

Research Article

Ant Colony Optimization-Based Streaming Feature Selection: An Application to the Medical Image Diagnosis

Labiba Gillani Fahad ¹, Syed Fahad Tahir,² Waseem Shahzad,¹ Mehdi Hassan,² Hani Alquhayz,³ and Rabia Hassan¹

¹Intelligent Knowledge Mining and Analytics (IKMA) Lab, Department of Computer Science, National University of Computer & Emerging Sciences, Islamabad, Pakistan

²Department of Computer Science, Faculty of Computing & Artificial Intelligence, Air University, E-9, PAF Complex, Islamabad, Pakistan

³Department of Computer Science and Information, College of Science in Zulfi, Majmaah University, Al-Majmaah, 11952, Saudi Arabia

Correspondence should be addressed to Labiba Gillani Fahad; labiba.fahad@nu.edu.pk

Received 6 January 2020; Revised 26 August 2020; Accepted 20 September 2020; Published 7 October 2020

Academic Editor: Manuel E. Acacio Sanchez

Copyright © 2020 Labiba Gillani Fahad et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Irrelevant and redundant features increase the computation and storage requirements, and the extraction of required information becomes challenging. Feature selection enables us to extract the useful information from the given data. Streaming feature selection is an emerging field for the processing of high-dimensional data, where the total number of attributes may be infinite or unknown while the number of data instances is fixed. We propose a hybrid feature selection approach for streaming features using ant colony optimization with symmetric uncertainty (ACO-SU). The proposed approach tests the usefulness of the incoming features and removes the redundant features. The algorithm updates the obtained feature set when a new feature arrives. We evaluate our approach on fourteen datasets from the UCI repository. The results show that our approach achieves better accuracy with a minimal number of features compared with the existing methods.

1. Introduction

The space-time complexity in processing and evaluation of data increases with the increase in dimensions, referred as the curse of dimensionality [1–4]. Because of the curse of dimensionality, different similarity measures and learning algorithms may not maintain their level of accuracy and performance [5, 6]. Learning algorithms tend to overfit because of the large set of features and a small number of datapoints. Dimensionality reduction techniques such as feature selection and feature extraction need to be applied to deal with this problem [7]. Given a set of all features in advance, traditional feature selection methods tend to pick a subset of relevant features by eliminating redundant and irrelevant information [4, 8]. Wrapper feature selection methods use learning-based classifiers to select a subset of

features from the available feature space. Wrapper methods, although select good features, may not be feasible in the case of streaming features because every time a new feature arrives, we need to retrain the classifier to measure the performance of the model. On the other hand, filter-based feature selection methods that measure the performance of each feature independently are computationally less expensive; however, it is possible that the individually weak features may perform better when combined [8–10]. Therefore, filter-based methods also may not provide a complete model for streaming features.

In various real-world applications, data are generated continuously and grow at an exponential rate over time [11]. For example, in the case of bioinformatics, it may be expensive to conduct wet lab experiments to acquire a complete feature set, and features may be infeasible to store.

Other examples may include tweets classification in the social media [12] and image classification in video surveillance [8, 13]. The availability of a complete feature set in such applications, before the selection starts, may not be possible. We need to consider the available data in a different perspective, i.e., streaming data and streaming features, where data are generated sequentially. Selection of relevant features from such data is called streaming feature selection (SFS). SFS is an emerging field, which also provides benefits to traditional feature selection methods. Since it works in an online manner, it performs feature selection without storing the huge data. By selection of relevant features from the set of features seen so far, time to evaluate each incoming feature can be reduced. The SFS model is incremental in nature and learns with the changes in the environment. The model includes the relevant and discards the irrelevant features. It also checks for any feature that may become redundant with time.

We propose a hybrid SFS approach that combines the wrapper and the filter methods. We exploit the wrapper method, ant colony optimization (ACO), for the feature selection. For the early termination of the selection method, we find the association among features by exploiting the filter method, symmetric uncertainty (SU), which is a modification of information gain [14]. The proposed approach is incremental in nature, where a complete retraining is not required if a new feature arrives. Thus, the computational time compared to a pure wrapper method can be reduced. Unlike the existing forward only search-based wrapper approaches that consider the incoming features only once, the proposed approach provides a forward-backward search to select the most appropriate feature set. Thus, we are also able to identify and remove redundant features from the already selected feature set. ACO-SU is compared with two existing streaming feature selection approaches OSFS [15] and alpha-investing [16], in the classification task using three classification methods: J48 [17], JRip [18], and decision table [19]. The evaluation of the proposed approach on fourteen datasets from the UCI repository [20] through comprehensive evaluation metrics shows improved performance in correct classification of data compared with the existing feature selection approaches.

The rest of the paper is organized as follows. Section 2 discusses the related work on feature selection and classification approaches. In Section 3, we discuss the proposed ACO-SU approach. In Section 4, the proposed approach is validated and compared with existing feature selection and classification approaches using 11 datasets. Finally, Section 5 draws conclusions.

2. Related Work

The exponential growth of information has increased the importance of data analytic techniques in data mining and machine learning. To study large databases and to extract useful information, it is necessary to acquire a useful subset of features from the entire search space.

An alpha-investing approach is proposed to reduce the false discovery rate and to avoid the overfitting in streaming features [16]. The approach describes the problem caused by pairwise interactions of the candidate features. For the removal of irrelevant features, an adaptive alpha-investing is introduced using linear and logistic regression. The approach is based on the forward search that only adds those features that seem to be useful. Since every feature is considered only once, it does not account for removing any previously added feature that may not be useful as the time passes [16]. The results show that the order of incoming features is important. The performance is better when all the useful features are presented first in the stream. A stepwise gradient descent approach of grafting is proposed, where loss function is logistic regression binomial negative likelihood, used for the binary class classification problem [21]. The loss function penalizes the irrelevant features, while it has no effect on the redundant and weakly relevant features. The loss function is convex, thus guarantees to find the global optimum. Each time a feature is added, it also performs a gradient test on all the previously added features and reoptimizes the whole model. The grafting technique is compared with a traditional stepwise gradient approach, where grafting proves to be fast; however, error rate remains similar as the numbers of features are increased. In comparison with alpha-investing, stepwise regression shows better results. However, the limitation of this approach is the significant increase in optimization time, as the number of features increases. Both the alpha-investing [16] and grafting [21] account for the relevant features only while the redundant features are ignored; however, removal of redundant features is crucial in the case of streaming features. An extensive study of swarm intelligence-based feature selection algorithms has been presented in [4] which shows that swarm intelligence-based approaches can provide a near optimal solution to the NP-hard computational problems. A fast-OSFS algorithm is proposed that categorizes incoming features into four categories: strongly relevant, weakly relevant, irrelevant, and redundant features [15, 19]. The OSFS algorithm has two main phases. First, it selects the relevant features, which may be strong or weak, and second, it removes the redundant features from the selected subset. In comparison with grafting and alpha-investing, OSFS shows better prediction accuracy and compactness.

Significant features for the text classification are selected using the firefly algorithm, while SVM is applied for classification [22]. Firefly performed better than the existing techniques of genetic algorithm (GA) and particular swarm optimization. A hybrid approach combining GA and local search using correlation-based filter ranking is discussed in [23], where the local filter method refines the population by selecting relevant features using symmetric uncertainty. A binary binomial cuckoo search-based feature selection is presented in [24] to avoid the premature convergence and maintain the diversity in population. Moreover, the stability issue of the meta-heuristic method is addressed by using a data transformation method based on principal component analysis and independent component analysis. A financial crisis prediction approach is discussed in [25], where grey

wolf optimization is integrated with the tumbling effect to select optimal features, and then the fuzzy neural classifier is used for prediction. A variant of ant colony optimization is introduced in [26], which instead of following the entire graph, traverses only a directed path to reduce the computation time and memory requirement while achieving better performance. An ant colony optimization-based approach is discussed in [27] to correctly classify e-mail messages from the data stream. In [28], improvements have been suggested in a feature selection approach clustering-based ant colony optimization [29], where features are first divided into clusters in the entire search space and then ACO is used for the subset of optimal features. In the modified approach, initialization of pheromones is based on measuring the relevance of each feature to the target class. Moreover, the evaluation function is updated using multiple discriminant analysis to assign low probability to the irrelevant and redundant features. A hybrid approach combining the wrapper and filter methods is discussed in [30], where first instance-based learning is used to find a candidate feature set (CFS), and then the wrapper method is applied for further evaluation of CFS and guidance of integration process.

Feature selection in an unsupervised environment is more challenging since much time is required in extracting the label information. An unsupervised filter-based feature selection method is discussed in [31], where the search space is represented as a graph and then features are ranked in relevance to the target class using the ant colony optimization algorithm. Another filter-based unsupervised feature selection approach using ant colony optimization is presented in [32] to find the optimal feature subset iteratively instead of using the learning algorithm. The feature relevance is computed by measuring the similarity between the features, thus leading to minimized redundancy. A streaming feature selection approach for the social media performs unsupervised feature selection [33, 34]. The approach exploits the link information to determine which features are closely related. Users of the same interest, such as football and cricket, usually share the same information and use the same words. Therefore, words are considered as features to represent a group. A feature selection approach based on group performance is discussed in [35]. The algorithm performs intragroup and intergroup selections, which to some extent is similar to k-means clustering.

An online streaming feature selection approach is discussed in [36] using rough set theory. The benefit of mining features using rough set theory is that it does not require any prior domain knowledge except the given dataset. A feature selection method is proposed in [37], where mutual information is combined with ant colony optimization for better performance. A hybrid method of feature selection techniques is proposed in [38], where filter- and wrapper-based search techniques are combined to provide an effective balance between exploration and exploitation of ants in search. Another hybrid approach comprising a classifier and a filter is discussed in [14], where ACO is used as a classifier. The classifier uses a statistical measure symmetric uncertainty as a heuristic function to measure the significance of

selected subsets, which is used to check the relevancy among the selected features. The redundancy among features is removed through conditional mutual information. Our proposed SFS approach is inspired by [14], where we have exploited ACO for incremental selection of streaming features. The proposed approach is comparable with the existing state-of-the-art approaches, Alpha-investing [16] and OSFS [15], which are discussed in detail here and are also used for comparison with the proposed approach.

3. Proposed Approach

We propose a hybrid ant colony optimization approach for streaming feature selection (ACO-SU) that exploits both the wrapper and filter feature selection approaches. We discuss the traditional ACO and the proposed ACO-SU in the next sections.

3.1. Ant Colony Optimization (ACO). Ant colony optimization, a branch of swarm intelligence, is inspired by the food foraging behavior of ants [39, 40]. Ants can find the shortest path between food and their nest without any direct communication. The ant places a chemical named pheromone along its path between the food source and the nest. This pheromone is the basis of indirect communication between ants. The path with the highest pheromone concentration is selected by other ants to follow the shortest path. Pheromone concentration and heuristic information both help the ant to select its path. ACO algorithm uses the goodness measure to select the path making it suitable for solving the shortest path problems such as the traveling salesman problem (TSP). The problem is represented using a graph. Referring to the TSP, each node represents a city. An agent must complete its tour from source to destination following the shortest possible path. After the source, an agent (ant) must select the next node at each level using a goodness measure. Equation (1) is used to calculate the selection probability from the i^{th} to the j^{th} node given as

$$P_{i,j} = \frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum_{k \in S} (\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}, \quad (1)$$

where $\tau_{i,j}$ is the concentration of pheromone along the path from i to j (equation (2)), $\eta_{i,j}$ is the value of the heuristic function that describes the worth of selecting j and α and β control the importance of pheromone and heuristic function. The equation is divided by the sum of the product of pheromone and heuristic values of all the nodes that are connected to the i^{th} node. The next node is selected based on the $P_{i,j}$ value of a node. When an ant completes its tour, the pheromone is updated along the path given as

$$\tau'_{i,j} = \tau_{i,j} + \tau_{i,j} \cdot \left[\left(1 - \frac{1}{1 + P_{i,j}} \right) \right], \quad (2)$$

where $\Delta_{i,j}$ represents the fitness of the ant on a path and $\tau_{i,j}$ is the previous value of the pheromone. The more the ants travel on a path, the higher the concentration of pheromones. The ant algorithm not only updates the pheromone

concentration on each path, but it also evaporates some of the pheromones on every path, so that the algorithm does not converge locally given as

$$\tau_{i,j} = (1 - \rho) \cdot \tau'_{i,j} + \Delta\tau_{i,j}, \quad (3)$$

where ρ is the percentage of the pheromone to be evaporated. After several iterations, the algorithm stops when a convergence criterion is met. For the application of ACO, we need the following [41]:

- (i) A representation of a solution
- (ii) A method to determine the fitness of the solution
- (iii) A heuristic measure for the solutions' component

3.2. Hybrid Ant Colony Optimization Algorithm (ACO-SU). ACO has the property of finding the shortest path; thus, with modifications, it can also be used for feature selection. A modified version of the ant colony algorithm is proposed to solve the problem of streaming feature selection. We introduce a termination path at each node; thus, unlike traditional ACO, an ant can select a termination path and complete its tour at any stage. Considering nodes as features, we do not have to include additional features to complete the path. Next, unlike ACO, we consider the symmetric distance between the i^{th} and j^{th} nodes since distance between features does not need to be asymmetric. Thus, we only need a 1D matrix to store all the distances between nodes. The fitness function used is the rule quality formula [42].

Figure 1 shows the overall flow of our approach. The incoming features are first presented to the hybrid ant algorithm. After selecting the best 10 subsets from the entire search space, these selected subsets are presented to classifiers. The final subset is selected with the highest prediction accuracy. The algorithm will continue to select more features until there are any incoming features. In the hybrid ant algorithm, apt for each level, an ant has to select a feature node based on a heuristic measure of the solution component. We use symmetric uncertainty (SU) for the heuristic component same as in [14], which is a modified version of information gain (IG). SU is a correlation-based filter method that is more efficient in eliminating the redundant features [23] than IG. SU handles a pair of features symmetrically. It measures the association between two features, given as

$$SU(X|Y) = \frac{2 \times IG(X,Y)}{H(X) + H(Y)}, \quad (4)$$

where $IG(X, Y)$ is the information gain of the selected feature X calculated with class attribute Y and $H(X)$ and $H(Y)$ are the entropies of the feature and class attributes, respectively. Note that there can be variations in calculating mutual information between features. SU is used to assign weight to every feature, while features with SU less than a threshold are removed. A feature with a high value of SU is assigned more weight and thus is used to initialize the population.

To measure the significance of a selected feature subset, the performance metric used is given as

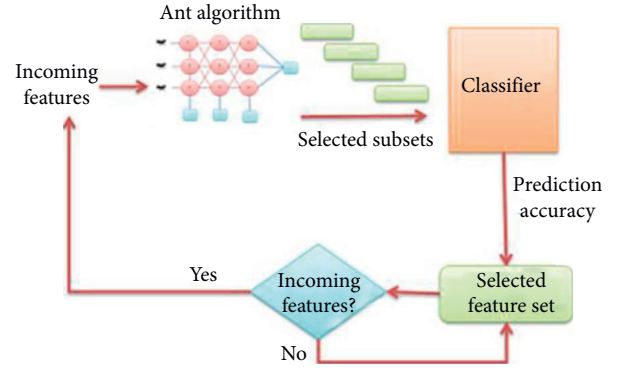


FIGURE 1: Flow diagram of the proposed approach.

$$\text{MERIT}(S) = \frac{(N - S) \times \left[\sum_{i=1}^K \text{SU}(X_i) \right]}{N}, \quad (5)$$

where N is the total number of candidate features, S is the number of features selected by an ant, and $\text{SU}(X_i)$ is the value of symmetric uncertainty of a feature selected by an ant along its path. The performance metric favors the subsets that have a minimum number of features with higher symmetric uncertainty. In the case of streaming features, features are generated dynamically, and the total number of features is unknown before the training starts. With each new incoming feature, nodes are dynamically added in the search space without affecting the previous learning of the algorithm, thus making the algorithm incremental in nature.

Algorithm 1 shows the steps of the proposed approach. The algorithm starts with the initial set of available features. Ants are generated after calculating the number of features in hand. Each ant is required to complete its path according to the defined criteria. At the end of each iteration, all ants update the pheromone concentration value on their paths. The best solutions found so far are saved. After a fixed number of iterations, the algorithm checks if any new features are available, initializes those nodes with the initial pheromone value, and constructs ants according to the new number of features. Nodes are added to the directed graph at each new arrival, and the algorithm continues from that point without reinitializing previous feature values. Ants may have converged on some paths, and it will be easy to add any new nodes in the converged path, if beneficial. In this way, our algorithm continues to select the best subset of features from the features seen so far. At each new generation, our algorithm also accounts for previously rejected features, if at any time some of the previously selected features are excluded from the candidate set; the algorithm can opt for a different set of features without affecting the whole search. To reduce the overall search space over time, we can also remove those features from the search space that are not opted by any ant or are below a certain threshold. Hence, we are able to perform backward search.

4. Evaluation and Comparisons

The proposed approach is compared with the existing streaming feature selection approaches: OSFS [15] and alpha-investing [16] since these are representative streaming


```

(1) Add initial features in search space
(2) while new incoming features do
(3)   Generate ants
(4)   Initialize pheromone
(5)   while stopping criteria not met do
(6)     Ants create rules (subset of features)
(7)     Evaluate selected subsets (using equation (4))
(8)     Update pheromone trial of each ant (using equation (2))
(9)     Evaporate pheromone (using equation (3))
(10)    Return the best solution (using equation (1))
(11)   end while
(12)   Return best 10 subsets selected
(13)   Remove redundant features from the subsets (using equation (5))
(14)   Run 4 classifiers on selected subsets
(15)   Returns one best solution
(16) End while

```

ALGORITHM 1: Proposed algorithm.

feature selection approaches commonly used for comparison in the state of the art. Alpha-investing tends to select a larger number of features, while it has limited predictive accuracy. OSFS has a better prediction accuracy but slower performance compared with alpha-investing as the number of relevant features increases [43]. The selected features from the proposed approaches, OSFS [15] and alpha-Investing [16], are evaluated using three classifiers, namely, J48 [17], JRip [18], and decision table [19], on fourteen datasets from the UCI repository [20, 44].

4.1. Experimental Setup. The proposed approach is implemented in the C++ language in the Linux environment on a dual core Intel Processor with 2 GB RAM and 40 GB hard disk. The initial parameters used in the ant colony optimization are the same as mentioned in [14]. We perform the parameter tuning using cross validation on the training set. Table 1 shows the parameter settings of the proposed approach. Initial pheromone value of each path is set to 0.5. Values of α and β both are set to 1. Pheromone evaporation factor is set to 0.15. The values of pheromones are scaled between 0 and 1, after updating and evaporation. The convergence criteria are set to the maximum number of iterations reached, i.e., 100. The number of ants created for feature selection is three times the total number of candidate features. The existing approaches for comparison are implemented in MATLAB [15] using the same implementation and parameter values as discussed in respective papers. A data mining tool WEKA is used for the classification [45]. We apply 10-fold cross validation.

4.2. Evaluation Measures. The performance evaluation metrics comprise precision, recall, F1 score, and accuracy [46].

Precision is the ratio of correctly assigned datapoints out of all the assigned datapoints to a class, given as

TABLE 1: Parameter setting of the proposed ACO-SU approach.

Parameter	Value
Number of ants	1000
Max. uncovered cases	10
Evaporation rate	0.15
No. of rules converged	10
Alpha	1
Beta	1

$$\text{precision} = \frac{TP}{TP + FP}, \quad (6)$$

where TP is the number of “true positives” and FP is the number of “false positives”. A high precision value shows the high number of correctly assigned labels out of the total assignments by a classifier.

Recall is the ratio between correctly assigned datapoints and all the datapoints in a class given as

$$\text{recall} = \frac{TP}{TP + FN}, \quad (7)$$

where FN is the number of false negatives. A high value of recall shows that a classifier returns most of the correct labels in a class. It is also known as “sensitivity” or “true positive rate”.

F1 score returns the weighted average of precision and recall, given as

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \quad (8)$$

The highest F1score, near to 1, represents the best performance, whereas 0 indicates the worst performance.

Accuracy is the amount of correct classifications from the total datapoints, in all classes, given as

$$\text{accuracy} = \frac{TP + TN}{N}, \quad (9)$$

where N is the total number of datapoints.

These measures describe the goodness of a classifier, in differentiating between positive and negative classes in a dataset using the selected features.

4.3. Analysis and Discussion. Table 2 describes the fourteen publicly available datasets used in our evaluation. Datasets from 1 to 9 are taken from the UCI repository [20], while 10–14 are from [44]. All the datasets contain variety in terms of the number of features and samples. All datasets are real and continuous. Datasets have been discretized using the WEKA unsupervised discretization filter, and missing values have been corrected. The number of features selected by the proposed approach is compared with those of the existing approaches, OSFS [15] and alpha-investing [16]. It can be observed that the datasets 1–3 have few features ($\text{Feature_Count} < 10$). The datasets 4–8 are mostly binary class datasets with moderate number of features ($10 < \text{Feature_Count} < 100$). The datasets 9–14 are more challenging multiclass datasets with larger sets of features ($\text{Feature_Count} > 100$). The test-lung dataset has 7 classes while the remaining four datasets have 3 classes.

Tables 3, 4, and 5 show the performance of feature selection approaches with three classifiers. In the case of datasets with *fewer features*, the proposed approach selects up to 20% of the total features and shows better performance than other two selection approaches. In the iris dataset, it can be observed that the alpha-investing approach selects four features compared to two and one features selected by OSFS and proposed ACO-SU approaches, respectively. The results show that from a total of four features, only one feature selected by ACO-SU was significant enough to contribute to the correct classification of this dataset, while the remaining three features can be ignored as redundant or irrelevant. In the case of datasets with *moderate number of features*, alpha-investing selects a larger set of features compared to the proposed approach and OSFS, whereas OSFS tends to select a smaller number of features, in most of the datasets. The number of features selected by the proposed approach remains mostly in between the other two approaches. Alpha-investing with 50% of selected features shows the least performance. In the house-votes dataset, ACO-SU and OSFS both select three most discriminative features from a total of 16 features, while alpha-investing selects almost all the features. The results show that three features in this case cover most of the variations, and remaining features can be ignored because of being redundant or irrelevant features. The proposed ACO-SU selects the minimum number of most significant features, followed by OSFS and alpha-investing. Finally, in the case of datasets with *large number of features*, the proposed ACO-SU and OSFS show comparable results with moderate number of selected features. In the clean-musk dataset, again with a larger set of features, the performance of alpha-investing decreases. In the test-lung dataset, the proposed ACO-SU selects five features in comparison with the four features by OSFS. On the other hand, alpha-investing shows the least performance since

alpha-investing selects only one feature. The reason of alpha-investing in dropping a smaller number of features could be due to a fixed threshold.

Table 3 shows the results of the J48 classifier on the selected features. J48 uses information gain for node splitting and feature selection, which is similar to symmetric uncertainty used in the proposed approach. The ACO-SU algorithm outperforms in 11 out of 14 datasets in classification accuracy. In the diabetes and house-votes datasets, the accuracy of the proposed approach is lower than the OSFS; it can be observed that the proposed approach has selected the minimum number of features. In the case of F1 score based on the precision and recall, it can be observed that in 11 datasets, the proposed ACO-SU approach shows better performance compared with the rest of alpha-investing and OSFS approaches while remains comparable in the remaining 3 datasets. Moreover, the TP rate of the proposed approach is also better than the other approaches. In particular, the proposed approach outperforms the alpha-investing and OSFS approaches in the lung-cancer, SPECT, leukemia, lymphoma, and test-lung datasets in terms of TP, F1 score, and accuracy.

Table 4 shows the results of the JRip classifier on the selected features. JRip is a rule-based algorithm, which adds one rule at a time and continues to add conditions to it. It also uses information gain to select a feature. Since information gain is more inclined towards the features having a greater number of values, symmetric uncertainty in the proposed approach measures the exclusivity of each feature. Therefore, we see a little performance degradation using this classifier. The ACO-SU algorithm outperforms OSFS [15] and alpha-investing [16] in 9 out of 14 datasets in classification accuracy. Alpha-investing has performed well in three datasets, where it has selected the highest number of features. Similarly, in the case of F1 score, performance of the proposed approach remains better, followed by OSFS and alpha-investing approaches on the selected features (Table 3). Moreover, the TP rate of the proposed approach is also better than the other approaches. In particular, the proposed approach outperforms alpha-investing and OSFS approaches in the liver, labor, SPECT, clean-musk, lymphoma, and test-lung datasets in all evaluation measures.

Table 5 shows the results of the decision table on the selected features. Decision table identifies the actions that can be performed based on the given conditions. Decision tables can be expressed as decision trees with actions as features and the decisions as outputs. ACO-SU outperforms other feature selection methods in 9 out of 14 datasets. The performance of features selected by the proposed approach for the decision table is similar to that of JRip, a modification of the decision table classifier; however, accuracy is higher than that of JRip on the same datasets. Moreover, recall remains better for the proposed approach followed by OSFS and alpha-investing. It can also be observed from Table 4 that the proposed ACO-SU approach achieves better F1 score in most of the datasets. In particular, the proposed approach attains a much better F1 score in iris, liver, labor, colic-horse, lung-cancer, SPECT, leukemia, and lymphoma datasets.

TABLE 2: Datasets and the number of features selected by the proposed ACO-SU approach compared with the alpha-investing and OSFS.

S. No.	Dataset	Number of instances	Classes	Number of features	ACO-SU	OSFS	Alpha-investing
1	Iris	150	3	4	1	2	4
2	Liver disorder	345	2	6	1	1	2
3	Diabetes	768	2	8	2	4	8
4	House-votes	435	2	16	3	3	14
5	Labor	57	2	16	4	3	6
6	Horse colic	300	2	28	6	3	18
7	Lung-cancer	32	3	56	6	3	6
8	SPECT	267	2	28	2	2	1
9	Clean-musk	476	2	169	3	8	34
10	Test-colon	62	3	183	5	2	3
11	Leukemia	72	3	269	8	2	1
12	Nci9	60	3	106	5	3	1
13	Lymphoma	96	3	141	5	3	1
14	Test-lung	73	7	326	5	4	1

TABLE 3: Performance metric comparison of features selected by the proposed approach, OSFS [15], and alpha-investing [16] given as an input to the J48 classifier.

Datasets	Precision			Recall			F1 score			Accuracy			All feats
	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	
Iris	96.2	96.2	94.0	96.0	96.0	94.0	96.0	96.0	94.0	96.0	96.0	93.9	93.9
Liver	33.8	33.8	33.8	58.1	58.1	58.1	42.7	42.7	42.7	58.2	58.2	58.2	58.2
Diabetes	74.7	74.6	77.8	75.1	75.1	78.3	73.5	74.8	77.9	75.5	76.1	77.3	77.3
House-votes	95.8	95.4	96.3	95.6	95.4	96.3	95.7	95.4	96.3	95.6	95.3	96.3	84.5
Labor	87.8	73.0	76.7	87.7	73.7	72.7	87.5	73.1	76.7	89.2	75.9	83.5	86.8
Colic-horse	81.7	72.7	77.9	77.7	73.7	76.3	74.0	71.1	73.0	77.8	74.7	76.8	76.4
Lung-cancer	78.5	39.9	50.5	78.1	43.8	50.0	78.0	40.9	50.0	74.1	47.6	48.8	40.8
SPECT	81.8	63.5	45.6	75.0	67.5	67.5	69.1	62.4	54.4	75.0	69.1	66.5	70.4
Clean-musk	99.6	31.9	99.6	99.6	56.5	99.6	99.6	40.8	99.6	99.7	56.5	99.7	99.7
Colon	72.8	76.1	64.5	71.0	74.2	61.3	70.0	74.4	61.7	72.7	74.4	62.1	54.7
Leukemia	59.6	40.6	21.0	59.7	43.1	45.8	59.4	40.8	28.8	55.2	45.8	45.7	36.6
Nci9	61.4	61.4	61.4	78.3	78.3	78.3	68.8	68.8	68.8	78.3	77.8	78.3	73.8
Lymphoma	46.4	41.2	16.7	47.9	41.7	32.3	45.2	41.3	19.5	46.5	40.6	31.8	36.9
Test-lung	49.8	43.5	19.4	47.9	49.3	32.9	48.2	45.7	24.4	48.6	46.7	34.3	46.7

TABLE 4: Performance metric comparison of features selected by the proposed approach, OSFS [15], and alpha-investing [16] given as an input to the JRip classifier.

Datasets	Precision			Recall			F1 score			Accuracy			All feats
	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	
Iris	96.2	96.2	94.0	96.0	96.0	94.0	96.0	96.0	94.0	96.0	96.0	94.1	94.1
Liver	63.4	54.4	48.6	62.2	57.8	53.8	55.9	48.9	47.9	61.7	56.9	53.7	53.8
Diabetes	75.8	75.9	76.3	75.9	75.9	76.8	74.1	75.9	76.4	75.5	77.0	77.4	77.4
House-votes	95.8	96.3	96.0	95.6	96.3	95.9	95.7	96.3	95.9	95.8	95.5	95.6	83.6
Labor	89.4	76.7	68.4	89.5	77.2	68.4	89.3	76.3	68.4	90.4	81.0	76.6	83.3
Colic-horse	74.7	75.9	70.9	75.0	76.0	72.3	74.8	75.9	70.8	75.1	76.1	72.1	72.4
Lung-cancer	60.4	67.3	66.3	59.4	65.6	62.5	58.5	66.0	62.7	61.2	50.2	56.8	47.2
SPECT	81.8	66.1	45.6	75.0	68.8	67.5	69.1	66.0	54.4	75.0	68.8	65.4	63.8
Clean-musk	95.2	60.9	60.9	94.7	61.6	61.3	94.7	60.1	61.0	94.4	61.8	62.3	69.3
Colon	61.2	76.0	65.3	58.1	74.2	62.9	57.6	74.4	63.5	64.4	73.9	58.7	63.0
Leukemia	43.0	36.1	21.0	44.4	48.6	45.8	42.9	40.4	28.8	48.3	46.7	45.7	40.7
Nci9	68.8	73.1	61.4	78.3	78.3	78.3	72.7	75.6	68.8	77.3	78.2	78.3	68.7
Lymphoma	42.5	30.3	13.3	42.7	38.5	36.5	37.5	31.8	19.5	45.7	35.5	36.2	37.9
Test-lung	38.8	35.6	14.2	46.6	45.2	30.1	41.7	37.8	15.6	47.8	40.6	29.2	59.3

It can be observed that the datasets 1–9 are binary class datasets with few features ($10 < \text{Feature_Count} < 30$) (Table 2). The proposed approach selects up to 20% of the total features and shows better performance than other two

selection approaches. In the case of datasets with moderate number of features, alpha-investing with 50% of selected features shows the least performance. The datasets 10–14 are more challenging multiclass datasets with larger sets of

TABLE 5: Performance metric comparison of features selected by the proposed approach, OSFS [15], and alpha-investing [16] given as an input to the decision table classifier.

Datasets	Precision			Recall			F1 score			Accuracy			
	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	ACO	OSFS	Alpha	All feats
Iris	96.2	96.2	94.0	96.0	96.0	94.0	96.0	96.0	94.0	96.0	96.0	93.8	93.8
Liver	54.3	33.8	48.9	57.3	58.1	57.3	52.1	42.7	44.3	56.8	56.9	58.0	54.7
Diabetes	74.3	74.8	77.8	74.7	75.0	78.3	73.0	74.9	77.9	74.7	75.9	77.7	77.7
House-votes	95.9	95.8	94.3	95.6	95.9	94.3	95.9	95.7	94.3	95.4	96.0	94.8	83.7
Labor	93.0	70.9	78.6	93.0	71.9	78.9	93.0	70.5	78.7	90.1	81.0	75.9	85.4
Colic-horse	74.7	74.7	71.9	75.0	75.0	73.0	74.8	74.8	72.0	74.7	76.1	74.7	74.6
Lung-cancer	72.0	41.9	39.1	68.8	46.9	43.8	68.4	42.6	37.9	65.8	42.7	41.0	44.3
SPECT	81.8	69.1	45.6	75.0	70.0	67.5	69.1	63.0	54.4	75.0	71.0	66.8	65.5
Clean-musk	95.2	56.5	99.6	94.7	61.6	61.3	94.7	61.8	94.7	95.1	63.0	95.1	95.1
Colon	67.2	74.2	61.3	66.1	74.2	62.9	65.9	74.5	61.6	72.6	71.6	59.7	54.2
Leukemia	46.2	38.4	21.0	44.4	44.4	45.8	45.0	40.3	28.8	45.7	49.7	42.0	37.6
Nci9	62.4	78.3	78.3	78.3	78.3	78.3	69.5	73.0	68.8	78.2	74.5	78.3	65.2
Lymphoma	44.5	33.6	26.7	47.9	38.5	36.5	42.7	34.7	28.1	41.9	40.2	34.8	33.1
Test-lung	42.6	42.6	20.1	43.8	49.3	32.9	42.8	43.9	24.8	42.8	48.9	34.6	39.3

features (Feature_Count > 100). Test-lung dataset has 7 classes, while the remaining four datasets have 3 classes. Alpha-investing shows the least performance on these datasets since alpha-investing selects only one feature. On the other hand, the proposed ACO-SU and OSFS show comparable results with moderate number of selected features. The obtained results show that the features selected by the proposed approach have better performance with all the three classifiers on the selected benchmark datasets especially in the recall, accuracy, and F1 score compared with the other feature selection methods.

The features selected by the proposed approach have better performance with all the three classifiers on the selected benchmark datasets in the recall, accuracy, and F1 score compared with the other feature selection methods. In some datasets, alpha-investing does not remove redundant or irrelevant features at all, while in others, it selects the least number of features; however, in both cases, its performance declines. A smaller number of selected features affects classification accuracy in the case of OSFS. The OSFS approach performs in between the alpha-investing and the proposed ACO-SU.

5. Conclusion

We proposed a streaming feature selection approach that identifies the most appropriate features from those seen so far. It also removes the features that become redundant over time. The approach is based on the ant colony optimization algorithm that combines the benefits of wrapper- and filter-based methods. The approach is incremental in nature, hence reduces the computational cost as new features are added to the model. Computational cost of training a classifier is reduced by selecting a subset of features, which is especially beneficial for the data with a larger set of features. The proposed approach selects the most relevant subset of features with an improved prediction accuracy compared with the existing approaches, which is evident from the comparable or better classification performance in the case of smaller and larger benchmark datasets. The average

accuracy of the proposed approach is up to 72.69% on the given classifiers. Further, the accuracy of the selected features can vary with different classification algorithms. Future work includes dealing with the missing values and continuous attributes which now widely exist in image processing and medical datasets.

Data Availability

Datasets used in this article are publicly available in the UCI repository [20] at the link <https://archive.ics.uci.edu/ml/index.php>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] S. C. H. Hoi, J. Wang, P. Zhao, and R. Jin, "Online feature selection for mining big data," in *Proceedings of the International workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, Beijing, China, 2012.
- [2] E. Keogh and M. Abdullah, "Curse of dimensionality," *In Encyclopedia of Machine Learning*, Springer, New York, NY, USA, 2010.
- [3] X. Hu, P. Zhou, P. Li, J. WANG, and X. WU, "A survey on online feature selection with streaming features," *Frontiers of Computer Science*, vol. 12, no. 3, pp. 479–493, 2018.
- [4] L. Brezocnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, pp. 1521–1552, 2018.
- [5] R. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.
- [6] L. G. Fahad, S. F. Tahir, and M. Rajarajan, "Activity recognition in smart homes using clustering-based classification," in *Proceedings of the IEEE International Conference on Pattern Recognition*, Stockholm, Sweden, 2014.
- [7] L. G. Fahad, S. F. Tahir, and M. Rajarajan, "Feature selection and data balancing for activity recognition in smart homes,"

- in *Proceedings of the IEEE International Conference on Communications*, London, UK, 2015.
- [8] S. F. Tahir and A. Cavallaro, "Cost-effective features for reidentification in camera networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1362–1374, 2014.
 - [9] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: a review," *Algorithms and Applications*, vol. 37-70, 2014.
 - [10] J. Li, K. Cheng, S. Wang et al., "A feature selection: a data perspective," *Journal of ACM Computing Surveys*, vol. 55, no. 6, p. 94, 2016.
 - [11] S. Larabi, M. Sainte, and A. Nada, "Firefly algorithm based feature selection for Arabic text classification," *Journal of King Saud University*, vol. 32, no. 3, pp. 320–328, 2020.
 - [12] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proceedings of the IEEE International Conference on Information and Communication Technology, Electronics and Microelectronics*, Opatija, Croatia, 2015.
 - [13] X. Xindong Wu, K. Kui Yu, W. Wei Ding, H. Hao Wang, and X. Xingquan Zhu, "Online feature selection with streaming features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178–1192, 2013.
 - [14] S. Ali and W. Shahzad, "A feature subset selection method based on symmetric uncertainty and ant colony optimization," in *Proceedings of the IEEE International Conference on Emerging Technologies*, Islamabad, Pakistan, 2012.
 - [15] K. Yu, J. Wang, W. Ding, and X. Wu, *A Library for Online Streaming Feature Selection*, 2015.
 - [16] J. Zhou, D. Foster, Robert Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proceedings of the ACM International Conference on Knowledge Discovery in Data Mining*, New York, NY, USA, 2005.
 - [17] E. Karabulut, S. A. Ozel, and T. Ibrikli, "A comparative study on the effect of feature selection on classification accuracy," *Procedia Technology*, vol. 1, no. 1, pp. 323–327, 2012.
 - [18] D. Khairan, "Rajab, new hybrid features selection method: a case study on websites phishing," *Security and Communication Networks*, vol. 1155, p. 9838169, 2017.
 - [19] R. Kohavi, "The power of decision tables," in *Proceedings of the European Conference on Machine Learning*, Heraclion, Greece, 1995.
 - [20] M. Lichman, *UCI Machine Learning Repository*, University of California, School of Information and Computer Science, Irvine, CA, USA, 2013, <http://archive.ics.uci.edu/ml>.
 - [21] S. Perkins and T. James, "Online feature selection using grafting," in *Proceedings of the International Conference on Machine Learning*, Washington, DC, USA, 2003.
 - [22] S. Larabi Marie-Sainte and A. Nada, "Firefly algorithm based feature selection for Arabic text classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 28, 2020.
 - [23] S. Senthamarai Kannan and N. Ramaraj, "A novel hybrid feature selection via symmetrical uncertainty ranking based local memetic search algorithm," *Knowledge-Based Systems*, vol. 23, no. 6, pp. 580–585, 2010.
 - [24] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, "Feature selection method based on hybrid data transformation and binary binomial cuckoo search," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 2, pp. 719–738, 2020.
 - [25] S. Sankhwar, D. Gupta, K. C. Ramya, S. Sheeba Rani, K. Shankar, and S. K. Lakshmanaprabu, "Improved grey wolf optimization-based feature subset selection with fuzzy neural classifier for financial crisis prediction," *Soft Computing*, vol. 24, no. 1, pp. 101–110, 2020.
 - [26] B. Chen, L. Chen, and Y. Chen, "Efficient ant colony optimization for image feature selection," *Signal Processing*, vol. 93, no. 6, pp. 1566–1576, 2013.
 - [27] K. Jan, P. Juszczuk, and B. Probierz, "The hybrid ant colony optimization and ensemble method for solving the data stream e-mail foldering problem," *Neural Computing and Applications*, vol. 19, no. 1, Article ID 04672, 2020.
 - [28] 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 Systems*, vol. 159, no. 1, pp. 270–285, 2018.
 - [29] P. Moradi and M. Rostami, "Integration of graph clustering with ant colony optimization for feature selection," *Knowledge-Based Systems*, vol. 84, no. 1, 2015.
 - [30] A. Ben Brahim and M. Limam, "A hybrid feature selection method based on instance learning and cooperative subset search," *Pattern Recognition Letters*, vol. 69, no. 1, pp. 28–34, 2016.
 - [31] S. Tabakhi and P. Moradi, "Relevance-redundancy feature selection based on ant colony optimization," *Pattern Recognition*, vol. 48, no. 9, pp. 2798–2811, 2015.
 - [32] S. Tabakhi, P. Moradi, and F. Akhlaghian, "An unsupervised feature selection algorithm based on ant colony optimization," *Engineering Applications of Artificial Intelligence*, vol. 32, no. 1, pp. 112–123, 2014.
 - [33] J. Li, X. Hu, J. Tang, and H. Liu, "Unsupervised streaming feature selection in social media," in *Proceedings of the ACM International Conference on Information and Knowledge Management*, Melbourne, Australia, 2015.
 - [34] A. Roy, "Automated online feature selection and learning from high-dimensional streaming data using an ensemble of Kohonen neurons," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, Killarney, Ireland, 2015.
 - [35] J. Wang, Z.-Q. Zhao, X. Hu, Y.-ming Cheung, M. Wang, and X. Wu, "Online group feature selection," in *Proceedings of the International Conference on Artificial Intelligence*, Beijing, China, 2013.
 - [36] S. Eskandari and M. M. Javidi, "Online streaming feature selection using rough sets," *International Journal of Approximate Reasoning*, vol. 69, no. 0, pp. 35–57, 2016.
 - [37] C.-K. Zhang and H. Hu, "Feature selection using the hybrid of ant colony optimization and mutual information for the forecaster," in *Proceedings of the IEEE International Machine Learning and Cybernetics*, Guangzhou, China, 2005.
 - [38] M. M. Kabir, M. Shahjahan, and K. Murase, "A new hybrid ant colony optimization algorithm for feature selection," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3747–3763, 2012.
 - [39] J. Kennedy and E. Russell, *Particle Swarm Optimization*, IEEE, Portland, OR, USA, 1995.
 - [40] Z. Khan, S. Fang, A. Koubaa, P. Fan, F. Abbas, and H. Farman, "Street-centric routing scheme using ant colony optimization-based clustering for bus-based vehicular ad-hoc network," *Computers & Electrical Engineering*, vol. 86, pp. 106736–106746, 2020.
 - [41] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, Hoboken, NJ, USA, 2005.
 - [42] K. M. Salama and A. M. Abdelbar, "Exploring different rule quality evaluation functions in ACO-based classification algorithms," in *Proceedings of the IEEE International Symposium on Swarm Intelligence*, Paris, France, 2014.

- [43] D. You, X. Wu, L. Shen et al., "Online feature selection for streaming features using self-adaption sliding-window sampling," *IEEE Access*, vol. 7, pp. 16088–16100, 2019.
- [44] H. Fuhui Long, C. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.
- [45] M. Hall, E. Frank, G. Holmes, B. Pfahringer, R. Peter, and H. Ian, *Witten the WEKA Data Mining Software: An Update*, ACM, New York, NY, USA, 2009.
- [46] L. Gillani Fahad, A. Khan, and M. Rajarajan, "Activity recognition in smart homes with self-verification of assignments," *Neurocomputing*, vol. 149, pp. 1286–1298, 2014.