contain information redundancy.

2.2.2 Wrapper Method

Wrapper methods differ from filter methods in that they iteratively evaluate feature subsets using a fitness function to search for the optimal combination. Since label information is unavailable in unsupervised settings, it is essential to employ a fitness function that can evaluate underlying structural properties of the given dataset [1,17–19].

Wrapper-based approaches typically adopt evolutionary algorithms that mimic biological evolution to search for optimal feature subsets. A typical example is GA, which evolves candidate solutions through selection, crossover, and mutation operators inspired by biological reproduction [4,5,20–22]. GA encodes possible feature combinations as chromosomes and evaluates each combination using a fitness function. By introducing stochastic elements into the search process, GA increases the selection probability of high-fitness parent individuals through a roulette wheel selection mechanism and generates offspring through crossover operations. In addition, mutation, which flips certain values randomly, helps alleviate the local optimum problem.

Kim et al. introduce four heuristic fitness criteria in conjunction with the k-means clustering algorithm [23]. Specifically, F_{within} evaluates how well clusters are compact by measuring the distances between cluster centers and the features belonging to each cluster. $F_{between}$ measures the separability between clusters by calculating the distances between cluster centers. $F_{clusters}$ and $F_{complexity}$ assess the appropriate number of clusters and the number of features, respectively, thereby providing an overall evaluation of clustering performance.

In addition, UFS based on Ant Colony Optimization (UFSACO) does not use a fitness function but utilizes an iterative population-based search mechanism similar to the wrapper method's optimization. UFSACO imitated the process by which ants use pheromones to find food by optimal routes. This method proposes a cosine similarity metric to quantify pairwise feature similarity between features [24]. Features are represented as nodes, and cosine similarity between features forms the edges of an undirected graph, which serves as the search space. During the search process, ants probabilistically prefer features with low similarity and high pheromone values. Ultimately, features with the highest pheromone values are sequentially selected for the final subset.

Particle Swarm Optimization (PSO) has strong global search capability and fast convergence rates, but is sensitive to parameter settings. To address these limitations, a filter-based barebone PSO (FBPSO) that does not require parameter adjustment has been proposed [25]. FBPSO is a hybrid approach that incorporates filter-based preprocessing steps within the wrapper-based PSO search framework. In this work, particles represent a subset of features, which move towards achieving stochastic optimal performance based on their own experience and the global best performance of the cluster. FBPSO first removes irrelevant or weakly related features based on feature correlations as a preprocessing step. After that, the algorithm repeatedly refines the feature subset using the DEL operator and the ADD operator based on the mutual information between the features. However, the filter component cannot capture higher-order dependence between the features, relying solely on pair-by-pair feature correlations. More broadly, existing hybrid methods tend to

use informational measurements only as an evaluation criterion rather than an active guiding mechanism in the search process, and early convergence problems are likely to occur in high-dimensional feature spaces. Redundancy reduction strategies such as maximizing relevance while minimizing redundancy have been extensively studied in supervised settings [7], yet their systematic application in UFS remains limited.

Since wrapper methods repeatedly evaluate feature subsets, they have the advantage of producing optimized feature subsets with lower information redundancy compared to filter methods. However, they require substantial computational cost and time, and they also suffer from a convergence problem, where the search becomes trapped in a prematurely converged local optimum and fails to escape toward a globally superior solution [6]. Hybrid approaches integrate the strengths of both filter and wrapper methods by combining preprocessing steps with evolutionary search, aiming to reduce computational cost while maintaining high-quality feature subsets [25]. However, such sequential combinations lack a systematic workflow that interleaves global search with fine-grained local refinement, leaving the convergence problem fundamentally unresolved. To address the inherent convergence issues of metaheuristic wrapper approaches, we propose a hybrid framework that augments the global search capabilities of GA with a specialized local refinement module. This integrated approach is designed to refine the search trajectory through analytical guidance, thereby ensuring the selection of more robust and diverse feature subsets. For a clearer comparison, Table 1 categorizes the filter and wrapper methods based on their advantages, limitations, and practical applications, supplemented by statistical benchmarks reported in previous studies.

Table 1: Comparison of representative UFS methods.

Method Type	Method	Advantages	Disadvantages	Applications	Statistical Results
Filter	Maximal Information Compression Index [13]	Low cost of calculation	Sensitive to similarity thresholds	Image recognition	Higher accuracy than competition methods
Filter	Dependency-based FS [14]	Precise noise feature exploration	Only global dependency measurements, Possible exclusion of useful features	Healthcare, Biodiversity	Error 23.07%p reduced
Filter	Entropy-based FS [15]	Low parameter sensitivity	Exhaustive search costs	Genetics	Select the same optimal subset as the full navigation
Filter	Laplacian Score + Distance-based Entropy [16]	High stability, Fast processing time	Parameter setting required	Medical image analysis	Improved accuracy on medical image datasets
Wrapper	GA [4,5,20–22]	Avoids local optimum	High computational costs, Premature convergence	Finance, Healthcare	100% accuracy (lung cancer)
Wrapper	ELSA [23]	Explore the number of clusters and feature subset at the same time	Low performance in complex data	Breast cancer prediction, Target marketing	Consistently outperforms greedy search on models with fewer features

(Continued)

Table 1 (continued)

Method Type	Method	Advantages	Disadvantages	Applications	Statistical Results
Wrapper	UFSACO [24]	Applicable to higher-dimensional data	A number of parameters	Cancer diagnosis	Average error 19.84%
Wrapper	FBPSO [25]	No parameter tuning required	Unable to capture higher order dependencies	Medical diagnosis	Accuracy up to 89.5%

3 The Proposed Method

3.1 Preliminary

UFS aims to identify an informative subset of features that best represents the intrinsic structure of the data without relying on label information. Given a data matrix $X \in \mathbb{R}^{n \times d}$, where n denotes the number of samples and d denotes the number of features, the goal is to find a subset $S \subseteq \{1, 2, \ldots, d\}$ ($|S| \ll d$) that maximizes the representativeness and diversity of information contained in X. In the absence of labels, the evaluation of feature subsets is often based on intrinsic information measures such as mutual information. Consequently, the optimization objective becomes non-differentiable and combinatorial, motivating the use of heuristic search-based methods.

GA provides an effective meta-heuristic framework to address such non-convex and discrete optimization problems [26]. In the context of UFS, each chromosome encodes a candidate feature subset, and the population evolves toward higher-quality subsets through evolutionary operators such as selection, crossover, and mutation. Each chromosome is represented as a binary vector, where a value of 1 indicates that the corresponding feature is selected. The initial chromosomes are generated by randomly setting a fixed number of bits, corresponding to the initial subset size, to 1. Parent selection is performed using a roulette-wheel mechanism, where the probability of selection is proportional to the fitness value. Crossover generates two offspring from two selected parents by exchanging segments of the parent chromosomes at a specific crossover point. The generated offspring then undergo mutation, in which the value of a randomly selected bit is changed from 0 to 1 or from 1 to 0. Unlike gradient-based methods that require differentiable objective functions, GA can explore discontinuous search spaces and avoid being trapped in local minima through stochastic recombination and mutation [23]. Moreover, the population-based nature of GA allows for parallel exploration of multiple candidate subsets, which enhances global search capability. Owing to these advantages, GA has been widely applied to FS problems as a flexible and general-purpose optimizer for discrete feature subset selection [3].

However, the standard crossover and mutation operations in a GA are not inherently suitable for FS, since the best individual features do not necessarily constitute the best feature subset [7]. In particular, the crossover operator tends to preserve individual features that appear favorable in isolation, without considering their joint redundancy or complementarity. To overcome these limitations and to enhance the discriminative capability of the selected subset while controlling its size, we introduce an information-theoretic local refinement mechanism applied to each chromosome in the genetic population. This refinement adaptively adds or removes features based on mutual information criteria, thereby guiding the evolutionary process toward more informative and compact feature subsets. The proposed framework consists of a GA-based global search module and an information-theoretic local refinement module. The GA conducts global exploration through selection, crossover, and mutation operators, while the local refinement module applies DEL and ADD operations based on joint entropy to the top-ranked elite chromosomes. This

ensures both broad search coverage and precise local optimization simultaneously. A clustering-based fitness function combining F_{within}, F_{between}, and $F_{\text{complexity}}$ is employed to evaluate the quality of each candidate feature subset.

The amount of uncertainty of a random variable X can be quantified by the Shannon's entropy, defined as

$$H(X) = -\sum P(X)\log P(X), \tag{1}$$

where $P(X)$ denotes the probability mass function of X. Entropy measures the expected amount of information required to describe the outcome of X, and thus serves as a fundamental quantity in information theory for uncertainty. The mutual information between two random variables X and Y represents the amount of information shared between them, and is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{2}$$

where $H(X,Y)$ denotes the joint entropy of X and Y. Mutual information can also be expressed as the reduction in uncertainty of one variable due to knowledge of the other. The conditional entropy quantifies the remaining uncertainty of one variable given that the other is known, and is defined as

$$H(X|Y) = H(X,Y) - H(Y) = H(X) - I(X;Y). \tag{3}$$

Conditional entropy is useful for measuring how much additional information is needed to describe X when Y is observed.

To evaluate fitness, we construct a fitness function by combining the heuristic metrics F_{within}, F_{between}, and $F_{\text{complexity}}$ from an evolutionary local selection algorithm [23]. These metrics are combined to simultaneously consider cluster compactness, separability, and model complexity. It is defined as follows, where the weights of each metric, α, β, and γ, are set to 1.

$$\alpha F_{\text{within}} + \beta F_{\text{between}} + \gamma F_{\text{complexity}} \tag{4}$$

F_{within} is a metric to evaluate how well clusters are cohesive. It is computed by measuring the distance between each cluster center and the features belonging to that cluster after running K-means. F_{between} is a metric to evaluate the separation between clusters. It is computed by measuring the distances between each cluster center and the data points belonging to other clusters. Since the goal is to derive a subset with low redundancy and high performance, a larger F_{within} and F_{between} indicate a superior subset. $F_{\text{complexity}}$ is a metric to evaluate the number of features selected in the subset, defined as follows, where d denotes the number of features included in the subset and D denotes the total number of features in the dataset. Since the objective is to derive a high-performing subset composed of the minimal number of informative features, a larger value indicates a better subset.

$$F_{\text{within}} = 1 - \frac{1}{nd}\sum_{i=1}^{n}\left\|X_{\text{sel}}^{(i)} - \text{centers}_{\text{labels}[i]}\right\|^2 \tag{5}$$

$$F_{\text{between}} = \frac{1}{n(k-1)d}\sum_{i=1}^{n}\sum_{\substack{c=1 \\ c\neq\text{labels}[i]}}^{k}\left\|X_{\text{sel}}^{(i)} - \text{centers}_{c}\right\|^2 \tag{6}$$

$$F_{\text{complexity}} = 1 - \frac{d-1}{D-1} \tag{7}$$

3.2 Local Refinement

3.2.1 Motivation for Local Refinement

Let $F = \{f_1, f_2, \cdots, f_d\}$ be the original dataset. From an information-theoretic perspective, the goal of unsupervised FS is to minimize the information loss between the full set F and the selected subset S, that is, to minimize the difference between $H(F)$ and $H(S)$. Since $H(F)$ is constant and $H(S) \leq H(F)$, the objective function can be written as

$$\arg\max_{S} H(S). \tag{8}$$

In the context of a GA, the subset S is encoded as a binary chromosome, where each bit indicates whether a corresponding feature is selected. To refine a given chromosome, features can either be removed from or add to S. From the information-theoretic viewpoint, the reduction in entropy caused by feature removal is designed to be minimized, whereas the increase in entropy resulting from feature addition is maximized. This principle forms the basis of the proposed local refinement strategy, which adaptively updates chromosomes to preserve maximal information content while maintaining compactness.

3.2.2 DEL Operation for a Chromosome

In the case of removing a feature from the subset S, the resulting subset $S - \{f^-\}$ is selected to minimize the information loss, measured as the difference between $H(S)$ and $H(S - \{f^-\})$. Let S^- be subset obtained by removing a feature f^- from S. The corresponding optimization problem can be written as

$$\arg\max_{f^- \in S} H(S - \{f^-\}). \tag{9}$$

However, since $H(S - \{f^-\})$ represents a high-dimensional joint entropy, it is generally impractical to estimate it accurately due to the exponential growth of the joint probability space. To address this estimation, we approximate $H(S - \{f^-\})$ using lower-order joint entropy terms, which provide a tractable yet informative surrogate for the true joint entropy.

Lee and Kim [27] define sum of the k-cardinality entropy as

$$U_k(X) = \sum_{Y \in X'_k} H(Y), \tag{10}$$

where X' is the power set of X, and $X'_k = \{e | e \in X', |e| = k\}$. Based on this definition, the upper bound of Han's inequality [28] can be written as

$$H(X) \leq \frac{1}{n-1} U_{n-1}(X'), \tag{11}$$

where n is the number of variables in X. From this upper bound, Lee and Kim [27] derive the Lemma 1 as:

Lemma 1: Let $U_k(S')$ be the k-cardinality entropy of given variable sets S. Then the lower bound and the upper bound of $U_k(S')$ can be defined as

$$\frac{1}{k-1}\left(kU_k(S') - \binom{n-1}{k-1}U_1(S')\right) \leq U_k(S') \leq \left(\frac{n-k+1}{k-1}\right)U_{k-1}(S'). \tag{12}$$

Lemma 1 indicates that the upper bound of $U_k(S')$ is determined by the $(k-1)$-cardinality entropy term. From this Lemma, Seo et al. [29] obtained the k-cardinality approximation of the high-dimensional joint entropy $H(X)$ as:

Theorem 1: Upper bound of the $H(X)$ with k-cardinality entropy is

$$H(X) \leq \left(\prod_{i=1}^{b} \frac{i}{n-1} \right) U_k(X'), \tag{13}$$

where $b = \min(n-k, k-1)$.

Theorem 1 implies that the approximation becomes tighter as the value of K increases, leading to a more accurate estimation of the overall entropy of $H(X)$ [30]. Based on this observation, the joint entropy term $H(S, f^+)$ can be approximated by the sum of k-cardinality entropies. In the proposed local refinement scheme, k is fixed to two in order to balance computational efficiency and approximation accuracy. If k is fixed to one, then it is univariate entropy. Accordingly, the objective function (9) can be approximated as

$$\begin{aligned} J_{\mathrm{DEL}} &= \arg\max_{f^- \in S} H(S - \{f^-\}) \\ &\approx \arg\max_{f^- \in S} \left(\prod_{i=1}^{b} \frac{i}{|S|-2} \right) U_2(S - \{f^-\}) \\ &= \arg\max_{f^- \in S} U_2(S - \{f^-\}) \\ &= \arg\max_{f^- \in S} \sum_{f_i, f_j \in S \setminus f^-} H(f_i, f_j) \end{aligned} \tag{14}$$

where $b = 1$ and the term related to b is independent to objective function. This objective function deselects f^- where the joint entropy between features in $S \setminus f^-$ is maximized.

Note that the total pairwise entropy over the subset S can be decomposed as

$$U_2(S) = \sum_{f_i, f_j \in S \setminus f^-} H(f_i, f_j) + \sum_{f_i \in S \setminus f^-} H(f_i, f^-). \tag{15}$$

Since $U_2(S)$ remains constant for a fixed subset S, maximizing the first term $\sum_{f_i, f_j \in S \setminus \{f^-\}} H(f_i, f_j)$ is equivalent to minimizing the second term $\sum_{f_i \in S \setminus \{f^-\}} H(f_i, f^-)$. To simultaneously minimize the individual information loss of f^-, we incorporate its self-entropy $H(f^-) = H(f^-, f^-)$ into the objective. The optimization problem can thus be extended over the entire subset S as:

$$\arg\min_{f^- \in S} \sum_{f_i \in S} H(f_i, f^-). \tag{16}$$

This reformulation allows for highly efficient matrix-based vectorization, ensuring the removal of the feature with the least individual information and synergy.

3.2.3 ADD Operation for a Chromosome

In addition to the objective function designed for deletion, an analogous objective function can also be formulated for the addition operation. In the case of adding a feature from the subset S, the resulting subset $\{S, f^+\}$ is aimed at maximizing the joint entropy. The corresponding optimization problem can be written as

$$\arg\max_{f^+\notin S} H(S, f^+). \tag{17}$$

From the Theorem 1, the objective function (17) can be approximated as

$$\begin{aligned} J_{\mathrm{ADD}} &\approx \arg\max_{f^+} \prod_{i=1}^{b} \frac{i}{|S|+1-i} U_2(\{S', f^+\}') \\ &= \arg\max_{f^+} \frac{1}{|S|} U_2(\{S', f^+\}'), \end{aligned} \tag{18}$$

where $b = 1$. Regardless of whether the candidate feature f^+ is included in the subset S or not, (18) can be reformulated as

$$J_{\mathrm{ADD}} \approx \arg\max_{f^+} \frac{1}{|S|} (U_2(S') + U_2(f^+ \times S')), \tag{19}$$

where $\times$ denotes the Cartesian product between two sets. Since $U_2(S')$ is independent of f^+, it can be removed. Therefore, the final objective function can be written as

$$\begin{aligned} J_{\mathrm{ADD}} &\approx \arg\max_{f^+} U_2(f^+ \times S') \\ &= \arg\max_{f^+} \sum_{f\in S} H(f^+, f). \end{aligned} \tag{20}$$

Finally, the proposed local refinement process can be summarized by two complementary operations: feature removal (DEL) and addition (ADD), which are defined as

$$\mathrm{DEL}: f^- = \arg\min_{f^-\in S} \sum_{f_i\in S} H(f_i, f^-), \tag{21}$$

$$\mathrm{ADD}: f^+ = \arg\max_{f^+\notin S} \sum_{f\in S} H(f^+, f), \tag{22}$$

where the DEL operation removes the feature contributing the least entropy within the current subset, and the ADD operation selects the feature that maximizes the joint entropy with the already selected features. These two operators enable the adaptive refinement of chromosomes toward subsets that preserve higher information content while maintaining compactness. The number of selected features is kept constant during the refinement process.

3.3 Proposed Algorithm

Specifically, Algorithm 1 invokes Algorithm 2 in each generation by passing the top h elite chromosomes as input. For each chromosome C_i, Algorithm 2 first applies m_i DEL operations to remove the most redundant features and then applies m_i ADD operations to incorporate the most informative unselected features. The number of operations is determined as $m_i = \max(1, m)$, where $m = \lfloor c/2 \rfloor$ and c denotes the current subset size. The refined chromosomes $N_1(t)$ are merged with the offspring generated by the GA, $N_2(t)$, and the top p individuals are selected to form the next generation $P(t)$.

Algorithm 1: Proposed algorithm

Input: p: size of population, h: number of generated chromosomes by local refinement, g: number of generated chromosomes by genetic operator, max_iter: maximum number of generation
procedure *PROPOSED ALGORITHM*
initialize and evaluate $P(0)$
for $t = 1$ **to** max_iter **do**
select top-h chromosomes C in $P(t-1)$
create $N_1(t)$ from C using Algorithm 2
create $N_2(t)$ from $P(t-1)$ using genetic operators
evaluate $N_1(t)$ and $N_2(t)$
add $N_1(t)$ and $N_2(t)$ to $P(t-1)$
select $P(t)$ from $P(t-1)$
end
end

Algorithm 2: Local refinement

Input: C: some chromosomes in $P(t)$, m: number of DEL and ADD operations
procedure *LOCAL REFINEMENT(C)*
for $i = 1$ **to** $|C|$ **do**
$m_i \leftarrow \max(1, m)$
for $j = 1$ **to** m_i **do**
apply DEL to C_i using Eq. (21)
end
for $j = 1$ **to** m_i **do**
apply ADD to C_i using Eq. (22)
end
end
end

We adopt a memetic search framework in which a GA conducts global exploration while an information-theoretic local refinement operator performs on top-ranked individuals. In each iteration, the algorithm selects the top-h chromosomes from the current population and applies DEL/ADD operations guided by lower-order entropy surrogates. The refined elites, together with GA-generated offspring, are then merged and down-selected to form the next generation, preserving the population size p and enforcing elitism. Total algorithm is represented in Algorithm and the algorithm for the local refinement is in Algorithm 2. The overall flow of the proposed method can be seen in Fig. 1, where the outer box represents the overall framework of Algorithm 1 and the inner dashed box highlights the local refinement procedure of Algorithm 2.

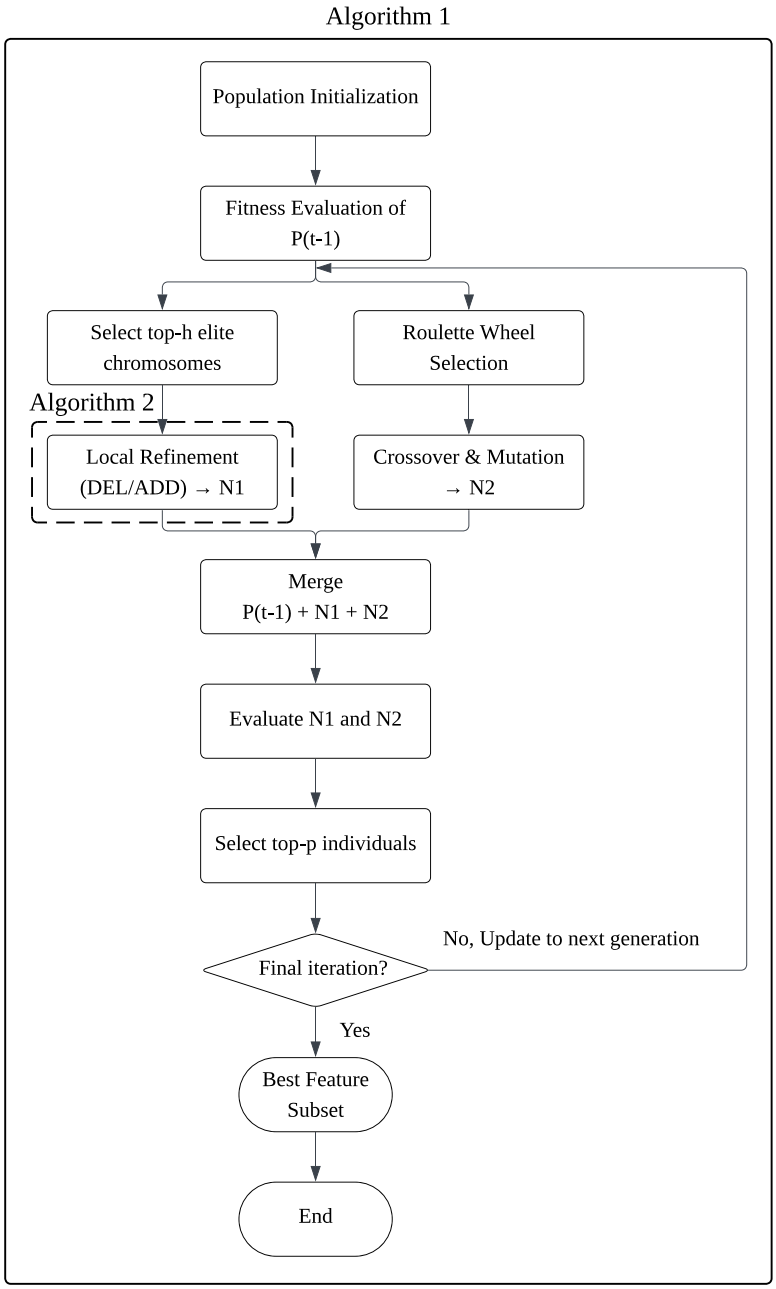


Figure 1: Overview of the proposed method.

3.4 Computational Complexity Analysis

Let n denote the number of samples, d the total number of features, p the population size, h the number of chromosomes subject to refinement, m the number of ADD and DEL operations per chromosome, and $|S|$ the average number of selected features in a chromosome. The complexity of the proposed method can be analyzed in three main parts: the initialization, the genetic evolution, and the local refinement process.

1) Initialization and Evaluation.

The initial evaluation of fitness values computes the clustering-based heuristic metrics. Assuming k-means clustering is used, the fitness evaluation takes $O(p \cdot I \cdot k \cdot n \cdot |S|)$, where I is the number of clustering iterations and k is the number of clusters.

2) Genetic Operations.

In each generation, crossover and mutation operations are performed on p chromosomes, resulting in a computational cost of $O(p \cdot d)$. The offspring evaluation adds another $O(p \cdot |S|)$, which is typically dominated by the evaluation of entropy- or information-based metrics. The selection step requires sorting or ranking the population, incurring an additional cost of $O(p \log p)$.

3) Local Refinement.

The proposed refinement scheme operates on the top-h chromosomes using both DEL and ADD operations. For the DEL operation, removing a feature with the smallest entropy value from S requires $O(|S|)$ operations per refinement, resulting in $O(h \cdot m \cdot |S|)$ in total. The ADD operation evaluates each candidate feature $f^+ \notin S$ by computing the pairwise joint entropy with all features in S, i.e.,

$$f^+ = \arg\max_{f^+ \notin S} \sum_{f \in S} H(f^+, f),$$

which has a complexity of $O((d - |S|) \cdot |S|)$ per ADD operation. Hence, the total complexity of the local refinement step is $O(h \cdot m \cdot (d - |S|) \cdot |S|)$ for exhaustive search. If a candidate sampling strategy of size $r \ll d$ is adopted, the complexity can be reduced to $O(h \cdot m \cdot r \cdot |S|)$.

4) Overall Complexity.

Considering all components, the computational complexity of a single generation can be approximated as

$$O\Big(p \cdot d + h \cdot m \cdot (d - |S|) \cdot |S| + p \log p\Big).$$

If the local refinement uses sampling or pre-computed entropy tables, the dominant term is $O(p \cdot d)$, making the algorithm relatively scalable to high-dimensional datasets. The pre-computation of all pairwise entropies $H(f_i, f_j)$ incurs an additional one-time cost of $O(d^2)$, which can be amortized over multiple generations.

5) Memory Complexity.

The memory requirement is mainly determined by the storage of pairwise entropy values, which is $O(d^2)$ in the full setting. To alleviate this cost, block-wise computation or sparse storage strategies can be applied by excluding feature pairs with negligible mutual information.

Overall, the proposed algorithm maintains a reasonable computational complexity while enhancing the convergence stability and solution quality through information-theoretic local refinement.

4 Experimental Results

4.1 Experimental Settings

In this study, to validate the performance of the proposed method, three wrapper-based UFS techniques UFSACO [24], Evolutionary local selection algorithm based GA (ELSA) [23], and FBPSO [25] were selected as comparison models. There are five data used in the experiment: Glioma, Prostate GE, TOX 171, Yale,

and warpAR10P. Normalized Mutual Information (NMI) and Clustering Accuracy Score (AC) are used as evaluation indicators.

ELSA follows a general GA algorithm and uses a clustering evaluation index as a fitness function. This function also has the same fitness function used in the method we propose. FBPSO generates a feature subset by evaluating the dissimilarity between selected features and the similarity between unselected features. UFSACO evaluates the similarity of features using cosine, through which a numerical value called pheromone is obtained and features with high values are sequentially selected.

All datasets are normalized to the range of [0, 1] using MinMaxScaler before the FS process. The parameters of the comparative methods were set as follows. For UFSACO, the maximum number of generations was set to 20, the number of ants in each generation was set to six, and the number of features explored by each ant was set equal to the initial subset size used in the proposed method. For ELSA and FBPSO, the number of generations was set to 20, and the population size for each generation was set to six. In particular, for a fair comparison with the proposed method, the number of offspring generated per generation in ELSA was fixed at six. This is because the proposed method generates two offspring and four additional individuals through local refinement in each generation. The number of ADD and DEL operations, denoted by m, was determined based on the number of selected features c in the chromosome undergoing local refinement. The main parameters of the proposed method were set as p:6, m:$\frac{c}{2}$, h:4, g:2, and max_iter:20.

To evaluate the clustering performance of the selected feature subsets, we conduct k-means clustering and measure the quality of the selected features. The number of clusters, k, is set equal to the true number of classes in the dataset. Since the k-means algorithm initializes cluster centers randomly, the clustering results may vary across different runs. Therefore, to ensure the reliability of the evaluation, k-means is executed independently 10 times for each feature subset, and the average of the results is used as the final metric. We employ NMI and AC as evaluation metrics [31–34] as

$$\mathrm{NMI} = \frac{2I(Y;\hat{Y})}{H(Y) + H(\hat{Y})} \tag{23}$$

$$\mathrm{AC} = \frac{1}{n} \max_{\pi} \sum_{i=1}^{k} M_{i,\pi(i)} \tag{24}$$

Both metrics take values between 0 and 1, and the evaluation is performed based on the clustering results obtained using k-means, a representative algorithm for unsupervised learning. The first metric, NMI, measures how statistically similar the clustering results are to the true class label structure of the data from an information-theoretic perspective. A higher NMI value indicates that the selected feature subset preserves the original class structure of the data well. $H(Y)$ and $H(\hat{Y})$ denote the information entropy used to normalize the mutual information value into the range between 0 and 1, and are computed as follows:

$$H(Y) = -\sum_{y \in Y} p(y) \log p(y) \tag{25}$$

$I(Y;\hat{Y})$ represents mutual information, which quantifies the amount of shared information between the true labels and the predicted results, and is defined as:

$$I(Y;\hat{Y}) = H(Y) - H(Y|\hat{Y}) \tag{26}$$

AC measures the matching ratio between cluster labels and true labels. Since the cluster labels obtained from the k-means algorithm are assigned arbitrarily, the Hungarian algorithm is applied to perform an

optimal one-to-one mapping between cluster labels and true labels. The AC is computed as follows, where $M_{i,\pi(i)}$ denotes the number of samples belonging to the true class i that are assigned to the predicted cluster $\pi(i)$.

These two evaluation metrics have been widely adopted in numerous prior studies on UFS [35–38]. Specifically, NMI measures the statistical dependency between clustering results and the actual class structure from an information-theoretic perspective. AC, on the other hand, directly measures the proportion of correctly classified samples through the Hungarian algorithm. However, each metric has its own limitations. NMI captures statistical dependency well but does not directly reflect misclassifications at the individual sample level. AC measures sample-level accuracy but does not account for differences in information content across clusters. Therefore, by employing both metrics together, the quality of unsupervised feature subsets can be evaluated more reliably from two complementary perspectives: statistical agreement and sample-level classification accuracy.

Table 2 summarizes the five data sets employed in this study, which comprise three biological and two facial image data sets. The biological group includes the Glioma [39] data set, which provides information on glioblastoma from 50 subjects, the Prostate GE [40] data set, containing 102 tissue samples, and TOX 171, which features gene expression profiles under various toxic conditions. To further validate the proposed method, two facial image datasets were incorporated: Yale and warpAR10P, consisting of images from 15 and 10 individuals, respectively. All datasets were sourced from the corresponding repository [41].

Table 2: Information about datasets.

Datasets	No. of Instances	No. of Features	No. of Classes
Glioma	50	4434	4
Prostate GE	102	5966	2
TOX 171	171	5748	4
Yale	165	1024	15
warpAR10P	130	2400	10

All experiments were conducted using Python 3.12.13 in the Google Colab environment, which provides cloud-based computing resources. The implementation utilized the scikit-learn library for k-means clustering, NMI calculation, and MinMaxScaler normalization, NumPy for matrix operations, and the SciPy library for the Hungarian algorithm used in AC calculation.

4.2 Comparison Results

Tables 3 and 4 present the NMI and AC results of the proposed method and the comparative algorithms, respectively. The bold values represent the best performance for each dataset. In both evaluation metrics. In particular, for the NMI metric, the proposed method on the TOX 171 dataset significantly outperformed FBPSO and UFSACO. In terms of AC, the proposed method achieved a score of 0.5007 on the Yale dataset, showing a substantial performance improvement compared to UFSACO, which recorded the lowest performance, and also outperforming ELSA. These results objectively demonstrate that the proposed method effectively captures the interactions and information among multivariate features, thereby deriving a more optimized feature subset.

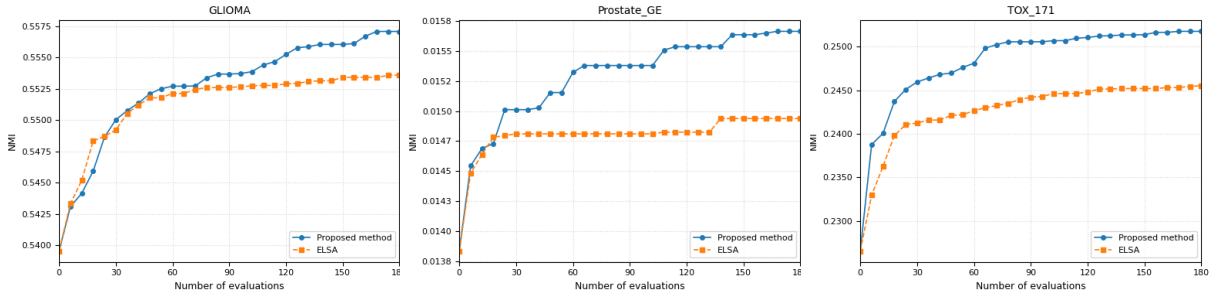
Table 3: Comparison of NMI results between the proposed method and comparative algorithms.

Datasets	ELSA	FBPSO	UFSACO	Proposed
Glioma	0.5528	0.5236	0.4130	**0.5547**
Prostate GE	0.0148	0.0136	0.0126	**0.0155**
TOX 171	0.2446	0.2253	0.1955	**0.2510**
Yale	0.5423	0.5174	0.4774	**0.5434**
warpAR10P	0.3100	0.2707	0.2697	**0.3118**

Table 4: Comparison of AC results between the proposed method and comparative algorithms.

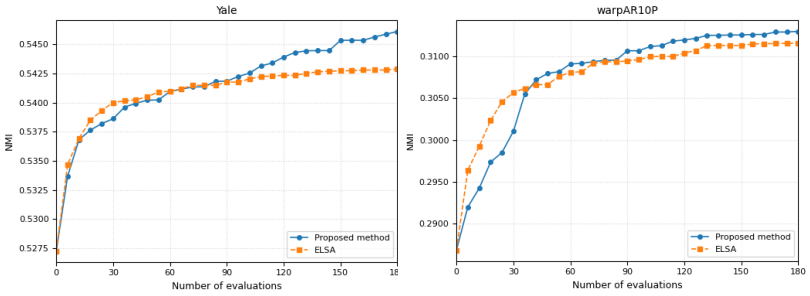
Datasets	ELSA	FBPSO	UFSACO	Proposed
Glioma	0.6056	0.5849	0.5511	**0.6078**
Prostate GE	0.5718	0.5691	0.5664	**0.5733**
TOX 171	0.4650	0.4513	0.4260	**0.4682**
Yale	0.4973	0.4728	0.3842	**0.5007**
warpAR10P	0.3180	0.2907	0.2813	**0.3218**

Figs. 2 and 3 illustrate the convergence processes of NMI and AC, respectively. The convergence comparison was limited to ELSA, which shares the same population size and fitness function as the proposed method, enabling a fair comparison based on the number of fitness evaluations. The horizontal axis represents the cumulative number of evaluations of newly generated individuals as the generations progress, while the vertical axis indicates the performance of the best individual found by the algorithm up to that evaluation point. By comparing the convergence processes of ELSA and the proposed method, the performance differences across generations can be clearly observed. As the number of evaluations increases, the proposed method consistently converges to higher NMI and AC values than the conventional ELSA across all five datasets. This pattern demonstrates that the local refinement process in the proposed method effectively alleviates the convergence problem of the traditional GA.



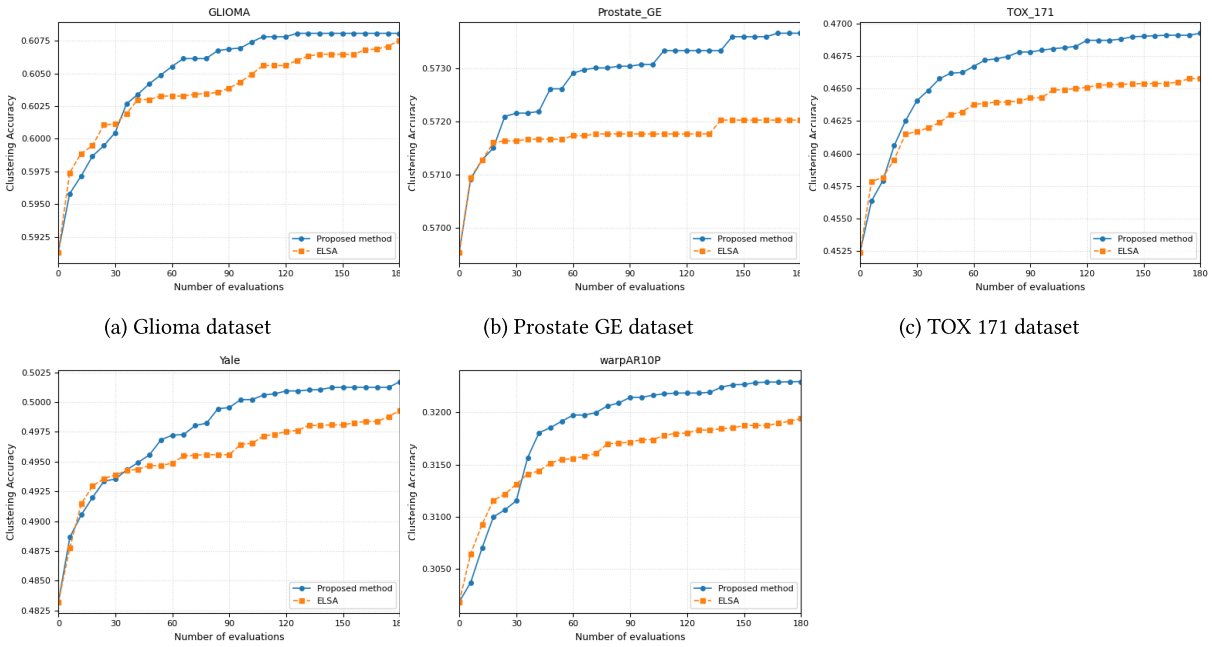
(a) Glioma dataset (b) Prostate GE dataset (c) TOX 171 dataset

Figure 2: (Continued)



(d) Yale dataset (e) warpAR10P dataset

Figure 2: Comparison results of the convergence between the proposed method and ELSA in terms of NMI (higher value indicates better performance).



(a) Glioma dataset (b) Prostate GE dataset (c) TOX 171 dataset

(d) Yale dataset (e) warpAR10P dataset

Figure 3: Comparison results of the convergence between the proposed method and ELSA in terms of AC (higher value indicates better performance).

4.3 Analysis of the Proposed Method

Tables 5 and 6 present the experimental results conducted to examine how sensitive the fitness function of the proposed method is to changes in the weights. Specifically, $\alpha = 1$ and $\gamma = 1$ are fixed, while β is varied from 0.01 to 100 to observe differences in performance. This aims to quantitatively analyze the contribution of the separability metric in selecting features that enhance discrimination between classes. The results show that, for most datasets, the highest NMI and AC values are achieved when $\beta = 1$. This suggests that appropriately considering inter-cluster separability contributes to improving clustering quality. Furthermore, since the performance does not significantly degrade even under extreme conditions where β is very small or very large, it can be confirmed that the proposed method is not highly sensitive to parameter variations.

Table 5: Sensitivity analysis of the penalty weight (β) across five datasets. Results are reported as mean NMI.

β	Glioma	Prostate GE	TOX 171	Yale	warpAR10P
0.01	0.5277	0.0144	0.2309	0.5193	0.2876
0.1	0.5287	0.0146	0.2280	0.5182	0.2891
1	0.5547	0.0155	0.2510	0.5434	0.3118
10	0.5267	0.0146	0.2293	0.5265	0.2891
100	0.5267	0.0145	0.2274	0.5264	0.2891

Table 6: Sensitivity analysis of the penalty weight (β) across five datasets. Results are reported as mean AC.

β	Glioma	Prostate GE	TOX 171	Yale	warpAR10P
0.01	0.5885	0.5707	0.4536	0.4718	0.3046
0.1	0.5836	0.5712	0.4510	0.4730	0.3049
1	0.6078	0.5733	0.4682	0.5007	0.3218
10	0.5855	0.5712	0.4499	0.4774	0.3037
100	0.5865	0.5709	0.4491	0.4764	0.3032

Table 7 presents the total execution time of the proposed method and the comparative algorithm, ELSA. Despite including the preprocessing overhead of constructing an initial joint entropy lookup table, the proposed method demonstrates computational efficiency. Although ELSA, which does not require the table construction process, generally records shorter execution times, it takes longer execution time on some datasets compared to the proposed method. This shows that the vectorized approach based on matrix operations reduces computational complexity in high-dimensional data, thereby enabling the local refinement process of the proposed method to be executed efficiently.

Table 7: Execution time results of ELSA and the proposed method (s).

Datasets	ELSA	Proposed
Glioma	10.16	9.95
Prostate GE	20.28	17.63
TOX 171	20.63	21.00
Yale	18.99	23.02
warpAR10P	18.16	21.27

Tables 8 and 9 compare the performance of the univariate entropy-based search method and the proposed multivariate entropy-based search method in the local refinement process. In particular, for the Yale dataset, NMI increased from 0.5307 to 0.5434 and AC improved from 0.4840 to 0.5007 showing the most significant performance gain. However, since warpAR10P is a high-dimensional image dataset with strong correlations among pixels, multivariate entropy may reflect redundant information, which can result in lower performance compared to univariate entropy. This suggests that multivariate entropy may be less effective in datasets with strong feature redundancy. Nevertheless, these results demonstrate that the proposed method, which considers interactions between features using joint entropy, derives more clustering-optimized feature subsets than the univariate entropy-based approach.

Table 8: NMI comparison of univariate and multivariate entropy based local refinement.

Datasets	Univariate Entropy	Multivariate Entropy
Glioma	0.5478	0.5547
Prostate GE	0.0141	0.0155
TOX 171	0.2474	0.2510
Yale	0.5307	0.5434
warpAR10P	0.3409	0.3118

Table 9: AC comparison of univariate and multivariate entropy based local refinement.

Datasets	Univariate Entropy	Multivariate Entropy
Glioma	0.6027	0.6078
Prostate GE	0.5701	0.5733
TOX 171	0.4636	0.4682
Yale	0.4840	0.5007
warpAR10P	0.3433	0.3218

5 Conclusion

This study proposes a joint entropy-based local search GA that considers interactions among multivariate features to address the convergence problem in UFS. To validate the effectiveness of the proposed method, experiments were conducted on five datasets, comparing its performance with ELSA, FBPSO, and UFSACO. The proposed method outperformed the comparative approaches in both NMI and AC across all datasets. This demonstrates that the joint entropy-based search effectively overcomes the limitations of conventional wrapper methods. Unlike conventional GA, which relies only on probabilistic operators, the proposed method incorporates information-theoretic measures, enabling precise local optimization even in unsupervised environments.

The proposed method achieves strong performance on the gene expression datasets Glioma and Prostate GE, suggesting that it can potentially be applied to disease detection and biomarker identification in clinical settings. Results obtained on the facial image datasets Yale and warpAR10P demonstrate its applicability to high-dimensional visual data processing. Moreover, it can be particularly suitable for high-dimensional data mining tasks that are expensive or impossible to obtain class labels such as customer segmentation, anomaly detection, and pattern recognition. The proposed framework can also be applied to cyber security applications such as network intrusion detection and malicious code classification. This is because high-dimensional, unlabeled data is common in these fields. Sensor data generated in IoT environments can also be utilized for unsupervised anomaly detection, as such data is typically collected without class labels. Future research will focus on integrating deep learning-based feature representation and cluster intelligence methods to further improve search efficiency and scalability.

However, the generation of a joint entropy table for the local refinement process generates preprocessing overhead, which requires more computational time as the dataset becomes larger, and its effectiveness can be limited in the presence of strong correlations between features, such as warpAR10P. Furthermore, since the performance evaluation in this study is based on the results of k-means clustering, the results can also be influenced by the random initialization of the cluster centroid. While the proposed method has been shown to effectively alleviate the convergence problem of the wrapper method, it still has limitations in not ensuring

global optimization. Therefore, in the future, we will consider an adaptive method that automatically adjusts the weights of the fitness functions according to the dataset characteristics. We will develop an evaluation approach that does not rely on clustering algorithms, and expand our experiments to various large datasets to further validate their performance. And we will explore the generalization of the local refinement process so that it can be applied to a wider range of optimization algorithms.

Acknowledgement: None.

Funding Statement: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (RS-2020-NR049579).

Author Contributions: Conceptualization, Hyunki Lim; methodology, Seyeon Son and Hyunki Lim; software, Seyeon Son; validation, Seyeon Son; formal analysis, Seyeon Son and Hyunki Lim; investigation, Seyeon Son and Hyunki Lim; resources, Hyunki Lim; data curation, Seyeon Son; writing—original draft preparation, Seyeon Son and Hyunki Lim; writing—review and editing, Seyeon Son and Hyunki Lim; visualization, Seyeon Son; supervision, Hyunki Lim; project administration, Hyunki Lim; funding acquisition, Hyunki Lim. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Details of the dataset and materials used in this study are provided in the main text.

Ethics Approval: Not applicable.

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

References

1. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF. A review of unsupervised feature selection methods. Artif Intell Rev. 2020;53(2):907–48. doi:10.1007/s10462-019-09682-y.
2. Bashir S, Khattak IU, Khan A, Khan FH, Gani A, Shiraz M. A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches. Complexity. 2022;2022(1):8190814. doi:10.1155/2022/8190814.
3. Sadeghian Z, Akbari E, Nematzadeh H, Motameni H. A review of feature selection methods based on meta-heuristic algorithms. J Exp Theor Artif Intell. 2025;37(1):1–51. doi:10.1080/0952813x.2023.2183267.
4. Alhijawi B, Awajan A. Genetic algorithms: theory, genetic operators, solutions, and applications. Evol Intell. 2024;17(3):1245–56. doi:10.1007/s12065-023-00822-6.
5. Taha ZY, Abdullah AA, Rashid TA. Optimizing feature selection with genetic algorithms: a review of methods and applications. arXiv:2409.14563. 2024.
6. El Aboudi N, Benhlima L. Review on wrapper feature selection approaches. In: Proceedings of 2016 International Conference on Engineering & MIS (ICEMIS); 2016 Sep 22–24; Agadir, Morocco. Piscataway, NJ, USA: IEEE; 2016. p. 1–5.
7. Peng H, Long F, Ding C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell. 2005;27(8):1226–38. doi:10.1109/tpami.2005.159.
8. Ali M, Mazhar T, Al-Rasheed A, Shahzad T, Ghadi Y, Khan M. Enhancing software defect prediction: a framework with improved feature selection and ensemble machine learning. PeerJ Comput Sci. 2024;10(17):e1860. doi:10.7717/peerj-cs.1860.
9. Xue Y, Zhang C. A novel importance-guided particle swarm optimization based on MLP for solving large-scale feature selection problems. Swarm Evol Comput. 2024;91(2):101760. doi:10.1016/j.swevo.2024.101760.
10. Bouchlaghem Y, Akhiat Y, Amjad S. Feature selection: a review and comparative study. E3S Web Conf. 2022;351(1):01046. doi:10.1051/e3sconf/202235101046.

11. Zuo X, Zhang W, Wang X, Dang L, Qiao B, Wang Y. Unsupervised feature selection via maximum relevance and minimum global redundancy. Pattern Recognit. 2025;164(3):111483. doi:10.1016/j.patcog.2025.111483.
12. Ming H, Heyong W. Filter feature selection methods for text classification: a review. Multimed Tools Appl. 2024;83(1):2053–91. doi:10.1007/s11042-023-15675-5.
13. Mitra P, Murthy C, Pal S. Unsupervised feature selection using feature similarity. IEEE Trans Pattern Anal Mach Intell. 2002;24(3):301–12. doi:10.1109/34.990133.
14. Talavera L. Dependency-based feature selection for clustering symbolic data. Intell Data Anal. 2000;4(1):19–28. doi:10.3233/ida-2000-4103.
15. Dash M, Choi K, Scheuermann P, Liu H. Feature selection for clustering—a filter solution. In: Proceedings of 2002 IEEE International Conference on Data Mining; 2002 Dec 9–12; Maebashi City, Japan. Piscataway, NJ, USA: IEEE; 2002. p. 115–22.
16. Liu R, Yang N, Ding X, Ma L. An unsupervised feature selection algorithm: laplacian score combined with distance-based entropy measure. In: Proceedings of 2009 Third International Symposium on Intelligent Information Technology Application; 2009 Nov 21–22; Nanchang, China. New York, NY, USA: IEEE; 2009. p. 65–8.
17. Liu Z, Yang J, Wang L, Chang Y. A novel relation aware wrapper method for feature selection. Pattern Recognit. 2023;140(1):109566. doi:10.1016/j.patcog.2023.109566.
18. Njoku U, Abello A, Bilalli B, Bontempi G. Wrapper methods for multi-objective feature selection. In: Proceedings of 26th International Conference on Extending Database Technology (EDBT 2023); 2023 Mar 28–31; Ioannina, Greece. p. 697–709.
19. Li G, Yu Z, Yang K, Lin M, Chen CLP. Exploring feature selection with limited labels: a comprehensive survey of semi-supervised and unsupervised approaches. IEEE Trans Knowl Data Eng. 2024;36(11):6124–44.
20. Gen M, Lin L. Genetic algorithms and their applications. In: Pham H, editor. Springer handbook of engineering statistics. London, UK: Springer; 2023. p. 635–74. doi:10.1007/978-1-4471-7503-2_33.
21. Naaman D, Ahmed B, Mahmood I. Optimization by nature: a review of genetic algorithm techniques. Indones J Comput Sci. 2025;14(1):268–84.
22. Altarabichi MG, Nowaczyk S, Pashami S, Mashhadi PS. Fast genetic algorithm for feature selection—a qualitative approximation approach. Expert Syst Appl. 2023;211(3):118528. doi:10.1016/j.eswa.2022.118528.
23. Kim Y, Street WN, Menczer F. Feature selection in unsupervised learning via evolutionary search. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2000 Aug 20–23; Boston, MA, USA. New York, NY, USA: ACM; 2000. p. 365–9.
24. Tabakhi S, Moradi P, Akhlaghian F. An unsupervised feature selection algorithm based on ant colony optimization. Eng Appl Artif Intell. 2014;32:112–23. doi:10.1016/j.engappai.2014.03.007.
25. Zhang Y, Hai-Gang L, Wang Q, Chao P. A filter-based bare-bone particle swarm optimization algorithm for unsupervised feature selection. Appl Intell. 2019;49(8):2889–98. doi:10.1007/s10489-019-01420-9.
26. Leardi R, Boggia R, Terrile M. Genetic algorithms as a strategy for feature selection. J Chemometr. 1992;6(5):267–81.
27. Lee J, Kim DW. Mutual information-based multi-label feature selection using interaction information. Expert Syst Appl. 2015;42(4):2013–25. doi:10.1016/j.eswa.2014.09.063.
28. Han TS. Nonnegative entropy measures of multivariate symmetric correlations. Inf Control. 1978;36(2):133–56. doi:10.1016/s0019-9958(78)90275-9.
29. Seo W, Kim DW, Lee J. Generalized information-theoretic criterion for multi-label feature selection. IEEE Access. 2019;7:122854–63. doi:10.1109/access.2019.2927400.
30. Seo W, Lee J. Unsupervised feature selection towards pattern discrimination power. In: Kiyavash N, Mooij JM, editors. Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence; 2024 Jul 15–19; Barcelona, Spain. Cambridge, MA, USA: PMLR; 2024. p. 3180–97.
31. Jerdee M, Kirkley A, Newman M. Normalized mutual information is a biased measure for classification and community detection. Nat Commun. 2025;16(1):11268. doi:10.1038/s41467-025-66150-8.
32. Agrawal U, Rohatgi V, Katarya R. Normalized mutual information-based equilibrium optimizer with chaotic maps for wrapper-filter feature selection. Expert Syst Appl. 2022;207(1):118107. doi:10.1016/j.eswa.2022.118107.

33. Suraya S, Sholeh M, Lestari U. Evaluation of data clustering accuracy using K-means algorithm. Int J Multidiscip Approach Res Sci. 2023;2(1):385–96. doi:10.59653/ijmars.v2i01.504.
34. Xue D, Pang SY, Liu N, Liu SK, Zheng WM. Phase-angle-encoded snake optimization algorithm for K-means clustering. Electronics. 2024;13(21):4215. doi:10.3390/electronics13214215.
35. Liu W, Ning Q, Liu G, Wang H, Zhu Y, Zhong M. Unsupervised feature selection algorithm based on $L_{2,p}$-norm feature reconstruction. PLoS One. 2025;20(3):1–25. doi:10.1371/journal.pone.0318431.
36. Perera K, Chan J, Karunasekera S. Group based unsupervised feature selection. In: Lauw HW, Wong RCW, Ntoulas A, Lim EP, Ng SK, Pan SJ, editors. Advances in knowledge discovery and data mining. Cham, Switzerland: Springer International Publishing; 2020. p. 805–17.
37. Li W, Chen H, Li T, Wan J, Sang B. Unsupervised feature selection via self-paced learning and low-redundant regularization. Knowl Based Syst. 2022;240(11):108150. doi:10.1016/j.knosys.2022.108150.
38. Lin Z, Needell D. Kernel alignment for unsupervised feature selection via matrix factorization. arXiv:2403.14688. 2024.
39. Chin L, Meyerson M, Aldape K, Bigner D, Mikkelsen T, VandenBerg S, et al. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. Nature. 2008;455(7216):1061–8. doi:10.1016/s0513-5117(09)79089-1.
40. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, et al. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell. 2002;1(2):203–9. doi:10.1016/s1535-6108(02)00030-2.
41. Li J, Cheng K, Wang S, Morstatter F, Trevino RP, Tang J, et al. Feature selection: a data perspective. ACM Comput Surv. 2017;50(6):94. doi:10.1145/3136625.