

# Adaptive Parameter Control Strategy for Ant-Miner Classification Algorithm

Hayder Naser Khraibet AL-Behadili<sup>1</sup>, Rafid Sagban<sup>2</sup>, Ku Ruhana Ku-Mahamud<sup>3</sup>

<sup>1</sup>Computer Science Department, Shatt Alarab University College, Basra, Iraq

<sup>2</sup>Department of Software, University of Babylon, Babylon, Iraq

<sup>3</sup>School of Computing, Universiti Utara Malaysia, Kedah, Malaysia

---

## Article Info

### Article history:

Received Jun 27, 2019

Revised Feb 14, 2020

Accepted Mar 11, 2020

---

### Keywords:

Rule Induction

Data Mining

Parameter Control

Metaheuristic

Swarm Intelligent

Ant Colony Optimization

---

## ABSTRACT

Pruning is the popular framework for preventing the dilemma of overfitting noisy data. This paper presents a new hybrid Ant-Miner classification algorithm and ant colony system (ACS), called ACS-AntMiner. A key aspect of this algorithm is the selection of an appropriate number of terms to be included in the classification rule. ACS-AntMiner introduces a new parameter called importance rate (IR) which is a pre-pruning criterion based on the probability (heuristic and pheromone) amount. This criterion is responsible for adding only the important terms to each rule, thus discarding noisy data. The ACS algorithm is designed to optimize the IR parameter during the learning process of the Ant-Miner algorithm. The performance of the proposed classifier is compared with related ant-mining classifiers, namely, Ant-Miner, CAnt-Miner, TACO-Miner, and Ant-Miner with a hybrid pruner across several datasets. Experimental results show that the proposed classifier significantly outperforms the other ant-mining classifiers.

Copyright © 2019 Institute of Advanced Engineering and Science.  
All rights reserved.

---

## Corresponding Author:

Hayder Naser Khraibet AL-Behadili,  
Computer Science Department,  
Shatt Alarab University College, Basea, Iraq;  
Email: [haider872004@gmail.com](mailto:haider872004@gmail.com)

---

## 1. INTRODUCTION

Machine learning (ML) is a data analysis technique and is intensively used for automating analytical model construction. ML performs a particular task by utilizing discovered patterns and inference without using explicit instructions. This technique is considered a branch of artificial intelligence. ML uses algorithms and statistical models to create a mathematical model on the basis of a given dataset. This model is known as a training model and will be used to make decisions or predictions for real-world problems [1]. In the ML term, a learning process is classified into two main instances, namely, unsupervised and supervised learning [2]. Unsupervised learning, also known as a clustering task, aims to analyze the given data into groups according to the behavior and similarities of data [3]–[5]. In contrast, supervised learning, also known as classification, is a powerful learning algorithm, where the target is the class of a specific data [6]. Classification is one of the studies in the field of ML with a wide variety of industrial and commercial applications. These applications include medical diagnosis, detection of spam emails, determination of bankruptcy, detection of intrusion, determination of handwritten digits, and face detection [7]–[10].

The algorithm that performs ML classification is known as a classifier algorithm. Numerous classifiers, such as decision trees, rule-based models, linear models, lazy evaluation methods, probabilistic, and nonlinear models, have been developed in the literature [11]. In the concepts of understandability, certain classifier algorithms produce classification models with high accuracy; however, the structures of the models are highly complex and difficult to understand. Other classification algorithms, including rule-based classification, produce an understandable model for users. A number of methods and techniques have been proposed; among which, the most successful is ant colony optimization (ACO), also known as the ant-mining

algorithm [12]–[14]. Ant-Miner uses ACO power to discover classification rules presented as IF <conditions> THEN <class>, where each single condition, also known as term, is represented as an attribute = value (e.g., blood pressure = high). Thus, the discovered rules have a natural language representation and are easy to interpret.

However, ant-mining algorithms face problems of over-fitting and under-fitting. Over-fitting occurs when the ant-mining algorithm effectively fits the data. However, generalizing the model to new data is difficult. Meanwhile, under-fitting occurs when the classification model cannot capture the underlying data trend [15]. Rule pruning is the framework used in ant-mining algorithms to avoid under-fitting and over-fitting problems [16]. The implementation of rule pruning is developed on the basis of different strategies, namely, post-pruning, pre-pruning, and hybrid pruning). The majority of ant-mining algorithms use post-pruning techniques, where the rule is discovered and undergoes a pruning procedure. This method is used to remove a term from the rule while improving its classification accuracy. This method scans the dataset to check the improvement in quality of the rule in each particular pruning. This procedure is repeated until a term is lifted in the rule, or no more improvement appears [17]. This procedure is time-consuming and complex when the dataset includes a large number of instances and attributes. Another important limitation of post-pruning is the use of greedy search biases to remove one term at a time rather than finding and pruning interrelated terms. Other studies have proposed a pre-pruning method in which irrelevant terms are removed in the construction process by using a threshold [18], [19]. The threshold value is part of the criteria, which is responsible for selecting the right terms during the construction. The threshold value used in the existing literature is based on information gain, which is data-dependent according to information theory. Selecting the right threshold is considered a formidable task while remaining constant at different search stages and for all datasets used in the experiments. Another research study proposed a hybrid pruner technique that integrated pre- and post-pruning methods [20]. The pre-pruning method integrates a user-defined parameter, which is responsible for determining the number of terms allowed in the construction rule. Then, the discovered rule undergoes the original post-pruning technique used in the majority of ant-mining classifiers. Given the limitations and drawbacks in post-pruning, the number of term parameters would seriously affect the discovered rule, which is critical. Any mistake would lead to an inappropriate model where a worse result would be reported.

This paper presents a novel hybrid Ant-Miner classification algorithm and ant colony system (ACS), called ACS-AntMiner. The important aspect of this algorithm is the selection of the appropriate number of terms to be included in the classification rule. ACS-AntMiner presents a new strategy to enhance the Ant-Miner algorithm by using an adaptive parameter control strategy. This study proposes a new parameter called IR ( $\zeta$ ), which is responsible for preventing unfit terms from inclusion. This parameter is based on the strength and importance of the particular term that needs to be included. Then, we propose a different search space to improve the selection of an appropriate IR ( $\zeta$ ) parameter value as a pre-pruning criterion. This mechanism considers the feedback from the rule construction process to select the parameter value and decide whether the value prevails in the search space. Post-pruning in the original Ant-Miner will be applied if further pruning is required. Therefore, the drawbacks of the post-pruning technique will be overcome, and the irrelevant terms are discarded by using our proposed technique.

The remaining sections of this paper are structured as follows. Section 2 introduces related works which includes the ACO algorithm and its applications to different domains. This also presents the main Ant-Miner components to generate the classification rule. The proposed modification is described in Section 3. The details about the materials and methods used in our experiments are provided in research method section. Analysis of the experimental results is presented in Section 5. Finally, section 6 discusses the conclusion and future research direction.

## 2. RELATED WORKS

ACO is a swarm-intelligence metaheuristic algorithm used to solve combinatorial optimization problems. This algorithm was first applied to the traveling salesman problem and has since been widely applied to a variety of NP-hard optimization problems, such as quadratic assignment problems, vehicle routing with time windows, grid computing, and data mining [21], [22]. The first ACO algorithm is the ant system, proposed by Dorigo et al. in the 1990s [23]. Real ants use the stigmergy method to find good paths when searching for a food source and deposit pheromones to indicate these paths. The amount of laid pheromones depends on the food quality and distance. As each ant randomly moves, other ants can detect previously visited paths and decide whether to visit them. The pheromones are then deposited along these paths. The process is characterized by positive feedback loop and stochastic behavior. Therefore, indirect communication among ants searching for food provides them with the ability to find the optimal path between the food source and the ants nest [24].

Artificial ants also behave as real ants and simulate their natural counterparts to seek for an optimal solution to the problem at hand. Each ant employs a stochastic constructive behavior and adopts two factors,

namely, the pheromone amount and heuristic information, to formulate its solution. Artificial ants also exhibit other features that cannot be found in real ants. Each ant has an internal memory to trace their previous action, and the local search procedure may be used to improve the quality of the solution produced by each ant [25]. The algorithmic outline of the ACO metaheuristic algorithm consists of three main procedures. The first procedure is to construct the solution, where each ant builds a full solution to the problem at hand by using the probabilistic state transition rule. The second procedure is the daemon action (found in many ACO variants), which is essential to obtain competitive results. An example of the daemon action is the local search procedure. The local search starts with an individual solution and iteratively applies changes to improve it. The third procedure updates the pheromone amount in two phases. The first phase deals with the pheromone amount that evaporates from every path, whereas the second increases the pheromone amount on the path that leads to high-quality solutions [26].

The concept of ACO classification has been used to express useful information from data. The information is presented as a rule-based model in the form of IF <conditions> THEN < class>. The rule antecedent (IF) consists of a set of conditions, which is connected by a logical conjunction operator (AND), that is, IF term1 AND term2 AND... Each term is a triple <attribute, operator, value> (i.e., <Gender = Male>). The first developed classifier algorithm, called the Ant-Miner, was introduced by Parpinelli, Lopes, and Freitas (2002) to discover classification rules by using the concept and principles of the ACO algorithm, divide and conquer strategy and ML [27]–[29]. Ant-Miner uses certain ACO procedures to generate a number of rules and determine the best option for a given set of training instances. The process consists of creating one rule for each ant and selecting the best option to include in the discovered rule list. The instances covered by the best rule are separated from the set of training instances. The process is repeated until all training instances are covered by the discovered rule list. The algorithmic outline of the Ant-Miner algorithm contains three importance components, known as construction rule, pruning procedure, and pheromone update procedure.

Rule construction refers to the routines that ants use to incrementally construct a classification rule. After the starting term is selected, one term is added to the ant path. Term selection depends on the available pheromone amount and heuristic information. The ants' solutions are chosen based on the values of pheromone and heuristics information. The higher the values of those factors, the more chance the solution to be selected to be included in the classification rules. The construction graph in this algorithm is fully connected where each term is presented as a node, and the possible paths connect the nodes. Therefore, each ant creates its own path, which represents a classification rule. Figure 1 shows an example of two solutions constructed by ants, which represents the following classification rules:

1. IF attribute 1 = v1, 1 AND attribute 2 = v2, 1 AND attributeN = vn, 1 THEN class = Class1;
2. IF attribute 1 = v1, 2 AND attribute2 = v2, 2 AND attributeN = vn, n THEN class = Class3.

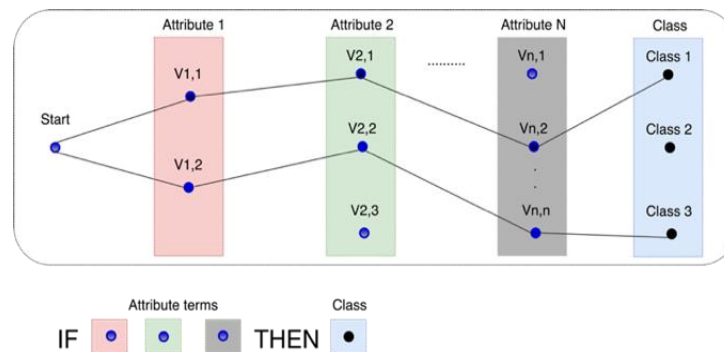


Figure 1. Generation of the classification rule in Ant-Miner

Rule pruning, a popular procedure in the rule-discovery process, aims to reduce the complexity of the discovered rules by avoiding irrelevant terms, thereby boosting the algorithm's performance. The pruning method removes irrelevant terms after the rules are developed to their maximum sizes. An example of a rule from the ML repository of the University of California Irvine (UCI) dataset called United States Congressional Voting is shown in Figure 2. The rule is produced using the Ant-Miner classifier without the use of rule pruning. The dataset includes 16 attributes, all of which occur in the discovered rule. The pruning procedure deletes one term at a time from the rule to improve its quality. This procedure will be iterated as long as more than one term is left in the rule, or additional improvement occurs. An example of a pruned rule is shown in Figure 3, which presents the same rule in Figure 2. The rule is simple and exhibits few terms but covers additional instances and is accurate.

```

IF HandicappedInfants = 'n' AND WaterProjectCostSharing = 'y' AND AdoptionBudgetResolution
= 'n' AND PhysicianFeeFreeze = 'y' AND ELsalvadorAid = 'y' AND Religious-groupsInSchools =
'y') AND AntiSatelliteTestBan = 'n' AND Aid-to-NicaraguanContras = 'n' AND mx-Missile = 'n'
AND Immigration = 'y' AND Synfuels-Corporation-Cutback = 'n' AND Education-Spending = 'y
' AND SuperfundRightToSue = 'y' AND crime = 'y' AND Duty-Free-Exports = 'n' AND
ExportAdministrationActSouthAfrica = 'y' THEN Republican

```

Figure 2. Classification rule using Ant-Miner algorithm without pruning step

```

IF PhysicianFeeFreeze = 'y' THEN Republican

```

Figure 3. Example of a pruned classification rule

Pheromone updating is the indirect mechanism used to achieve communication among ants in the colony. This procedure is responsible for awarding the pheromone amount to good-quality rules in two steps. First, the pheromone values are decreased to avoid an unlimited pheromone amount from accumulating which allow the ant colony to forget previously constructed low-quality rules. The pheromone is then deposited to good rules constructed by the ant. During the learning process, the ants converge to high-quality rules.

A number of modifications have been proposed in constructing ACO rule-based classification. Liu et al. (2002) introduced a less accurate heuristic function to reduce the algorithm's computational time [30]. An extension was proposed to increase the exploration of the search space by adapting the state transition strategy [31]. Jiang et al. (2005) proposed three major modifications on the pheromone update strategy, heuristic function, and transition rule [32]. Another study proposed two strategies, namely, punishing and mutation operators [33]. Chan and Freitas (2006) proposed a modification in the rule-pruning procedure for use with big data [20]. Another modification was performed by Chan and Freitas (2006b) for multilabel data classification [34]. Smaldon and Frietas (2006) proposed a strategy to determine the class consequent in advance and then generate classification rules for each particular class [35]. Another work introduced the contribution of multiple ant colonies rather than a single one using the original Ant-Miner [36]. Antminer+ has been developed with different aspects in which the pheromone is initialized with the upper trail limit, and upper and lower boundaries are introduced to the amount of deposited pheromones. The pheromones are updated using a strong elitist strategy, which can be an iteration of the best ant or the global-best ant. Antminer+ also exhibits a different heuristic function, new pheromone updating strategy, and new self-adaptive mechanism to weigh the  $\alpha$  and  $\beta$  parameters. Lastly, Antminer+ defines the environment as a directed acyclic graph, where all other algorithms use a fully connected graph [37], [38]. The threshold criterion was proposed to determine the importance of each term for including or rejecting them from inclusion [18], [19]. Another study was proposed to handle the continuous attributes on the run of the algorithm rather than discretization in the pre-processing step [39], [40]. A study was proposed to modify the rule construction process by considering the correlation between terms [41]. Another research was conducted in medical diagnosis in which the Ant-Miner was combined with the fuzzy system [42]. Agravat et al. (2010) proposed Ant-Miner for intrusion detection, with three modifications, namely, pheromone update procedure, transition rule, and fitness function [43]. Salama and Abdelbar (2010) produced a multi-pheromone concept [44]. Liang et al. (2011) introduced a multistage rule-choosing method to select the rule set [45]. Another research study proposed a hybrid between the Ant-Miner and simulated annealing algorithm by incorporating the latter in the rule construction process [46]. A study proposed the implementation of ant population rather than a single ant to discover the classification rule in each iteration [47]. Baig and Shahzad (2012) introduced a different heuristic function according to the correlation between the terms. The study proposed a random selection method to determine the values of the  $\alpha$  and  $\beta$  parameters [48]. Other studies have determined the class consequent in advance and then constructed the rule for this class, allocating different types of pheromone for each class [49]. Different quality functions, and each ant in the colony must select the quality functions before the rule construction process, as proposed by Rajpiplawala & Singh (2014) [50]. The Laplace correction heuristic function was proposed to replace the original heuristic function in the Ant-Miner algorithm [51]. Another study proposed a heuristic function that considers the correlation between attributes and instance coverage [52]. A related study also proposed different pheromone update procedures that used an evaporation factor which dynamically updates during the search process [17].

This study aims to provide an adaptive strategy to control the important terms online to be included in the construction rule on the basis of the existing literature in the area of ACO algorithms for rule-based classification. This study also aims to improve the stability of the algorithm and classification performance and simplify the discovered rules through the proposed strategy.

### 3. THE PROPOSED ACS-ANTMINER CLASSIFICATION ALGORITHM

Our proposed technique introduces a new parameter called the importance rate (IR), responsible for adding only the important terms to each rule. The IR is a pre-pruning criterion based on the probability (heuristic and pheromone) amount. If the probability amount of the selected term is below the IR criterion, then the term is rejected from inclusion; otherwise, it is added. This technique handles irrelevant terms during learning instead of finding a complete rule and then pruning it. Each ant is assigned its own IR value during the model learning process. The IR value is modified over time to ensure diversity and consider the different search stages. The other terms that can be added to the rules are then explored, and the learning process is prevented from being stuck in local optima.

The ant colony system (ACS) algorithm is designed to optimize the IR parameter. An ant solution represents the IR parameter of the Ant-Miner classifier. The classification performance of the Ant-Miner classifier on the training dataset is used to design the feedback collection strategy of the ACS. Figure 4 shows the main stages in the implementation of the proposed ACS-based parameter selection for the Ant-Miner classifier.

1. Pheromone is initialized on a feedback collection table of the ACS-based parameter selection algorithm.
2. Parameters of the IR value are selected for each ant in the Ant-Miner classifier.
3. Ant-Miner constructs a classification rule on the basis of the selected parameter.
4. Feedback collection starts to collect local and global feedback on the basis of the quality of the discovered rule.
5. ACS algorithm uses the feedback to reward the parameter value according to its performance in the Ant-Miner classifier. The stages in Figure 4 are discussed in detail in the following paragraphs.

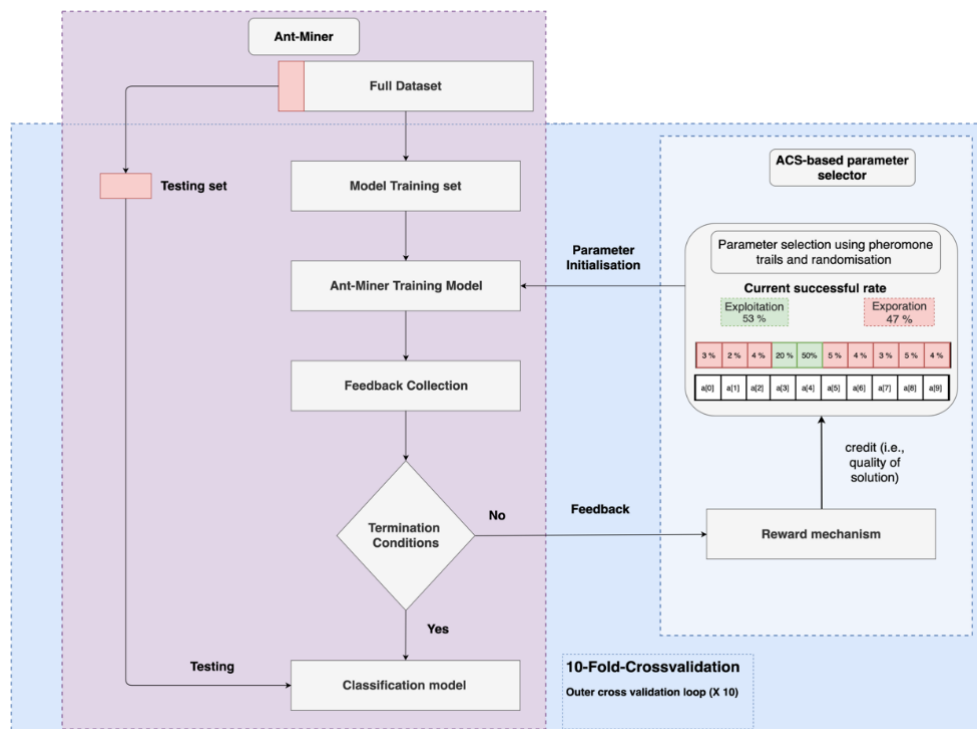


Figure 4. System architectures of the proposed ACS-AntMiner classification algorithm

The adaptive parameter selection strategy describes the application of ACO concept which collects the feedback from the classification process to adjust the threshold of pre-pruning in the Ant-Miner classifier. This dynamic adjusted threshold is called *importance rate* ( $\zeta$ ). This parameter controls the irrelative terms during the rules construction phase of the classifier according to its probability (i.e., the heuristic and the pheromone information). To understand the strategy of  $\zeta$  selection, three vectors have been developed, a vector of threshold values is denoted as  $ParametersValue(vn) = \{v1, v2, v3, \dots, vn\}$  which represents various  $\zeta$  values, and another vector of probabilities for each parameter value  $vn$ , denoted as  $ParametersProbabilities(vn) = \{p(v1), p(v2), p(v3), \dots, p(vn)\}$ , which determines the selection probability of each parameter value, and a vector to collect the local and global quality of the threshold values, denoted as  $feedbackCollection(vn) = \{q(v1), q(v2), \dots, q(vn)\}$ . When each threshold value is used  $vn$ , then its current quality  $q(vn)$  is updated.

Initialization: The pheromone initialization are required to design the transition probabilities of the ACS-Ant-Miner in establishing the classification model. First, the feedback collection vector is initiated with a small value using equation.

$$\tau_n(t = 0) = \frac{1}{n} \tag{1}$$

where  $n$  is the size of feedback vector.

Parameter selection: The new ACS selection transition rule is introduced as shown in (2). This is in accordance with the pheromone value that is considered for parameter value selection in our proposed ACS-based parameter selection algorithm.

$$P(v_n) = \begin{cases} \text{arcm}ax_{v_n} \in P & \text{if } q < q_0 \text{ (exploitation)} \\ R, & \text{otherwise ( exploration)} \end{cases} \tag{2}$$

where  $q_0$  is a parameter of the controlling state transition rule which will gradually increase between  $[0.1, \dots, 0.9]$ . The maximum (max) and minimum (min) values of the range identified experimentally in this work by using the following equations.

$$q_0 = (\text{max}) e^{-t \frac{\ln(a)}{a}} \tag{3}$$

where  $a$  is the number of ants and  $t$  is the index of the current ant in the ACS. This equation is set as  $q_0$  with min value to increase the exploration of the ACS algorithm at the beginning of the construction process. During the construction process, each ant is assigned its own  $q_0$  value. Over time, the value of  $q_0$  will gradually increase and be stopped at the value of max.  $q$  which is a random value uniformly distributed in between  $[0, \dots, 1]$ .  $R$  is a randomly chosen threshold value which is a pure exploration instead of the biased exploration calculated by the original formula of the state transition rule [53]. Figure 5, below, shows the process of controlling the state transition rule by the  $q_0$  parameter.

$P_n$  is the probability of selecting a specific value from available values and is simplified in Figure 6, which is calculated by using Equation 4.

$$P_n = \frac{[\tau_n(t)]}{\sum_{n=1}^t [\tau_n(t)]} \tag{4}$$

where  $[\tau_n(t)]$  is the amount of the quality associated to each threshold value to probabilistically select one of them.

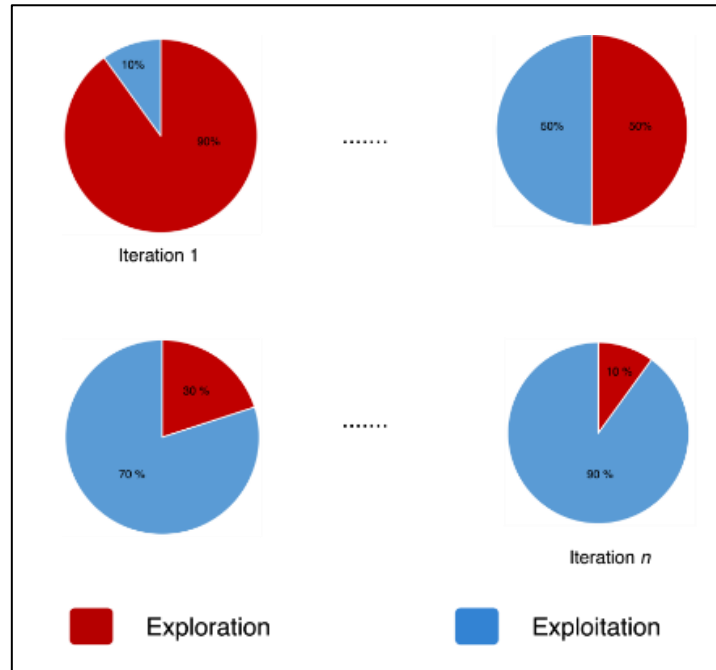


Figure 5. Process of controlling the state transition rule

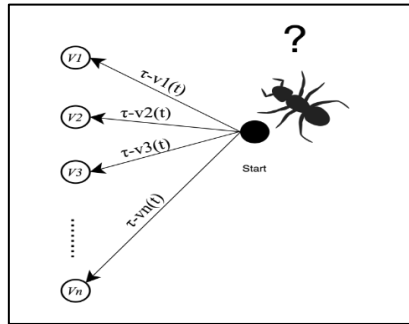


Figure 6. Construction graph with probabilities employed by the adaptive pre-pruning selection technique

An example of the probabilities result is shown in Table 1. The first column represents the available threshold values while the second column represents the quality of the threshold values based on the feedback collected locally and globally. The last column represents the probability of obtaining a specific threshold value. Thus, the use of the probabilities was able to help in deriving conclusions about the causes of the successful or unsuccessful performance in different threshold values. For example, the success rate of the threshold value is  $v_7$ , which is higher than the other. This means that threshold value  $v_7$  has a higher probability of being selected.

Table 1. Conditional probabilities of threshold values

Threshold	Quality	Probabilities
$v_1$	0.01	0.34
$v_2$	0.5	16.95
$v_3$	0.22	7.46
$v_4$	0.3	10.17
$v_5$	0.4	13.56
$v_6$	0.2	6.78
$v_7$	0.9	30.51
$v_8$	0.21	7.12
$v_9$	0.1	3.39
$v_{10}$	0.11	3.73

Ant-Miner classifier: The Ant-Miner starts to construct one classification rule from the training dataset and adds it to a list of discovered rules. The instances covered by the construction rule are then removed from the training set. This process is iterated until the number of uncovered instances is larger than the maximum uncovered cases. The discovered classification rule process includes three main stages, called, rule construction, pruning procedure, and the pheromone update procedure.

Rule construction: Each ant will create one rule by adding one term at a time in the rule. The term selection uses (5) which is based on the heuristic information and pheromone amount associated with each term. Addition of terms will stop when all the attributes have been visited, or any additional term inserted into the construction rule would cover less than the allowed number of minimum cases per rule. A term is selected using the Ant-Miner probability [50].

$$\text{Probability} = \frac{[\tau_{ij}(t)] [\eta_{ij}]}{\sum_{i=1}^a x_i \cdot \sum_{j=1}^{b_i} [\tau_{ij}(t)] [\eta_{ij}]} \tag{5}$$

where  $[\tau_{ij}(t)]$  is the pheromone amount associated with term  $ij$  at iteration  $(t)$ ;  $[\eta_{ij}]$  is the heuristic information for each term;  $a$  is the attribute number in the dataset;  $b_i$  is the number of different values for each attribute; and  $x_i$  is set to 1 if attribute  $a_i$  is not yet visited by ant, or 0 otherwise.

The heuristic function and the pheromone amount are used to select terms in the rule construction process. The heuristic function in the Ant-Miner classifier is inspired from information theory. This function computes the amount of information contained in each term (entropy). The heuristic function is defined by (6)

$$\eta_{ij} = \frac{\log_2 k - H(W|A_i = V_{ij})}{\sum_{i=1}^a \sum_{j=1}^{b_i} (\log_2 k - H(W|A_i = V_{ij}))} \tag{6}$$

where  $a$  and  $b_i$  have the same meaning as illustrated above;  $k$  is the number of classes; and  $W$  is the class attribute, that is, the attribute whose domain consists of the classes to be predicted.

Rule pruning: The rule constructed by each ant is pruned to delete irrelevant terms. This procedure utilizes one term elimination method to delete irrelevant terms from the current rule. The predictive class of the pruning rule can be changed if the number of instances covered by the pruning rule is different from the instances covered by the original rule. Thus, the classifier determines the predictive class by assigning the majority class among the instances covered by the rule. The quality of the new rule is then calculated. This process will be repeated while improving the rule quality, or only one term is left in the rule.

The ACS-based parameter selection collects feedback using the Ant-Miner construction rule by applying our proposed local feedback collection as in (7).

$$\tau_{x(t+1)} = (1 - \rho) \cdot \tau_{x(t)} + \rho \cdot Q(t) \quad (7)$$

where the  $\rho$  value indicates the pheromone evaporation rate for avoiding unlimited accumulation on specific parameter value, and  $Q(t)$  is the quality of the discovered rule.

Ant-Miner pheromone updating: The pheromone amount for each term is updated, thereby depositing pheromone in the term, followed by each ant according to rule quality. The effect of pheromone evaporation for unused terms is carried out by dividing the value of each  $\tau_{ij}$  by the summation of all values of  $\tau_{ij}$ . The pheromone update is defined as following formulas:

$$\tau_{ij(t+1)} = \tau_{ij(t)} + \tau_{ij(t)} \cdot Q \quad (8)$$

$$Q = \frac{TP}{TP+FN} * \frac{TN}{FP+TN} \quad (9)$$

$$\tau_{ij(t+1)} = \frac{\tau_{ij(t)}}{\sum_{i=1}^a \sum_{j=1}^{b_i} \tau_{ij(t)}} \quad (10)$$

where  $a$  and  $b_i$  have the same meaning as illustrated above;  $Q$  is the quality of the discovered rule;  $TP$  is the case that is covered and correctly predicted by the rule. In contrast,  $FP$  is the case that is covered and predicted incorrectly by the rule,  $FN$  is the case that is uncovered but predicted correctly by the rule and  $TN$  is the case that is uncovered and predicted incorrectly by the rule.

Once the first ant finishes this process, the second ant starts with a different  $IR$  ( $\zeta$ ) value. This process will be iterated until all ants in the colony have discovered all rules, or the current ant builds a rule that is exactly similar to the rule constructed by the previous ant, thereby increasing the number and exceeding the pre-defined number of converged rules.

The best rule among all discovered rules of all ants in the current iteration will be added into the list of discovered rules. The ACS-based parameter selection collects global feedback according to the best rule discovered in the current iteration using the proposed (11).

$$\tau_{x(t_{best})} = (1 - \rho) \cdot \tau_{x(t_{best})} + \rho \cdot Q(t_{best}) \quad (11)$$

where  $\rho$  is the pheromone decay parameter, and  $Q(t_{best})$  is the quality of the best discovered rule. A new iteration will start with the same activities.

#### 4. RESEARCH METHOD

A ten-fold cross-validation method is used in our experiments. In this method, the dataset is split into ten (10) subsets. Each subset is equally sized, where nine (9) are used for the training process. The remaining subset is used in the testing stage. This process is repeated ten times with a different subset for training and testing to ensure that all subsets will be used in training and testing. Subsequently, the performance of all folds is averaged, and the standard deviations are computed. The ten-fold cross-validation method has been also adopted and adapted and used in other ant-mining classifier studies [14], [54].

The evaluation in this study is performed on the basis of three criteria. First, the classification accuracy in discovering the rule list is called the correct classification rate. This criterion is based on the correctly classified instances in the test data. Each time, the training subsets consist of  $n$  number of instances, the classifier constructs the training and test subsets that will be used to test the performance. The correct classification instances will determine the performance of the proposed classifier. Second, the size in discovering the rule list, which is measured by the amount of terms per rule. The number of terms (conditions) refers to the number of antecedents carried by each rule. Third, the observation of the performance of the algorithms in the classification accuracy and model size. The average classification accuracy rank versus the



average model size rank of all classifiers is used in our experiments. A lower average rank implies better algorithm performance.

We perform several experiments using 16 UCI benchmark datasets to test the performance of our hybrid classification algorithm. The benchmark datasets are selected in accordance with Ant-mining literature. These benchmarks are secondary datasets that have been chosen from UCI [55]. These datasets demonstrate different attribute numbers, which lie between 4 and 60. The attributes exhibit continuous and categorical types. The datasets differ in size of instance number within the range of 148–8124 and class label numbers. The main descriptions of the experimental datasets are listed in Table 2.

Table 2. Characteristics of datasets used in the experiment

Datasets Name	Description		
	Instances Number	Attributes Number	Classes Number
Balance Scale	625	4	3
Breast Cancer (Ljubljana)	286	9	2
Breast Cancer (Wisconsin)	699	9	2
Credit-a	690	15	2
Credit-g	1000	20	2
Diabetes	768	8	2
Heart (Cleveland)	303	13	5
Heart (Statlog)	270	13	2
Hepatitis	155	19	2
Ionosphere	351	34	2
Iris	150	4	3
Lymphography	148	18	4
Mushroom	8124	22	2
Segment	2310	19	7
Sonar	208	60	2
Tic-tac-toe	958	9	2

Furthermore, we use different ant-mining classifiers to determine the effect of the proposed strategy on the classification accuracy and the classification model size against related and state-of-the-art classifiers in literature. The applied classifiers are Ant-Miner and other related ant-mining algorithms, namely, CAnt-Miner, TACO-Miner, and Ant-Miner with a hybrid pruner. The CAnt-Miner classification algorithm is an extended version of the Ant-Miner classifier that integrates an entropy discretization method in handling continuous features during construction [39]. The TACO-Miner provides a threshold value on the basis of the information gain of each term. If the information associated with the selected term is below the threshold value, then the term will be rejected. The threshold value is considered a pre-pruning criterion that determines the acceptance or rejection of terms [18], [19]. The Ant-Miner with a hybrid pruner presents a new hybrid rule pruner that hybrids the original Ant-Miner's rule pruner with a rule pruner based on two aspects, namely, information gain and number of terms allowed in the rule called  $r$  value. The new method is applied to each rule that exceeds the number of terms allowed in a rule. The number of terms in the selected rule is then reduced until it is equal to the  $r$  value. This method is implemented on the basis of information gain and roulette wheel for selection. Thereafter, the procedure of the Ant-Miner's original rule pruner is applied [20].

Finally, the parameter values for the Ant-Miner classifiers are adopted from Raul et al. (2016) and López-Ibáñez et al. (2016) to ensure that each classifier will work with the same parameter values and provide a fair evaluation of the results [56], [57]. The parameters used for the Ant-Miner classifiers are listed in Table 3.

Table 3. Experimental parameters

Parameter	Description	Value
NA	Number of ants in the colony	10
MCR	Minimum number of cases that each rule must cover	5
MUC	Maximum of uncovered cases by the discovered rules	10
RC	Number of rules for convergence	10
NI	Number of iterations	10
$\rho$	Evaporation rate	0.32
$\alpha$	Alpha	1
q0	The parameter of exploration/exploitation	0.53

### 5. RESULTS AND DISCUSSION

The results of classification accuracy are depicted in Figure 7 which shows that the proposed ACS-AntMiner obtained the best result in 12 out of 16 datasets. The proposed algorithm achieves second-best in two datasets, namely, lymphography and segment. Our classifier is competitive in two datasets, namely, Credit-g and Mushroom where the second best results have been obtained. CAnt-Miner classifier was able to obtain the best result in three (3) datasets. Figure 8 depicts the summary of the best classification accuracies for all algorithms.

With regard to the size of the classification model, rule-based classification complexity refers to the number of terms included in a given classification model. As shown in Figure 9 & Figure 10, the TACO-Miner, which discovers the classification model, obtains the lowest size in ten (10) out of 16 datasets. Our proposed algorithm achieves the lowest size in six (6) over 16 datasets. In addition, our proposed algorithm obtains the second-best lowest size in five (5) datasets, namely, Balance Scale, Heart (Cleveland), Heart (Statlog), Sonar, and Tic-tac-toe. However, the no-free-lunch theorem states that no algorithm performs best on all datasets [12], [58].

In summary, the results achieved by the ACS-AntMiner are good because the classification accuracy achieves the best and outperforms the other classifiers in the majority of the datasets. The classification model size indicates that the ACS-AntMiner only performs slightly lower than the TACO-Miner and significantly better than the remaining compared classifier.

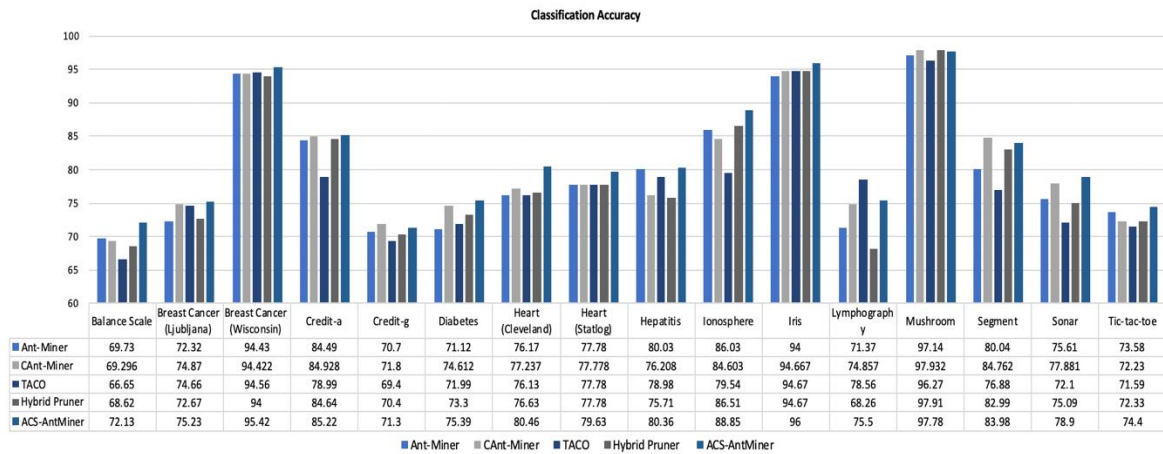


Figure 7. Comparison of the classification accuracy

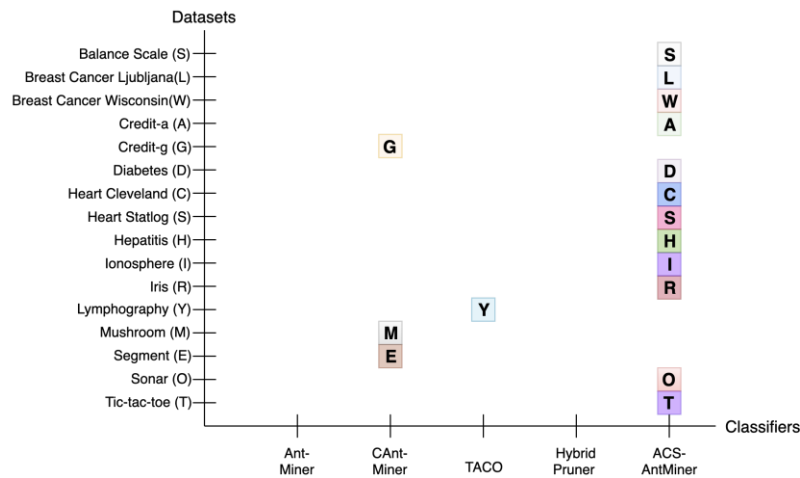


Figure 8. Performance summary of classification accuracy

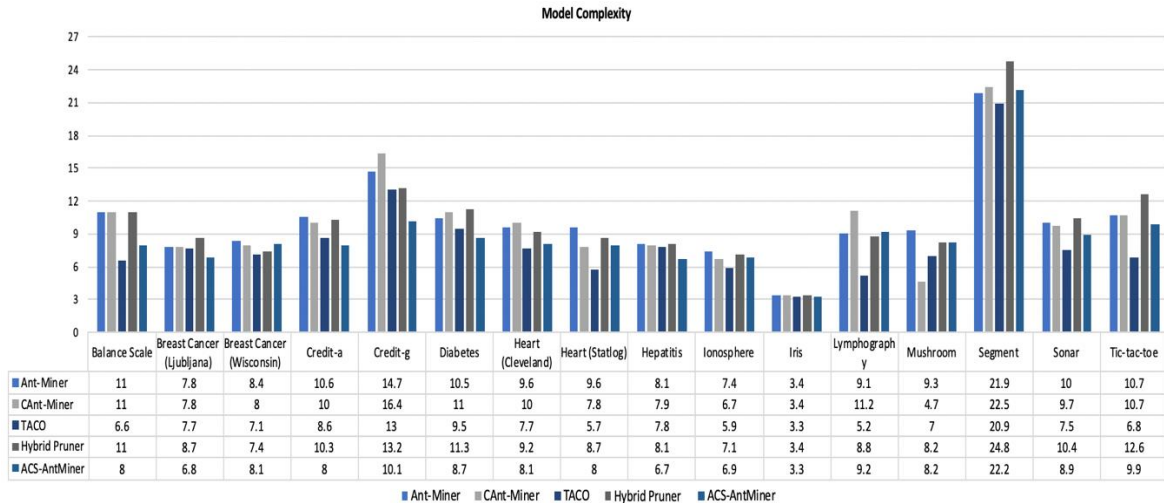


Figure 9. Comparison of the number of terms

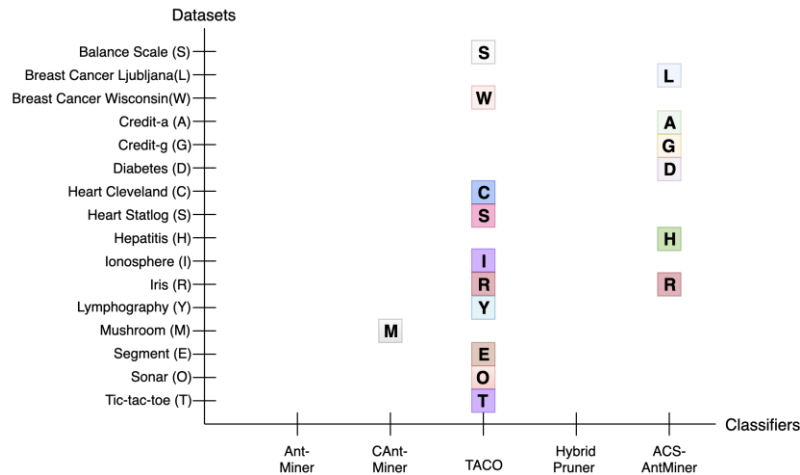


Figure 10. Performance summary of the number of terms in a rule

Table 4 shows the results averaged over all the datasets, with the best performances shown in bold. This table includes the average classification accuracy and rule term numbers. The proposed classifier exhibits the best classification accuracy than the other classifiers. The TACO-Miner obtains the best result for a number of terms. The ACS-AntMiner achieves the second-best in the average number of terms in the overall datasets.

Performance	Ant-Miner	CAnt-Miner	TACO	Hybrid Pruner	ACS-AntMiner
Accuracy	79.65	80.50	78.67	79.46	81.90
Terms #	10.13	9.92	8.14	10.2	8.81

We assign the average accuracy ranking against the average model size rank over the different datasets Figure 11. A lower average rank indicates a better performance of the algorithms. Figure 11 proves that the results obtained by the ACS-AntMiner dominate other classifiers when considering the classification accuracy and model size ranks. The ACS-AntMiner only performs slightly worse than TACO in terms of model size, but it is significantly better than TACO and the other classifiers in terms of classification accuracy. Therefore, the ACS-AntMiner is the best classifier that balances between the classification accuracy and model size. This balance aims to overcome the problems of overfitting and under-fitting, considering that a less accurate rule in the training process that covers numerous training instances is better than an accurate rule that covers only one instance. This case helps to avoid the overfitting problem and increases the generalisation of the classification model. Our classifier aims to find the appropriate pruning criteria for each dataset by considering the feedback

(local and global) in the learning process on the basis of the classification accuracy and avoiding the learning process from constructing a classification model that is too simple to describe a given set of data. Understanding these two phenomena allows the development of a classification algorithm that balances two extremes. Therefore, the ACS-AntMiner outperforms other classifiers when the classification accuracy and model are balanced.

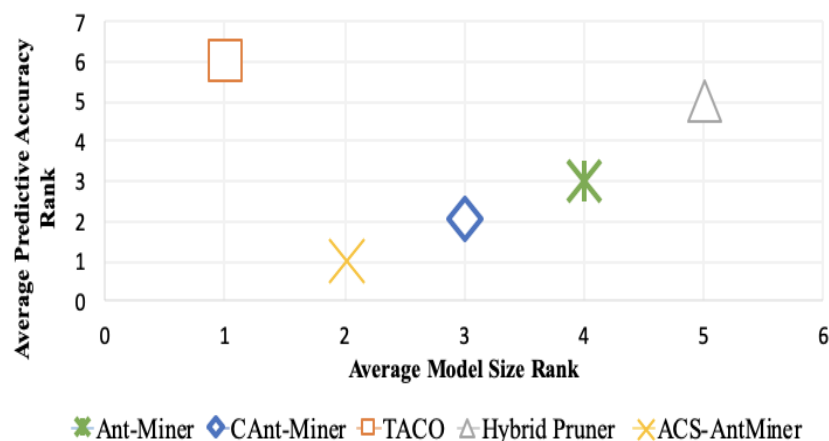


Figure 11. Average rank test

## 6. CONCLUSION

In this study, a novel ACO-based rule classification algorithm, called ACS-AntMiner, is presented which balances between high classification accuracy and low model size. A new parameter, called IR ( $\zeta$ ), is introduced to control the relative importance to the terms included in the construction rule. The ACS optimization algorithm is then designed to control this parameter online using three main features. The first feature is the parameter selection probability, which considers the balance between exploration and exploitation. This method can effectively guide our classifier in selecting a better parameter value to select the terms included in the construction rule. The second feature is the local feedback collection procedure, which monitors the effectiveness of the selected parameter value. This feature evidently reinforces the performance of the classification algorithm. The last one is the global feedback collection, which keeps track of the best parameter value used in the current iteration. These new features are implemented in our algorithm, and 16 datasets are used to evaluate the performance of ACS-AntMiner classifier. The evaluation result with four most related ant-mining classifiers, namely, Ant-Miner, CAnt-Miner, TACO-Miner, and Ant-Miner with hybrid pruner, shows that our proposed classifier performs best in the classification accuracy and achieves preferable simplicity. This due to the usage of the ACS algorithm to optimise the IR ( $\zeta$ ) value. Certain potential directions in future research in our classifier are discovered. Such a notion could draw inspiration from other swarm-intelligent optimization algorithms, namely, PSO, ABC, BA, and FA, to optimize the IR ( $\zeta$ ) parameter online, which may guide ants to find enhanced solutions. Another research direction is to adapt techniques that automatically adjust the parameters of the search methods on-the-fly in the field of the stochastic local search algorithm.

## ACKNOWLEDGMENTS

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

## REFERENCES

- [1] A. Holzinger, M. Plass, K. Holzinger, G. C. Crisan, C.-M. Pintea, and V. Palade, "A glass-box interactive machine learning approach for solving NP-hard problems with the human-in-the-loop," *arXiv Prepr. arXiv1708.01104*, pp. 1–26, 2017.
- [2] Y. Drias and S. Kechid, "Dynamic Web information foraging using self-interested agents: Application to scientific citations network," *Concurr. Comput.*, no. August, pp. 1–17, 2017.
- [3] 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.
- [4] A. M. Jabbar and K. R. 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.
- [5] 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.
- [6] 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.
- [7] P. G. Espejo, S. Ventura, and F. Herrera, “A survey on the application of genetic programming to classification,” *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 40, no. 2, pp. 121–144, 2010.
- [8] J. Uthayakumar, T. Vengattaraman, and P. Dhavachelvan, “Swarm intelligence based classification rule induction (CRI) framework for qualitative and quantitative approach: An application of bankruptcy prediction and credit risk analysis,” *J. King Saud Univ. - Comput. Inf. Sci.*, 2017.
- [9] H. N. K. Al-behadili, “Classification Algorithms for Determining Handwritten Digit,” *Iraq J. Electr. Electron. Eng.*, vol. 12, no. 1, pp. 96–102, 2016.
- [10] J. Wahid and H. F. A. Al-Mazini, “Classification of Cervical Cancer Using Ant-Miner for Medical Expertise Knowledge Management,” in *Knowledge Management International Conference (KMICe)*, 2018.
- [11] A. Gavrilovski *et al.*, “Challenges and Opportunities in Flight Data Mining: A Review of the State of the Art,” *AIAA Infotech @ Aerosp.*, no. January, 2016.
- [12] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, “Classification with ant colony optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, 2007.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] X. Zhang and W. Sun, “An Adaptive Ant Colony Algorithm for Classification Rule Mining,” in *2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE2016)*, 2016, vol. 133, pp. 295–299.
- [18] 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.
- [19] 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.
- [20] 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.
- [21] R. Sagban, K. R. Ku-Mahamud, and M. A. B. Shahbani, “Nature-inspired Parameter Controllers for ACO-based Reactive Search,” *Res. J. Appl. Sci. Eng. Technol.*, vol. 11, no. 1, pp. 109–117, 2015.
- [22] A. M. Jabbar, “Controlling the Balance of Exploration and Exploitation in ACO Algorithm,” *J. Univ. Babylon*, vol. 26, no. 4, pp. 10–17, 2018.
- [23] M. Dorigo and T. Stutzle, “The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances,” *Handb. Metaheuristics SE - Int. Ser. Oper. Res. Manag. Sci.*, vol. 57, pp. 250–285, 2003.
- [24] M. Dorigo and T. Stützle, *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004.
- [25] M. Dorigo, M. Birattari, and T. Stützle, “Ant Colony Optimization: Artificial Ants as a Computational Intelligence Technique,” Bruxelles, Belgium, 2006.
- [26] M. Lopez-Ibanez, T. Stutzle, and M. Dorigo, “Ant Colony Optimization: A Component-Wise Overview,” in *Springer*, Cham: Springer International Publishing, 2016, pp. 1–37.
- [27] 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.
- [28] 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.
- [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] Liu, H. A. Abbass, and B. Mckay, “Density-Based Heuristic for Rule Discovery With Ant-Miner,” *6th Australas. Jt. Work. Intell. Evol. Syst.*, pp. 1–6, 2002.
- [31] B. Liu, H. A. Abbass, and B. Mckay, “Classification rule discovery with ant colony optimization algorithm,” in *IEEE Computational Intelligence Bulletin*, 2004, vol. 7435 LNCS, no. 1, pp. 678–687.
- [32] W.-J. Jiang, Y. Xu, and Y. Xu, “A novel data mining algorithm based on ant colony system,” in *Fourth International Conference on Machine Learning and Cybernetics*, 2005, no. August, pp. 18–21.
- [33] J. Ji, N. Zhang, C. Liu, and N. Zhong, “An Ant Colony Optimization Algorithm for Learning Classification Rules,” *WI '06 Proc. 2006 IEEE/WIC/ACM Int. Conf. Web Intell.*, pp. 1034–1037, 2006.
- [34] A. Chan and A. a. Freitas, “A new ant colony algorithm for multi-label classification with applications in bioinformatics,” in *the 8th annual conference on Genetic and evolutionary computation*, 2006, pp. 27–34.

- [35] J. Smaldon and A. a Frietas, "A New Version of the Ant-Miner Algorithm Discovering Unordered Rule Sets," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation. ACM*, 2006, no. 6, pp. 43–50.
- [36] J. He, D. Long, and C. Chen, "An improved ant-based classifier for intrusion detection," *Third Int. Conf. Nat. Comput. ICNC 2007*, vol. 4, no. 60273062, pp. 819–823, 2007.
- [37] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck, and B. Baesens, "Classification With Ant Colony Optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 5, pp. 651–665, 2007.
- [38] W. Verbeke, D. Martens, C. Mues, and B. Baesens, "Building comprehensible customer churn prediction models with advanced rule induction techniques," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2354–2364, 2011.
- [39] 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.
- [40] F. Otero, A. Freitas, and C. Johnson, "Handling continuous attributes in ant colony classification algorithms," in *2009 IEEE Symposium on Computational Intelligence and Data Mining, CIDM 2009 - Proceedings*, 2009, pp. 225–231.
- [41] W. Shahzad and A. R. Baig, "Compatibility As a Heuristic for Construction of Rules By Artificial Ants," *J. Circuits, Syst. Comput.*, vol. 19, no. 01, pp. 297–306, 2010.
- [42] S. Madhusudhanan, M. Karnan, and K. Rajivgandhi, "Fuzzy based ant miner algorithm in datamining for hepatitis," *2010 Int. Conf. Signal Acquis. Process. ICSAP 2010*, pp. 229–232, 2010.
- [43] D. Agravat, U. Vaishnav, and P. B. Swadas, "Modified Ant Miner for intrusion detection," in *ICMLC 2010 - The 2nd International Conference on Machine Learning and Computing*, 2010, pp. 228–232.
- [44] 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.
- [45] S. C. Liang, Y. C. Lee, and P. C. Lee, "The application of ant colony optimization to the classification rule problem," *IEEE Int. Conf. Granul. Comput. GrC 2011*, pp. 390–392, 2011.
- [46] R. Saian and K. R. Ku-Mahamud, "Hybrid Ant Colony Optimization and Simulated Annealing for Rule Induction," *2011 UKSim 5th Eur. Symp. Comput. Model. Simul.*, no. March 2016, pp. 70–75, 2011.
- [47] H. Wu and K. Sun, "A simple heuristic for classification with ant-miner using a population," in *the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2012*, 2012, vol. 1, pp. 239–244.
- [48] A. R. Baig and W. Shahzad, "A correlation-based ant miner for classification rule discovery," *Neural Comput. Appl.*, vol. 21, no. 2, pp. 219–235, 2012.
- [49] K. M. Salama and F. E. B. Otero, "Using a unified measure function for heuristics, discretization, and rule quality evaluation in Ant-Miner," *2013 IEEE Congr. Evol. Comput. CEC 2013*, pp. 900–907, 2013.
- [50] S. Rajpiplawala and D. K. Singh, "Enhanced cAntMiner PB Algorithm for Induction of Classification Rules using Ant Colony Approach," *IOSR J. Comput. Eng.*, vol. 16, no. 3, pp. 63–72, 2014.
- [51] B. Fakhar, "A Novel Method for Extracting Classification Rules based on Ant-Miner," *J. Math. Comput. Sci.*, vol. 8, pp. 377–386, 2014.
- [52] Z. Liang, J. Sun, Q. Lin, Z. Du, J. Chen, and Z. Ming, "A novel multiple rule sets data classification algorithm based on ant colony algorithm," *Appl. Soft Comput. J.*, vol. 38, pp. 1000–1011, 2016.
- [53] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, 1997.
- [54] R. Parpinelli, H. Lopes, and A. A. A. Freitas, "Data mining with an ant colony optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 321–332, 2002.
- [55] 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>.
- [56] 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.
- [57] M. López-Ibañez, T. Stützle, and M. Dorigo, "Ant Colony Optimization: A Component-Wise Overview," Bruxelles, Belgium, 2016.
- [58] R. P. Datta and S. Saha, "Applying rule-based classification techniques to medical databases : an empirical study," *Int. J. Bus. Intell. Syst. Eng.*, vol. 1, no. 1, pp. 32–48, 2016.