International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 1
ISSN: 2088-5334

Genetic-based Pruning Technique for Ant-Miner Classification Algorithm

Hayder Naser Khraibet AL-Behadili[a,1], Ku Ruhana Ku-Mahamud[b,2], Rafid Sagban[c,3]

[a]*Computer Science Department, Shatt Alarab University College, Basra,61001, Iraq*
[b]*School of Computing, Universiti Utara Malaysia, Kedah, 06010 Sintok, Kedah, Malaysia*
[c]*Department of Software, University of Babylon, Babylon,51002, Iraq*

Corresponding author: [1]haider872004@gmail.com, [2]ruhana@uum.edu.my, [3]rsagban@uobabylon.edu.iq

Abstract*— Ant colony optimization (ACO) is a well-known algorithm from swarm intelligence that plays an essential role in obtaining rich solutions to complex problems with wide search space. ACO is successfully applied to different application problems involving rules-based classification through an ant-miner classifier. However, in the ant-miner classifier, rule-pruning suffers from the problem of nesting effect origins from the method of greedy Sequential Backward Selection (SBS) in term selection, thereby depriving the opportunity of obtaining a good pruned rule by adding/removing the terms during the pruning process. This paper presents an extension to the Ant-Miner, namely the genetic algorithm Ant-Miner (GA-Ant Miner), which incorporates the use of GA as a key aspect in the design and implementation of a new rule pruning technique. This pruning technique consists of three fundamental procedures: an initial population Ant-Miner, crossover to prune the rule, and mutation to diversify the pruned classification rule. The GA-Ant Miner performance is tested and compared with the most related ant-mining classifiers, including the original Ant-Miner, ACO/ PSO2, TACO-Miner, CAnt-Miner, and Ant-Miner with a hybrid pruner, across various public available UCI datasets. These datasets are varied in terms of instance number, feature size, class number, and the application domains. Overall, the performance results indicate that the GA-Ant Miner classifier outperforms the other five classifiers in the classification accuracy and model size. Furthermore, the experimental results using statistical test prove that GA-Ant Miner is the best classifier when considering the multi objectives (i.e., accuracy and model size ranks).

***Keywords*—Ant colony optimization; genetic algorithm; metaheuristic; rules-based classification; swarm intelligence.**

Manuscript received 11 Jan. 2020; revised 17 Oct. 2020; accepted 3 Nov. 2020. Date of publication 28 Feb. 2021.

I. INTRODUCTION

Data mining, also known as knowledge discovery, is the operation of unveiling hidden insights from data. Different institutions and companies consider it as the most crucial opportunity to raise revenue. Data mining is widely used in various fields, such as medicine, science, recognition, business, and engineering [1]. In data mining, two types of learning are available: supervised techniques and unsupervised learning approaches [2]–[6]. Unsupervised learning techniques discover patterns from data. These techniques work without any previous knowledge from the data (i.e., unlabeled class) [7]–[10].

Conversely, supervised learning techniques use labeled data to build the data mining model [11]. Such techniques can be considered a powerful approach with an accurate and rapid result in a wide range of applications (e.g., businesses). One of the supervised learning techniques that gains significant attention is the rules-based classification which extracts classification rules from the data. One of the prominent algorithms used for rules-classification is ant colony optimization (ACO) for rules classification of Ant-Miner variants [12], [13]. The Ant-Miner produces a comprehensive classification model by finding a list of classification rules fashion (IF-THEN) from the data. The advantages of these rules can be easily translated to natural language.

The Ant-Miner [14] is inspired by the real behavior of an actual ant colony. The Ant-Miner is a metaheuristic, swarm-based, stochastic, and separate-and-conquer approach. This consisted of three major stages, namely, rule building, pruning rule, and updating pheromone. In the rule building stage, each specific ant begins to add terms to be included in the rule. This term acts as a particular duo (attribute and value) from the attribute in the dataset, and each term can be added only once under the building rule. The Ant-Miner classifier adds terms that increase classification performance according to its pheromone concentration and amount of information.

In rule pruning, the overfitting problem can be avoided by decreasing the length and increasing the constructed rules' simplicity. The procedure removes one term at a time whilst enhancing quality. The pruning repeats until improvement ceases. The pheromone update has two main stages: updating the pheromone amount for all terms in the current rule based on its quality and updating all terms that do not appear in the current rule.

The Ant-Miner pruning technique has the nesting effect originating from a greedy sequential backward selection method in feature selection. The pruning starts from a complete set of terms and erases one term at a time with no ability to add the eliminated terms again. It deprives the opportunity to obtain a good pruned rule to restoring the removed terms [15]–[18]. This paper proposes a new pruning technique based on the genetic algorithm's search behavior (GA) to find the optimal pruning rule and introduce a new rules-classification algorithm called the GA-Ant Miner. The GA-AntMiner has a flexible rule pruning technique for adding/dually removing the terms.

II. Material and Method

A. Implementation of GA-Ant Miner Classifier

The GA-Ant Miner classifier begins to discover one classification rule from training instances. This discovered rule is then inserted in the rule list, in which every instance covers this rule antecedent and have class predicted by the consequent rule are removed from the training instances set. These operations stop when all the training data cases are lower than the prespecified constant values knows as the maximum number of uncovered cases. This approach has three major stages, called rule building, pruning rule, and updating pheromone.

The initial procedure is the construction rule, where every ant begins to insert terms to be included in the rule. The ant inserts one term to improve the classification accuracy according to its probability value. The probability of each term to be selected in the particular rule is provided by Equation (1) [19] as follows:

$$\text{Probability} = \frac{\left[\tau_{ij(t)}\right]\,\left[\eta_{ij}\right]}{\sum_{i=1}^{a} xi \,.\sum_{j=1}^{bi}\left[\tau_{ij(t)}\right]\,\left[\eta_{ij}\right]} \qquad (1)$$

where $\left[\boldsymbol{\tau_{ij(t)}}\right]$ represent the value of pheromone for each term at iteration (t); $\left[\eta_{ij}\right]$ is the problem depending upon heuristic function; a is the attribute number in the dataset; bi is the number of different values for each attribute; and xi equal to 1 (while the attribute is not yet used by the current ant); or 0 (otherwise). The heuristic amount and the pheromone amount are used to decide on the term selected. In the GA-Ant Miner, the heuristic function is inspired by information theory. The GA-Ant Miner computes the amount of information contained in each term (entropy). The heuristic function is given by Equations (2) and (3) [19], as follows:

$$\eta_{ij} = \frac{\log_2 k - H(W|A_i = V_{ij})}{\sum_{i=1}^{a} xi \,.\sum_{j=1}^{bi}(\log_2 k - H(W|A_i = V_{ij}))} \qquad (2)$$

$$H(W|A_i = V_{ij}) = -\sum_{w=1}^{k}\left[\frac{P(W|A_i = V_{ij})}{T_{ij}}\right] * \log_2\left[\frac{P(W|A_i = V_{ij})}{T_{ij}}\right] \qquad (3)$$

where w is the class attribute, and k is the class number; $P\left(W|A_i = V_{ij}\right)$ is the instances partition, where: each attribute Ai has values Vij with from class w. $|Tij|$ is the total number of instances in partition Tij and a present the total attributes number. bi is the values number in the particular of attribute i. This process is repeated, while specific attributes are not used yet, or the prespecified minimum number of uncovered instances by the constructed rule. Once the rule is completed, the classifier chooses the (then) part of the rule by assigning the majority class among the instances covered by the rule.

This study proposed a new pruning technique using the GA concept. Three algorithmic components are added (population initialization, crossover, and mutation) in the proposed technique. The modification aims to minimize the number of terms in the discovered rule and maximize the classification accuracy. The pruning technique's algorithmic components in the GA-Ant Miner classifier are population initialization, crossover, mutation, updating of instance list, determination of consequent rule, calculation of rule quality, and stopping criteria. The GA-Ant Miner generates the classification rule as an integer 1D array with a size equal to the number of features in the dataset and consists of two components. In the first component (antecedent), each bit is associated with the dataset feature. If the bit of this array equals a positive integer number, then one term of that feature can participate in the classification rule. Otherwise, if the bit of this array equals a negative value, then the terms of that feature are excluded. Meanwhile, the second component represents the classification class label. In population initialization, the proposed technique is to add a 1D array (rule) of negative values in all elements with the same size as the original rule, as described in Fig. 1.

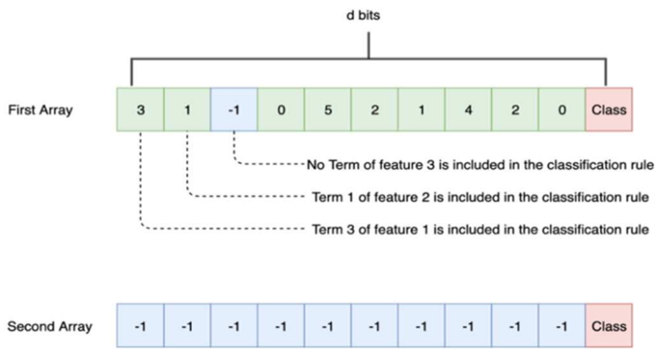

Fig. 1 Chromosome of Genetic-based post pruning technique

The term elimination processes occur implicitly through crossover and mutation between the two chromosomes, and each eliminated term can be re-added. The crossover operator is how parent chromosomes (rules) exchange genetic information to create the best pruning rule. The crossover rate parameter is performed to decide if the rules should have a crossover. The parameter of the crossover rate is compared with a random number to perform the crossover operation. Besides, different methods are used for trading genetic information between two individuals. The crossover operation used in this study includes two single-point crossover operations. The first point is the first term in the rule, while the second point corresponds to the high correlation term that improves the pruning rule quality. The pseudocode (Fig. 2) of the crossover method is implemented as follows:

```
Crossover Pseudocode

FOR each term in the rule
      IF CrossoverRate > Random ();
           FirstTerm = SelectFirstTerm();
           SecondTerm= SelectSecondTerm();
           Offspring = Crossover (FirstTerm,
           SecondTerm, FirstParent, SecondParent);
       ELSE: Offspring= (FirstParent, SecondParent);
       END IF
END Loop
```

Fig. 2 Crossover operator pseudocode

A mutation operator is used to maintain genetic diversity from one generation of a rule pruning to the next. The mutation rate parameter is used to perform mutation in a similar approach to the crossover operator. If the mutation rate is greater than the random number, then each gene has an equal chance of being mutated during the mutation stage. The mutation operator selects a random bit in the parent chromosomes and flips the value of this bit. Fig. 3 shows the pseudocode of the mutation operator. Besides, examples of two single point crossovers, and one single point mutation operator used in the pruning technique are shown in Fig. 4.

```
Mutation Pseudocode

IF MutationRate > Random ();
        MutatedTerm = SelectMutatedTerm();
        Offspring = Mutation (MutatedTerm,FirstParent,
        Second Parent);
ELSE: Offspring= (FirstParent, SecondParent);
END IF
```

Fig. 3 Mutation operator pseudocode

The number cases covered by the pruning rule are checked using the update instance list procedure. If the number of instances changes, then the classifier selects the consequence ('then' part) of the rule by giving the majority class that appears in the cases covered by the rule. The quality of the original rule is compared with the pruned one. If the pruning rule's quality is higher than the uprunning rule, then the former rule takes the place of the original one. This process is iterated until the termination condition, which is a fixed number of the eliminations using crossover and mutation operators, is satisfied.

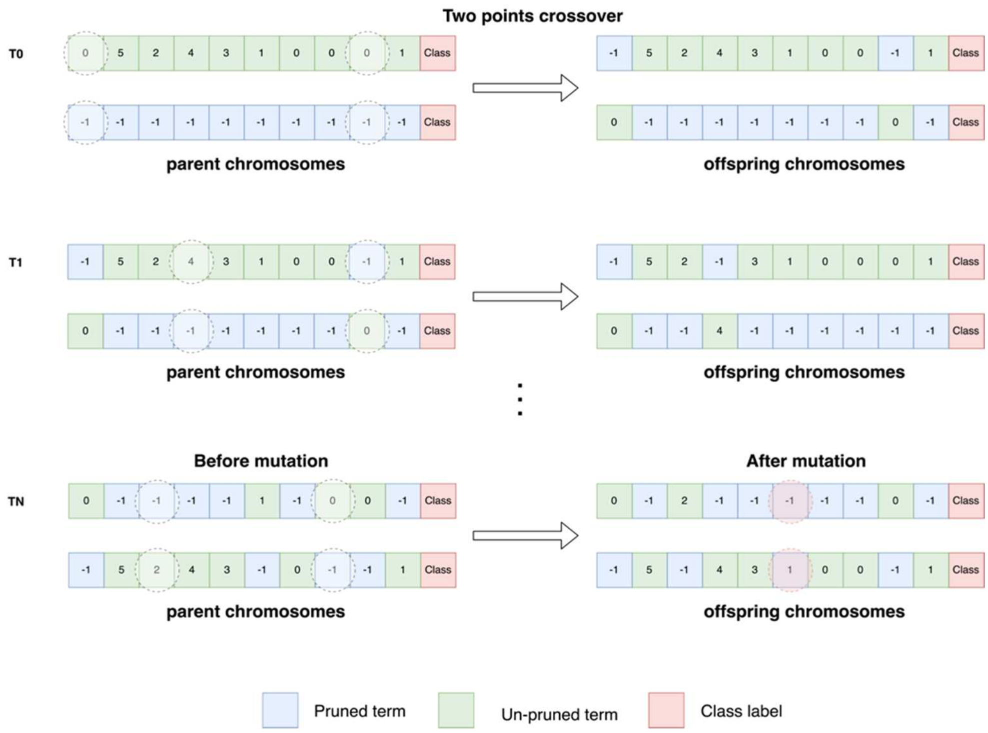

Fig. 4 . Crossover operation with two single points and mutation operation with one single point

The pheromone is updated after rule construction and prune procedures. The approach of pheromone update has two main procedures. Firstly, growing the pheromone for all terms that occurs in the construction rule according to rule quality by Equations (4) and (5) [19].

$$\tau_{ij(t+1)} = \tau_{ij(t)} + \tau_{ij(t)} . Q \tag{4}$$

$$Q = + \frac{TP}{TP+FN} * \frac{TN}{FP+TN} \tag{5}$$

Where *TP* represents the true positive instances; *FN* represents the false-negative instances; *TN* represents the true negative instances; and *FP* represents the false-negative instances. Secondly, evaporating each term does not represent in the constructed rule by normalizing unused terms. Another rule will be built by another ant derived from the updated pheromone amount. The process is accomplished based on the following stopping conditions are satisfied. In the first condition, the number of discovered rules must be equal to the number of ants. According to the number of the rule convergence that statically determined, the second condition is where the ant converges to a particular rule by building one precisely the same as previously constructed. The best rule constructed will be added to the list of discovered rules. Fig. 5 displays a high-level pseudocode of the GA–Ant Miner algorithm.

```
GA-AntMiner as post-pruning technique

Input: arff dataset
Output: classification rule

Training Database = {all instances};
RuleList Initialization =[];
WHILE(Training Database > MaxNumber of uncovered instances)
Ant Number=1;
Convergence Number=1;
Pheromone Initialization();
REPEAT
RuleConstructs;

        // Genetic-based pruning technique start here
PopulationInitialization;
While (termination condition not met);
    Crossover;
    Mutation;
    UpdateInstancesList;
    DetermineRuleConsequent;
    EvaluateRule;
    IF Quality ( PruneRule ) > Quality (Rule);
    Rule= PruneRule;
    End IF
END-WHILE
        // Genetic-based pruning technique end here
Pheromone Updating ();
IF (Current constructed rule=Previous constructed rule)
THEN Convergence index number = Convergence index number + 1;
ELSE Convergence index number t = 1;
END IF Ant number = Antnumber + 1;
UNTIL (Ant number>=limit number) OR (Convergence index number >= Rule Convergence limit)
Best rule selection ();
Add Best rule to Discovered ruleList ();
Training Database = Training Database - {Instances Covered by Best rule};
END-WHILE
```

Fig. 5 GA-Ant Miner pseudocode

B. Experiments

A 10-fold cross-validation procedure is used to evaluate the anti-mining classification algorithms. In this procedure, the dataset is split into ten groups. Each group is equally sized, where nine groups are used for the training process. The remaining group is used in the testing stage. This process is repeated ten times with a different group for training and testing to ensure that all groups are used. Subsequently, the performance of all folds is averaged, and the standard deviations are computed. The well-known 10-fold cross-validation technique is used in other anti-mining classifier studies [20], [21].

C. Performance Evaluation

The evaluation is performed based on three criteria. Firstly, the classification accuracy in discovering the rule list is called the accurate classification rate. This criterion is based on the accurately classified instances in the test data. Each time, the training subsets consist of n number of instances, and the classifier constructs the training and test subsets that are used to test the performance. The accurate classification instances determine the performance of the proposed classifier. Secondly, the size of the rule list is computed by the number of terms in the constructed rules. The term number (conditions) refers to the number of antecedents carried by each rule. Thirdly, the algorithms' performance in the classification accuracy against the complexity of the model is observed. The average rank of classification accuracy and model size is used in our experiments. A low rank implies good algorithm performance.

D. Databases

Benchmark datasets are used to compare the proposed algorithms with the commonly related ant-mining classification algorithms in the literature. Benchmark datasets are selected in accordance with the ant-mining literature. This benchmark includes secondary datasets selected from UCI [22]. The datasets diverse in terms of the number of instances (lie between the range of 150–8124), attributes (range of 4–60) and class labels. In addition, the attributes consist of categorical and continuous types.

The selected datasets are as follows: Balance Scale, Breast Cancer (Ljubljana), Breast Cancer (Wisconsin), Credit-a, Credit-g, Diabetes, Heart (Cleveland), Heart (Stat log), Hepatitis, Ionosphere, Iris, Lymphography, Mushroom, Segment, Sonar, and Tic-Tac-Toe. The main features of each dataset are summarized in Table 1. The features include the name of datasets, number of instances, number of attributes, number of values in each class attribute, and type of attributes.

TABLE I
MAIN DATASET FEATURES IN THE EXPERIMENTS

Data Sets Name	Attributes Number	Instances Number	Type of Attributes	Classes Number
Balance Scale	4	625	Categorical	3
Breast Cancer (Ljubljana)	9	286	Categorical	2
Breast Cancer (Wisconsin)	9	699	Continuous	2
Credit/A Dataset	15	690	Categorical, Continuous	2
Credit/G Dataset	20	1000	Categorical, Continuous	2
Diabetes disease	8	768	Continuous	2
Heart/Clevelan d disease	13	303	Categorical, Continuous	5
Heart/Statlog diabetic disease	13	270	Categorical, Continuous	2
Hepatitis disease	19	155	Categorical, Continuous	2
Ionosphere dataset	34	351	Continuous	2
Iris dataset	4	150	Continuous	3
Lymphography medical imaging	18	148	Categorical, Continuous	4
Mushroom dataset	22	8124	Categorical	2
Segment dataset	19	2310	Continuous	7
Sonar dataset	60	208	Categorical, Continuous	2
Tic/tac/toe	9	958	Categorical	2

E. Classifiers

The compared classifiers include the original Ant-Miner, CAnt-Miner, ACO/PSO2, Ant-Miner with a hybrid pruner, and TACO-Miner. The first classifier is CAnt-Miner, an Ant-Miner version and can handle continuous attributes during training model construction [23]. ACO/PSO2 is a hybrid swarm intelligence metaheuristic algorithm for rules-based classification. The pruning procedures of ACO/PSO2 are applied to discover the best rule for each iteration. ACO/PSO2 uses two pruning procedures. The first procedure is the original Ant-Miner pruning procedure and applied to the best rule discovered a whose number of terms is less than 20. If the constructed rule entails more than 20 terms for each rule, then the pruning iterates to remove the unimportant or detrimental terms from the classification rule until the number is decreased to 20 terms.

The Ant-Miner pruning procedure is then implemented subsequently [24]. The TACO-Miner classifier consists of a predefined value of threshold criterion based on each term's information gain. If the information gain value related to the term is lower than the threshold value, the term is declined in the inclusion process [25], [26]. The threshold is considered a preprinting criterion and used to accept or reject terms. On the other hand, the ant-miner with a hybrid pruner introduces a new rule into the pruning procedure and entails the hybridization of the original Ant-Miner's rule pruner with another rule pruner the basis of two aspects. The aspects are the information gain of terms and a new parameter to determine the acceptable number of terms to be included in the rule called r. The first procedure is utilized for each rule that overrides the number of acceptable terms allowable in a rule. The number of terms in the selected rule is then reduced until its value reaches the value of r. This selection method is executed based on the roulette wheel technique and the value of each term's information gain. After that, the second procedure, which is the Ant-Miner's same prune ring procedure, is applied [27].

F. Parameter Setting

This subsection introduces the parameter values used in all experiment steps adopted to ensure fair comparison results when each classifier works with similar parameter values [28]–[30]. The list of parameters used for all classifiers are listed in Table 2.

TABLE II
EXPERIMENTAL PARAMETERS

Parameter	Description	Value
Ant Number	Total number of ants	10
MICR	Mini instances number covered by the rule	5
MI	Max instances number not covered by the rule	10
Convergence Number	Convergence limit number	10
Iteration Number	Iteration number	10
β	Beta	1
α	Alpha	1
CR	Crossover Rate	0.8
MR	Mutation Rate	0.1

III. RESULTS AND DISCUSSION

This section compares the GA-AntMiner classifier results with those of related classifiers with different rule pruning procedures. These classifiers are the original Ant-Miner, CAnt-Miner, ACO/PSO2, TACO-Miner and Ant-Miner with a hybrid pruner. Experiments on 16 datasets from the UCI repository are conducted for all classification algorithms. The experiments use 10 folds of the cross-validation technique based on the previous section's benchmark scenarios. In the first method, Tables 3 and 4 show the experimental results of the average classification accuracy and model size. The first row presents the average classification accuracy in each table and the standard deviations after the symbol "+/−." For each table in the experiment, the best result is clarified in bold. The second row displays the performance rank for each dataset. The experimental results in Tables 3 and 4 are used to determine the best classifiers.

Table 3 shows that the GA-Ant Miner is better than the Ant-Miner in all datasets. The GA-Ant Miner is better than TACO and hybrid pruner in 15 datasets. Furthermore, the GA-Ant Miner outperforms the CAnt-Miner and ACO/PSO2 in 13 and 12 datasets, respectively. In comparison with other classifiers, the GA-Ant Miner achieves the highest result in 10 datasets. The GA-Ant Miner obtains the second-best performance in four datasets (Credit-g, Diabetes, Segment and Tic-tac-toe). The second-best classifier is ACO/PSO2 with three datasets. The CAnt-Miner achieves the best result in two datasets, and the TACO classifier obtains the best result in one dataset. The Ant-Miner and hybrid pruner acquire the lowest results across all datasets.

Table 4 shows that the GA-Ant Miner achieves the better result for model size in all datasets in comparison with the Ant-Miner classifier. By using the same token, the GA-Ant Miner achieves the best result in 15 datasets compared with the CAnt-Miner and hybrid pruner classifiers. The GA-Ant Miner gains over 14 datasets in contrast to ACO/PSO2. However, the GA-Ant Miner and TACO classifiers are like the highest result in eight datasets. In comparison with other classifiers, the GA-Ant Miner achieves the best result in nine datasets. The GA-Ant Miner obtains the second-best result in four datasets (Balance Scale, Heart-Cleveland, Heart-Stat log, and Mushroom). The second-best classifier is TACO with five datasets. ACO/PSO2 and the CAnt-Miner achieve the best result in two datasets and one dataset, respectively. Furthermore, the Ant-Miner and hybrid pruner classifiers obtain lower results than other classifiers.

The GA-Ant Miner has obtained the best classification accuracy and best model size. Under these circumstances, the GA-Ant Miner dominates the other classifiers in all evaluation criteria. This result is due to the enhancement process achieved by utilizing the GA ability to refresh the eliminated terms during the pruning process.

TABLE III
AVERAGE CLASSIFICATION ACCURACY (AVERAGE +/− STANDARD DEVIATION, PERFORMANCE RANK) OBTAINED USING 10 FOLDS CROSS-VALIDATION METHODS FOR ALL CLASSIFIERS AND GA-ANT MINER

Dataset		Ant-Miner	CAnt-Miner	ACO/PSO2	TACO	Hybrid Pruner	GA-Ant Miner
Balance Scale	Accuracy	69.73% +/- 1.58%	69.29 % +/- 1.112	68.66 % +/- 4.97	66.65% +/- 2.1%	68.62% +/- 1.21%	**71.53% +/- 1.46%**
	Rank	2	3	4	6	5	**1**
Breast Cancer (Ljubljana)	Accuracy	72.32% +/- 1.73%	74.87% +/- 1.846	70.94 % +/- 5.37	74.66% +/- 2.52%	72.67% +/- 2.52%	**75.53% +/- 2.59%**
	Rank	5	2	6	3	4	**1**
Breast Cancer (Wisconsin)	Accuracy	94.43% +/- 1.17%	94.42% +/- 0.889	93.86 % +/- 4.56	94.56% +/- 0.85%	94% +/- 1.06%	**94.71% +/- 1.4%**
	Rank	3	4	6	2	5	**1**
Credit/A Dataset	Accuracy	84.49% +/- 1.04%	84.92% +/- 1.063	84.69 % +/- 4.39	78.99% +/- 2.59%	84.64% +/- 1.06%	**85.8% +/- 0.68%**
	Rank	5	2	3	6	4	**1**
Credit/G Dataset	Accuracy	70.7% +/- 1%	**71.80% +/- 0.841**	71.0 % +/- 4.52	69.4% +/- 2.16%	70.4% +/- 0.81%	71.5% +/- 1.51%
	Rank	4	**1**	3	6	5	2
diabetic disease	Accuracy	71.12% +/- 2.01%	74.61% +/- 2.197	**76.31 % +/- 4.32**	71.99% +/- 1.49%	73.3% +/- 1.68%	75% +/- 1.12%
	Rank	6	3	**1**	5	4	2
Heart/Cleveland disease	Accuracy	76.17% +/- 2.85%	77.23% +/- 1.652	78.51 % +/- 6.16	76.13% +/- 2.32%	76.63% +/- 1.49%	**79.27% +/- 1.81%**
	Rank	5	3	2	6	4	**1**
Heart/Stat log diabetic disease	Accuracy	77.78% +/- 2.41%	77.77% +/- 2.869	78.89 % +/- 7.78	77.78% +/- 2.14%	77.78% +/- 2.59%	**80% +/- 1.67%**
	Rank	4	6	2	4	4	**1**
Hepatitis disease	Accuracy	80.03% +/- 3.68%	76.20% +/- 2.034	76.13 % +/- 8.34	78.98% +/- 3.65%	75.71% +/- 2.89%	**81.93% +/- 2.71%**
	Rank	2	4	5	3	6	**1**
Ionosphere dataset	Accuracy	86.03% +/- 1.77%	84.60% +/- 1.074	65.51 % +/- 7.46	79.54% +/- 1.89%	86.51% +/- 1.77%	**87.22% +/- 1.35%**
	Rank	3	4	6	5	2	**1**
Iris dataset	Accuracy	94% +/- 1.85%	94.66% +/- 1.663	94.0 % +/- 8.14	94.67% +/- 1.94%	94.67% +/- 1.66%	**96% +/- 1.47%**
	Rank	5.5	4	5.5	2.5	2.5	**1**
Lymphography medical imaging	Accuracy	71.37% +/- 1.87%	74.85% +/- 3.475	77.19 % +/- 12.59	**78.56% +/- 2.89%**	68.26% +/- 2.59%	75.49% +/- 3.52%
	Rank	5	4	2	**1**	6	3
Mushroom dataset	Accuracy	97.14% +/- 0.42%	97.93% +/- 0.561	**100.0 % +/- 0.0**	96.27% +/- 0.75%	97.91% +/- 0.45%	97.85% +/- 0.31%
	Rank	5	2	**1**	6	3	4
Segment dataset	Accuracy	80.04% +/- 1.4%	**84.76% +/- 0.846**	82.08 % +/- 4.64	76.88% +/- 0.85%	82.99% +/- 1.24%	83.33% +/- 1.07%
	Rank	5	**1**	4	6	3	2
Sonar dataset	Accuracy	75.61% +/- 2.64%	77.88% +/- 2.482	54.86 % +/- 3.87	72.1% +/- 4.01%	75.09% +/- 3.63%	**78.42% +/- 2.73%**
	Rank	3	2	6	5	4	**1**
Tic/tac/toe	Accuracy	73.58% +/- 1.72%	72.23% +/- 1.361	**100.0 % +/- 0.0**	71.59% +/- 1.57%	72.33% +/- 1.4%	75.45% +/- 2.1%
	Rank	3	5	**1**	6	4	2

TABLE IV
AVERAGE MODEL SIZE (AVERAGE +/− STANDARD DEVIATION, PERFORMANCE RANK) OBTAINED USING 10 FOLDS CROSS-VALIDATION METHOD FOR ALL CLASSIFIERS AND GA-ANT MINER

Dataset		**Ant-Miner**	**CAnt-Miner**	**ACO/PSO2**	**TACO**	**Hybrid Pruner**	**GA-Ant Miner**
Balance Scale	Accuracy	11 +/-0	11 +/-1	52 +/- 0	**6.6 +/- 0.48**	11 +/- 0	9.2 +/- 0.61
	Rank	4	4	6	**1**	4	2
Breast Cancer (Ljubljana)	Accuracy	7.8 +/- 0.29	7.80 +/- 1.14	26.8 +/-6.196	7.7 +/- 0.5	8.7 +/- 0.62	**6.7 +/- 0.42**
	Rank	3.5	3.5	6	2	5	**1**
Breast Cancer (Wisconsin)	Accuracy	8.4 +/- 0.22	8.0 +/- 0.94	17.1 +/- 2.42	7.1 +/- 0.38	7.4 +/- 0.31	**6.5 +/- 0.22**
	Rank	5	4	6	2	3	**1**
Credit/A Dataset	Accuracy	10.6 +/- 0.4	10.0 +/- 1.83	70.6 +/- 7.6	8.6 +/- 0.5	10.3 +/- 0.58	**8.3 +/- 0.8**
	Rank	5	3	6	2	4	**1**
Credit/G Dataset	Accuracy	14.7 +/- 0.58	16.4 +/- 4.24	30.5 +/-16.33	13 +/- 0.56	13.2 +/- 0.77	**12.2 +/- 0.51**
	Rank	4	5	6	2	3	**1**
diabetic disease	Accuracy	10.5 +/- 0.4	11 +/- 1.94	112.5 +/- 9.312	**9.5 +/- 0.4**	11.3 +/- 0.75	**9.5 +/- 0.37**
	Rank	3	4	6	**1**	5	**1**
Heart/Cleveland disease	Accuracy	9.6 +/- 0.69	10 +/- 2.49	28.3 +/- 4.347	**7.7 +/- 0.54**	9.2 +/- 0.8	8.4 +/- 0.64
	Rank	4	5	6	**1**	3	2
Heart/Stat log diabetic disease	Accuracy	9.6 +/- 0.58	7.8 +/- 1.39	25.9 +/- 4.30	**5.7 +/- 0.26**	8.7 +/- 0.62	7.5 +/- 0.37
	Rank	5	3	6	**1**	4	2
Hepatitis disease	Accuracy	8.1 +/- 0.48	7.9 +/- 1.44	11.6 +/- 2.31	7.8 +/- 0.65	8.1 +/- 0.71	**7.5 +/- 0.52**
	Rank	4.5	3	6	2	4.5	**1**
Ionosphere dataset	Accuracy	7.4 +/- 0.64	6.7 +/- 0.94	**2.2 +/- 0.42**	5.9 +/- 0.92	7.1 +/- 0.46	6.2 +/- 0.42
	Rank	6	4	**1**	2	5	3
Iris dataset	Accuracy	3.4 +/- 0.27	3.4 +/- 0.84	3.3 +/- 0.94	3.3 +/- 0.21	3.4 +/- 0.27	**3 +/- 0.15**
	Rank	5	5	2.5	2.5	5	**1**
Lymphography medical imaging	Accuracy	9.1 +/- 0.5	11.2 +/- 2.39	42.8 +/-6.48	**5.2 +/- 0.39**	8.8 +/- 0.65	8.9 +/- 0.77
	Rank	4	5	6	**1**	2	3
Mushroom dataset	Accuracy	9.3 +/- 1.25	**4.7 +/- 0.94**	33.4 +/- 2.87	7 +/- 0.26	8.2 +/- 0.47	7 +/- 0.45
	Rank	5	**1**	6	2.5	4	2.5
Segment dataset	Accuracy	21.9 +/- 0.69	22.5 +/- 4.30	59.3 +/- 7.9	20.9 +/- 1.39	24.8 +/- 1.2	**20.1 +/- 0.82**
	Rank	3	4	6	2	5	**1**
Sonar dataset	Accuracy	10 +/- 0.49	9.7 +/- 1.15	**0.9 +/- 1.97**	7.5 +/- 0.37	10.4 +/- 0.62	9 +/- 0.56
	Rank	5	4	**1**	2	6	3
Tic/tac/toe	Accuracy	10.7 +/- 1.69	10.7+/- 4.34	53.6 ± 7.306	6.8 +/- 0.81	12.6 +/- 1.31	**4.3 +/- 0.7**
	Rank	3	3	6	2	5	**1**

Table 5 and Fig. 6 show the result of Holm's post hoc and Friedman's nonparametric test to illustrate the second benchmark scenario. For this evaluation test, the average classification accuracy rank and average model size rank of the statistical results across the 16 datasets are computed and listed in Table 5.

TABLE V
TEST RESULTS OF THE NONPARAMETRIC TEST FOR GA-ANT MINER AND OTHER CLASSIFIERS

	Ant-Miner	CAnt-Miner	ACO/PSO 2	TACO	Hybrid Prune	GA-Ant-Miner
Accuracy	4.09	3.12	3.59	4.53	4.09	**1.56**
Terms	4.34	3.81	5.15	1.78	4.21	**1.68**

Fig. 6 shows that the results obtained by the GA-Ant Miner classifier outperform those of the other five classifiers in terms of classification accuracy and the number of discovered rules. Therefore, the GA-Ant Miner has a dominant result in comparison with other classifiers.

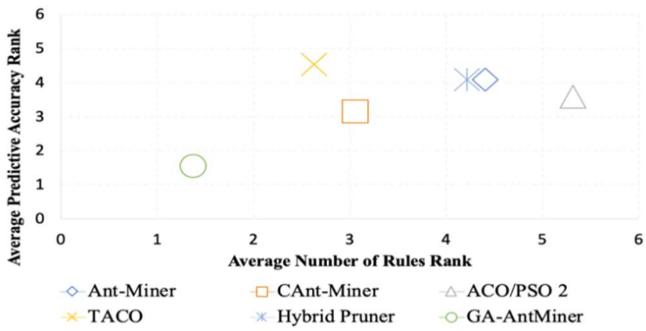

Fig. 6 Results of GA-Ant Miner on the average classification accuracy rank versus the average model size rank

Fig. 6 proves that the result obtained by the GA-Ant Miner outperforms those of the other classifiers when considering the classification accuracy and model size ranks. The GA-Ant Miner only performs slightly better than the TACO classifier in terms of model size. Still, it is significantly better than TACO and the other classifiers in terms of classification accuracy. Therefore, GA-Ant Miner is the dominant classifier that balances the classification accuracy and model size. This result is due to the enhancement of the post-pruning technique by using the GA algorithm concepts (i.e., crossover and mutation) to overcome the nesting effect's problem and find the best fitting rule by minimizing the number of terms based on the classification accuracy.

IV. Conclusion

This research introduced a new ACO-based rule classification algorithm, that is, the GA-AntMiner. The experimental results showed that our proposed GA-AntMiner significantly outperforms the well-known Ant-Miner, ACO/PSO2, TACO-Miner, CAnt-Miner, and Ant-Miner with hybrid pruner classification algorithms in terms of classification accuracy and model size. Moreover, using the new pruning technique based on the GA concept enabled the GA-AntMiner to be more flexible than the other classifiers. Future research directions are to adapt the parameter value (i.e., mutation rate and crossover rate) on the fly rather than maintaining a constant value to find the best classification rule. This task is essential in the rule classification technique to adjust the dataset's parameter values in designing a classification model.

Acknowledgment

The authors would like to thank the Malaysian Ministry of Higher Education for funding this study under the Transdisciplinary Research Grant Scheme, TRGS/1/2018/UUM/02/3/3 (S/O code 14163).

References

[1] H. N. K. Al-behadili, "Intelligent Hypothermia Care System using Ant Colony Optimization for Rules Prediction," *J. Univ. Babylon*, vol. 26, no. 2, pp. 47–56, 2018.

[2] L. Yang, K. Li, W. Zhang, and Z. Ke, "Ant colony classification mining algorithm based on pheromone attraction and exclusion," *Soft Comput.*, pp. 1–13, 2016.

[3] 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.

[4] 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.

[5] 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.*, vol. 98, no. 04, pp. 657–671, 2020.

[6] N. C and S. V, "A Study on Applications of Machine Learning Techniques in Data Mining," *Shodhshauryam, Int. Sci. Ref. Res. J.*, vol. 1, no. 3, pp. 31–34, 2005.

[7] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "An improved ACS algorithm for data clustering," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 17, no. 3, pp. 1506–1515, 2020.

[8] A. M. Jabbar and K. Ku-Mahamud, "Ant-based sorting and ACO-based clustering approaches: A review," in *In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 217–223.

[9] A. M. Jabbar, R. Sagban, and K. R. Ku-Mahamud, "Balancing Exploration and Exploitation In ACS Algorithms for Data Clustering," *J. Theor. Appl. Inf. Technol.*, vol. 97, no. 16, pp. 4320–4333, 2019.

[10] 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.

[11] H. N. K. Al-behadili, "Classification Algorithms for Determining Handwritten Digit," *Iraqi J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.

[12] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Ant colony optimization algorithm for rule-based classification: Issues and potential solutions," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 21, pp. 7139–7150, 2018.

[13] H. N. K. Al-behadili, K. R. Ku-Mahamud, and R. Sagban, "Rule pruning techniques in the ant-miner classification algorithm and its variants: A review," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018, pp. 78–84.

[14] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 321–332, 2002.

[15] B. Venkatesh and J. Anuradha, "A Review of Feature Selection and Its Methods," *Cybern. Inf. Technol.*, vol. 19, no. 1, pp. 3–26, 2019.

[16] Y. B. W. Wah, N. Ibrahim, H. A. Hamid, S. Abdul-Rahman, and S. Fong, "Feature selection methods: Case of filter and wrapper approaches for maximising classification accuracy," *Pertanika J. Sci. Technol.*, vol. 26, no. 1, pp. 329–340, 2018.

[17] A. A. Abdoos, P. K. Mianaei, and M. R. Ghadikolaei, "Combined VMD-SVM based feature selection method for classification of power quality events," *Appl. Soft Comput. J.*, vol. 38, pp. 637–646, 2016.

[18] 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.

[19] R. Parpinelli, H. Lopes, and A. A. AFreitas, "Data Mining With an Ant Colony Optimization Algorithm," *IEEE Trans. Evol. Comput.*, vol. 47, no. 6 (4), pp. 321–332, 2002.

[20] R. Saian and K. R. Ku-Mahamud, "Ant colony optimization for rule induction with simulated annealing for terms selection," *Proc. - 2012 14th Int. Conf. Model. Simulation, UKSim 2012*, no. March 2012, pp. 33–38, 2012.

[21] K. M. Salama and A. M. Abdelbar, "Extensions to the Ant-Miner classification rule discovery algorithm," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6234 LNCS, pp. 167–178, 2010.

[22] D. Dua and T. Karra, "UCI Machine Learning Repository," *Irvine, CA: University of California, School of Information and Computer Science*, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[23] F. Otero, A. A. Freitas, and C. G. Johnson, "cAnt-miner: An ant colony classification algorithm to cope with continuous attributes," in *International Conference on Ant Colony Optimization and Swarm Intelligence.*, 2008, pp. 48–59.

[24] N. Holden and A. A. Freitas, "A Hybrid PSO/ACO Algorithm for Discovering Classification Rules in Data Mining," *J. Artif. Evol. Appl.*, vol. 2008, pp. 1–11, 2008.

[25] K. Thangavel and P. Jaganathan, "Rule Mining Algorithm with a New Ant Colony Optimization Algorithm," *Int. Conf. Comput. Intell. Multimed. Appl. ICCIMA 2007*, vol. 1, pp. 561–563, 2007.

[26] S. Tripathy, S. Hota, and P. Satapathy, "MTACO-Miner : Modified Threshold Ant Colony Optimization Miner for Classification Rule Mining," in *Emerging Reserch in Computing, Information, Communication and Application*, 2013, pp. 1–6.

[27] A. Chan and A. Freitas, "A new classification-rule pruning procedure for an Ant Colony Algorithm," in *In International Conference on Artificial Evolution (Evolution Artificielle)*, 2006, vol. 3871 LNCS, pp. 25–36.

[28] R. Robu, C. Vacar, N. Robu, and S. Holban, "A study on Ant Miner parameters," in *6th International Conference on Information, Intelligence, Systems and Applications*, 2016.

[29] M. López-Ibáñez, T. Stützle, and M. Dorigo, "Ant Colony Optimization: A Component-Wise Overview," Bruxelles, Belgium, 2016.

[30] M. L. Raymer, W. Punch, E. Goodman, L. Kuhn, and A. Jain, "Dimensionality reduction using genetic algorithms - Evolutionary Computation," *IEEE Trans. Evol. Comput.*, vol. 4, no. 2, pp. 164–171, 2000.