Shahid Bahonar
University of Kerman

Journal of Mahani Mathematical Research

Print ISSN: 2251-7952 Online ISSN: 2645-4505

ENSEMBLE OF SEMI-SUPERVISED FEATURE SELECTION ALGORITHMS TO REINFORCE HEURISTIC FUNCTION IN ANT COLONY OPTIMIZATION

F. Karimi, M.B. Dowlatshahi ✉, and A. Hashemi

Article type: Research Article

(Received: 24 April 2024, Received in revised form 31 July 2024)

(Accepted: 27 September 2024, Published Online: 12 October 2024)

Abstract. Feature selection (FS) is a well-known dimensionality reduction method that chooses a hopeful subset of the original feature collection to diminish the influence the curse of dimensionality phenomenon. FS improves learning performance by removing irrelevant and redundant features. The significance of semi-supervised learning becomes obvious when labeled instances are not always accessible; however, labeling such data may be costly or time-consuming. Many of the samples in semi-supervised learning are unlabeled. Semi-supervised FS techniques overcome this problem, simultaneously utilizing information from labeled and unlabeled data. This article presents a new semi-supervised FS method called ESACO. ESACO uses a combination of ACO algorithm and a set of heuristics to select the best features. Ant colony optimization algorithm (ACO) is a metaheuristic method for solving optimization problems. Heuristic selection is a significant part of the ACO algorithm that can influence the movements of ants. Utilizing numerous heuristics rather than a single one can improve the performance of the ACO algorithm. However, using multiple heuristics investigates other aspects to attain optimal and better solutions in ACO and provides us with more information. Thus, in the ESACO, we have utilized the ensemble of heuristic functions by integrating them into Multi-Criteria Decision-Making (MCDM) procedure. So far, the utilization of multiple heuristics in ACO has not been studied in semi-supervised FS. We have compared the performance of the ESACO using the KNN classifier with variant experiments with eight semi-supervised FS techniques and 15 datasets. Considering the obtained results, the efficiency of the presented method is significantly better than the competing methods. The article's code link on GitHub can also be found at the following: https://github.com/frshkara/ESACO.

Keywords: Ant Colony Optimization; Ensemble of Heuristics; Semi-Supervised Learning; Ensemble Feature Selection; Multi-Criteria Decision-Making.
2020 MSC: 68T20

✉ dowlatshahi.mb@lu.ac.ir, ORCID: 0000-0002-4862-0626

https://doi.org/10.22103/jmmr.2024.23194.1607
Publisher: Shahid Bahonar University of Kerman
How to cite: F. Karimi, M.B. Dowlatshahi, A. Hashemi, *Ensemble of semi-supervised feature selection algorithms to reinforce heuristic function in ant colony optimization*, J. Mahani Math. Res. 2025; 14(1): 283-326.

1. Introduction

Due to the increase in the number of dimensions in massive data, a phenomenon called the curse of dimensionality arises, which causes problems for data mining and machine learning algorithms. The curse of dimensionality causes the overfitting of algorithms, decreases the accuracy of learning models, and increases the learning time and computation complexity. The principal challenge in machine learning and data mining is extracting knowledge from this large quantity of data (Hashemi, et al., 2020; Hashemi, et al., 2022).

Generally, data does not always contain beneficial features, and many of them are irrelevant and redundant. The redundant features are almost a combination of other features and do not provide new information. Also, the irrelevant features do not influence achieving the result. Due to the presence of these features in data, learning algorithms are time-consuming and suffer from a decrease in learning accuracy. Feature selection is one of the most efficient approaches that chooses a subset of the original features and deletes redundant and unrelated features to improve learning efficiency (Cai et al. 2018; Chandrashekar & Sahin 2014; Dey 2024b, 2024a; Hashemi et al., 2023; Hira & Gillies 2015; Khalid et al., 2014; Miao & Niu 2016; Miri et al., 2022a; Venkatesh & Anuradha 2019).

Based on interaction with the learning algorithms, FS techniques can be categorized into three types: wrapper, embedded, and filter (Hashemi et al., 2021; Miri et al., 2022b). Filter techniques are independent of learning algorithms and assess the features according to statistical criteria (Bayati et al., 2022). In the wrapper techniques, a classification algorithm is used to evaluate the possible feature subsets. Embedded methods learn a model once and then assess the features based on the learned model, similar to filter techniques (Chandrashekar & Sahin 2014; Hashemi & Dowlatshahi 2023; Miao & Niu 2016; Venkatesh & Anuradha 2019; Xue et al. 2016).

Considering the interaction with the class label, FS techniques are categorized into three classes: unsupervised, supervised, and semi-supervised (Hashemi, Joodaki, et al. 2022; Hashemi, Dowlatshahi, and Nezamabadi-pour 2021; Karimi, Dowlatshahi, and Hashemi 2023). The features are evaluated based on the level of interaction with the class label in the supervised FS methods (Hashemi et al. 2021). We have no information about the class label in unsupervised FS approaches. Only a tiny portion of the data is labeled in semi-supervised techniques (Beiranvand et al., Dowlatshahi 2022; Dowlatshahi & Hashemi 2023; van Engelen & Hoos 2020; Sheikhpour et al. 2017). Fig. 1 demonstrates the general categorization FS techniques.

FS can be considered an optimization process for selecting a subset of optimal features. Hybrid optimization problems are NP-hard since deterministic polynomial computation cannot solve them in a limited time. Therefore,

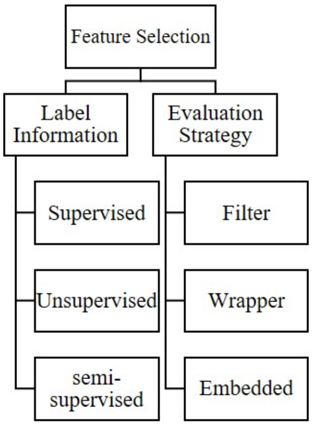


Figure 1. the general categorization of FS classification

solving complex problems by using approximate techniques is necessary. Metaheuristic methods can solve many optimization problems and quickly attain near-optimal solutions in a reasonable time (Mazyavkina et al., 2021).

ACO is one of the famous metaheuristic methods with excellent performance on FS problems. ACO focuses on the inherent behavior of real ants in nature and is a population-based metaheuristic technique (Sheikhpour et al., 2018). Random search is the main procedure of the ACO algorithm. The principal part of the ACO algorithm, which is used for the possible sampling of search space, is a chemical called pheromone (Jia et al., 2020). The ants collaborate in ACO to discover the optimal route. Every ant deposits several pheromones along the track while accidentally moving from available paths and updates the pheromone in its route. Finally, the route with more pheromones is selected as the optimal path (Ma et al., 2011; Sugiyama et al., 2010).

On the other hand, ensemble learning combines the results of several machine learning techniques to achieve better performance rather than using one alone (Miri et al., 2022). Generally, utilizing ensemble methods is better than using one technique alone. Thus, this paper uses an ensemble of heuristics in ACO to enhance the accuracy of semi-supervised FS.

Initial Research Questions: 1. Can using multiple heuristics in the ACO algorithm improve its accuracy of semi-supervised feature selection? 2. Does combining non-linear and linear heuristics in the ACO algorithm for semi-supervised learning enhance its accuracy? 3. Does integrating ACO heuristics or ensemble heuristics) with an MCDM algorithm improve the accuracy of semi-supervised feature selection? 4. Does modeling the integration of ACO heuristics within an MCDM algorithm, represented as a graph, increase the accuracy of semi-supervised feature selection?

This paper presents a new semi-supervised FS algorithm that uses an ensemble of heuristics in ACO for choosing practical features. Based on our knowledge, no semi-supervised FS method has utilized an ensemble of heuristics in ACO. The proposed ESACO uses MCDM modeling for combining different heuristics. MCDM makes the decision-making process more transparent and efficient by analyzing and comparing various options. For the aggregation of heuristics, we have used various FS techniques. Therefore, the features as alternatives and the heuristics (FS techniques) are assumed as criteria in this modeling. The principal difference between the presented algorithm and other methods in the paper is that the features are not assessed according to one heuristic function. In other words, considering that none of the proposed semi-supervised feature selection techniques have used ensemble heuristics in the ACO algorithm, we decided to use this idea for our proposed algorithm. Also, to increase the efficiency of our proposed method, we have integrated heuristics into the MCDM algorithm.

The principal properties of the proposed method are as below:

- A semi-supervised FS technique has been presented according to the ensemble of heuristic functions in ACO for the first time.
- The proposed algorithm uses both linear and non-linear approaches within its heuristics, allowing for the effective utilization of information from both labeled and unlabeled samples.
- The proposed algorithm integrates ACO heuristics within the MCDM framework, leading to high accuracy in semi-supervised feature selection.
- The presented algorithm is competitive with the current semi-supervised FS approaches and performs in a reasonable run-time.
- The results demonstrate that the proposed method is more efficient than competing techniques when 20 and 40 percent of the training instances are labeled.

To demonstrate the efficiency and optimality of our method, we benchmarked it against eight existing semi-supervised FS strategies using a collection of 15 real-world datasets. According to advanced techniques, the results show the excellence of the presented algorithm over the other competing techniques. The article's following sections are presented: Section 2 represents the related methods. Section 3 presents the technique's elemental notions. Section 4 explains the characteristics of the presented algorithm. Sections 5 and 6 describe the experimental setup and the obtained results, respectively. Finally, a conclusion is proposed in section 7.

2. Related Works

In this section, some semi-supervised FS approaches are reviewed.

Supervised FS methods perform weakly due to overfitting when only a small count of labeled instances is accessible. SELF is a semi-supervised FS technique suggested by Sugiyama et al. (Sugiyama et al., 2010) that retains the global structure of unlabeled samples to detach labeled instances in variant handles from others. Semi-supervised Fisher's local discriminant analysis (SELF), is computed according to a particular analysis and includes an analytic construction of the globally optimal solution.

The sudden increase in digital pictures requires efficient methods to manage these pictures. Ma et al. (Ma et al., 2011) presented a method to exploit unlabeled and labeled data to learn manifold structure and choosing the most effective features using sparse models. This structure can simultaneously learn a robust classifier for image annotation by selecting distinct features related to semantic concepts.

CLS, which stands for Constrained Laplacian Score, is a presented technique for semi-supervised FS. Since CLS utilizes unlabeled and labeled samples to select the most relevant features, the critical problem is to obtain information from labeled instances (described by pairwise constraints). The presence of noise in constraints has been shown to hinder the efficiency of the learning process. Hindawi et al. (Benabdeslem & Hindawi, 2011) presented a random subspace method according to the ensemble of data resampling (bagging). This technique analyzes multiple perspectives within the unlabeled and labeled data to generate a global ranking of features. It achieves this by combining multiple constraint Laplacian scores.

For regression problems, Sheikhpour et al. (Sheikhpour et al., 2018) suggested a repetitive construction founded on mixed non-convex and convex $l_{2,p}$ $(0 < p \leq 1)$ regularization and graph Laplacian. This method uses a special matrix (a semi-supervised graph Laplacian-based scatter matrix) to encode the local structure of both unlabeled and labeled samples.

NMF stands for non-negative matrix factorization. Jia et al. (Jia et al., 2020), utilized an NMF construction to find the best subset of features by modeling the label information. The proposed technique to improve grouping performance allows the production of low-dimensional displays. Specifically, some complementary regularizers and similarities are included in traditional NMF to guide decomposition. Additionally, it restricts the dissimilarity and similarity of the low-dimensional presentations of unlabeled and labeled samples.

If we have binary labeled and unlabeled instances, labeling all unlabeled data points as positive or negative based on certain criteria. In this procedure, practical and academic studies present powerful results for FS through feature ranking and hypothesis testing. Sechidis et al. (Sechidis and Brown, 2018) obtained two unique strategies (Semi-IAMB and Semi-JMI), which have been shown to significantly outperform competing methods when combined with prior "soft" domain knowledge. They perform well if the class label is absent, but not by accident.

Liu et al. (Liu et al., 2010) suggested a method that analyzes the dispensation of unlabeled and labeled data with a particular label propagation technique. Then, to optimize the tracking proportion criterion, an efficient method directly identifies the ideal set of features. This technique utilizes label propagation to assign soft labels to unlabeled samples by analyzing the distribution of both labeled and unlabeled data. Then, to choose the optimal feature subset the ratio optimization measure is utilized.

Table 1 shows a summary of related methods that have been compared with the proposed method.

Following are the recently proposed metaheuristic algorithms that we will try to use in the future. The Binary Waterwheel Plant Optimization Algorithm is a novel metaheuristic algorithm inspired by the foraging behavior of the waterwheel plant. It's designed for FS problems, particularly those involving high-dimensional and complex datasets (Anon n.d.). The Weighted Superposition Attraction Optimization Algorithmis a novel metaheuristic algorithm for FS in machine learning, inspired by the concept of particle attraction and superposition in quantum mechanics. It's particularly well-suited for addressing high-dimensional and complex FS problems Ganesh et al., 2023). Asghari Varzaneh Liu et al. (Asghari Varzaneh et al., 2022) suggested a new two-step FS method for high-dimensional data. First, it uses mRMR to score features and select the most relevant ones. Then, it applies an improved version of the Equilibrium Optimizer (IMEO) algorithm to further refine the FS and avoid getting stuck in suboptimal solutions. This approach combines filter-based (mRMR) and wrapper-based (IMEO) techniques for effective FS (Ganesh et al. 2023).

3. Basic Conceptions

3.1. **Ant Colony Optimization (ACO).** ACO algorithm was recommended by Dorigo et al. (Dorigo, Birattari, and Stutzle 2006). ACO according to the innate behavior of real ants, is a random search and swarm intelligence metaheuristic method and applicable to solving optimization problems. The main section of the ACO algorithm is pheromone that utilizes the search space for possible sampling. The ant deposits a continuous trail of pheromone as they travel from the nest to the food source. The value of the pheromone, to prevent ants from being trapped in the local optimum, evaporates over time. Based on this fact, when more ants travel along a path, and the pheromone value is more intense the probability of choosing an optimal solution is higher. (Dorigo and Blum 2005; Monteiro, Fontes, and Fontes 2012; Paniri, Dowlatshahi, and Nezamabadi-pour 2020, 2021).

The ACO algorithm, in addition to the attributes of real ants, adds other abilities to artificial ants. They have an internal state that maintains the history of their former activities and evaluates the excellence of the solution generated by each ant. They update the pheromone trail locally and globally, to gain better solutions. Also, the ACO algorithm can use some heuristic information to perform the search process (Dorigo & Blum 2005; Monteiro et al. 2012).

Table 1. Summery of Related Works.

Algorithm	Method	Technique	Advantages	Disadvantages
TRCFS	Filter	scatter matrix trace ratio criterion	Handles limited labeled data, incorporates both labeled and unlabeled data,	higher computational complexity
SELF	Filter	Projects data to lower dimensions, preserving local structure and class separation.	Effective dimensionality reduction, handles limited labeled data, preserves local structure	Sensitive to outliers, higher computational complexity
ENCLS	Filter	Ensemble of Laplacian Scores	Handles limited labeled data, incorporates graph-based information	Tuning Parameters, Increased complexity Limited validation
GSSNMF	Filter	Graph Laplaciane, Incorporating label information through an auxiliary loss term	Effective dimensionality reduction, preserves local structure and discriminative information	Sensitive to outliers, higher computational complexity
SFSGL	Filter	Graph Laplaciane, mixed convex and non-convex l_2 , $p - norm$ regularization	incorporates graph-based information, promotes sparsity and structure preservation	Parameter tuning, Computational complexity Limited validation
SFSS	Filter	multi-layer structure joint l_2, 1_norm minimization	Effectively for regression problems, enhances robustness, incorporates structural information.	Computational complexity, Limited validation Tuning parameters
Semi-JMI	Filter	conditional mutual information (JMI)	effectively rank features with positive labels	Limited Information Use, Time Complexity
Semi-MIM	Filter	Based on the IAMB (Incremental Association Markov Blanket)	effectively find a set of relevant feature with positive and negative labels	Require a large amount of data

The general procedure of the ACO algorithm, after initializing the parameters, is constructed as follows: In each iteration, until the stopping criteria are satisfied, the ants formulate solutions using the specific pheromone formulation. After the ants complete their solution, the pheromone trail evaporates uniformly at all points. Each ant updates the pheromone vector according to its solutions (Dorigo & Blum, 2005; Monteiro et al., 2012).

To represent the ACO-based FS problem, the problem must be formulated suitably.The nodes illustrate features, and the edges represent relationships among them. Until the stopping criterion is met, Optimal search is when ants visit the minimum number of nodes in the graph. For instance, attaining high accuracy with fewer features than the initial features set can be the stopping metric (Kanan, Faez, and Taheri 2007).

Suppose we have a graph G, with a set of features $F = \{f_1, f_2, \ldots, f_m\}$ as graph nodes. Initially, m ants are move on the n features of the problem space with some initial pheromone values. Each ant on graph G follows a stochastic greedy rule for its movements. Thus, While ants terminate their trips, the global updating rule updates the pheromone values of features (Hashemi, Joodaki, et al. 2022; Paniri et al. 2020).

In a probabilistic approach, ant k uses the probabilistic action selection rule based on the equation below to transmit from featurei to feature j (Paniri et al. 2020):

$$(1) \qquad p_{ij}^{k}(t) = \frac{[\tau_i(t)]^{\alpha}[\eta_{(F_i,F_j)}]^{\beta}}{\sum_{u \in N_i}[\tau_u]^{\alpha}[\eta_{(F_i,F_u)}]^{\beta}} \quad \forall j \in N_i^k, \text{ if } q > q_0$$

Where $p_{ij}^{k}(t)$ is a transition possibility to move from feature i to feature j at time t for k-th ant, τ_i is the allocated pheromone value to feature i, $\eta(F_i, F_j)$ is value of the heuristic information between features (F_i, F_j). N_i^k s the set of neighbor features for feature i, q is a random variable that is uniformly dispensed in $[0, 1]$, and q_0 is a constant variable $(0 \leq q_0 \leq 1)$. The value of the parameter β is between $[0, 1]$, Setting the pheromone value and the level significance of heuristic information. If $\beta = \mathit{0}$, the heuristic information is neglected and the decision is taken based on the previous action history. Greedily, every ant visits the next feature based on the equation below (Paniri et al., 2020):

$$(2) \qquad i = \arg\max_{u \in N^k} \left\{[\tau_u][\eta(F_i, F_u)]^{\beta} \mid q \leq q_0\right\}$$

The interchange between the exploration and exploitation abilities of the algorithm is balanced with the parameters q and q_0. Whenever an ant chooses its next feature (from feature i), it creates a random number q. Every feature can be selected proportional to its probability (Exploration) when $q > q_0$. utilizing Eq.1. Otherwise, by Eq.2 (Exploitation), the most beneficial feature is selected to use. The combination of Eq.1 and Eq.2, named the "pseudo-random proportional" law is The state transition rule. The pheromone routes are updated by every ant separately after ants travel some features on the graph. based on the equation below The global updating rule updates the pheromone values (Paniri et al., 2020):

$$\tau_i(t+1) = (1-\rho)\tau_i(t) + \Delta\tau_i \tag{3}$$

The pheromone values of feature i at times t and t+1, individually, are $\tau_i(t+1)$, $\tau_i(t)$, and the pheromone decay rate parameter is ρ. According to some criteria, $\varDelta\tau_i$ determines the pheromone enhancement value for feature i.

Eventually, a feature with a high pheromone value is a beneficial feature. Thus, the optimal feature subset is determined, based on the value of pheromones, by choosing the m $(m(m > d)$ top features specified.

3.2. **MCDM and MOORA Algorithm.** MCDM stands for Multi-Criteria Decision Making, which provides potent decision-making in domains where selecting the best option is complex. MCDM is the strategy to find the best solution among various options according to multiple criteria. In decision-making problems, no option is often better than others in all criteria. Thus, the decision-makers generally endeavor to discover an acceptable solution (Aruldoss 2013; Hashemi et al. 2022).

There are several methods for solving MCDM problems. One of the most famous is the MOORA method. MOORA stands for Multi-Objective Optimization based on Ratio Analysis, was presented by Zavadskas (Karel et al., 2006), which compounds the reference point and the ratio system procedure (Hashemi, Joodaki, et al., 2022). We must initialize a weight vector and a decision matrix to perform an MCDM process. Therefore, the decision matrix (X) in MOORA is constructed based on the rating X_{ij} of m options according to n criteria, as follows:

$$\mathbf{X} = \begin{pmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \dots & \mathbf{X}_{1n} \\ \mathbf{X}_{21} & \mathbf{X}_{22} & \dots & \mathbf{X}_{2n} \\ \dots & \dots & \dots & \dots \\ \mathbf{X}_{m1} & \mathbf{X}_{m2} & \dots & \mathbf{X}_{mn} \end{pmatrix} \tag{4}$$

Sometimes, in an MCDM, every criterion has a variant grade of an impression on the results. Therefore, the weights of the criteria are represented by the weight vector, which is described below:

$$\mathbf{W} = \big(w_1, w_2, \dots, w_n\big) \tag{5}$$

Since the ranking of options for every metric may be in different ranges, this matrix is generally normalized first before giving matrix X to the MCDM algorithm. Generally, in MOORA the vector normalization technique is utilized, the formula of which is displayed below:

$$X^*_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m}(x_{ij})^2}} \tag{6}$$

where i and j, respectively, are the criterion and alternative numbers. The normalization matrix is constructed as below:

$$\mathbf{X}^* = \begin{pmatrix} \mathbf{X}^*_{11} & \mathbf{X}^*_{12} & \dots & \mathbf{X}^*_{1n} \\ \mathbf{X}^*_{21} & \mathbf{X}^*_{22} & \dots & \mathbf{X}^*_{2n} \\ \dots & \dots & \dots & \dots \\ \mathbf{X}^*_{m1} & \mathbf{X}^*_{m2} & \dots & \mathbf{X}^*_{mn} \end{pmatrix} \tag{7}$$

Not all factors in MCDM are equally important. So, we assign different weights to each factor using a normalization matrix. The MOORA algorithm for calculate the utility of each alternative utilizes the arithmetic weighted aggregation operator. Therefore, the weighted sum of beneficial criteria is subtracted from the non-beneficial criteria as below:

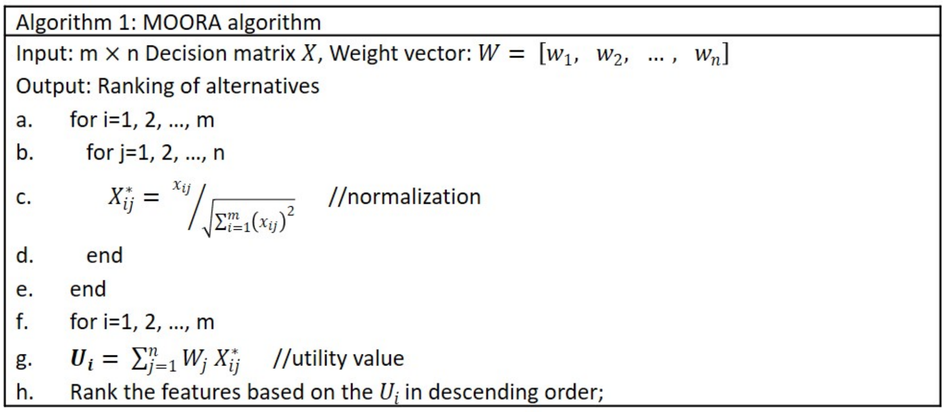
$$U_i = \sum_{j=1}^{g} W_j X^*_{ij} - \sum_{j=g+1}^{n} W_j X^*_{ij} \tag{8}$$

where (n - g) and g show the numbers of non-beneficial and beneficial criteria, respectively. All metrics used in this article are beneficial. Therefore, the profit of the i-th option is computed by the equation below:

$$U_i = \sum_{j=1}^{n} W_j X^*_{ij} \tag{9}$$

The utility vector, denoted by , represents the utility scores of each alternative (i-th option). The option that maximizes utility is considered the best alternative. A detailed explanation of the MOORA algorithm, broken down into individual steps, can be found in Fig. 2.

The description of the MOORA algorithm is as follows: Lines 1-3: a decision matrix (X) with dimensions m * n and a weighted matrix W in one dimension with size n are given as input to the algorithm. The decision matrix is a table that contains the scores of each alternative (option) being considered on each criterion (factor). The weight vector assigns a weight to each criterion, indicating its relative importance. Lines 5-9: This loop iterates over each row (i) in the decision matrix. Lines 6-8: This section is commented out and appears to be incomplete. It likely intended to normalize the decision matrix. Normalization transforms the values in the decision matrix into a common



```
Algorithm 1: MOORA algorithm
Input: m × n Decision matrix $X$, Weight vector: $W = [w_1, w_2, \ldots, w_n]$
Output: Ranking of alternatives
a.  for i=1, 2, …, m
b.    for j=1, 2, …, n
c.      $X_{ij}^{*} = {x_{ij}} \Big/ \sqrt{\sum_{i=1}^{m} (x_{ij})^2}$      //normalization
d.    end
e.  end
f.  for i=1, 2, …, m
g.  $\boldsymbol{U_i} = \sum_{j=1}^{n} W_j \, X_{ij}^{*}$     //utility value
h.  Rank the features based on the $U_i$ in descending order;
```

Figure 2. MOORA Algorithm.

scale so that criteria measured in different units can be compared. Lines 10-14: This loop iterates over each column (j) in the decision matrix. Line 11: This line is also commented out. Lines 15-19: This loop iterates over each row (i) in the decision matrix again. Line 16: This line calculates a utility value (U_1) for each alternative (i) by multiplying the weight vector (W) with the corresponding row in the decision matrix (X). Line 17: This line is commented out. It likely intended to rank the alternatives based on the utility values (U_1).

4. **Presented algorithm**

As discussed earlier, incorporating multiple heuristic functions can lead to improved ACO performance relative to a single heuristic function.This section presents a new semi-supervised FS technique that uses an ensemble of heuristics based on integrating the MCDM procedure in the ACO. A powerful combination technique is required to reach more dependable results when some FS techniques are used for ranking features. We use an ACO algorithm with multiple heuristics to identify a consistent order for the features by analyzing the ranking agreement from all feature selection algorithms. In this method, by considering the features as alternatives and FS techniques as criteria, the ensemble of heuristics has been modeled into an MCDM procedure. In other words, we have used multiple FS techniques in ACO as various heuristics to attain the best agreement. By using the MOORA algorithm and based on the results of different FS techniques, a final discovery can be achieved that indicates the importance and ranking of each feature. The general construction of the presented method is demonstrated in Fig. 4. The pseudo code of the presented method is displayed in Fig. 5.

4.1. **Pheromone Initialization.** To create a good solution the principal part of the ACO algorithm is the pheromone trail. In the presented technique, every ant adds some pheromone to the nodes of the graph (features) while passing on them and creates a vector, denoted by τ , with d dimensions. This vector represents the pheromone sequence. In this article, the initial value of the pheromone is set to τ_0. This means the initial values (denoted by τ_0) of the pheromone vector for all links between nodes j and i are initialized with a fixed value of 0.5.

4.2. **Basic FS methods.** ACO algorithms usually use one heuristic function, while using several heuristics instead of one can lead to better results. Thus, the presented method has used an ensemble of heuristic functions instead of one heuristic. In the following, we mention the FS methods used in ensemble learning of the ESACO.

The first FS technique is cosine similarity, which computes redundancy between features. The matrix flcos with dimensions $d * d$ is the redundancy matrix. To calculate the cosine coefficient, we can use this formula: (Lee & Wander 1988):

$$\text{Cosine}(X, Y) = \left| \frac{\sum_{i=1}^{n}(X_i Y_i)}{\sqrt{\sum_{i=1}^{n} X_i^2}\sqrt{\sum_{i=1}^{n} Y_i^2}} \right| \tag{10}$$

where Y and X are two n-dimensional feature vectors, in which n is the count of instances. Since our proposed method is designed for semi-supervised data, to compute the redundancy between features in both labeled and unlabeled instances it uses cosine similarity.

The cosine similarity value is in ranges $[0, 1]$. Two random variables are dependent if the cosine similarity value is nearer to 1, and they are independent if it is closer to 0.

The next FS function is MIC. MIC (Maximal Information Coefficient) computes relevancy between features. A vector named flcorr captures the relevance of features, with a dimension of d. The MIC is computed as below (Zhu et al., 2019):

$$\text{MIC}(X, Y) = \max \left\{ |X||Y| < T, \, \frac{\text{MI}(X, Y)}{\log_2(\min(|X|, |Y|))} \right\} \tag{11}$$

Here, Y and X represent two sets of features, each with n dimensions. Here, $|Y|$ and $|X|$ represent the number of data points falling into each bin on the Y-axis and X-axis, respectively. T represents the upper limit of the grid partition for the joint space of X and Y. The mutual information between X and Y is denoted by $MI(X, Y)$.

The entropy of two variables and one variable can be calculated as below (Zhu et al., 2019):

$$H(X) = -\sum_{i=1}^{n} p(X_i) \log_2(p(X_i)) \tag{12}$$

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} p(X_i, Y_j) \log_2(p(X_i, Y_j)) \tag{13}$$

To compute the relevance between the class label and the features of labeled samples we have used the MIC.

MIC demonstrates the linear and non-linear relevancy among two variable pairs. The range of the MIC value is $[0, 1]$, where 1 indicates the maximum relationship between variables, and 0 denotes independence (Cao et al., 2021). MIC has two great attributes: equality and generality, and to have the most relevancy, it is looking for a smaller subset of the primary feature collection. It can specify various relevancy, including non-functional, functional, non-linear, and linear. Via experimental comparison, MIC indicates good versatility and fairness (Pan, 2021).

Since most instances in semi-supervised learning are unlabeled, ESACO uses an unsupervised FS algorithm to rank the features. The third FS technique is the LLCFS algorithm. Given that the proposed method leverages both labeled and unlabeled data, ESACO has used the LLCFS algorithm to identify information between unlabeled samples. Kernel Learning and FS for Local Learning-Based Clustering (LLCFS) were presented by Hong Zeng (Zeng & Cheung, 2011), which is a suitable data presentation method through kernel learning or FS within the structure of technique LLC (Wu & Schölkopf, 2006). Local Learning-based Clustering, or LLC, is a way to group data points together. In LLC, each point's group is determined by looking at its neighbors. The LLCFC merges the FS into the LLC. It should be mentioned that learning the local regression model, which is trained only with the points in every neighborhood, is the principal component of the LLC. LLCFS involves two main phases (Du and Shen 2015): 1. In the weighted feature space, build the k-nearest neighbor graph 2. Conduct joint clustering and feature weight learning.

In short it can be said: in our proposed method, we employed the cosine similarity to quantify the redundancy among unlabeled features, aiming to select those with the least redundancy. Additionally, we utilized the MIC correlation coefficient to assess the relationship between labeled features, which was, in fact, employed to select those features that exhibit the highest degree of mutual relevance. Also, due to the semi-supervised nature of the proposed method and the high number of unlabeled samples, it was decided to use the LLCFS algorithm, which selects the features that have the least redundancy by adjusting the weighting of the features. Next, the features selected by each algorithm are sorted and the process of selecting suitable features is continued by the MCDM matrix, which is explained below.

Various methods were tested to select best features, and the mentioned methods had the best results among the others.

4.3. **Decision Matrix Construction.** We obtain the ranks assigned by the feature selection algorithms to gain the last rank of features. In the presented method, we investigate the features according to assessing some feature ranking techniques. We assume this process as an MCDM problem, then to gain the final heuristic, MOORA algorithm is then used to analyze the outputs from various FS techniques and identify the most effective approach.

As a first step, the method, to gain the rank matrix (decision matrix), multiple FS methods rank the features as below:

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \dots & \mathbf{R}_{1n} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \dots & \mathbf{R}_{2n} \\ \dots & \dots & \dots & \dots \\ \mathbf{R}_{m1} & \mathbf{R}_{m2} & \dots & \mathbf{R}_{mn} \end{pmatrix} \tag{14}$$

where the rank allocated to m-th feature with n-th filter technique is indicated by R_{mn}.

Since our optimization goal is maximization, and the decision matrix is constructed using feature ranks, normalization becomes necessary. Thus, considering that feature's ranking we determine a score for every feature. The top-ranked feature, identified by a value of 1 in matrix R, is assigned the maximum score, equivalent to the number of features (m). Following the ranking scheme of matrix R, the feature in the second best (value of 2) is assigned a score of m minus one. This process continues as long as all features have been allocated new scores. The normalization decision matrix, is matrix R^*, is built as below:

$$\mathbf{R}^* = \begin{pmatrix} \mathbf{R}^*_{11} & \mathbf{R}^*_{12} & \dots & \mathbf{R}^*_{1n} \\ \mathbf{R}^*_{21} & \mathbf{R}^*_{22} & \dots & \mathbf{R}^*_{2n} \\ \dots & \dots & \dots & \dots \\ \mathbf{R}^*_{m1} & \mathbf{R}^*_{m2} & \dots & \mathbf{R}^*_{mn} \end{pmatrix} \tag{15}$$

Now, for our MCDM procedure, we have our decision matrix in which the columns (FS techniques) are considered as criteria (experts), where each row of R^* (representing a feature) corresponds to an alternative. We have determined an equivalent weight for every criterion $(1/n)$. Therefore, to rank the features according to the matrix R^*, the MOORA technique is utilized. To evaluate the utility (U) of every feature, we have utilized Algorithm 1. The results of using the MOORA algorithm are constrcuted as follows:

$$\mathbf{U} = \big(u_1, u_2, \dots, u_m\big) \tag{16}$$

where m is the feature number.

The performance summary of the MCDM routine can be expressed as follows: A consolidated feature matrix is constructed by combining and ranking feature subsets generated from various selection methods. Weighting is applied to the features within this matrix, and subsequent ranking determines the final feature set. Next, the selected features are ranked by the MOORA algorithm and given to the ACO algorithm as a heuristic matrix.

Considering we have our heuristic vector (U), to discover the optimal features we can use the ACO algorithm. In Eq.(1), we require to initialize some parameters for applying ACO, which are written below:

$$P_i^k(t) = \frac{([\tau_i][\eta_i(U_i)]^\beta)}{\sum_{u \in N_i^k}([\tau_u][\eta_u(U_u)]^\beta)} \tag{17}$$

In Eq.(17), the possibility for ant k to travel to feature i at time t is indicated by $P_i^k(t)$. The pheromone value of feature i is indicated by τ_i, the neighbors of ant k are illustrated by N_i^k, and the heuristic information of the feature is η_i.

As noted earlier, utilizing one heuristic function in the ACO algorithm cannot effectively handle complex problems in the long run. Therefore, $\eta_i(U_i)$ combines information from various heuristics to evaluate the importance of feature i. Now, between pheromone and heuristic information we assume equal importance and the β parameter will be set to 1.

At the end of every iteration, the pheromone matrix, $\boldsymbol{\tau}$, is updated. According to the Eq.(2), the value of pheromones is updated, the pheromone values, $\tau_i(t+1)$ and $\tau_i(t)$, represent the values for feature $i+1$ and i at the current time $t+1$ and t compared to feature i at the same time, respectively. The pheromone decay rate parameter (represented by the symbol ρ) is set to 0.1 in this case. $\Delta\tau_i$ denotes the pheromone enhancement value for feature i. After the ACO algorithm finishes, the features are arranged in descending order based on their final pheromone values.

4.4. **Summary of the ESACO algorithm.** The ESACO algorithm is a ensemble FS algorithm that uses ACO to select a set of effective features for classification. The algorithm uses three FS techniques (LLCFS, MIC, and COS) to compute the weight of each feature, and then uses ACO to select a set of features with maximum classification performance. The steps of the ESACO algorithm are as follows: Step 1: This step involves preprocessing the data to prepare it for the algorithm. The data is divided into two sets: the labeled (X_L) and the unlabeled samples $(X = X_L\ U\ X_U)$. The features are also standardized to have the same scale. Step 2: This step computes redundancy vector between all features in the dataset X using cosine similarity. This matrix captures the degree of similarity between each pair of features, including both labeled and unlabeled samples. Step 3: This step computes the relevancy vector between the class label and the features of the labeled samples using MIC algorithm.

$flCOS$ is used to measure the strength of the relationship between each feature and the class label. $flMIC$ is an information-based FS algorithm that uses the mutual information between features and the class label to select features. Step 4: This step computes the feature weights using $LLCFS$ techniques. $LLCFS$ is a clustering-based FS algorithm that uses the minimum distance between clusters to select features. Step 5: This step constructs the decision matrix which shows the weights of the features for each sample. The weights are computed using a combination of the weights computed by the $LLCFS$, $flMIC$, and $flCOS$ FS algorithms. Step 6: This step initializes the weights for each FS technique to 1/3. Step 7: This step uses ACO to select a set of features with maximum classification performance. The ACO algorithm uses pheromone to guide the ants to select features. Step 8: This step selects the top d features as the final feature set. These features are the most important features for classifying the data.

To illustrate the concept, a portion of the algorithm's calculations can be demonstrated with a small dataset as a numerical example in Fig. 3. Finally, the utility vector is fed into the ACO algorithm, which then selects the best features based on their pheromone values (with the highest values indicating the most favorable features).

5. **Experimental Tuning**

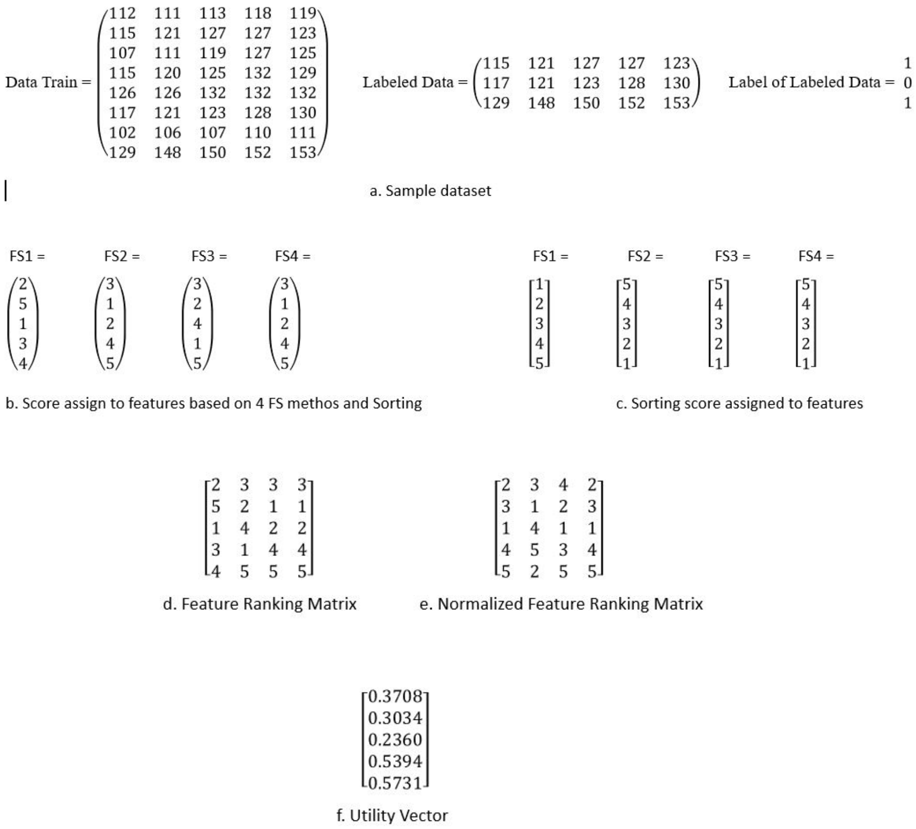
This section determines the datasets, simulation environment, assessment metrics, parameter tuning, and classifiers. The experiments were run on a computer with Windows 10, an Intel Core i7 processor, and 16GB of RAM. We used MATLAB software (version R2021a) for the analysis. . We ran each experiment 30 separate times for every algorithm on each dataset. This repetition (30 runs) helps to achieve statistically significant and consistent results. Using hold-out validation, in each run a split of 70% for the training set and 30% for the test set was used on the dataset instances. Considering the semi-supervised nature of the technique, experiments were performed separately with accidental labeling on 40 % and 20 % of the training instances.

5.1. **Datasets Specification.**

In this part, we contrast the efficiency of the presented method ESACO with some techniques in the literature: SFSS (Ma et al. 2011), SELF (Sugiyama et al. 2010), EnsCLS (Hindawi, Elghazel, & Benabdeslem 2013), SFSGL (Sheikhpour et al. 2018), GSSNMF (Jia et al. 2020), Semi-JMI and Semi-MIM (Sechidis & Brown 2018) and TRCFS (Liu et al. 2010).

5.2. **Classifier Specification.**

To assess the classification performance of the presented technique, we have utilized the K-Nearest Neighbour classifier (KNN) (Liao and Vemuri 2002). The k parameter of the KNN classifier is set to 5. We evaluate the efficiency of the semi-supervised FS techniques using a range of feature numbers from 10 to 100 $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$.



$$\text{Data Train} = \begin{pmatrix} 112 & 111 & 113 & 118 & 119 \\ 115 & 121 & 127 & 127 & 123 \\ 107 & 111 & 119 & 127 & 125 \\ 115 & 120 & 125 & 132 & 129 \\ 126 & 126 & 132 & 132 & 132 \\ 117 & 121 & 123 & 128 & 130 \\ 102 & 106 & 107 & 110 & 111 \\ 129 & 148 & 150 & 152 & 153 \end{pmatrix} \quad \text{Labeled Data} = \begin{pmatrix} 115 & 121 & 127 & 127 & 123 \\ 117 & 121 & 123 & 128 & 130 \\ 129 & 148 & 150 & 152 & 153 \end{pmatrix} \quad \text{Label of Labeled Data} = \begin{matrix} 1 \\ 0 \\ 1 \end{matrix}$$

a. Sample dataset

$$FS1 = \begin{pmatrix} 2 \\ 5 \\ 1 \\ 3 \\ 4 \end{pmatrix} \quad FS2 = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 4 \\ 5 \end{pmatrix} \quad FS3 = \begin{pmatrix} 3 \\ 2 \\ 4 \\ 1 \\ 5 \end{pmatrix} \quad FS4 = \begin{pmatrix} 3 \\ 1 \\ 2 \\ 4 \\ 5 \end{pmatrix}$$

b. Score assign to features based on 4 FS methos and Sorting

$$FS1 = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} \quad FS2 = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \quad FS3 = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix} \quad FS4 = \begin{bmatrix} 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

c. Sorting score assigned to features

$$\begin{bmatrix} 2 & 3 & 3 & 3 \\ 5 & 2 & 1 & 1 \\ 1 & 4 & 2 & 2 \\ 3 & 1 & 4 & 4 \\ 4 & 5 & 5 & 5 \end{bmatrix} \qquad \begin{bmatrix} 2 & 3 & 4 & 2 \\ 3 & 1 & 2 & 3 \\ 1 & 4 & 1 & 1 \\ 4 & 5 & 3 & 4 \\ 5 & 2 & 5 & 5 \end{bmatrix}$$

d. Feature Ranking Matrix

e. Normalized Feature Ranking Matrix

$$\begin{bmatrix} 0.3708 \\ 0.3034 \\ 0.2360 \\ 0.5394 \\ 0.5731 \end{bmatrix}$$

f. Utility Vector

Figure 3. Example of the steps of the ESACO algorithm

5.3. **Performance Evaluation Metrics.** We used F-score and accuracy metrics to evaluate the efficiency of the suggested technique and competitive strategies (Hashemi, Dowlatshahi, et al., 2021). First, follow the following concepts: 1. True Positive (TP): The ratio of positive samples that are correctly categorized. 2. False Positive (FP): The ratio of negative samples that are incorrectly categorized as positive. 3. True Negative (TN): The ratio of negative samples that are correctly categorized. 4. False Negative (FN): The ratio of positive samples that are incorrectly categorized as negative.

F-score and Accuracy metrics are described below:

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \tag{18}$$

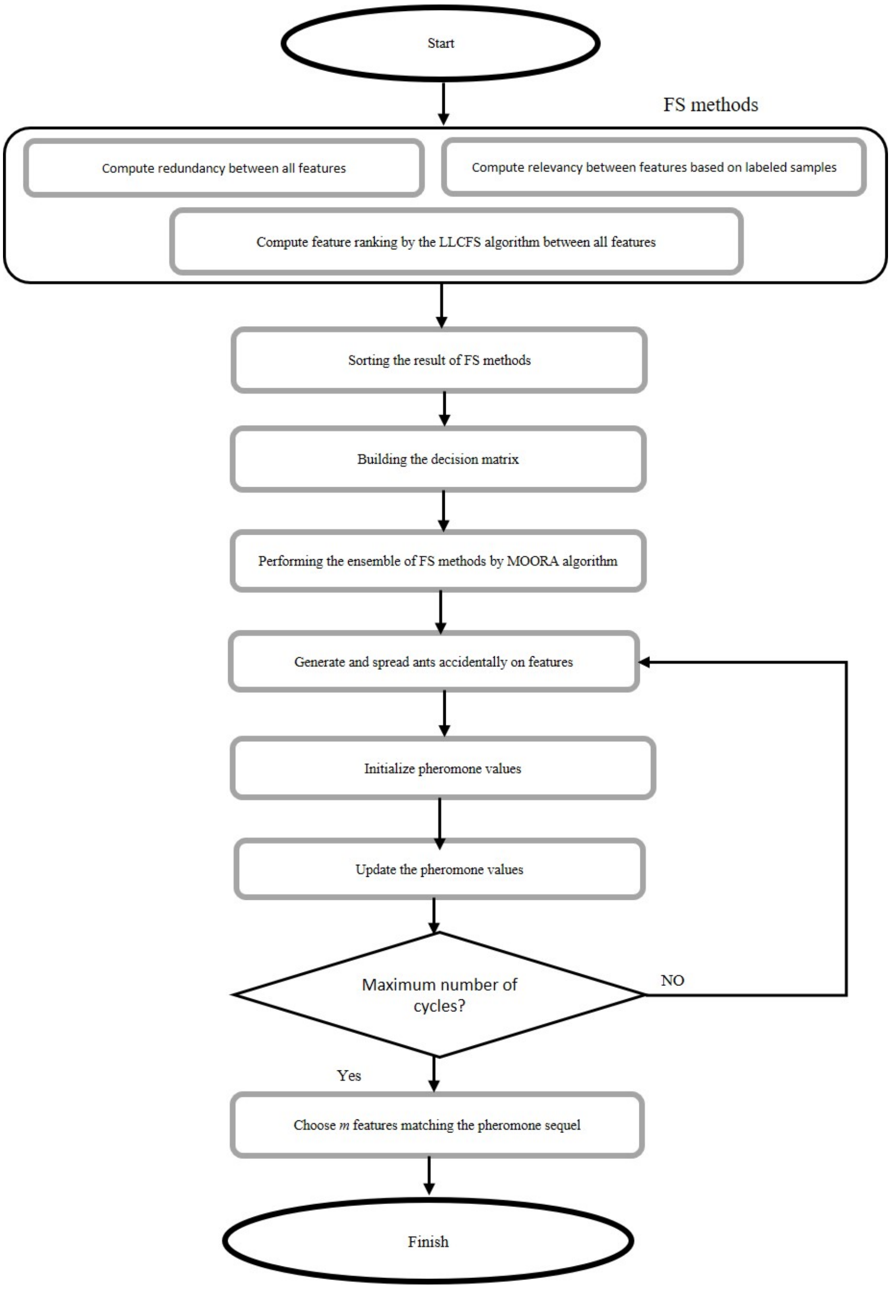
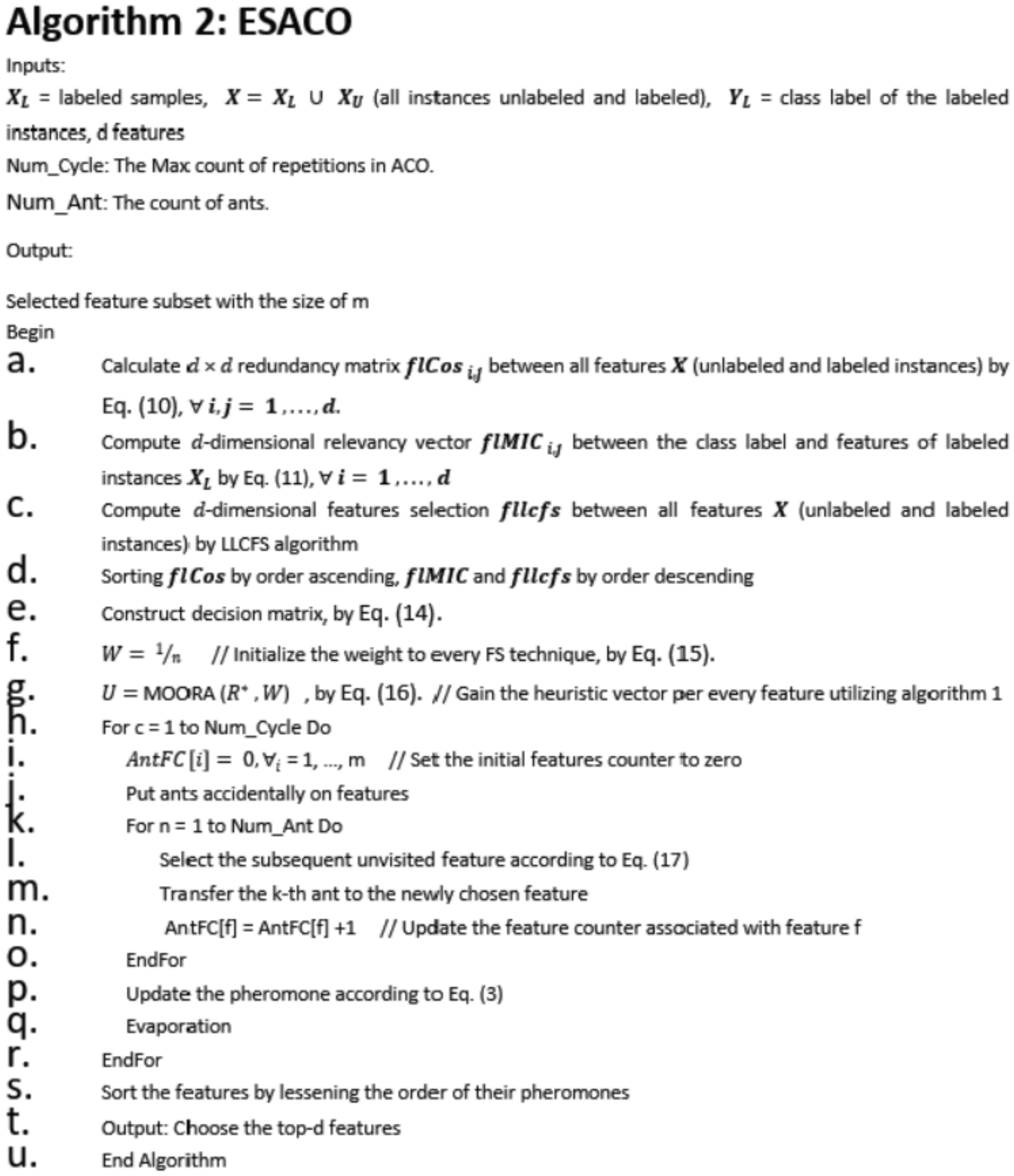

Start
FS methods
Compute redundancy between all features
Compute relevancy between features based on labeled samples
Compute feature ranking by the LLCFS algorithm between all features
Sorting the result of FS methods
Building the decision matrix
Performing the ensemble of FS methods by MOORA algorithm
Generate and spread ants accidentally on features
Initialize pheromone values
Update the pheromone values
Maximum number of cycles?
NO
Yes
Choose *m* features matching the pheromone sequel
Finish

Figure 4. The overall framework of the ESACO



```
Algorithm 2: ESACO
Inputs:
$X_L$ = labeled samples, $X = X_L \cup X_U$ (all instances unlabeled and labeled), $Y_L$ = class label of the labeled instances, d features
Num_Cycle: The Max count of repetitions in ACO.
Num_Ant: The count of ants.
Output:
Selected feature subset with the size of m
Begin
a.  Calculate $d \times d$ redundancy matrix $flCos_{i,j}$ between all features $X$ (unlabeled and labeled instances) by Eq. (10), $\forall i, j = 1, \ldots, d$.
b.  Compute $d$-dimensional relevancy vector $flMIC_{i,j}$ between the class label and features of labeled instances $X_L$ by Eq. (11), $\forall i = 1, \ldots, d$
c.  Compute $d$-dimensional features selection $fllcfs$ between all features $X$ (unlabeled and labeled instances) by LLCFS algorithm
d.  Sorting $flCos$ by order ascending, $flMIC$ and $fllcfs$ by order descending
e.  Construct decision matrix, by Eq. (14).
f.  $W = 1/n$    // Initialize the weight to every FS technique, by Eq. (15).
g.  $U$ = MOORA $(R^*, W)$ , by Eq. (16). // Gain the heuristic vector per every feature utilizing algorithm 1
h.  For c = 1 to Num_Cycle Do
i.      $AntFC[i] = 0, \forall_i = 1, \ldots, m$    // Set the initial features counter to zero
j.      Put ants accidentally on features
k.      For n = 1 to Num_Ant Do
l.          Select the subsequent unvisited feature according to Eq. (17)
m.          Transfer the k-th ant to the newly chosen feature
n.          AntFC[f] = AntFC[f] +1    // Update the feature counter associated with feature f
o.      EndFor
p.      Update the pheromone according to Eq. (3)
q.      Evaporation
r.  EndFor
s.  Sort the features by lessening the order of their pheromones
t.  Output: Choose the top-d features
u.  End Algorithm
```

Figure 5. ESACO Algorithm.

$$F - score = \frac{(TP)}{(TP + 1/2 * (FP + FN))} \tag{19}$$

5.4. **Parameters Tuning.** The parameter tuning in swarm intelligence algorithms are typically determined experimentally. Table 3 presents these parameter values. Num_Cycle, which represents the number of iterations for the ants, is assigned to 30 (Num_Cycle = 30), Num_Ant, which represents the number of ants, is set 30 and the pheromone decay rate is set to 0.15 is assigned to 30 (Num_Ant = 30 and $\rho = 0.15$). The power parameter of the multinomial decay rate and the parameter β are both assiged to 0.7 and 1 ($\beta = 1$), respectively. The final number of features to be selected (m) can range from 10 to 100. However, you can assigned it to any value between 1 and the total number of features (d), i.e., $1 \leq$ m $\leq$ d. For consistency with competing techniques, the presented parameter values are assumed for all techniques, according to the corresponding referenced papers.

Table 2. The specifications of datasets.

Dataset	Feature Type	Num of Instances	Num of Features	Num of Classes	Reference
Arcene	Continuous	200	10000	2	(Hashemi, et al., 2021)
Chiaretti	Continuous	217	1413	3	[1]
Golub	Discrete	72	7129	2	[2]
Jaffe	Discrete	213	676	10	[3]
Khan	Continuous	63	2308	4	[4]
Leukemia	Discrete	72	7070	2	(Li et al., 2017)
Lung	Continuous	203	3312	5	(Li et al., 2017)
Lymphoma	Discrete	96	4026	9	(Li et al., 2017)
NCI60	Continuous	64	6830	14	[5]
ORL	Discrete	400	1024	40	(Samaria, & Harter, 1994)
Orlraws10p	Discrete	100	10304	10	[6]
Prostate-GE	Continuous	102	5966	2	Li et al., 2017)
Semeion	Discrete	1593	256	2	[7]
Sorlie	Continuous	85	456	5	[8]
SRBCT	Discrete	83	2308	4	[9]

6. Result and Discussion

In this section, we compare the efficiency of the presented algorithm ESACO with some techniques in the literature: SFSS (Ma et al. 2011), SELF (Sugiyama et al. 2010), EnsCLS (Hindawi et al. 2013), SFSGL (Sheikhpour et al. 2018),

Table 3. Parameters determining ESACO.

Parameter	Explanation	Value
ρ	Evaporating rate of pheromones	0.1
Num_Cycle	The number of iteratations required for the algorithm to find a solution	30
Num_Ant	The number of ants that search in the space of featurest	30
β	The exchange between pheromones and heuristic information	0.7
m	The number of most remarkable features that select	$10 \leq m \leq 100$
q	exploration-exploitation coefficient	0.7

GSSNMF (Jia et al. 2020), Semi-JMI and Semi-MIM (Sechidis & Brown 2018) and TRCFS (Liu et al. 2010).

Tables 3 and 4 demonstrate the run-time of the techniques in states 20% and 40% of the training instances are labeled. Figs. 5 to 12 demonstrate the classification performance for F-score and Accuracy measures.

6.1. **Comparison with other semi-supervised FS methods.** Based on the results and observations, it can be observed that the ESACO technique achieved better classification performance compared to competing semi-supervised FS strategies in the literature. The mean accuracy of the ESACO when 20% of the training instances are labeled is demonstrated in Figs. 6 and 7. It can be observed that the presented method in all cases, compared to other strategies, has gained better classification accuracy, particularly for Leukemia, Chiaretti, Arcene, Golub, NCI60, Lymphoma, Khan, Prostate-GE, Lung, Sorlie, Orlraws10P, and SRBCT datasets.

When 20% of the training instances are labeled the mean F-score of the ESACO is shown in Figs. 8 and 9. It can be seen that the presented method in all states, compared to other strategies, has gained better F-score value, particularly for Chiaretti, Arcene, Golub, Khan, Lung, Leukemia, Lymphoma, ORL, Prostate-GE, Semeion, Sorlie, and SRBCT datasets.

When 40% of the training instances are labeled the mean accuracy of the ESACO is shown in Figs. 10 and 11. The results of 40% labeling of training instances ESACO, Similar to the results of 20% labeling, has demonstrated better classification accuracy in all cases compared to other methods. particularly for Chiaretti, Arcene, Golub, Jaffe, Khan, Lung, NCI60, Orlraws10P, and Semeion datasets. ESACO has retained its high accuracy in both 40% and 20% labeling of training instances compared to other strategies in all datasets.

When 40% of the training instances are labeled the mean F-score of the ESACO is shown in Figs. 12 and 13. The results of 40% labeling of training

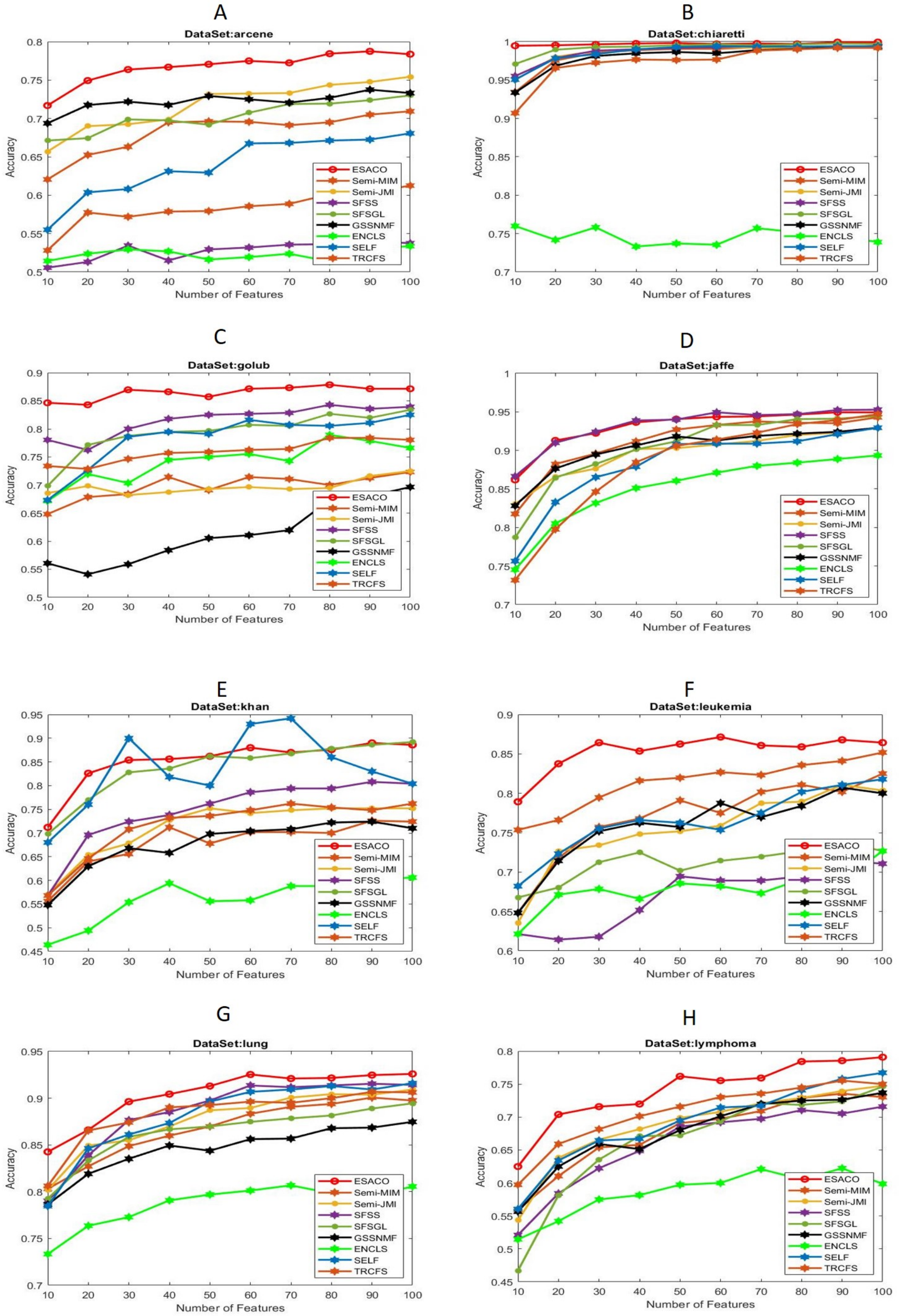


Figure 6. Accuracy metric for 20 percent labeled data

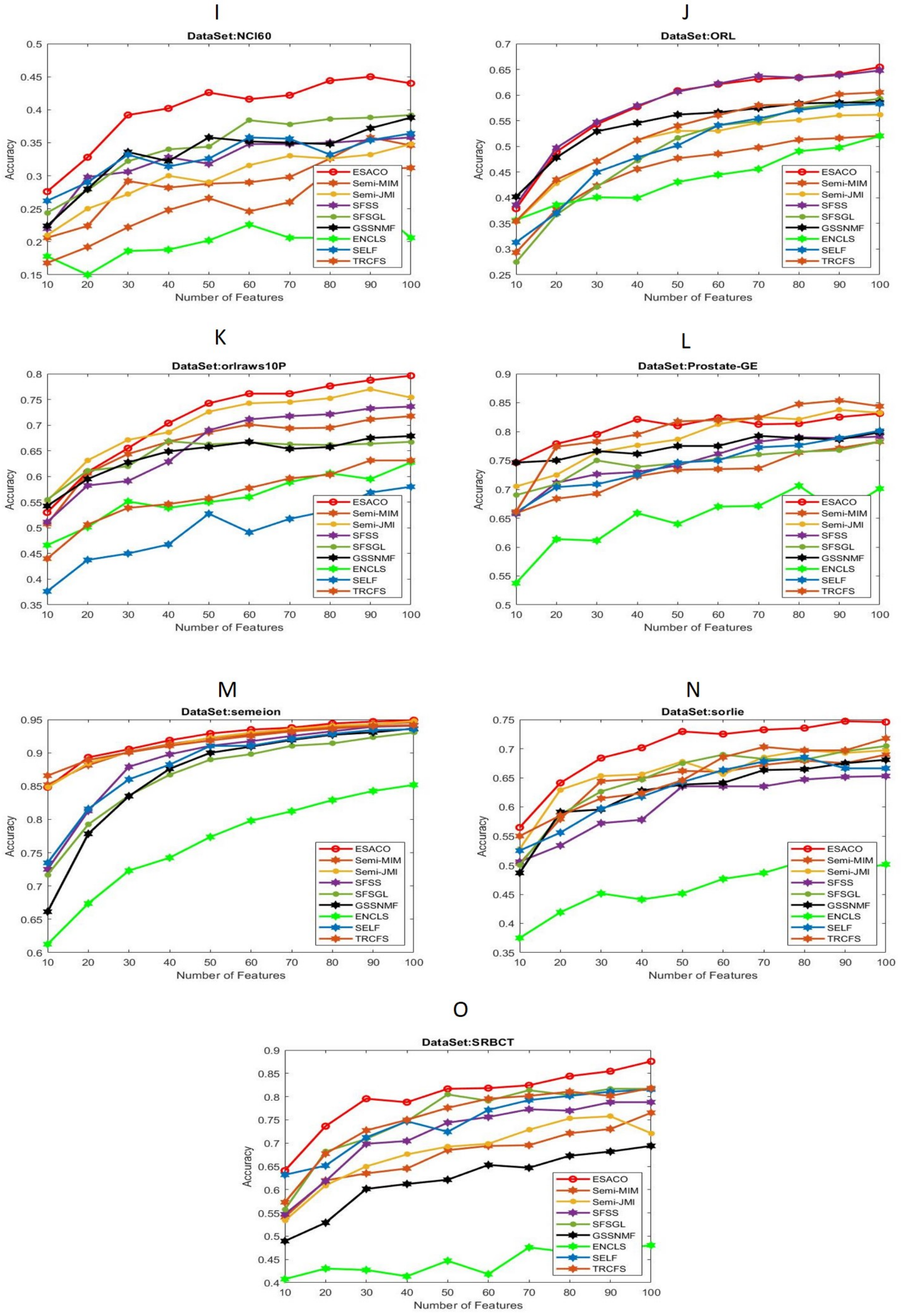


FIGURE 7. Continuous of Accuracy metric for 20 percent labeled data

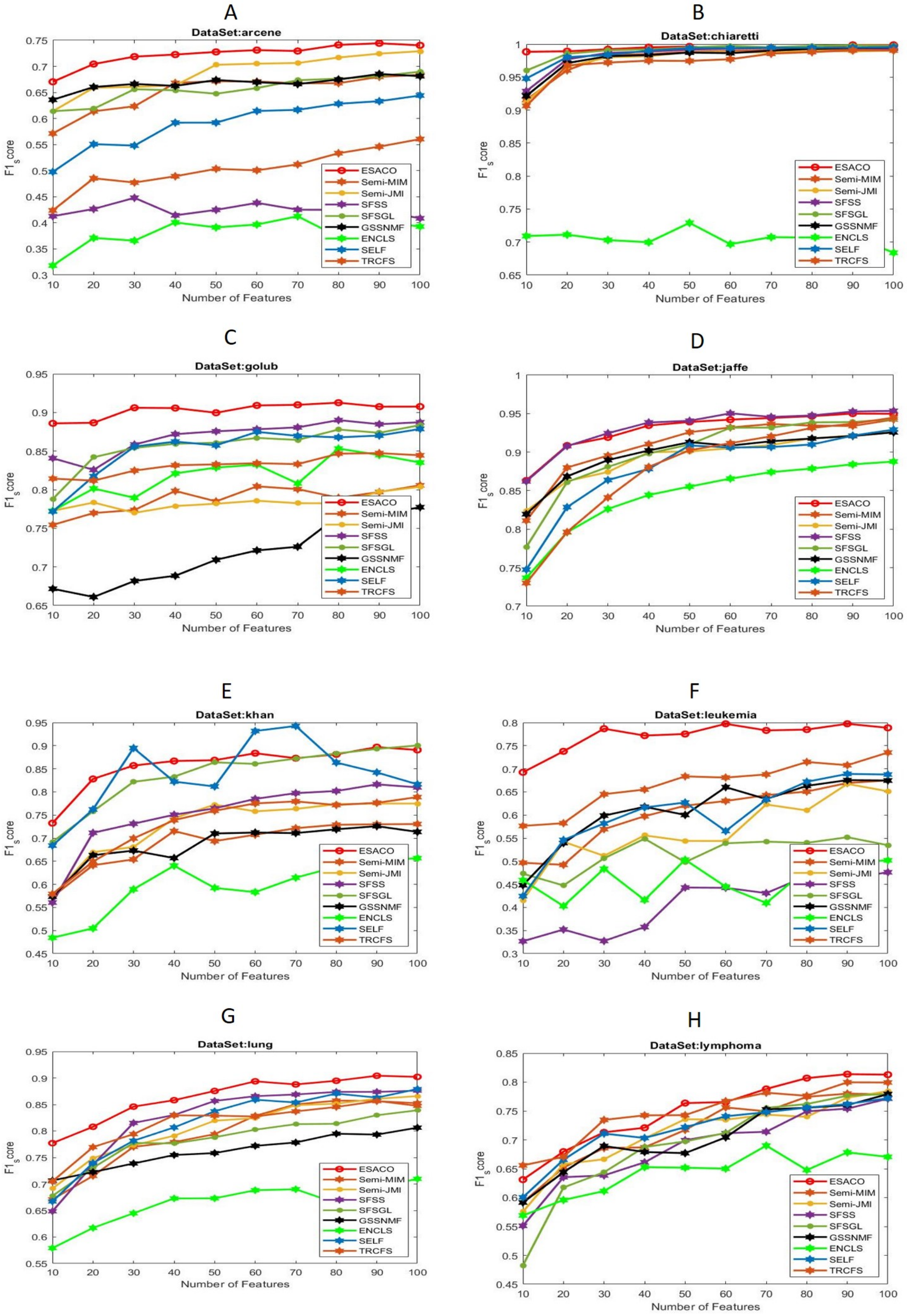


FIGURE 8. F-score metric for 20 percent labeled data

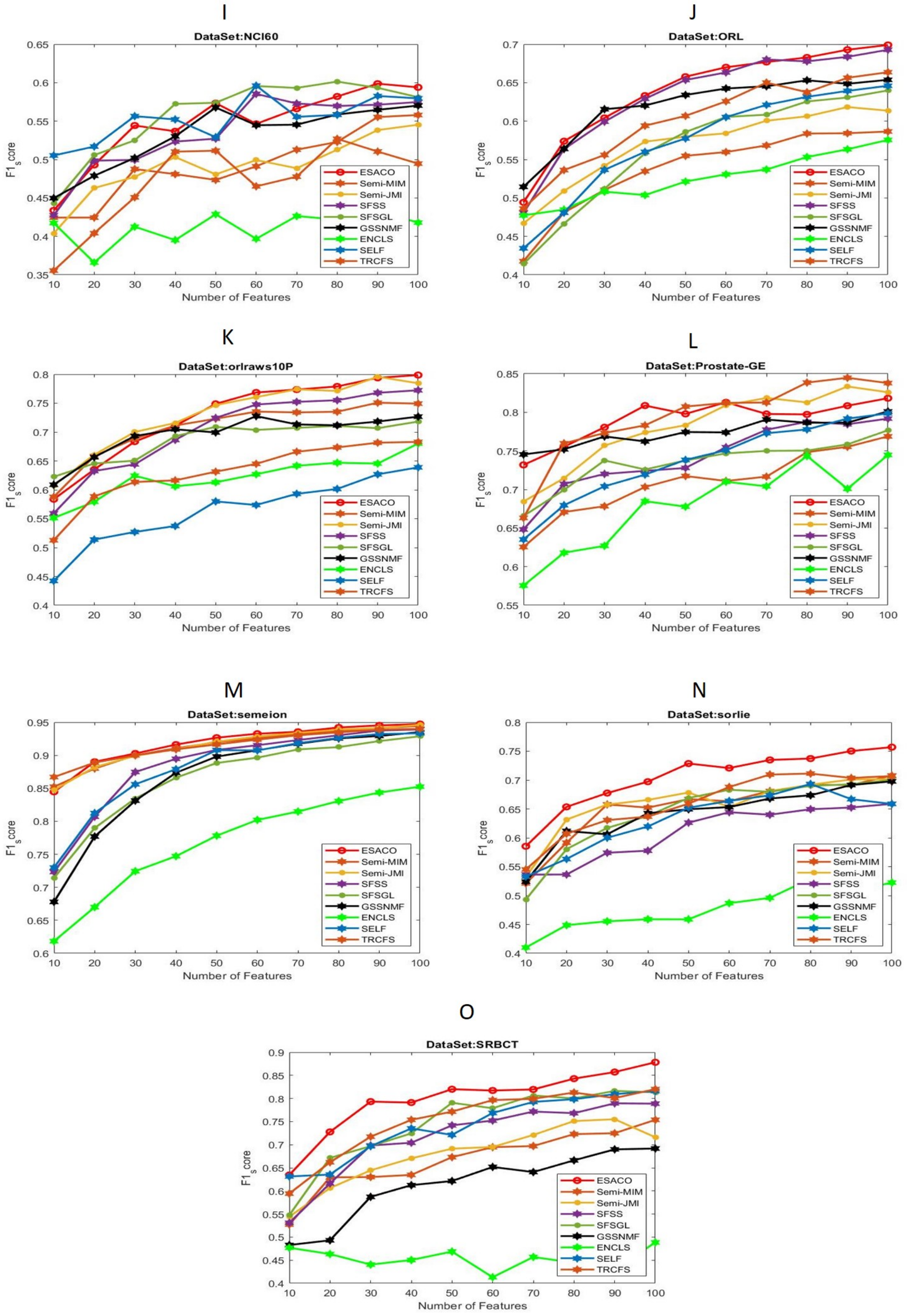


FIGURE 9. Continuous of F-score metric for 20 percent labeled data

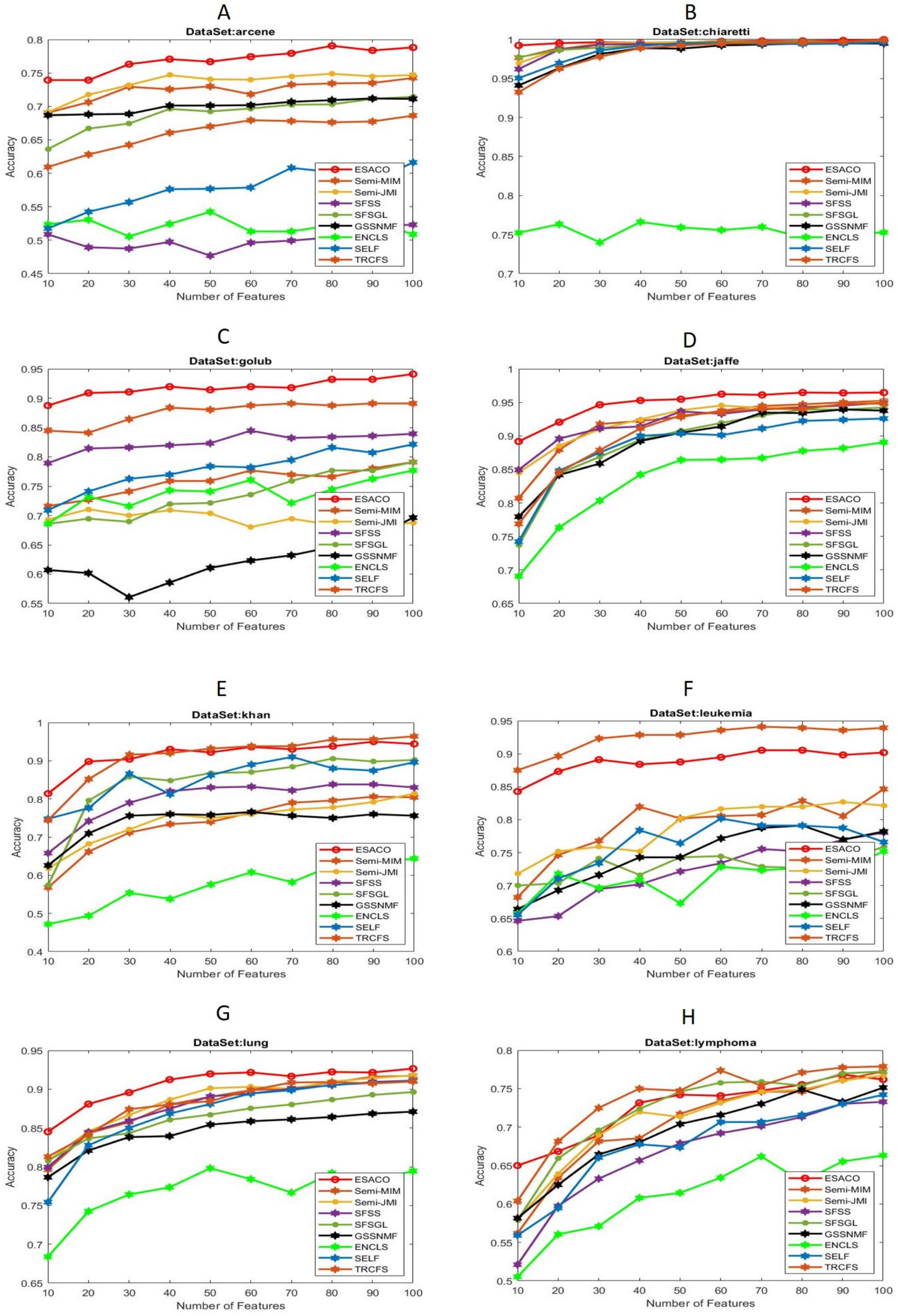


Figure 10. Accuracy metric for 40 percent labeled data

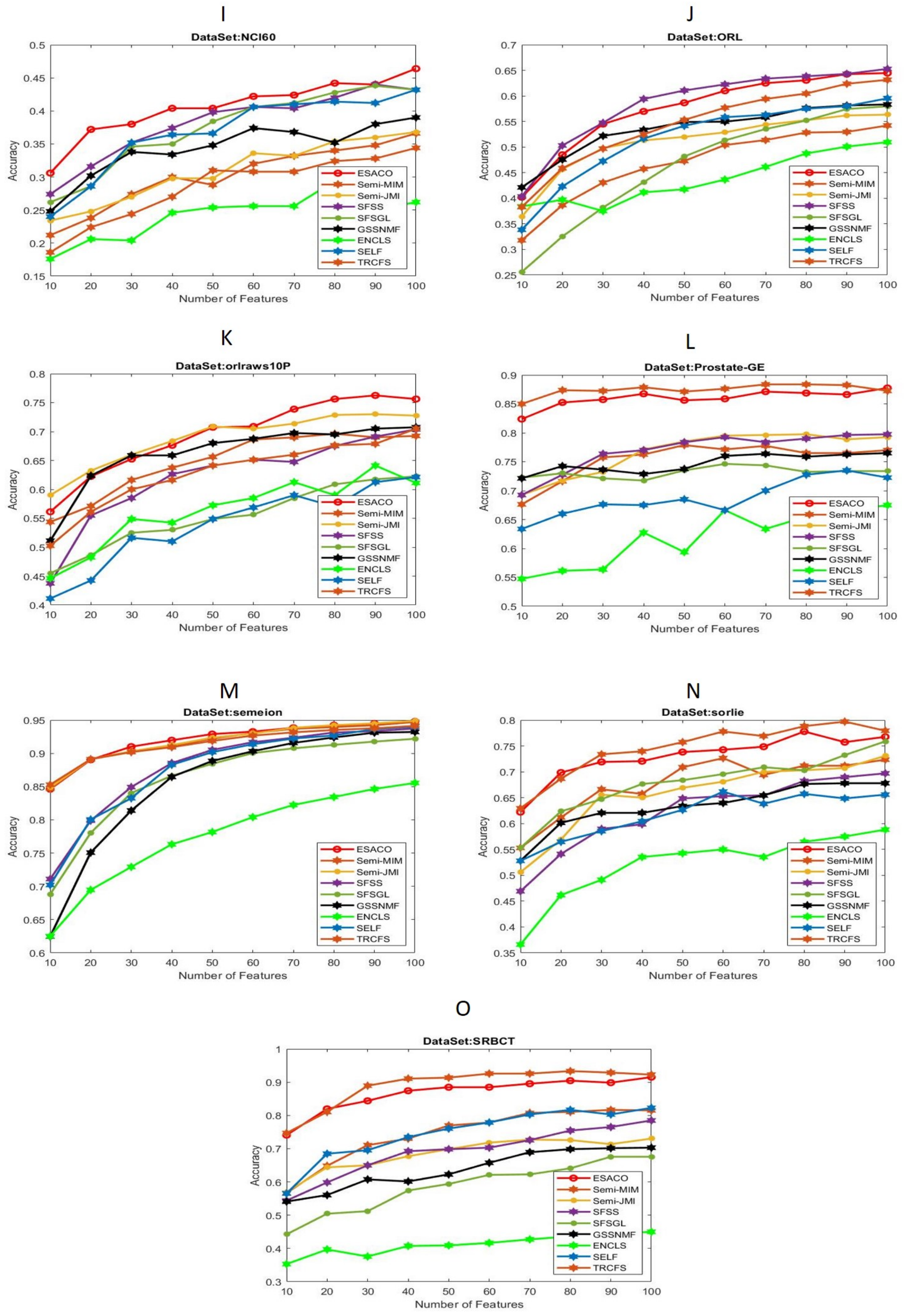


Figure 11. Continuous of Accuracy metric for 40 percent labeled data

instances ESACO, Similar to the results of 20% labeling, has demonstrated better classification F-score in all cases compared to other methods. particularly for Chiaretti, Arcene, Golub, Jaffe, Khan, Lung, NCI60, Orlraws10P, and Semeion datasets. Also, similar to the results of mean accuracy, ESACO has retained its high F-score in both 40% and 20% labeling of training instances compared to other strategies in all datasets.

We have also calculated the computation complexity of the proposed algorithm. Table 4 shows the computational complexity of different steps of the proposed method, considering the pseudo code presented in ESACO Algorithm.

Table 4. The computational complexity of ESACO algorithm.

Step	Computational Complexity (Big Omicron)
Step 1 (Redundancy matrix)	$O(nd^2)$
Step 2 (Relevancy vector)	$O(dl^2)$
Step 3 (LLCFS algorithm)	$O(nd^2)$
Step 4 (MCDM algorithm)	$O(nd + n^2 \log d)$
Steps 5-15 (ACO process)	$O(NumCycle \times NumAnt \times d^2)$
Step 16 (Sorting the features)	$O(d \log d)$
Overall	$O(nd^2 + dl^2 + n^2 \log d + NumCycle \times NumAnt \times d^2)$
Description of symbols	d: number of features, l: number of labeled samples, n: number of instances, NumCycle: number of iterations, NumAnts→number of ants.

Tables 5 and 6 display the run times (in seconds) when 20 % and 40 % of training instances are labeled. A non-parametric Friedman test [50] compares the results' importance according to the statistical procedure. We consider a p-value less than 0.05 to be statistically significant. These comparisons are demonstrated in Tables 7 to 11, and Table 9 displays the general win/tie/loss. In Tables 5 to 8, the (+) sign means that the presented ESACO is statistically better than other techniques. As well as, (-) and (=) signs demonstrate worst and equal efficiency, respectively.

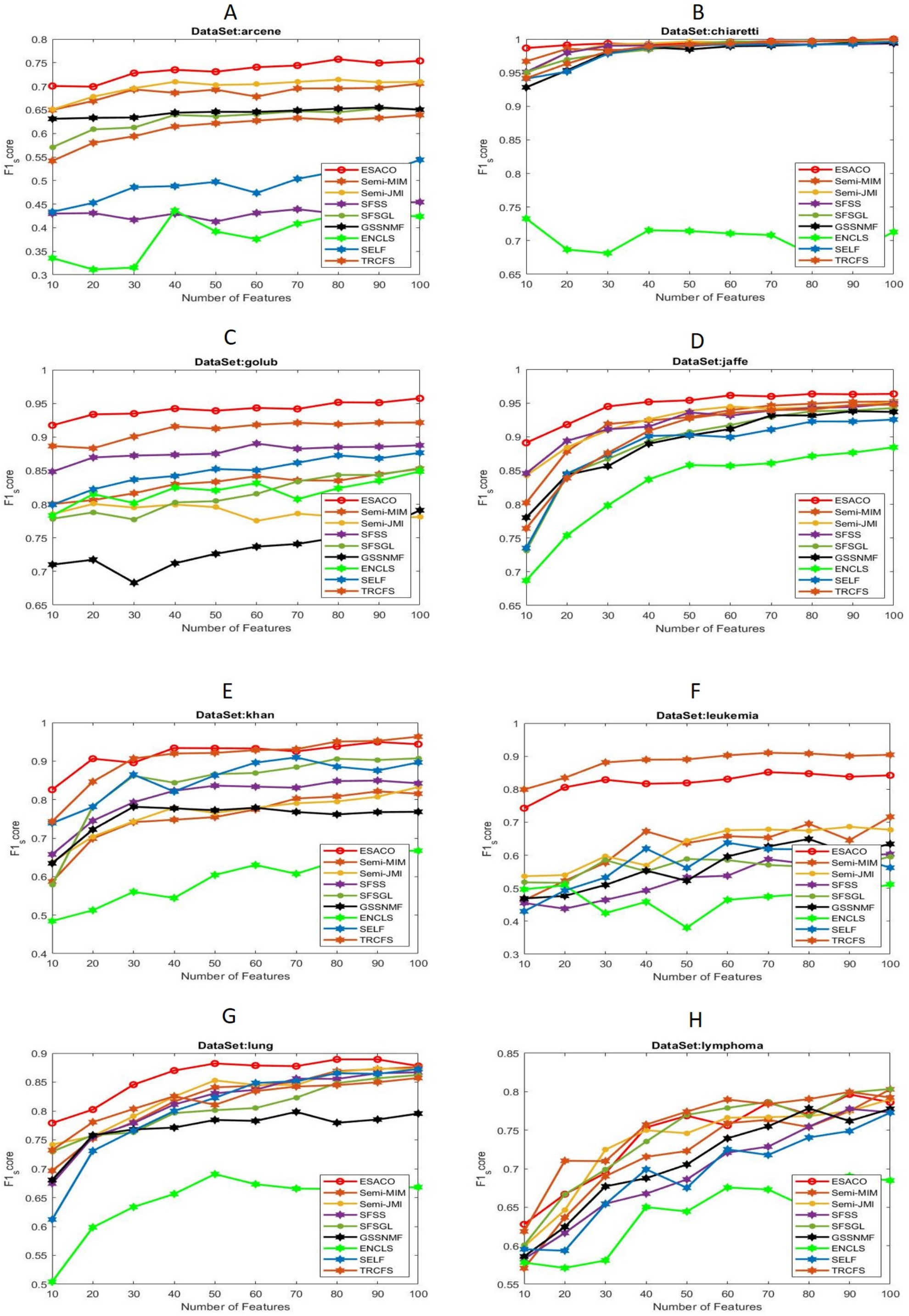


FIGURE 12. F-score metric for 40 percent labeled data

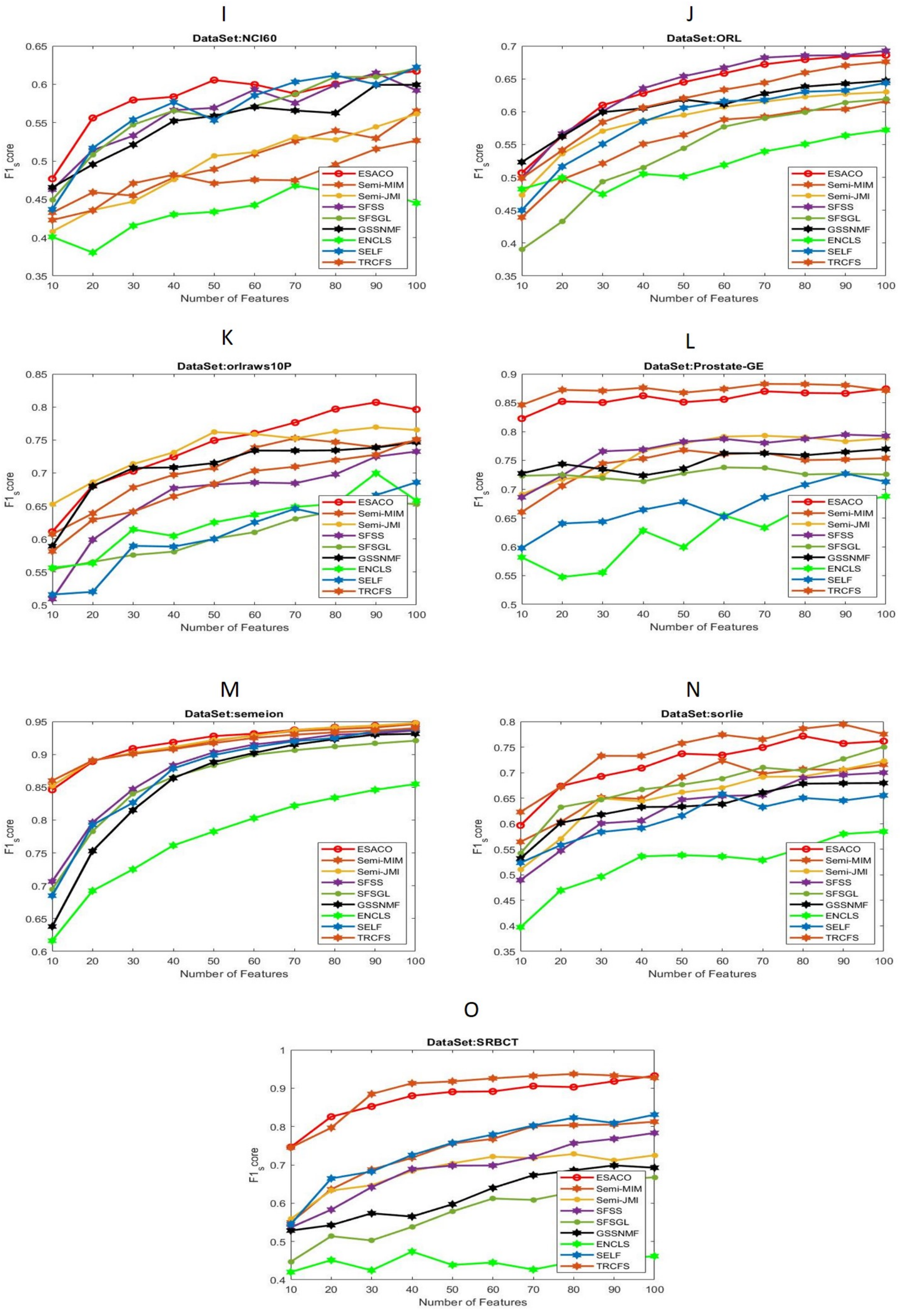


Figure 13. Continuous of F-score metric for 40 percent labeled data

Table 5. Execution time comparisons on 20 percent labeled data

Dataset	Algorithm								
	TRCFS	SELF	ENCLS	GSSNMF	SFSGL	SFSS	Semi-JMI	Semi-MIM	ESACO
Arcene	2.4106	223.5991	10.8828	0.2576	1.0586e+03	39.2320	1.0110e+04	1.0110e+04	9.8125
Chiaretti	0.1341	3.0651	8.8570	0.0525	7.0677	0.6865	146.2542	146.2542	1.8887
Golub	0.9523	291.0064	3.2690	0.0767	1.0747e+03	10.3789	5.5374e+03	5.5374e+03	4.3646
Jaffe	0.0212	0.7157	6.5901	0.0276	0.1391	0.0690	25.8568	25.8569	2.6198
Khan	2.0665	33.8233	0.0875	0.0322	24.3326	2.0407	639.7204	639.7204	1.6704
Leukemia	0.9754	348.3809	2.8181	0.0921	841.3363	39.9451	5.7033e+03	5.7033e+03	3.7441
Lung	0.2826	45.3454	13.2009	0.1430	102.8575	7.0634	1.1580e+03	1.1580e+03	3.0800
Lymphoma	0.3269	294.9491	4.1747	0.1203	197.4651	8.1868	1.7184e+03	1.7184e+03	2.7395
NCI60	0.9253	309.5145	2.5270	0.0958	775.7263	49.1071	4.5409e+03	4.5409e+03	2.9547
ORL	0.0548	1.4303	26.8676	0.1291	0.5390	0.1816	126.7509	126.7509	3.2790
Orlraws10p	2.0274	584.6595	3.9926	0.3086	4.5680e+03	52.0016	8.8094e+03	8.8094e+03	4.0027
Prostate-GE	0.8640	197.2423	4.5990	0.1142	810.2006	38.2078	4.8306e+03	4.8306e+03	2.0036
Semeion	0.4127	0.6385	725.1094	0.3217	2.6039	0.2967	19.2946	19.2946	5.7473
Sorlie	0.0108	1.9155	2.7092	0.0174	0.5128	0.0599	23.1184	23.1184	0.6916
SRBCT	0.0922	27.5736	2.6624	0.0294	29.5941	1.9439	440.8094	440.8094	1.0074

Table 6. Execution time comparisons on 40 percent labeled data

Dataset					Algorithm				
	TRCFS	SELF	ENCLS	GSSNMF	SFSGL	SFSS	Semi-JMI	Semi-MIM	ESACO
Arcene	2.3430	505.1395	13.2143	0.2794	1.4458e+03	39.9798	865.8015	865.8016	8.2353
Chiaretti	0.0924	10.0775	16.6324	0.0931	4.3854	1.3928	21.2724	21.2724	2.5095
Golub	0.8896	442.0393	2.4231	0.0643	163.4951	8.6186	307.9999	308.0000	6.2567
Jaffe	0.0363	1.7457	15.1251	0.0588	0.1730	0.1243	4.1754	4.1755	3.3599
Khan	0.1437	74.6716	4.0143	0.0452	62.5907	4.2744	62.3078	62.3079	2.3025
Leukemia	0.8355	757.9332	2.5230	0.1263	1.4366e+04	41.5873	308.8415	308.8422	6.0314
Lung	0.5089	240.2031	25.7691	0.2615	160.2422	13.4017	204.6004	204.6005	6.7009
Lymphoma	0.3652	164.0963	2.9120	0.0741	1.8168e+03	8.5217	96.9267	96.9268	4.2023
NCI60	0.8981	486.4758	3.6843	0.1251	1.0988e+03	73.6994	549.2767	549.2769	3.6623
ORL	0.1058	2.3717	366.5952	0.2207	0.8914	0.3266	276.8230	276.8231	4.3867
Orlraws10p	1.6026	1.2548e+03	3.9167	0.1927	824.7220	53.5960	657.9727	657.9727	6.3218
Prostate-GE	0.7360	285.9702	5.5476	0.1098	1.3860e+03	42.6458	288.7139	288.7140	2.9473
Semeion	0.3840	0.8446	889.1811	0.2615	2.9452	0.3019	1.7231	1.7232	4.7613
Sorlie	0.0119	4.5501	4.8377	0.0209	4.0180	0.0785	2.4465	2.4466	1.2993
SRBCT	0.1470	59.7695	4.6626	0.0553	96.6638	2.9199	78.1815	78.1816	1.9804

TABLE 7. The gained p-values for the accuracy metric according to 20 percent labeled data by the Friedman test

Dataset					ESACO against			
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Chiaretti	0.0004 (+)	0.0004 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0004 (+)	0.0004 (+)
Golub	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Jaffe	0.0016 (+)	0.0016 (+)	0.1616 (=)	0.0016 (+)	0.0016 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Khan	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.1573 (=)	0.0005 (+)	0.0005 (+)	0.2987 (=)	0.0005 (+)
Leukemia	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lung	0.0005 (+)	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)
Lymphoma	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0009 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)
NCI60	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)
ORL	0.0005 (+)	0.0018 (+)	0.4884 (=)	0.0005 (+)	0.0153 (+)	0.0018 (+)	0.0005 (+)	0.0018 (+)
Orlraws10p	0.0056 (+)	0.1153 (=)	0.0018 (+)	0.0377 (+)	0.0056 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Prostate-GE	0.0005 (+)	0.7290 (=)	0.0005 (+)	0.0005 (+)	0.0029 (+)	0.0005 (+)	0.0005 (+)	0.7290 (=)
Semeion	0.0377 (+)	0.0833 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0153 (+)
Sorlie	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
SRBCT	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0016 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)

Table 8. The gained p-values for the F-score metric according to 20 percent labeled data by the Friedman test

Dataset				ESACO against			
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Chiaretti	0.0004 (+)	0.0004 (+)	0.0143 (+)	0.0484 (+)	0.0005 (+)	0.0005 (+)	0.1616 (=)	0.0004 (+)
Golub	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Jaffe	0.0018 (+)	0.0005 (+)	0.1659 (=)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Khan	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.1659 (=)	0.0005 (+)	0.0005 (+)	0.2987 (=)	0.0005 (+)
Leukemia	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lung	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lymphoma	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0018 (+)	0.4884 (=)
NCI60	0.0005 (+)	0.0018 (+)	0.1659 (=)	0.2987 (=)	0.0153 (+)	0.0005 (+)	1 (=)	0.0005 (+)
ORL	0.0005 (+)	0.0018 (+)	0.0833 (=)	0.0005 (+)	0.0377 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
Orlraws10p	0.1659 (=)	0.7290 (=)	0.0056 (+)	0.0377 (+)	0.0833 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
Prostate-GE	0.0005 (+)	0.7290 (=)	0.0005 (+)	0.0005 (+)	0.0056 (+)	0.0005 (+)	0.0005 (+)	0.2987 (+)
Semeion	0.0377 (+)	0.1659 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0153 (+)
Sorlie	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
SRBCT	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0016 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)

TABLE 9. The gained p-values for the accuracy metric according to 40 percent labeled data by the Friedman test

Dataset					ESACO against			
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Chiaretti	0.0016 (+)	0.0133 (+)	0.0004 (+)	0.0016 (+)	0.0004 (+)	0.0005 (+)	0.0005 (+)	0.0016 (+)
Golub	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Jaffe	0.0005 (+)	0.0005 (+)	0.0005 (=)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Khan	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.2207 (=)
Leukemia	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lung	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lymphoma	0.0080 (+)	0.0229 (+)	0.0005 (+)	0.1659 (=)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0377 (+)
NCI60	0.0005 (+)	0.0005 (+)	0.0087 (+)	0.0056 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)
ORL	0.0005 (+)	0.0018 (+)	0.0153 (+)	0.0005 (+)	0.0056 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)
Orlraws10p	0.0018 (+)	0.4884 (=)	0.0005 (+)	0.0005 (+)	0.0377 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Prostate-GE	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0056 (+)
Semeion	0.2987 (=)	0.7290 (=)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0153 (+)
Sorlie	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0056 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0153 (+)
SRBCT	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0051 (+)

Table 10. The gained p-values for the F-score metric according to 40 percent labeled data by the Friedman test

Dataset					ESACO against			
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
Arcene	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Chiaretti	0.0005 (+)	0.2888 (=)	0.0016 (+)	0.0358 (+)	0.0004 (+)	0.0005 (+)	0.0004 (+)	0.0051 (+)
Golub	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Jaffe	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Khan	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	1 (=)
Leukemia	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)
Lung	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
Lymphoma	0.0833 (=)	0.0833 (=)	0.0005 (+)	0.2987 (=)	0.0018 (+)	0.0005 (+)	0.0018 (+)	0.0833 (=)
NCI60	0.0005 (+)	0.0005 (+)	0.0153 (+)	0.0833 (=)	0.0018 (+)	0.0005 (+)	0.2987 (=)	0.0005 (+)
ORL	0.0005 (+)	0.0018 (+)	0.1659 (=)	0.0005 (+)	0.0056 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)
Orlraws10p	0.0005 (+)	0.7290 (=)	0.0005 (+)	0.0005 (+)	0.0153 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
Prostate-GE	0.0005 (+)	0.0005 (=)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0056 (+)
Semeion	0.2987 (=)	0.4884 (=)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0153 (+)
Sorlie	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)	0.0018 (+)	0.0005 (+)	0.0005 (+)	0.0018 (+)
SRBCT	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0005 (+)	0.0377 (+)

Table 11. The overall win/tie/loss, according to the Friedman test

Dataset					ESACO against			
	Semi-MIM	Semi-JMI	SFSS	SFSGL	GSSNMF	ENCLS	Self	TRCFS
20% accuracy of labeled data	15/0/0	13/2/0	13/2/0	14/1/0	15/0/0	115/0/0	14/1/0	14/1/0
F-measure of 20% of labeled data	14/1/0	13/2/0	12/3/0	13/2/0	15/0/0	15/0/0	12/3/0	14/1/0
40% accuracy of labeled data	14/1/0	13/2/0	14/1/0	14/1/0	15/0/0	15/0/0	15/0/0	14/1/0
F-measure of 40% of labeled data	13/2/0	11/4/0	14/1/0	13/2/0	15/0/0	15/0/0	14/1/0	13/2/0
Overall	56/4/0	50/10/0	53/7/0	54/6/0	60/0/0	60/0/0	55/5/0	55/5/0

To evaluate the robustness of the proposed method to parameter variations, we conducted a sensitivity analysis on the available datasets using five distinct parameter configurations. Table 12 shows the accuracy results of the sensitivity test of the parameters used in the proposed method. Our results show that the proposed method is relatively insensitive to parameter changes. This means that even when the parameters are adjusted, the results remain stable. The parameters in the table are: *decay*: evaporation coefficient, *cycle*: number of executions of ESACO algorithm, *ant*: number of ants q is exploration-exploitation coefficient and β: the exchange between pheromones and heuristic information .

Table 12. Testing the sensitivity of parameters.

Dataset	decay = 0.25, cycle = 50, ant = 60, beta & q = 0.8	decay = 0.2, cycle = 40, ant = 50, beta & q = 0.7	decay = 0.15, cycle = 30, ant = 40, beta & q = 0.6	decay = 0.1, cycle = 20, ant = 30, beta & q = 0.5	decay = 0.05, cycle = 10, ant = 20, beta & q = 0.4
Arcene	0.80375	0.780625	0.765	0.776875	0.774375
Chiaretti	1	0.999418605	0.999418605	0.998837209	0.999418605
Golub	0.882142857	0.8875	0.898214286	0.851785714	0.864285714
Jaffe	0.952352941	0.948235294	0.952941176	0.951764706	0.954117647
Khan	0.848	0.886	0.912	0.898	0.878
Leukemia	0.844642857	0.85	0.828571429	0.857142857	0.842857143
Lung	0.927777778	0.924691358	0.916666667	0.919135802	0.922222222
Lymphoma	0.789473684	0.769736842	0.848214286	0.784210526	0.781578947
NCI60	0.392	0.418	0.446	0.424	0.434
ORL	0.6415625	0.613125	0.634375	0.63875	0.6365625
Orlraws10p	0.80375	0.78125	0.79875	0.77	0.79375
Prostate-GE	0.8575	0.83125	0.8375	0.82875	0.80375
Semeion	0.95	0.951805338	0.950156986	0.950627943	0.948744113
Sorlie	0.745588235	0.760294118	0.785294118	0.739705882	0.763235294
SRBCT	0.874242424	0.836363636	0.865151515	0.834848485	0.836363636

6.2. **Discussion.** The proposed method, ESACO, to check the best feature subset in semi-supervised data utilizes the ACO algorithm. ACO can solve combinatorial optimization problems, especially FS problems. The primary difference between this technique and other metaheuristic algorithms and other competitive techniques is the heuristic selection. The FS procedure in competitive techniques is based on a specific heuristic. However, the recommended algorithm used an ensemble of heuristic selection techniques. We have performed experiments to display the effect of the ensemble of heuristics in the semi-supervised FS.

The main characteristic of the suggested algorithm is using the ensemble of heuristics according to the MCDM process in ACO, which is the first time used in the semi-supervised FS. Thus, considering its high performance and power, we determined to use this procedure in semi-supervised FS.

Furthermore, as in Tables 3 and 4, the run-time of all techniques is displayed. According to these values, it can be observed that the presented algorithm is not the fastest. Due to the iterative framework of the ant algorithm, this matter is entirely rational. Because in this method, to achieve the best feature, we have assumed 30 iterations. Furthermore, in every iteration, the MCDM procedure is performed.

A non-parametric test is utilized according to the Friedman test to evaluate the validity of the results. Table. 9 suggests the general values of win/tie/loss by the statistical comparisons. We can see the excellence of the presented technique against the other methods in two different semi-supervised prospective and two classification metrics. The presented algorithm acts better statistically than the competitive techniques in most cases (443 cases between 480 cases), and the efficiency is the tie in the remnant of 37 cases.

7. Conclusion

Based on our knowledge, no ACO-based semi-supervised FS method equipped with an ensemble of heuristics has been presented so far. The presented technique uses an ensemble of heuristic functions instead of one heuristic function in ACO. The idea of using ensemble learning for semi-supervised FS is based on the fact that using multiple learning algorithms results in better performance and higher accuracy than using one single function alone. EASCO utilizes the MCDM procedure in the FS process based on the recommendations of numerous heuristics to find the best heuristic. EASCO utilizes unlabeled and labeled instances of information at the same time to discover the most beneficial features. The results of the KNN classifier show that the suggested method is better than comparing techniques on fifteen datasets. Furthermore, the results demonstrate that the presented algorithm is better efficient than other methods in two states of accidental 40% and 20% labeling of the training instances with proper run-time. One of the advantages of the ESACO algorithm is its high speed and accuracy. Also, in most cases, the ESACO algorithm has been able to show its superiority over competing algorithms. While the ESACO algorithm offers these advantages, it is also somewhat time-consuming, which is acceptable due to the iterative nature of the ACO algorithm.Considering the premier efficiency of EASCO, we desire to generalize this procedure to semi-supervised FSs: wrapper, embedded, and multi-label techniques.

References

[1] Aruldoss, M. (2013). "A Survey on Multi Criteria Decision Making Methods and Its Applications." 1(1):31–43.

[2] Asghari, V. Z., Soodeh, H., Ebrahimi, S., Javidi, & M. M. (2022). "A New Hybrid Feature Selection Based on Improved Equilibrium Optimization." Chemometrics and Intelligent Laboratory Systems 228:104618. doi: 10.1016/J.CHEMOLAB.2022.104618.

[3] Bayati, Hamid., Dowlatshahi, M. B., & Hashemi, A . (2022). "MSSL: A Memetic-Based Sparse Subspace Learning Algorithm for Multi-Label Classification." International Journal of Machine Learning and Cybernetics 2022 1–18. doi: 10.1007/S13042-022-01616-5.

[4] Beiranvand, F., Vahid M., & Dowlatshahi, M. B.. (2022). "Unsupervised Feature Selection for Image Classification: A Bipartite Matching-Based Principal Component Analysis Approach." Knowledge-Based Systems 250:109085. doi: 10.1016/J.KNOSYS.2022.109085.

[5] Benabdeslem, K, & Hindawi, M. (2011). "Constrained Laplacian Score for Semi-Supervised Feature Selection." Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 6911 LNAI:204–18. doi: 10.1007/978-3-642-23780-5_23.

[6] Cai, J,. Jiawei L,. Shulin W, & Sheng Yang. (2018). "Feature Selection in Machine Learning: A New Perspective." Neurocomputing 300:70–79. doi: 10.1016/j.neucom.2017.11.077.

[7] Cao, D,. Yuan C,. Jin C,. Hongyan Z,. & and Zheming Y. (2021). "An Improved Algorithm for the Maximal Information Coefficient and Its Application." Royal Society Open Science 8(2). doi: 10.1098/RSOS.201424.

[8] Chandrashekar, G,. & Ferat S. (2014). "A Survey on Feature Selection Methods." Computers and Electrical Engineering 40(1):16–28. doi: 10.1016/j.compeleceng.2013.11.024.

[9] Dey, N, ed. (2024a). "Applications of Ant Colony Optimization and Its Variants." doi: 10.1007/978-981-99-7227-2.

[10] Dey, N, ed. (2024b). "Applied Multi-Objective Optimization." doi: 10.1007/978-981-97-0353-1.

[11] Dorigo, M,. Birattari, M,. & Stutzle, T. (2006). "Ant Colony Optimatization." Intelligence Magazine, IEEE 321.

[12] Dorigo, M,. & Christian Blum. (2005). "Ant Colony Optimization Theory: A Survey." Theoretical Computer Science 344(2–3):243–78. doi: 10.1016/j.tcs.2005.05.020.

[13] Dowlatshahi, M. B., & Hashemi, A. (2023). “Unsupervised Feature Selection: A Fuzzy Multi-Criteria Decision-Making Approach.” Iranian Journal of Fuzzy Systems 20(7):55–70. doi: 10.22111/IJFS.2023.7630.

[14] Du, L,. & Yi D. (2015). “Unsupervised Feature Selection with Adaptive Structure Learning.” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 2015-August:209–18. doi: 10.48550/arxiv.1504.00736.

[15] van E,. Jesper, E., & Holger H. Hoos. (2020). “A Survey on Semi-Supervised Learning.” Machine Learning 109(2):373–440.

[16] Narayanan, G., Shankar, R, Čep, R., Chakraborty, S, & Kanak Kalita. (2023). “Efficient Feature Selection Using Weighted Superposition Attraction Optimization Algorithm.” Applied Sciences (Switzerland) 13(5). doi: 10.3390/APP13053223.

[17] Hashemi, A., Dowlatshahi, M. B, & Nezamabadi-pour, H. (2021). “An Efficient Pareto-Based Feature Selection Algorithm for Multi-Label Classification.” Information Sciences 581:428–47. doi: 10.1016/J.INS.2021.09.052.

[18] Hashemi, A., & Dowlatshahi, M. B.. (2023). “A Fuzzy Integral Approach for Ensembling Unsupervised Feature Selection Algorithms.” 2023 28th International Computer Conference, Computer Society of Iran, CSICC 2023. doi: 10.1109/CSICC58665.2023.10105330.

[19] Hashemi, A., Dowlatshahi, M. B, & Nezamabadi-pour, H. (2020). “MGFS: A Multi-Label Graph-Based Feature Selection Algorithm via PageRank Centrality.” Expert Systems with Applications 142:113024. doi: 10.1016/J.ESWA.2019.113024.

[20] Hashemi, A., Dowlatshahi, M. B, & Nezamabadi-pour, H. (2021). “Ensemble of Feature Selection Algorithms: A Multi-Criteria Decision-Making Approach.” International Journal of Machine Learning and Cybernetics 2021 13:1 13(1):49–69. doi: 10.1007/S13042-021-01347-Z.

[21] Hashemi, A., Joodaki, M., Joodaki, N. Z, & Dowlatshahi, M. B. (2022). “Ant Colony Optimization Equipped with an Ensemble of Heuristics through Multi-Criteria Decision Making: A Case Study in Ensemble Feature Selection.” Applied Soft Computing 124:109046. doi: 10.1016/J.ASOC.2022.109046.

[22] Hashemi, A., Pajoohan, M. R., & Dowlatshahi, M. B. (2022). “Online Streaming Feature Selection Based on Sugeno Fuzzy Integral.” 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2022. doi: 10.1109/CFIS54774.2022.9756477.

[23] Hashemi, A., Pajoohan, M. R., & Dowlatshahi, M. B. (2023). “An Election Strategy for Online Streaming Feature Selection.” 2023 28th International Computer Conference, Computer Society of Iran, CSICC 2023. doi: 10.1109/CSICC58665.2023.10105319.

[24] Hindawi, M, Elghazel, H., & Benabdeslem K. (2013). “Efficient Semi-Supervised Feature Selection by an Ensemble Approach.” International Workshop on Complex Machine Learning Problems with Ensemble Methods COPEM@ECML/PKDD’13, Prague, Czech Republic. pp.41-55. ⟨hal-01339249⟩.

[25] Hira, Zena M., & Duncan F. Gillies. (2015). “A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data.” Advances in Bioinformatics 2015. doi: 10.1155/2015/198363.

[26] Jia, Y., Kwong, S., Hou, J., & Wu, W. (2020). “Semi-Supervised Non-Negative Matrix Factorization with Dissimilarity and Similarity Regularization.” IEEE Transactions on Neural Networks and Learning Systems 31(7):2510–21. doi: 10.1109/TNNLS.2019.2933223.

[27] Kanan, H R., Faez, F., & Taheri, M. (2007). “Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in the Application of Face Recognition System.” Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4597 LNCS:63–76. doi: 10.1007/978-3-540-73435-2_6.

[28] Karel, W., Brauers, W., & Zavadskas, E. (2006). ”The MOORA method and its application to privatization in a transition economy”. Control and Cybernetics. http://eudml.org/doc/209425.

[29] Karimi, F., Dowlatshahi, M. B, & Hashemi, A. (2023). “SemiACO: A Semi-Supervised Feature Selection Based on Ant Colony Optimization.” Expert Systems with Applications 214:119130. doi: 10.1016/J.ESWA.2022.119130.

[30] Khalid, S., Khalil, T., & Nasreen, S. (2014). “A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning.” Proceedings of 2014 Science and Information Conference, SAI 2014 372–78. doi: 10.1109/SAI.2014.6918213.

[31] Lee, J., R., & Nicewander W. A. (2012). “Thirteen Ways to Look at the Correlation Coefficient.” American Statistician 42(1):59–66. doi: 10.1080/00031305.1988.10475524.

[32] Li, J., Cheng. K., Wang, S., Morstatter, F., Trevino, R P., Tang, J., & Liu, h. (2017). “Feature Selection: A Data Perspective.” ACM Computing Surveys 50(6). doi: 10.1145/3136625.

[33] Liao, Y, & Vemuri, V. R. (2002). “Use of K-Nearest Neighbor Classifier for Intrusion Detection.” Computers and Security 21(5):439–48. doi: 10.1016/S0167-4048(02)00514-X.

[34] Liu, Y., Nie, F., Wu, J., & Chen, L. (2010). “Semi-Supervised Feature Selection Based on Label Propagation and Subset Selection.” Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application 293–96. doi: 10.1109/ICCIA.2010.6141595.

[35] Ma, Zhigang, Yi Yang, Feiping Nie, Jasper Uijlings, and Nicu Sebe. (2011). “Exploiting the Entire Feature Space with Sparsity for Automatic Image Annotation.” MM’11 - Proceedings of the 2011 ACM Multimedia Conference and Co-Located Workshops 283–92. doi: 10.1145/2072298.2072336.

[36] Mazyavkina, Nina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. (2021). “Reinforcement Learning for Combinatorial Optimization: A Survey.” Computers and Operations Research 134. doi: 10.1016/j.cor.2021.105400.

[37] Miao, Jianyu, and Lingfeng Niu. (2016). “A Survey on Feature Selection.” Procedia Computer Science 91:919–26. doi: 10.1016/j.procs.2016.07.111.

[38] Miri, M., Dowlatshahi, M. B, & Hashemi, A. (2022a). “Evaluation Multi Label Feature Selection for Text Classification Using Weighted Borda Count Approach.” 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2022. doi: 10.1109/CFIS54774.2022.9756467.

[39] Miri, M., Dowlatshahi, M. B, & Hashemi, A. (2022b). “Feature Selection for Multi-Label Text Data: An Ensemble Approach Using Geometric Mean Aggregation.” 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems, CFIS 2022. doi: 10.1109/CFIS54774.2022.9756484.

[40] Miri, M., Dowlatshahi, M. B, Hashemi, A., Kuchaki., R. M, Gupta, B. B., & Alhalabi, W, S. (2022c). “Ensemble Feature Selection for Multi-Label Text Classification: An Intelligent Order Statistics Approach.” International Journal of Intelligent Systems. doi: 10.1002/INT.23044.

[41] Marta S.R., Dalila, B.M.M., & Fernando A.C.C. (2012). “Ant Colony Optimization: A Literature Survey.” FEP Working Papers 474, Universidade do Porto, Faculdade de Economia do Porto.

[42] Pan, W. (2021). “Feature Selection Algorithm Based on Maximum Information Coefficient.” IEEE Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) 2600–2603. doi: 10.1109/IAEAC50856.2021.9390868.

[43] Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2020). “MLACO: A Multi-Label Feature Selection Algorithm Based on Ant Colony Optimization.” Knowledge-Based Systems 192:105285. doi: 10.1016/j.knosys.2019.105285.

[44] Paniri, M., Dowlatshahi, M. B., & Nezamabadi-pour, H. (2021). “Ant-TD: Ant Colony Optimization plus Temporal Difference Reinforcement Learning for Multi-Label Feature Selection.” Swarm and Evolutionary Computation 64:100892. doi: 10.1016/j.swevo.2021.100892.

[45] Sechidis, K., & Brown, G. (2018). “Simple Strategies for Semi-Supervised Feature Selection.” Machine Learning 107(2):357–95. doi: 10.1007/s10994-017-5648-2.

[46] Sheikhpour, R., Agha Sarram, M., Gharaghani, S., & Zare Chahooki, M.A. (2017). “A Survey on Semi-Supervised Feature Selection Methods.” Pattern Recognition 64:141–58. doi: 10.1016/j.patcog.2016.11.003.

[47] Sheikhpour, R., Agha Sarram, M., & Sheikhpour, E. (2018). “Semi-Supervised Sparse Feature Selection via Graph Laplacian Based Scatter Matrix for Regression Problems.” Information Sciences 468:14–28. doi: 10.1016/j.ins.2018.08.035.

[48] Sugiyama, M., Idé, T., Nakajima, S., & Sese, J. (2010). “Semi-Supervised Local Fisher Discriminant Analysis for Dimensionality Reduction.” Machine Learning 78(1–2):35–61. doi: 10.1007/s10994-009-5125-7.

[49] Venkatesh, B., & J. Anuradha. (2019). “A Review of Feature Selection and Its Methods.” Cybernetics and Information Technologies 19(1):3–26. doi: 10.2478/CAIT-2019-0001.

[50] Wu, M., & Schölkopf, B. (2006). “A Local Learning Approach for Clustering.” Advances in Neural Information Processing Systems 19.

[51] Xue, B., Zhang, M., Browne, W. N., & Yao, X. (2016). “A Survey on Evolutionary Computation Approaches to Feature Selection.” IEEE Transactions on Evolutionary Computation 20(4):606–26. doi: 10.1109/TEVC.2015.2504420.

[52] Zeng, H., & Cheung, Y. M. (2011). “Feature Selection and Kernel Learning for Local Learning-Based Clustering.” IEEE Transactions on Pattern Analysis and Machine Intelligence 33(8):1532–47. doi: 10.1109/TPAMI.2010.215.

[53] Zhu, H., You, X., & Liu, S. (2019). "Multiple Ant Colony Optimization Based on Pearson Correlation Coefficient." IEEE Access 7:61628–38. doi: 10.1109/ACCESS.2019.2915673.

Fereshteh Karimi
Orcid number: 0000-0003-0105-904X
Department of Computer Engineering
Lorestan University
Khoramabad, Iran
Email address: karimi.fr@fe.lu.ac.ir

Mohammad Bagher Dowlatshahi
Orcid number: 0000-0002-4862-0626
Department of Computer Engineering
Lorestan University
Khoramabad, Iran
Email address: dowlatshahi.mb@lu.ac.ir

Amin Hashemi
Orcid number: 0000-0001-8019-1721
Department of Computer Engineering
Lorestan University
Khoramabad, Iran
Email address: hashemi.am@fe.lu.ac.ir