



Adaptive Technique for Feature Selection in Modified Graph Clustering-Based Ant Colony Optimization

Hassan Almazini^{1*} Ku Ruhana Ku-Mahamud¹

¹*School of Computing, University Utara Malaysia, Kedah, Malaysia*

* Corresponding author's Email: h3.almazini@gmail.com

Abstract: Modified graph clustering ant colony optimization (MGCACO) algorithm is an unsupervised feature selection (UFS) algorithm used in determining a subset of effective genes from microarray data. The feature subset construction is based on the ant colony optimization (ACO) algorithm, which guides the search process from clusters. However, the MGCACO algorithm is unable to choose all significant features from the clusters to form an optimal feature subset. This paper proposes an enhanced graph clustering ACO (EGCACO) to overcome the problem of feature selection in the MGCACO algorithm. A principal point of this algorithm is utilizing an adaptive selection technique that guides ACO for subset construction from the clusters of features. The adaptive technique for ant selection is based on the state of the search space. Experimental results indicated that the proposed EGCACO achieves the highest classification accuracy than five other common UFS algorithms on four classifiers, where it obtained 87.13%, 86.19%, 87.38% and 90.80% for support vector machine, k-nearest neighbor, decision tree and random forest classifiers, respectively. In particular, the proposed algorithm can select the genes of the deoxyribonucleic acid microarray with consideration of relevance and redundancy among the genes. Therefore, the proposed EGCACO can be implemented to handle the high dimension feature space, such as image processing, text classification, and microarray data processing, which is critical for good and reliable results.

Keywords: Unsupervised feature selection, Clustering, Parameter control, Ant colony optimization, Microarray.

1. Introduction

Feature selection (FS) is a procedure that removes irrelevant, redundant, or noisy data, and detects relevant features with the aim to improve predictive accuracy, increase comprehensibility, and accelerate data mining algorithms [1]. The selection of the features can be classified according to supervised, unsupervised, and semi-supervised [2, 3]. Supervised feature selection is employed when the class labels are available, while unsupervised feature selection (UFS) is used in the absence of class labels. Both unlabeled and labeled data are utilized in semi-supervised FS methods. Unsupervised feature selection is more difficult than semi-supervised and supervised, as it is not assisted by class labels [4]. Many UFS methods have been proposed in recent years by various scholars [5-9]. It has several advantages, such as being impartial as experts or data

analysts are not needed to categorize samples, and it can perform well even though no prior information is available. It is important for the exploratory analysis of biological data (e.g., the advances of deoxyribonucleic acid (DNA) microarray) and it provides an efficient way to discover the unknown meaningful insights into the classification of diseases [10]. The natural structure of DNA microarray data is high-dimensional with few records and many columns where they represent a well-known phenomenon called the “curse of dimensionality” [11]. Nevertheless, the genes (features) in the data are not entirely included for classification purposes as not all genes contain relevant information. Therefore, to achieve reliable, accurate, and effective performance, an effective data preprocessing process such as UFS should be implemented in DNA microarray classification [12]. In spite of this, UFS has several key disadvantages. It neglects the potential correlation between different features, and

therefore, the subsets generated may be suboptimal for the specific task of the discrimination. In addition, it depends on some mathematical principles without ensuring that the principles are universally valid for all data [13].

In recent years, more attention has been paid in particular to approaches based on metaheuristic as a search technique for optimization problems. Among those approaches are genetic algorithm (GA) [14, 15], ant colony optimization (ACO) [16–21], and particle swarm optimization (PSO) [22, 23]. Predominantly in high-dimensionality data, metaheuristic swarm search algorithms are very useful with regard to their global search capability [24]. ACO has been widely used as a metaheuristic swarm search algorithm for FS [25, 26]. In comparison to other swarm intelligence algorithms, ACO has several advantages including the capability of local and global search, long-term distributed memory, and learning reinforcement strategy [27–29]. In dealing with FS, ACO is appropriate for handling high-dimensional, noise, irrelevant, and redundant datasets [30]. Nevertheless, there are several shortcomings in the existing ACO-based algorithms [27]. First, these algorithms involve a complete graph including whole possible feature combinations leading to high computational costs. Second, a learning model is utilized to evaluate the feature subsets that lead to an increase in the complexity of their computations. Third, many of these algorithms may not conduct a redundancy analysis to rule out strongly correlated features that can reduce the efficiency of the classifier. Finally, the effectiveness of such algorithms relies greatly on the selected subset size and the relationship with the learning model. To overcome these issues, feature clustering is utilized as one of the UFS methods that considers the correlation among features. The feature clustering model can reduce the dimensionality in unsupervised high-dimensionality data as a powerful tool in increasing the effectiveness of search for the optimal feature subset. This is achieved by clustering the most associated features in the same groups. Furthermore, the performance of the UFS process will be improved when it uses the information obtained from the clusters.

One of the most popular algorithms that clusters the features is the modified graph clustering ACO (MGCACO), which is an extension from the graph clustering ACO (GCACO) algorithm [20, 22]. GCACO has achieved the highest performance accuracy for microarray dataset as compared to other well-known algorithms. This is due to the fact that the selection process is driven in such a manner that at least one feature is picked from each cluster along each ant search process, and comparatively less

correlated features are injected in a large percentage with regard to more correlated features to the consecutive iteration.

Both GCACO and MGCACO are based on the Louvain algorithm for community detection [32], which is used to detect the communities of the most correlated features by maximizing a modularity function and finding the local maximum. Such algorithms are easy and simple in determining the communities in large networks due to the greedy search method that utilizes modularity maximization as a goal to find the best community [33]. The GCACO algorithm is modified with the aim to improve the performance of GCACO. The new algorithm is called MGCACO and its performance outperforms GCACO and other well-known UFS algorithms. Its main disadvantage is the lack of setting (user-defined) the appropriate threshold value for a parameter that is responsible in guiding the ACO selection of the features from the clusters.

This paper presents a study on a new technique that guides the ants to pick relevant features from the clusters. The proposed technique is constructed in three main phases. In the first phase, the pheromone initialization is performed on two pheromone matrices i.e. for the feature search space and the ϵ value. This aims to reduce the size of the search space and overcomes the searching complexity. The second phase is where the ant selection process that employs both the greedy search method and the probabilistic state transition rules. In each selection process, the ϵ value is reinforced with the appropriate adaptive adjustment. The last phase is the pheromone updating rule for the goodness of the selected ϵ value. This pheromone value becomes the feedback in guiding the search to an optimal ϵ value within the iterations. Thus, the advantage of the proposed algorithm is that the ϵ value can be automatically determined, rather than being determined by the user.

The remainder of this paper is structured as follows. Sections 2 and 3 present the ACO-based UFS algorithms and the proposed method of ACO-based adaptive selection technique, respectively. The data and experimental design are presented in Section 4, while Section 5 displays the results and discussion. Finally, Section 6 summarizes the conclusions and future directions.

2. ACO-based unsupervised feature selection algorithms

Several bio-inspired algorithms based on the paradigm of swarm intelligence for UFS have been reviewed by S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad [34]. The

algorithms are unsupervised feature selection based on ACO (UFSACO) microarray gene selection based on ACO (MGSACO), relevance-redundancy feature selection based on ACO (RRFSACO), and unsupervised probabilistic FS using ACO (UPFS). Swarm intelligence involves a group of artificial agents that attempt to mimic the natural behaviors of a population of animals. Every agent conducts a search mission individually, and handles a complex issue collectively. In general, the most popular algorithms that use metaheuristic search to select features are artificial bee colony [35], PSO [24], ACO [36], grey wolf optimization [37], taboo search [38], and scatter [39]. Nevertheless, on the paradigm of metaheuristic based swarm intelligence, ACO is more flexible in the scope of FS than other algorithms [30, 39]. This is proven by the use of the distributed long-term memory, positive feedback, parallel nature implementation, similar function to the reinforcement learning scheme, and good exploration and exploitation capabilities due to greedy search and stochastic algorithm components [32].

S. Tabakhi, P. Moradi, and F. Akhlaghian [41] introduced one of the first bio-inspired swarm intelligence paradigm known as UFSACO to choose a feature subset with minimum similarity (low redundancy) between features. In this algorithm, a complete undirected graph was used where nodes represented features and the similarities between the features were the weights on the edges. Similar features were considered redundant. The amount of pheromone depended on the feature counter value on the edge of the graph. In each step, an ant built its candidate solution by iteratively selecting the edge with the lowest iterative similarity and the greatest pheromone value. The pheromone value would decrease as it took place based on the intensified pheromone rule. Until a predetermined stopping criterion was reached (iteration number), the ants interactively traversed the graph, preferring low similarities and high pheromone values. Finally, a specific number (specified by user) of features that had high pheromone value and low redundancy was chosen. The UFSACO algorithm was evaluated in terms of the number of selected features and classification accuracy to achieve the final feature subset. Several well-known classical classifiers i.e., random forest, naïve Bayes, decision tree, and support vector machine (SVM) were employed in the performance evaluation of UFSACO on benchmark datasets. The datasets included Arrhythmia, Spambase, Ionosphere, Hepatitis, Wisconsin Diagnostic Breast Cancer, and Wine from the University of California Irvine (UCI) machine learning repository [43], and Madelon dataset, which

was obtained from the NIPS2003 FS challenge [44]. Results showed the superior performance of UFSACO over the other algorithms.

Other algorithms that have a similar approach to UFSACO are MGSACO [36], RRFSACO [45], and UPFS [46], which have also been proposed to solve classification problems that involves UFS. For both RRFSACO and MGSACO, in addition to quantifying the redundancy feature as in UFSACO, each feature's relevance is further measured by its variance. Therefore, the aim of these algorithms is to choose attributes that reduce redundancy while maximizing relevance. The goal is to pick non-redundant attributes in UPFS by utilizing the similarity of cosine less than Pearson's correlation [47].

There are several limitations to the current ACO-based FS algorithms. In the search process, the possibility of dependency among the features is ignored. These ACO algorithms presume that the attributes are conditionally independent. Therefore, once the ant chooses the next attribute, the feature dependency on those already picked is ignored. Consequently, the organized subset might include redundant attributes. To avoid this issue, the feature clustering approach is suggested.

A hybrid UFS algorithm that depends on the Louvain algorithm for community detection [32] for modified binary ant system and clustering combination known as FSCBAS was presented by Z. Manbari, F. Akhlaghian Tab, and C. Salavati [40]. This algorithm dealt with data processing issues in a large dimension search space and offered global search as well as local search capabilities within and between clusters. FSCBAS, inspired by simulated annealing and genetic algorithm, introduced a novel redundancy rule for reduction to evaluate the correlation among chosen attributes and a damped mutation technique that prevented falling into a local optimum. However, its performance had low accuracy in terms of the number of chosen attributes from each dataset.

The GCACO algorithm proposed by P. Moradi and M. Rostami [27] worked on the attributes represented as a graph with features (nodes) and feature-similarities (edges). Then, the community detection method was applied to cluster features (nodes) [32] and select a minimum redundancy attribute subset between the attributes collected from the clusters. Community detection in the weighted graph was significant for the comprehension of the graph structures and analysis of its properties. In the FS field of study, the purpose of community detection is to cluster identically strong and correlated attributes together in the same group [31]. This method determined the communities within the graph

by maximizing a task of modularity. This method was convenient and effective to classify communities within large networks. The method detected communities in two stages of a network. Every node in the first step was assigned to a community selection based on the greedy search operation to maximize the specific modularity of the network. The second step simply created a new network by merging those of the communities previously found. The process then iterated until a significant improvement of the modularity of the network was achieved and the features were then represented in a complete graph in each cluster [27]. The ACO-based FS algorithm was applied to create the feature subset from the clusters. In every iteration, an ant randomly traveled to a cluster and chose at least one attribute from the cluster depending on the predefined threshold value that guided the ant to choose the next feature from a different cluster; otherwise, the ant would remain in the current cluster. The efficiency of every solution (i.e., a set of features) was estimated by implementing a separability index once all ants traversed completely on the graph. The value of pheromone of every node was adjusted by utilizing the intensified pheromone rule. Finally, the feature with the highest pheromone would become a candidate in the final subset. The result of GCACO outperformed other well-known algorithms in terms of the number of feature subset and classification accuracy. This was because every ant in the graph cluster attempted to travel among features for minimum similarity and maximum target class dependence.

Better performance than GCACO was obtained in a modified version called MGCACO, which was proposed by H. Ghimatgar, K. Kazemi, M. S. Helfroush, and A. Aarabi [31] with three changes. The first was the pheromone initialization that had been improved by considering the relevance of features to classes. Higher priority was given to more relevant features. Secondly, depending on the multiple discriminant analysis, MGCACO used an evaluation function that could effectively evaluate redundancy and the relevance of features. Finally, MGCACO utilized a better efficient cost mission, integrating redundancy and the relevance of features in a highly efficient manner, where features were sorted based on redundancy and relevance analyses. The MGCACO algorithm could address irrelevant as well as redundant features. However, the MGCACO algorithm showed sensitivity on determining the value of the parameter because a manual setting was used to set the appropriate value, which was between [0,1] to guide the ACO subset construction before execution [27, 31]. The value was considered very

critical and highly data-dependent. Users might not consider the relevance, dependency, and homogeneity among the features in the search spaces. Therefore, a fixed value might reduce the quality of the feature subset selection in MGCACO.

3. The proposed algorithm

The proposed algorithm is an enhancement of MGCACO, called EGCACO in which a new approach is used for the ants in selecting an appropriate value to guide the subset construction. This is in contrast to the MGCACO algorithm where a fixed threshold value is used.

The flowchart of EGCACO is shown in Fig. 1 where the main processes are graph representation, feature clustering, and ACO-based selection. The enhancements in MGCACO are highlighted in the figure, which are pheromone initialization, ant selection, and pheromone updating rule. Significant features would be chosen by the ants based on the value of a parameter, ϵ , to be included in the feature subset. In general, the flow of EGCACO began with the pheromone initialization of a vector each for the ϵ values and the feature search space. This was then followed by the ant selection execution where each ant obtained an ϵ value to select the features by using both probabilistic state transition rules and greedy method. Finally, the pheromone updating rule was applied for both the selected ϵ value and the selected features.

EGCACO followed the procedures in the ACO algorithm [48] for pheromone initialization, ant selection, and pheromone updating rule (feedback collection). A state transition probability function was used to select the ϵ value based on the amount of pheromone and heuristic. Stochastically, the selection of a single parameter value was conducted by ACO. Subsequently, during the learning process of the proposed algorithm, the quality collected for each discovered feature subset and the quality of the significant feature subset were determined as feedback collection. Then, to guide the search process, these feedback indicators were transformed into reward assignments. To determine whether the current ϵ value was dominated or not, the transition probability of ACO was utilized. The process was repeated in the learning process of the proposed algorithm. Pheromone initialization, ant selection, and pheromone updating rule are comprised in the ACO-based selection process. The three steps are described in the following flow chart.

Pheromone initialization: To establish the feature selection process, the pheromone initialization

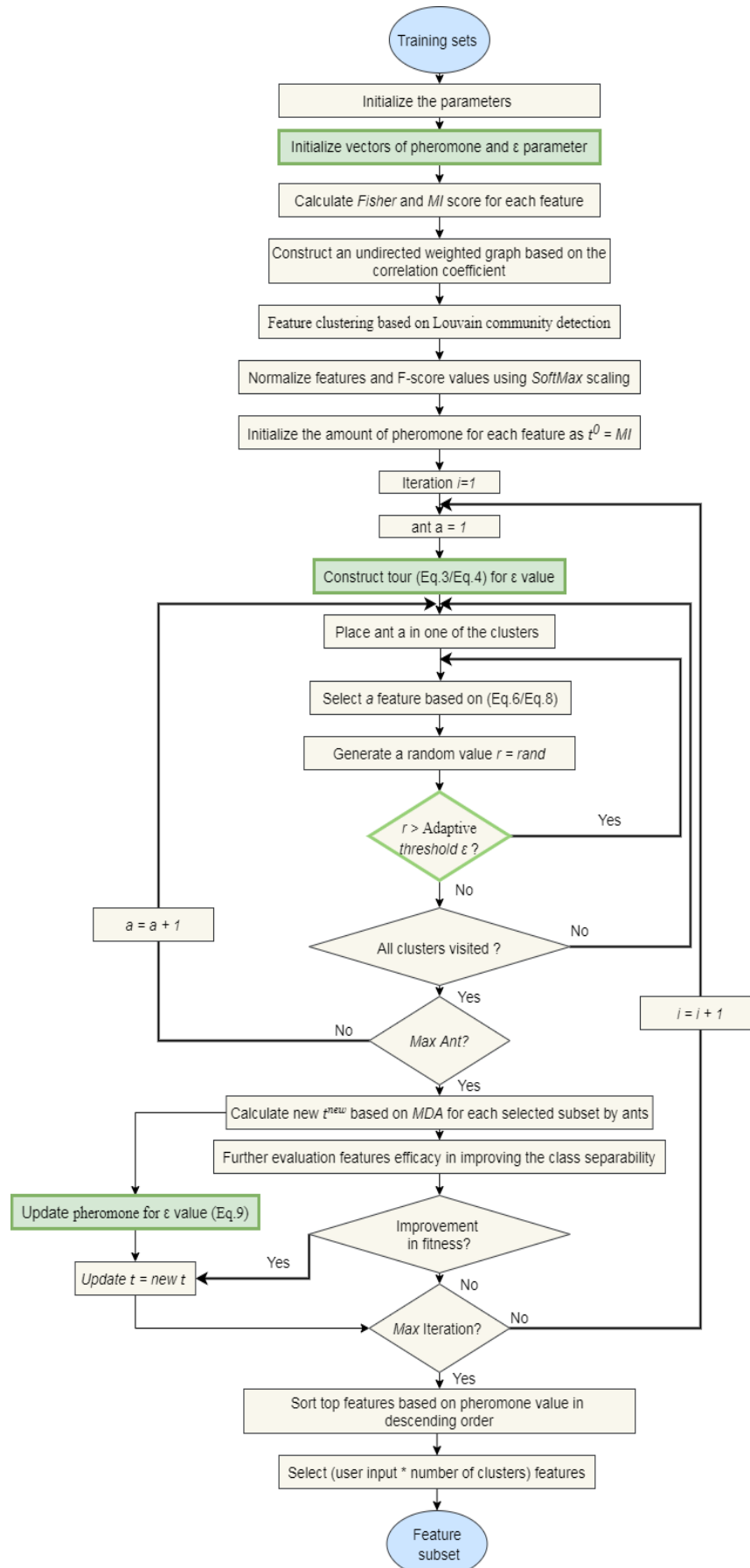


Figure.1 Flow chart of the EGCACO algorithm

required two vectors to implement the feedback collection of EGCACO. The first vector represented the quality of the threshold parameter values, referred to as $\tau p_n = \{\tau p_1, \tau p_2, \tau p_3, \dots, \tau p_n\}$, which reflected the different ε values that were initiated using the proposed equation.

$$\tau p_n = \varnothing \quad (1)$$

where \varnothing is a constant number at time $t = 0$ that can be set to any value in the range $[0,1]$. The second vector denoted the quality amount assigned to each feature and was represented as $\tau f_n = \{\tau f_1, \tau f_2, \tau f_3, \dots, \tau f_n\}$. At time $t = 0$, the pheromone initialization was determined by performing the relevance of attributes to classes. Higher priority was given to more relevant features. The mutual information among any feature (f_d) and classes was computed for this purpose as [31]:

$$MI(d) = \sum_m \sum_{k=1}^c p(f_d[m].k) \log \frac{p(f_d[m].p(k))}{p(f_d[m].k)} \quad (2)$$

where $f_d[m]$ is the discretizing value of the d th feature for bin $m = [1, 2, \dots, M]$, in which M (number of bins) is set to 10 in the MCGACO algorithm. The probability of the k th class is $p(k)$ and the number of classes is k .

Ant selection: When the ACO algorithm began to solve the problem, two selection stages were implemented. In the first stage, each ant selected a parameter (node) that represented its direction in time t , which was determined by the proposed probability Equations (3) and (4).

$$P_{(an)} = \begin{cases} \arg \max_{an \in P} & \text{if } q < q_0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where $P_{(an)}$ is denoted as the probability for each parameter value and $P_{(an)} = \{a_1, a_2, a_3, \dots, a_n\}$ determines the selection probability of each ε value. q is the parameter in the range $[0, 1]$. The predefined constant, q_0 , is a random number. In the probabilistic method, the ant selected the next parameter with a new probability of P_n calculated as follows:

$$P_n = \begin{cases} \frac{[\tau p_{n(t)}]^\alpha [\pi_n]^\beta}{\sum_{n=1}^L [\tau p_{n(t)}]^\alpha [\pi_n]^\beta} & \text{if } q > q_0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where α and β represent the significant predefined constant parameter value in the range $[0, 1]$ of both the pheromone and the heuristic information, respectively. $[\tau p_{n(t)}]$ is the quantity of quality correlated with each threshold value, while $[\pi_n]$ indicates the heuristic information that is calculated for this purpose as:

$$\pi_n = \frac{1}{P_{(vn)}} \quad (5)$$

where $P_{(vn)}$ indicates the threshold value for each parameter and $P_{(vn)} = \{v_1, v, v_3, \dots, v_n\}$ determines the initialization of each ε value between $[0,1]$. Thus, according to Equation (4), one of the parameters was selected corresponding to its probability value by using roulette wheel selection.

In the second stage, each ant began to construct the feature subset from a group of clusters that were generated based on the Louvain algorithm for community detection [32]. Each cluster contained the most correlated normalized features from the training dataset. Thus, in every iteration, in a fully connected undirected graph, a j th ant randomly selected at least one attribute from the cluster in its path. Then, by using probabilistic decision rules that depended on both probability and greedy method with the dependence on the selected parameter value (obtained from the first stage), the ant would directly pick the next feature from a different cluster; otherwise, the ant would remain in the current cluster. If q_0 was less than q , then the j th ant would pick the next feature by following the greedy method as [27]:

$$f_{next} = \arg \max_{f_d \in Uf_m^j} \{ \tau f^i(f_d)^\alpha \cdot \pi(f_d, Vf_j)^\beta \} \quad (6)$$

where Uf_m^j is the set of features that has not yet been reached by the j th ant in the existing cluster (m th cluster), τf^i is the pheromone quantity value correlated with feature (f_d), Vf_j indicates the previously selected features (visited features), and $\pi(f_d, Vf_j)$ denotes the heuristic information function. A modified heuristic information function was introduced in MGCACO [31], which is defined as follows:

$$\pi(f_d, Vf_j) = Fscore(f_d) = \frac{1}{\frac{1}{N_{Vf_j}} \sum_{m=1}^{N_{Vf_j}} w(f_d, f_m)} \quad (7)$$

Table 1. Characteristics of the datasets

Name of Dataset	Area Type	No. of Features	No. of Patterns	No. of Classes	Features	
					Values	
					Categorical	Numerical
Wine	Physical	13	178	3	-	√
Hepatitis	Life	19	155	2	√	√
Ionosphere	Physical	34	351	2	-	√
Spambase	Computer	57	4601	2	-	√
Arrhythmia	Life	279	452	16	√	√
Madelon	Artificial	500	4400	2	-	√
Colon	Microarray	2000	62	2	-	√
Leukemia	Microarray	7129	72	2	-	√

Table 2. Confusion Matrices

Actual	Predicted	
	Normal	Abnormal
Abnormal	FN	TP
Normal	TN	FP

Table 3. Parameter settings

Parameter	Description	Value
A	Ants number	100
I	Iterations	50
ρ	Evaporation coefficient	0.7
ϕ	Initial amount of pheromone	0.06
α	Importance of the pheromone parameter	0.8
β	Heuristic information parameter	0.2
q_0	Exploration/exploitation parameter	0.7

where $Fscore(f_d)$ denotes the score of feature f_d based on its relevance to the classes [49], N_{Vf_j} is the size of Vf , $w(f_d, f_m)$ indicates the values of the Pearson’s correlation [47] among feature f_d and feature f_m that are obtained from all the features (Vf_j) visited by the j th ant from all priors clusters. Therefore, in the greedy method, the features picked by the ants were those with maximum dependence and minimum parity to the features already selected on the target class. If q_0 was larger than q , a probabilistic method for each of the features not

visited yet in the current cluster ($f_d \in Uf_m^j$) is defined as follows [27]:

$$P(f_d) = \frac{[\tau f^i(f_d)]^\alpha [\pi(f_d, Vf_j)]^\beta}{\sum_{f_d \in Uf_m^j} [\tau f^i(f_d)]^\alpha [\pi(f_d, Vf_j)]^\beta} \text{ For } f_d \in Uf_m^j \quad (8)$$

According to the roulette wheel rule, the next feature would be chosen.

Pheromone updating rule: Once all the ants had executed their routes on the graph, and before ending each iteration i , the quantity of pheromone of each selected parameter and feature (*i.e.*, $\tau p_{n(t)}, \tau f^i(f_d)$) was updated in two stages. In the first stage, the proposed updating rule of the selected parameters is calculated as:

$$\tau p_{x(t+1)} = (1 - \rho) \cdot \tau p_{x(t)} + \sum_{k=1}^A \Delta_k^i(t) \quad (9)$$

where the ρ value indicates the pheromone decay parameter to escape infinite accumulation with a particular parameter value, $\tau p_{x(t)}$ and $\tau p_{x(t+1)}$ demonstrate the pheromone amounts on parameter $P_{(n)}$ at times (t) and ($t + 1$), respectively. A is the number of ants, $\Delta_k^i \tau p_{x(t)}$ is the additional pheromone boost to parameter $P_{(n)}$ by ant k , which is computed from the use of multiple discriminant analysis (MDA) [50] as follows:

$$\Delta_k^i(t) = \frac{\text{argmax}_{t \in MDA_{k=1}^A}}{MDA_n} \quad (10)$$

In the second stage, the pheromone updating of the selected features was performed as follows [31]:

$$\tau f^{i+1}(f_d) = (1 - \rho) \cdot \tau f^i(f_d) + \sum_{j=1}^A \Delta_j^i(f_d) \quad (11)$$

$$\Delta_j^i(f_d) = \begin{cases} \gamma_j^i, & f_d \in fS_j^i \\ 0, & f_d \notin fS_j^i \end{cases} \quad (12)$$

where fS_j^i is the feature chosen in the i th iteration by the j th ant, γ_j^i is denoted as MDA [50] identical in the i th iteration to the j th chosen subset. Suppose N_{fs} features should be selected. At each iteration, all the features were sorted using on Equation (13) [31]. N_{fs} features were chosen and the MDA values identical to these features were computed using Equation (14) [31]. If the MDA values were enhanced relative to the previous iteration, the pheromone values were permitted to be updated; otherwise, no change would be made.

$$Sf(d) = \tau f(d) \cdot Fscore(f_d) \quad d = 1, 2, \dots, D \quad (13)$$

$$Fitness = MDA(Feature(1, \dots, N_{fs})) \quad (14)$$

The processes (ant selection, pheromone updating rule) were repeated until the iterations were terminated. Finally, in representing the final feature subset, the features were sorted in descending order based on their pheromone values. The number of features that would be chosen was determined by multiplying a predetermined number of features, ω , with the number of clusters, K .

4. Data and experimental design

Experiments were conducted on different benchmark classification problems to demonstrate the credibility of EGACO. The datasets used were Arrhythmia, Spambase, Ionosphere, Hepatitis, and Wine, which are available in the UCI machine learning repository [43], and three other datasets, i.e., Madelon, Colon, and Leukemia. The Madelon dataset was obtained from the NIPS2003 FS challenge [44], while the Colon and Leukemia datasets were obtained from the Bioinformatics

Research Group at the Universidad Pablo de Olavide [51]. Table 1 summarizes the characteristics of those datasets. A non-linear method of normalization, i.e., SoftMax scaling [52], was utilized to scale attributes with extreme data values. The datasets varied in terms of the number of features/attributes (range of 13–7129), records, class labels, application area, and types of feature values. The size of the dataset was categorized based on the number of features into four categories: small, medium, large, and very large [53]. A quarter of the datasets were classified as small with sizes in the range of 0–20 features, while 13% of the datasets were in the medium category with the number of features in the range 20–50. The large datasets comprised 13% of the total datasets with feature sizes ranging from 50 to 100. The remaining datasets (50% of the total number) were very large datasets, containing 100 and more features. In terms of missing values, only the Hepatitis, Spambase, and Arrhythmia datasets had 175, 280, and 350 cells with missing values, respectively. The missing values were replaced with the mean values of each respective features. All datasets were then normalized. Each dataset was divided into two sets, where two-thirds were for training and one-third was for testing. The training set was utilized to determine the feature subset, while the testing set was to evaluate the accuracy of the selected features.

For each algorithm, ten independent runs were performed. The proposed algorithm was compared with other well-known ACO-based UFS algorithms (FSCBAS, MGCACO, GCACO, UFSACO, and MGSACO), where the filter method had been implemented in all the algorithms. To show the generality of the proposed algorithm, several well-known classifiers including random forest, k-nearest neighbor (k-NN), decision tree, and SVM were used. These classifiers were implemented in the WEKA [54] software package. The metric for comparison was the generated feature subsets accuracy, AC, computed using Equation (15) as follows:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \quad (15)$$

where TN, TP, FN, and FP are the confusion matrices that describe the classification results (true or false).

- TN: Correctly classified as normal
- FN: Intrusions that are classified as normal
- TP: Correctly classified as intrusion
- FP: Normal behavior but classified as intrusion

Table 2 shows the possible cases in identifying the cases.

Table 4. Average classification accuracy using SVM classifier

Dataset	# Selected features		EGCACO	FSCBAS [40]	MGCACO [31]	GCACO [27]	MGSACO [36]	UFSACO [41]
Wine	6	Acc Std	98.69 (1) ± 1.40	97.2 (3) ± 0	98 (2) ± 1.72	95.1 (5) ± 2.6	94.61 (6) ± 2.52	96.02 (4) ± 2.04
Hepatitis	6	Acc Std	78.85 (6) ± 2.10	81.55 (3) ± 3.74	83.85 (1) ± 4.82	83.64 (2) ± 2.83	81.1 (4) ± 1.13	80.78 (5) ± 0.4
Ionosphere	15	Acc Std	89.15 (1) ± 3.10	85.62 (4) ± 3.57	86.24 (3) ± 2.83	86.79 (2) ± 2.41	81.6 (6) ± 2.09	84.96 (5) ± 2.12
SpamBase	24	Acc Std	88.22 (2) ± 0.80	85.78 (3) ± 0.22	90.98 (1) ± 0.90	84.51 (4) ± 2.12	81.86 (6) ± 2.67	83 (5) ± 2.67
Arrhythmia	20	Acc Std	86.29 (1) ± 1.10	68 (3) ± 3.63	70.84 (2) ± 3.83	62 (4) ± 5.2	56.55 (5) ± 1.4	54.39 (6) ± 0.29
Madelon	40	Acc Std	64.73 (1) ± 3.32	61.2 (3) ± 2.07	58.98 (6) ± 2.83	64.61 (2) ± 5.58	60.98 (4) ± 0.27	60.75 (5) ± 0.24
Colon	40	Acc Std	92.38 (1) ± 4.85	87.59 (2) ± 1.31	84.76 (3) ± 6.31	81.99 (6) ± 2.23	83.88 (4) ± 2.94	82.1 (5) ± 3.26
Leukemia	40	Acc Std	98.75 (1) ± 1.90	95.42 (3) ± 0.67	95.83 (2) ± 3.22	93.67 (4) ± 2.35	90.14 (5) ± 5.17	89.45 (6) ± 3.28

Table 5. Average classification accuracy using K-NN classifier

Dataset	# Selected features		EGCACO	FSCBAS [40]	MGCACO [31]	GCACO [27]	MGSACO [36]	UFSACO [41]
Wine	6	Acc Std	97.38 (2) ± 1.82	97.2 (3) ± 0	98 (1) ± 2.02	95.68 (5) ± 2.78	94.61 (6) ± 2.52	96.02 (4) ± 2.04
Hepatitis	6	Acc Std	76.35 (6) ± 3.65	79.36 (5) ± 3.74	82.12 (1) ± 6.97	81.36 (2) ± 1.33	81.1 (3) ± 1.13	80.78 (4) ± 0.41
Ionosphere	15	Acc Std	90.84 (1) ± 2.00	85.62 (3) ± 3.75	86.5 (2) ± 2.24	85.36 (4) ± 4.14	81.6 (6) ± 2.1	84.96 (5) ± 2.12
SpamBase	24	Acc Std	89.37 (2) ± 0.70	85.78 (5) ± 0.22	89.6 (1) ± 1.26	87.04 (4) ± 1.32	81.86 (6) ± 2.68	88.33 (3) ± 0.68
Arrhythmia	20	Acc Std	81.56 (1) ± 3.66	57.31 (3) ± 1.33	64.07 (2) ± 3.14	53.75 (5) ± 5.38	55.25 (4) ± 1.84	50.87 (6) ± 1.2
Madelon	40	Acc Std	65.54 (4) ± 2.50	59.8 (6) ± 1.34	60.94 (5) ± 2.56	76.21 (1) ± 1.74	75.85 (2) ± 0.87	75.13 (3) ± 0.4
Colon	40	Acc Std	91.42 (1) ± 3.01	87.59 (2) ± 2.52	81.43 (4) ± 5.01	81.13 (5) ± 4.32	82.1 (3) ± 2.79	80.65 (6) ± 2.28
Leukemia	40	Acc Std	97.08 (1) ± 1.90	95.42 (2) ± 1.64	93.75 (3) ± 9.63	88.02 (5) ± 3.11	88.06 (4) ± 2.27	87.92 (6) ± 1.45

Experiments were carried out on a computer using MATLAB 2020a with a 3.4 GHz CPU and 8.00 GB of RAM. For a fair evaluation of the results, the parameter settings were adopted from the ACO-based UFS algorithms [27, 31, 39] as shown in Table 3.

5. Result and discussion

The main aim of the EGCACO based UFS is to maximize dependency and minimize redundancy from the datasets, this can be measured using the accuracy as a performance metric since increased in accuracy reflects decrease in computational cost [54]. Thus, for the performance of the EGCACO algorithm, the classification accuracy is used as an

evaluation measure in comparing the achievement of the algorithm. Tables 4–7 show the average classification accuracy (Acc) and the standard deviation (Std) of EGCACO and five other UFS algorithms with SVM, decision tree, k-NN, and random forest classifiers, respectively. The best result was highlighted for each dataset while the figures in parentheses indicated the rank of the algorithms.

Table 4 shows that the proposed EGCACO obtained the best result in six out of eight datasets when the SVM classifier was used. The proposed algorithm was unable to obtain the best classification accuracy with datasets that had missing values replaced and datasets that contained a combination of feature value types. In Table 5, EGCACO obtained

the best classification accuracy results on four datasets when the k-NN classifier was employed. K-NN classifier is known to work well with numerical data. This was the case for the Ionosphere, Arrhythmia, Colon, and Leukemia datasets. In spite of this, EGCACO was able to select significant features in the Arrhythmia dataset, which contained both types of data.

Table 7 displays the classification accuracy for the algorithms using decision tree classifier. The proposed algorithm obtained the highest classification accuracy in six out of eight datasets. Decision tree classifier can work with both types of feature values. The proposed algorithm was able to select significant features to produce good

classification accuracy even for databases with only continuous feature values such as the Colon database. The results in Table 8 showed that for the case of the random forest classifier, EGCACO was able to gain the highest accuracy in all the datasets except for the Wine dataset, which obtained the second rank with a very small percentage difference. Random forest classifier is known to operate well with datasets that have categorical data. Nevertheless, in this case, the proposed algorithm was able to select significant features for classification even for datasets that had both categorical and numerical data types. In all the experiments, EGCACO was able to obtain the best classification accuracy for large-sized datasets such

Table 7. Average classification accuracy using decision tree classifier

Dataset	# Selected features		EGCACO	FSCBAS [40]	MGCACO [31]	GCACO [27]	MGSACO [36]	UFSACO [41]
Wine	6	Acc Std	95.74 (1) ± 2.22	93.83 (2) ± 0	93.17 (4) ± 3.37	93.57 (3) ± 2.98	92.93 (6) ± 3.47	93.09 (5) ± 3.02
Hepatitis	6	Acc Std	75.19 (6) ± 4.95	79.36 (3) ± 0	78.08 (4) ± 6.71	80.31 (2) ± 1.14	80.26 (1) ± 0.92	77.04 (5) ± 1.22
Ionosphere	15	Acc Std	91.51 (1) ± 2.94	88.32 (5) ± 1.41	90.86 (2) ± 4.21	90.54 (3) ± 1.98	86.7 (6) ± 2.14	89.06 (4) ± 1.51
SpamBase	24	Acc Std	92.51 (1) ± 0.71	89.68 (4) ± 0.22	90.88 (2) ± 0.66	89.12 (5) ± 1.12	89.83 (3) ± 1.52	89.08 (6) ± 0.51
Arrhythmia	20	Acc Std	87.14 (1) ± 1.44	59.76 (3) ± 1.37	66.71 (2) ± 3.22	56.62 (4) ± 4.13	49.94 (6) ± 2.05	50.89 (5) ± 2.32
Madelon	40	Acc Std	69.45 (5) ± 4.40	67.4 (6) ± 1.25	71.42 (4) ± 3.50	81.77 (1) ± 2.38	80.12 (2) ± 0.57	79.49 (3) ± 0.67
Colon	40	Acc Std	90.48 (1) ± 5.32	86.13 (2) ± 0.83	78.09 (6) ± 7.73	84.23 (4) ± 2.39	85.17 (3) ± 2.82	82.91 (5) ± 4.18
Leukemia	40	Acc Std	97.08 (1) ± 2.76	93.62 (2) ± 0	90.83 (3) ± 3.63	85.74 (5) ± 1.52	90.28 (4) ± 4.29	77.92 (6) ± 2.96

Table 8. Average classification accuracy using random forest classifier

Dataset	# Selected features		EGCACO	FSCBAS [40]	MGCACO [31]	GCACO [27]	MGSACO [36]	UFSACO [41]
Wine	6	Acc Std	97.35 (2) ± 1.50	97.2 (3) ± 0	97.5 (1) ± 2.53	96.11 (4) ± 2.32	94.61 (6) ± 2.53	96.02 (5) ± 2.04
Hepatitis	6	Acc Std	86.57 (1) ± 2.40	81.39 (3) ± 0	84.43 (2) ± 4.30	81.3 (4) ± 1.76	81.1 (5) ± 1.14	80.78 (6) ± 0.41
Ionosphere	15	Acc Std	94.03 (1) ± 1.85	85.77 (3) ± 3.57	93.59 (2) ± 1.77	85.1 (4) ± 2.41	81.6 (6) ± 2.1	84.95 (5) ± 2.12
SpamBase	24	Acc Std	94.74 (1) ± 0.75	85.78 (5) ± 0.23	94.17 (2) ± 0.75	89.02 (4) ± 1.98	81.86 (6) ± 2.68	89.96 (3) ± 0.54
Arrhythmia	20	Acc Std	83.57 (1) ± 3.32	66.38 (3) ± 1.33	75.36 (2) ± 2.24	61.18 (4) ± 2.22	57.22 (5) ± 1.6	50.05 (6) ± 2.91
Madelon	40	Acc Std	83.31 (1) ± 1.28	61.19 (6) ± 1.63	76.51 (4) ± 2.75	72.27 (5) ± 1.72	83.29 (2) ± 0.48	83.24 (3) ± 0.49
Colon	40	Acc Std	88.09 (1) ± 4.87	87.59 (2) ± 2.53	84.29 (3) ± 4.28	81.22 (6) ± 3.95	82.91 (4) ± 2.18	81.78 (5) ± 4.93
Leukemia	40	Acc Std	98.75 (1) ± 2.66	95.42 (3) ± 0.67	98.75 (2) ± 1.90	89.69 (4) ± 3.61	89.36 (5) ± 5.16	86.12 (6) ± 3.53

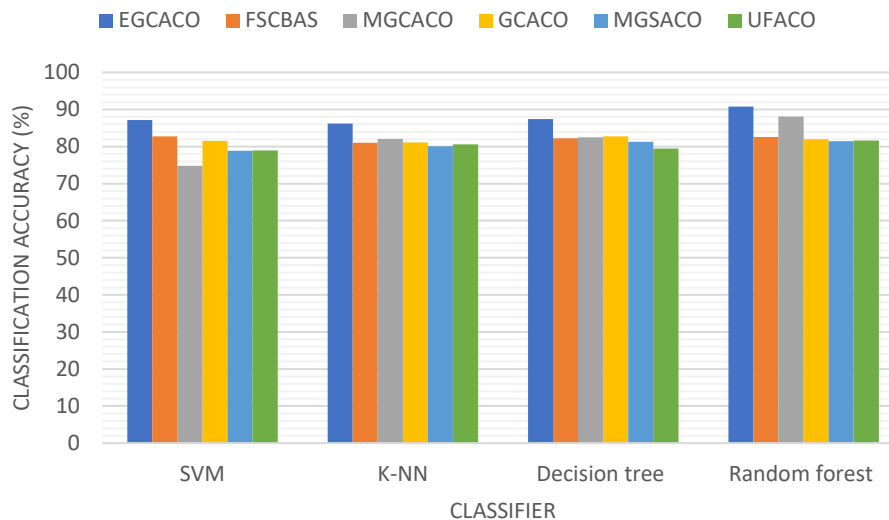


Figure.3 Average classification accuracy of all data

as microarray datasets (i.e., Colon and Leukemia) when all the classifiers were used. Fig. 3 depicts the summary of accuracy for the algorithms with respect to the classifiers. Overall, it can be seen EGCACO showed superior performance. The ACO-based adaptive selection technique had the ability to select the appropriate ϵ value to be used in selecting significant features. Therefore, the ACO-based UFS process considered the dependency of the chosen features and chose a feature subset with reduced redundancy between them. Thus, the best results obtained by EGCACO were reported for all classifiers.

6. Conclusion

The proposed EGCACO algorithm introduced an ACO-based adaptive feature selection technique that can maximize dependency and minimize redundancy among the features. The proposed technique is constructed with three main aspects/traits. The first aspect is pheromone initialization, where two pheromone matrices are initialized for the ϵ value and the feature search space, respectively. This overcomes the searching complexity where the search space is reduced in size. The second aspect is the ant selection procedure, which utilizes both probabilistic state transition rules and greedy search method. This aspect clearly reinforces the appropriate dynamic adjustment of the ϵ value in each selection. The last aspect is the pheromone updating rule for the quality of the selected ϵ value. This enables the search of an optimal ϵ value within the iterations. For most related ACO-based UFS methods, the evaluation outcome demonstrated that EGCACO is able to enhance classification accuracy with the proposed

ACO-based adaptive selection of the ϵ value, which depends on the feedback from the search behavior. This has enabled the proposed EGCACO to obtain the best classification accuracy compared to five other popular UFS algorithms for all four classifiers. Percentages of the best accuracies are 87.13%, 86.19%, 87.38%, and 90.80% for support vector machine, k-nearest neighbor, decision tree, and random forest classifiers, respectively. Future research direction is to focus on the adaptive adjustment of several parameters that are related to the ACO-based UFS algorithm.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The draft has been prepared by the 1st author while the review and editing has performed by the 2nd author.

Acknowledgments

The researchers thank the Malaysian Ministry of Higher Education for financially supporting this research under Grant Scheme (TRGS /1/2018/UUM/02/3/3 (S/O code14163).

References

- [1] V. Pavithra and V. Jayalakshmi, "Review of Feature Selection Techniques for Predicting Diseases", In: *Proc. of 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 1213–1217, 2020.

- [2] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective", *Neurocomputing*, Vol. 300, pp. 70–79, 2018, doi: 10.1016/j.neucom.2017.11.077.
- [3] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering", *IEEE Trans. Knowl. Data Eng.*, Vol. 17, No. 4, pp. 491–502, 2005, doi: 10.1109/TKDE.2005.66.
- [4] Z. Manbari, F. A. Tab, and C. Salavati, "Fast unsupervised feature selection based on the improved binary ant system and mutation strategy", *Neural Comput. Appl.*, Vol. 31, No. 9, pp. 4963–4982, 2019, doi: 10.1007/s00521-018-03991-z.
- [5] X. He, D. Cai, and P. Niyogi, "Laplacian Score for feature selection", *Adv. Neural Inf. Process. Syst.*, pp. 507–514, 2005.
- [6] Y. Jiang and J. Ren, "Eigenvalue sensitive feature selection", In: *Proc. of 28th Int. Conf. Mach. Learn. ICML*, No. August, pp. 89–96, 2011.
- [7] P. Padungweang, C. Lursinsap, and K. Sunat, "A discrimination analysis for unsupervised feature selection via optic diffraction principle", *IEEE Trans. Neural Networks Learn. Syst.*, Vol. 23, No. 10, pp. 1587–1600, 2012, doi: 10.1109/TNNLS.2012.2208269.
- [8] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou "ℓ_{2,1}-Norm regularized discriminative feature selection for unsupervised learning", In: *Proc. IJCAI Int. Jt. Conf. Artif. Intell.*, pp. 1589–1594, 2011, doi: 10.5591/978-1-57735-516-8/IJCAI11-267.
- [9] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning", In: *Proc. of ACM Int. Conf. Proceeding Ser.*, Vol. 227, pp. 1151–1157, 2007, doi: 10.1145/1273496.1273641.
- [10] R. Xu, S. Damelin, B. Nadler, and D. C. Wunsch, "Clustering of high-dimensional gene expression data with feature filtering methods and diffusion maps", In: *Proc. of 1st Int. Conf. Biomed. Eng. Informatics, BMEI 2008*, Vol. 1, No. May, pp. 245–249, 2008, doi: 10.1109/BMEI.2008.256.
- [11] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Fourth Edition, Academic Press, 4 edition, 2008..
- [12] C. Lazar, J. Taminou, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaezen, R. Duque, H. Bersini, and A. Nowe, "A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis", *EEE/ACM Transactions on Computational Biology and Bioinformatics*, Vol. 9, No. 4, pp. 1106–1119, 2012, doi: 10.1109/TCBB.2012.33.
- [13] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for Multi-Cluster data", In: *Proc. of ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 333–342, 2010, doi: 10.1145/1835804.1835848.
- [14] H. N. K. Al-Behadili, K. R. Ku-Mahamud, and R. Sagban, "Hybrid ant colony optimization and genetic algorithm for rule induction", *J. Comput. Sci.*, Vol. 16, No. 7, pp. 1019–1028, 2020.
- [15] I. S. Oh, J. S. Lee, and B. R. Moon, "Hybrid genetic algorithms for feature selection", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 26, No. 11, pp. 1424–1437, 2004, doi: 10.1109/TPAMI.2004.105.
- [16] A. M. Jabbar, K. R. Ku-mahamud, and R. Sagban, "An improved ACS algorithm for data clustering", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 17, No. 3, pp. 1506-1515, 2020.
- [17] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Modified ACS centroid memory for data clustering", *J. Comput. Sci.*, Vol. 15, No. 10, pp. 1439–1449, 2019.
- [18] H. N. K. Al-behadili, K. R. Ku-mahamud, and R. Sagban, "HYBRID ANT COLONY OPTIMIZATION AND ITERATED LOCAL SEARCH FOR RULES-BASED CLASSIFICATION", *J. Theor. Appl. Inf. Technol.*, 98, No. 04, pp.657-671, 2020.
- [19] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Ant-based sorting and ACO-based clustering approaches: A review", In: *Proc. of IEEE Symp. Comput. Appl. Ind. Electron.*, No. July, pp. 217–223, 2018, doi: 10.1109/ISCAIE.2018.8405473.
- [20] J. Wahid and H. F. A. Al-Mazini, "Classification of cervical cancer using ant-miner for medical expertise knowledge management", In: *Proc. of International Conf. (KMICe)*, pp. 25–27, 2018.
- [21] Y. Wan, M. Wang, Z. Ye, and X. Lai, "A feature selection method based on modified binary coded ant colony optimization algorithm", *Appl. Soft Comput. J.*, Vol. 49, pp. 248–258, 2016, doi: 10.1016/j.asoc.2016.08.011.
- [22] L. Y. Chuang, S. W. Tsai, and C. H. Yang, "Improved binary particle swarm optimization using catfish effect for feature selection", *Expert Syst. Appl.*, Vol. 38, No. 10, pp. 12699–12707, 2011, doi: 10.1016/j.eswa.2011.04.057.
- [23] L. Y. Chuang, C. H. Yang, and J. C. Li, "Chaotic maps based on binary particle swarm

- optimization for feature selection”, *Appl. Soft Comput. J.*, Vol. 11, No. 1, pp. 239–248, 2011, doi: 10.1016/j.asoc.2009.11.014.
- [24] M. Amoozegar and B. Minaei-Bidgoli, “Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism”, *Expert Syst. Appl.*, Vol. 113, pp. 499–514, 2018, doi: 10.1016/j.eswa.2018.07.013.
- [25] H. B. Alwan and K. R. Ku-Mahamud, “Mixed-variable ant colony optimisation algorithm for feature subset selection and tuning support vector machine parameter”, *Int. J. bio-inspired Comput.*, Vol. 9, No. 1, pp. 53–63, 2017.
- [26] A. K. Shukla, D. Tripathi, B. R. Reddy, and D. Chandramohan, “A study on metaheuristics approaches for gene selection in microarray data: algorithms, applications and open challenges”, *EVol. Intell.*, No. 0123456789, 2019, doi: 10.1007/s12065-019-00306-6.
- [27] P. Moradi and M. Rostami, “Integration of graph clustering with ant colony optimization for feature selection”, *Knowledge-Based Syst.*, Vol. 84, pp. 144–161, 2015, doi: 10.1016/j.knosys.2015.04.007.
- [28] H. N. K. Al-Behadili, R. Sagban, and K. R. Ku-Mahamud, “Adaptive parameter control strategy for ant-miner classification algorithm”, *Indones. J. Electr. Eng. Informatics*, Vol. 8, No. 1, pp. 149–162, 2020.
- [29] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, “Annealing strategy for an enhance rule pruning technique in ACO-based rule classification”, *Indones. J. Electr. Eng. Comput. Sci.*, Vol. 16, No. 3, pp. 1499–1507, 2019.
- [30] T. M. Fahrudin, I. Syarif, and A. R. Barakbah, “Ant colony algorithm for feature selection on microarray datasets”, In: *Proc. of Int. Electron. Symp. IES 2016*, pp. 351–356, 2016, doi: 10.1109/ELECSYM.2016.7861030.
- [31] H. Ghimatgar, K. Kazemi, M. S. Helfroush, and A. Aarabi, “An improved feature selection algorithm based on graph clustering and ant colony optimization”, *Knowledge-Based Syst.*, Vol. 159, pp. 270–285, 2018, doi: 10.1016/j.knosys.2018.06.025.
- [32] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, “Fast unfolding of communities in large networks”, *J. Stat. Mech. Theory Exp.*, No. 10, 2008, doi: 10.1088/1742-5468/2008/10/P10008.
- [33] J. Zhang, J. Ma, Z. Sun, and Q. Yan, “Research Review on Algorithms of Community Detection in Complex Networks”, *J. Phys. Conf. Ser.*, Vol. 1069, No. 1, 2018, doi: 10.1088/1742-6596/1069/1/012124.
- [34] S. Solorio-Fernández, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, “A review of unsupervised feature selection methods”, *Artif. Intell. Rev.*, 2019, doi: 10.1007/s10462-019-09682-y.
- [35] E. Zorarpaci and S. A. Özel, “A hybrid approach of differential evolution and artificial bee colony for feature selection”, *Expert Syst. Appl.*, Vol. 62, pp. 91–103, 2016, doi: 10.1016/j.eswa.2016.06.004.
- [36] S. Tabakhi, A. Najafi, R. Ranjbar, and P. Moradi, “Gene selection for microarray data classification using a novel ant colony optimization”, *Neurocomputing*, Vol. 168, pp. 1024–1036, 2015, doi: 10.1016/j.neucom.2015.05.022.
- [37] H. Almazini, and K. R. Ku-Mahamud, “Grey Wolf Optimization Parameter Control for Feature Selection in Anomaly Detection”, *International Journal of Intelligent Engineering and Systems*, Vol. 14, No. 2, pp. 474–483, 2021, doi: 10.22266/ijies2021.0430.43.
- [38] L. Shi, Y. Wan, X. Gao, and M. Wang, “Feature Selection for Object-Based Classification of High-Resolution Remote Sensing Images Based on the Combination of a Genetic Algorithm and Tabu Search”, *Comput. Intell. Neurosci.*, 2018, doi: 10.1155/2018/6595792.
- [39] F. García López, M. García Torres, B. Melián Batista, J. A. Moreno Pérez, and J. M. Moreno-Vega, “Solving feature subset selection problem by a Parallel Scatter Search”, *Eur. J. Oper. Res.*, Vol. 169, No. 2, pp. 477–489, 2006, doi: 10.1016/j.ejor.2004.08.010.
- [40] Z. Manbari, F. A. Tab, and C. Salavati, “Hybrid fast unsupervised feature selection for high-dimensional data”, *Expert Syst. Appl.*, Vol. 124, pp. 97–118, 2019, doi: 10.1016/j.eswa.2019.01.016.
- [41] S. Tabakhi, P. Moradi, and F. Akhlaghian, “An unsupervised feature selection algorithm based on ant colony optimization”, *Eng. Appl. Artif. Intell.*, Vol. 32, pp. 112–123, 2014, doi: 10.1016/j.engappai.2014.03.007.
- [42] M. Dorigo and T. Stützle, *Ant colony optimization: overview and recent advances*, Handbook of metaheuristics, pp. 311–351, 2019.
- [43] A. Asuncion and D. Newman, *UCI machine learning repository*, 2013.
- [44] I. Guyon, S. Gunn, A. Ben-Hur, G. Dror, “Result analysis of the NIPS 2003 feature selection challenge”, In: *Proc. of the Advances in Neural Information Processing Systems*, pp.

545–552, 2005.

- [45] S. Tabakhi and P. Moradi, “Relevance-redundancy feature selection based on ant colony optimization”, *Pattern Recognit.*, Vol. 48, No. 9, pp. 2798–2811, 2015, doi: 10.1016/j.patcog.2015.03.020.
- [46] B. Z. Dadaneh, H. Y. Markid, and A. Zakerolhosseini, “Unsupervised probabilistic feature selection using ant colony optimization”, *Expert Syst. Appl.*, Vol. 53, pp. 27–42, 2016, doi: 10.1016/j.eswa.2016.01.021.
- [47] R. Saidi, W. Bouaguel, and N. Essoussi, “Hybrid feature selection method based on the genetic algorithm and pearson correlation coefficient”, *Machine Learning Paradigms: Theory and Application*, Springer, Cham., pp. 3–24, 2019.
- [48] M. Dorigo, V. Maniezzo, and A. Coloni, “Ant system: optimization by a colony of cooperating agents” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 26, No. 1, pp. 29–41, 1996.
- [49] Q. Gu, Z. Li, and J. Han, “Generalized fisher score for feature selection”, *arXiv preprint arXiv:1202.3725*, 2012.
- [50] T. J. Bihl, K. W. Bauer, and M. A. Temple, “Feature Selection for RF Fingerprinting with Multiple Discriminant Analysis and Using ZigBee Device Emissions”, *IEEE Trans. Inf. Forensics Secur.*, Vol. 11, No. 8, pp. 1862–1874, 2016, doi: 10.1109/TIFS.2016.2561902.
- [51] V. Bolón-Canedo, N. Sánchez-Marroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, “A review of microarray datasets and applied feature selection methods”, *Inf. Sci. (Ny.)*, Vol. 282, pp. 111–135, 2014, doi: 10.1016/j.ins.2014.05.042.
- [52] M. Tokic and G. Palm, “Value-difference based exploration: adaptive control between epsilon-greedy and softmax”, *In: Proc. of Annual conference on artificial intelligence*, Springer, Berlin, Heidelberg, pp. 335–346, 2011.
- [53] M. Kudo and J. Sklansky, “Comparison of algorithms that select features for pattern classifiers”, *Pattern Recognit.*, Vol. 33, No. 1, pp. 25–41, 2000.
- [54] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software”, *ACM SIGKDD Explor. Newsl.*, Vol. 11, No. 1, p. 10, 2009, doi: 10.1145/1656274.1656278.
- [55] A. L. Blum and P. Langley, “Selection of relevant features and examples in machine learning”, *Artif. Intell.*, Vol. 97, No. 1–2, pp. 245–271, 1997.