



A rule-based fuzzy ant colony improvement (ACI) approach for automated disease diagnoses

Piyush Bagla¹ · Kuldeep Kumar²

Received: 8 June 2022 / Revised: 12 October 2022 / Accepted: 12 March 2023 /

Published online: 21 March 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

India has approximately seven qualified doctors for every 10,000 people, highlighting the several issues confronting the medical industry, including physician insufficiency, slower pace of diagnosis, and untimely provision of medical aid to common people. Because clinical decision-making requires the highest degree of precision in diagnosis, it is time-consuming and challenging work for physicians. Physicians may benefit significantly from using an automated computing system that aids them in disease diagnosis. Numerous research studies have developed an effective medical expert system to support physicians. However, reaching the highest level of precision remains a challenge. We developed an intelligent medical system based on machine learning and enhancement methodologies because we saw a need to improve diagnostic precision. This research proposes and tests a medical intelligent system design using four different medical datasets. The authors offer a fuzzy logic-based expert system for ACI (Ant Colony Improvement). To develop fuzzy categorization rules, this system uses the ACO approach. The artificial ants create possible fuzzy search criteria, and stochasticity the method encourages ants to develop more accurate rules. The fuzzy inference engine makes conclusions about testing patterns using these updated rules. The suggested system's performance is compared to those of various current systems. With an average accuracy of 93.27% on different datasets, it is discovered that the suggested approach outperforms all existing methodologies for different disease diagnosis datasets.

Keywords Machine learning · Ant Colony improvement · ACO · Fuzzy logic · Medical expert system

✉ Piyush Bagla
piyushb.cs.19@nitj.ac.in

Kuldeep Kumar
kuldeepkumar@nitkkr.ac.in

¹ Department of Computer Science and Engineering, Dr. B R Ambedkar National Institute of Technology, Jalandhar 144027, India

² Present address: Department of Computer Engineering, National Institute of Technology, Kurukshetra 136119, India

1 Introduction

The medical industry is confronted with new issues, such as emerging diseases, rising costs, novel medicines, and rapid decision-making. An automated system that assists in disease detection will benefit the medical industry. Since clinical decision-making requires the utmost precision in diagnosis, it is time-consuming and challenging work for physicians. This has prompted researchers to develop very precise medical decision-assistance systems. In the case of medical diagnostics, decision-making is a complex process. A significant number of overlapping structures, cases, distractions, fatigue, and visual system impairments might contribute to an incorrect diagnosis. Over the course of years, a vast number of different methodologies have been proposed to circumvent these limitations and arrive at correct and informed findings about disease diagnosis [13, 14, 19, 24, 44]. These methodological approaches may be represented as knowledge in a variety of forms, such as by employing AI (Artificial Intelligent), ML (Machine Learning,) and other statistical models, for instance, decision trees, rule-based models, linear and non-linear models, etc. In addition to these well-known methods, some of them are also often used for classification tasks such as ID3 (Iterative Dichotomiser), SVMs (Support Vector Machines), ACO (Ant Colony Optimization), KNN (K-Nearest Neighbour), and Neural Networks. Even though the classification models produced by approaches like neural networks and SVMs are very accurate, their architectures are extremely difficult to comprehend. Rather than generating the model all at once, decision trees assemble it together one feature at a time, employing a greedy heuristic approach to determine which characteristics are most important. Therefore, decision trees may provide an unsatisfactory classification model due to their failure to account for attribute interactions. In contrast to unlabeled examples, the K-nearest neighbor technique has a significant computing cost because of the huge number of neighbors.

A rule-based classification system named ACO has been proposed as a solution to the challenges that have been outlined above. It is used to solve a range of combinatorial optimization problems and is influenced by the natural phenomena of stigmergy, which is best shown by the behavior of ants [25, 26]. The ACO system makes use of a combination of two primary components. To begin, it makes use of a stochastic process, which enables it to investigate a significant section of the search spaces more thoroughly. Second, it uses an iterative adaptation strategy based on positive feedback [10, 34]. These two features provide a classification model that is not only intuitive but also effective, which helps to offset the constraints that were previously described. In the field of discovering new classification rules, ACO algorithms have seen extensive use. In addition, it can identify globally optimal classification rules and handle attribute interactions more efficiently than standard rule induction algorithms, which often display greedy behavior throughout the rule-creation process.

Fuzzy techniques have also emerged as a prominent classification problem solution. It improves the efficacy of the classification as well as the decision support system by rendering the use of overlap class definition in which a case is matched with more than one rule with different output classes. In addition, it has robust characteristics for managing ambiguity and uncertainty [48]. With the introduction of fuzzy if-then rules, the interoperability of the results enhances; subsequently, it gives useful insight into the classifier's architecture and decision-making procedure [47]. A pattern could also be provided to numerous classes, each of which has a different membership degree. The fuzzy classification rules are portrayed in language forms that are simple for people to comprehend and investigate [27].

In this work, we have integrated Fuzzy logic with ACO techniques to improve the classification rule's quality. This will enhance the performance of the conventional ACO classification method. We call this technique ACI using a fuzzy rule-based approach. In our proposed system, the artificial ants were employed to explore the search space and consequently develop candidate fuzzy rules. We made an effort to strike a balance between the cooperative and competitive tendencies of ants in such a manner that the ants are incentivized to provide their assistance to the colony in the formulation of more specific rules. In addition, we suggested a novel framework for fuzzy rule learning that is applicable to both continuous and nominal attributes. In our proposed system, the learning process is different from other approaches in the sense the learning process for each class is performed independently rather than for the entire class. The resulting classification system is used in four disease diagnostic datasets, including PID (Pima Indian Diabetes), WDBC (Wisconsin Diagnostic Breast Cancer), WPBC (Wisconsin Prognostic Breast Cancer), and the Heart-Disease. The findings suggest that this system is capable of detecting the disease signatures associated with diabetes and breast cancer with an accuracy that is on par with or even superior to the results obtained by past research in this area. Additionally, this approach has a high degree of comprehensibility since it generates just a small number of fuzzy if-then rules of relatively short length.

The remainder of this articles is organized as follows: Section 2 of this article presents literature pertinent to this investigation. Section 3 highlighted some of the most significant research gaps identified in the preceding literature. Our proposed work is described in Section 4, which is broken into various sub-sections. Section 5 describes the experimental setup necessary to create this Fuzzy-ACI system. In Section 6, the result and discussion are given. In Section 7, we have also mentioned the limitations of this study, followed by the conclusion and suggestion for further research in Section 8.

2 Related work

In this section, we will discuss some of the most prominent rule-based classification studies in the literature. In addition, we will emphasize the advantages and disadvantages of each research. Keeping in mind the aim of the research, we have emphasized our discussion on the technique that performs rule-based classification using ACO. In Table 1 in this section, we have compared the methodologies, benefits, limitations, and datasets utilized by various well-known research relevant to our proposed study in a more structured format.

ACO was first employed to carry out a data mining task in a study by [31], which resulted in the creation of a model called AntMiner1. The implementation of this model was based on the notion of the Ant System (AS) algorithm, which served as inspiration for this implementation. In the AS method, however, the pheromone is updated only after every Ant in the colony has built its respective solution. On the other hand, the pheromone in the AntMiner1 is brought up to date whenever an ant builds a solution. Because of this, the pheromone information may be used by the Ant that comes immediately after it. The Ant-Miner begins by initializing the pheromone, and then every Ant selects a term from the dataset using a random selection method; the notation of the term is $\langle \text{attribute}, \text{operator}, \text{value} \rangle$. To create the classification rule, the Ant begins at the term that serves as its starting point and then travels through the iterative process of moving from one term to an unvisited term. Rule formulation is complete when each Ant has traversed all the terms based on heuristic information and pheromones value. To achieve higher overall quality, individual terms are pruned one at a

Table 1 Comparison of different rule-based classification techniques that uses Ant Colony Optimization

S. No	Study Name	Dataset	Benefits	Limitations	Ref.
1	AntMiner1	Hepatitis, tic-tac-toe, breast cancer, Cleveland, dermatology	<ul style="list-style-type: none"> • An advancement of AS algorithm. • Introduced ACO in data mining tasks. • Smaller rule generation than CN2 	<ul style="list-style-type: none"> • Work only on the categorical attribute. • It has used IG (information gain) as a heuristic function, decreasing the computational accuracy. • Computationally expensive 	[31]
2	AntMiner2	Breast cancer	<ul style="list-style-type: none"> • It replaces IG with density estimation as a heuristic function. • Computationally less expensive. 	<ul style="list-style-type: none"> • Unbalance exploration and exploitation phases. • Gained similar accuracy when compared with AntMiner1. 	[22]
3	AntMiner3	Tic_Tac_Toe and Breast Cancer dataset.	<ul style="list-style-type: none"> • Updated the pheromone state transition process. • Balanced between exploration and exploitation • Results not only depend on heuristic and pheromone value but also depends on random number generation. 	<ul style="list-style-type: none"> • It requires a greater number of ants to converge to a solution. • Lower average accuracy than AntMiner1 	[23]
4	mAnt-Miner+	Vote, anneal, PIMA, zoo, monk, balance, breast cancer, tic tac toe.	<ul style="list-style-type: none"> • Uses simple heuristic strategies which solve the problem of local optima. • A large number of ants are used for rule discovery in each iteration. • Better accuracy. • Less processing time than AntMiner1 	<ul style="list-style-type: none"> • Work only on categorical attributes. • More processing time when compared with AntMiner2. • No performance evaluation with other pruning strategies • No comparison with other studies on Pheromone updation. 	[46]
5	New Rule Pruning Procedure	House votes, Chess, Web Mining Dataset	<ul style="list-style-type: none"> • Proposed a hybrid rule pruner which results in faster rule pruning. • Generation of shorter and simple classification rules. • Less computational time. 	<ul style="list-style-type: none"> • Decreased classification accuracy. • It can't use a multi-class classification system. • The best result can only be obtained when used with a dataset having large attributes. 	[6]
6	MuLAM (Multi-Label Ant-Miner)	Five biometric datasets on protein function.	<ul style="list-style-type: none"> • It can generate more rules in each iteration in place of a few rules. • It can be used in solving a multi-classification problem. 	<ul style="list-style-type: none"> • The comparison with other algorithms was made on their default parameter settings. 	[7]
7	Unordered AntMiner	Used the same dataset as in [31]	<ul style="list-style-type: none"> • No knowledge of previous rules is required in order to interpret new classification rules. • Encourages the analysis of newly found information, a crucial aspect of data mining 	<ul style="list-style-type: none"> • During rule construction by an ant, the output class is fixed for every rule. • No optimum parameter settings. • This approach cannot work on the continuous dataset. 	[39]

Table 1 (continued)

S. No	Study Name	Dataset	Benefits	Limitations	Ref.
8	TACO-Miner (Threshold Ant Colony Optimization)	Hepatitis, tic-tac-toe, breast cancer, and dermatology	<ul style="list-style-type: none"> Added threshold value at the rule construction step. Higher predictive accuracy Less computational complexity and cost. 	<ul style="list-style-type: none"> No deterministic approach to set threshold value. The rule list generated by TACO is large. The accuracy, when compared with other thresholds AntMiner technique such as MTACO, is less. 	[41]
9	MTACO (Modified Threshold Ant Colony Optimization)	Wisconsin Breast Cancer	<ul style="list-style-type: none"> Post rule pruning task is minimized, which reduces the complexity. Comprehensive and simple rule generation with high accuracy. The model competes with GA and simulated annealing. 	<ul style="list-style-type: none"> Mainly applicable for the discrete problem There are no proven criteria to select the threshold value. 	[42]
10	Ant-Tree Miner	Twenty-two publicly available datasets in UCI machine learning repository	<ul style="list-style-type: none"> Embedding decision tree with ACO technique. Achieved higher accuracy when compared with C4.5, CART, and cACDT. The size of the tree is minimized, which is equal to the number of leaf nodes in the construction graph. 	<ul style="list-style-type: none"> The comparison has been made with algorithms that were used with their default parameter settings. Overfitting problem 	[30]
11	cAnt-Miner	WDBC, CRX, Hepatitis, Glass, Ionosphere, Wine, Australian, Heart	<ul style="list-style-type: none"> Entropy-based variable discretization. It works on continuous variables, unlike other Ant Miners. It doesn't require discretization at the pre-processing stage. Extended the construction graph. Achieved higher accuracy when compared with the AntMiner algorithm. 	<ul style="list-style-type: none"> No lower and upper bound has been set for the threshold value, which is used for dynamically partitioning the continuous attribute. No comparison has been made for the algorithm of 	[28]
12	cAnt-Miner2	WDBC, CRX, Hepatitis, Glass, Ionosphere, Wine, Australian, Heart	<ul style="list-style-type: none"> Implements a new dynamic discretization process name Minimum Description Length (MDL). Achieved higher accuracy than cAnt-Miner. 	<ul style="list-style-type: none"> Need to evaluate a new pheromone update strategy. No comparison to continuous-variable classification algorithms. 	[29]
13	AntMiner _{mbc}	Nineteen publicly available data in the UCI repository.	<ul style="list-style-type: none"> It is a hybrid approach that takes advantage of multiple classifiers to build the final model. Proposed a new heuristic function 	<ul style="list-style-type: none"> More execution time. Need for parallel execution as each Ant generates a classification rule independently. To find a better solution, strategies like clustering 	[21]

Table 1 (continued)

S. No	Study Name	Dataset	Benefits	Limitations	Ref.
14	ACS-AntMiner (Ant Colony System)	Sixteen UCI benchmark dataset	<ul style="list-style-type: none"> • For test data classification, a mechanism named the weighted voting mechanism is used. • Introduction to new pre-pruning criteria called IR (Importance Rate). • Hybrid approach • High classification accuracy. • Small model size 	<ul style="list-style-type: none"> • can be embedded in the rule generation process. • The criteria for deciding the IR were not discussed properly. • Need to adjust the search parameter. • Need to use another swarm optimization algorithm to optimize the IR parameter 	[2]
15	AntMiner+	Credit scoring dataset	<ul style="list-style-type: none"> • Implements MMAS (Min Max Ant System) • It used DAG (Directed Acyclic Graph) to choose better paths for the ants. • It introduces weight parameters to calculate heuristic and pheromone values. 	<ul style="list-style-type: none"> • More computation time in comparison to other related techniques • It works on nominal attributes and requires a discretization phase for the continuous attribute. 	[43]

time. Rule pruning is a process of reducing the number of created rules to improve their objectivity. The process is repeated until there is no more opportunity for improvement or the rule still contains at least one term in it. Following that, the pheromones value will be updated in two phases. In the first phase, the rule's quality determines the quantity of pheromone deposited in all terms. In the second phase, the rule doesn't contain the pheromone as it was evaporated from every term. Following that, the ants begin to move closer and closer to the high-quality rule. At last, the most effective rules will be added to the list of predictive rules, while all instances that are addressed by the rule will be excluded from the dataset. Only the categorical attributes are taken into consideration while using the Ant-Miner algorithm. During the pre-processing, continuous attributes are converted into discrete ones.

There were several changes made to the Ant-Miner algorithm's AS variation to make it better. Multiple ant colonies, rather than just one for the original Ant-Miner, were introduced in [16]. Those colonies operate in parallel, producing a single set of rules that applies to all of them. Allowing individual colonies to deposit their own unique pheromones was a key change made to the pheromone updating process. The researchers in [18] proposed a new Ant-Miner rule mining algorithm that uses a cost-based discretization method and a multi-population parallel strategy. An additional heuristic function, with a simpler and less precise method, was proposed by [16]. Due to the improved heuristic function, the method can be run with a less computational expense. Another research [46] suggested that Ant-Miner may be made better by letting a swarm of ants figure out the rule on every iteration. Also, they have considered the algorithm's calculation time and added a new heuristic function to address it. With the increasing number of attributes in datasets, [6] proposes a rule-pruning approach that is computationally cheaper to utilize. A multi-label data categorization was introduced in [7], where the found rule includes more than one class label in the consequent component. Instead of producing a single rule at each iteration, the algorithm generates a set of rules. Furthermore,

the pheromone will be updated for each class attribute that appears in the found rule, and the number of pheromone matrices is proportional to the number of class attributes. In a different study introduced by [39], rule creation occurs just after the class consequent has been chosen. The Ant builds a rule for every class that follows and then proposes an unordered rule set. The hope is that each Ant would know the rule generation procedure and its implications and stick to it. Offline, step-by-step adjustments were made to the various parameters of this method. A threshold value is proposed in other papers [41, 42], both of which focus on the rule-building process. To accept or reject the terms for inclusion in the present rule, the threshold value must be considered. If a term's heuristic value is higher than a specified threshold, it will be included in the existing rule; otherwise, it will be removed.

In research conducted by [37], further development of Ant-Miner employs a choice of class consequent before rule building and assigns a unique sort of pheromone to each class. The researchers then investigate how the number of rules and the prediction power of rules changes depending on the quality measurement function used. Another study by [28] suggested cAnt-Miner, a variant of Ant-Miner with dynamic support for continuous characteristics. The entropy discretization technique was employed in this algorithm. The same team of researchers also unveiled a follow-up tool to cAnt-Miner called cAntMiner2 [29]. It is planned to add this new functionality to cAnt-Miner so that the algorithm can more naturally represent continuous qualities and place pheromones at the edges of the building graph rather than vertices. [38] introduces a novel approach to term selection during rule development by considering the connection between the candidate term and the previously selected term and the rule consequent. By considering both the instances coverage and the association across attributes, the heuristic information function presented in [21] is a useful tool. Work proposed in [36] introduces different forms of pheromones for each class consequent and also enables individual ants to keep track of their own past experiences. In addition, it allows for online measurement adaptation by allowing individual ants to choose their own values for α and β parameters. In [4], the authors developed an updated heuristic function for the Ant-Miner algorithm that takes into account the interdependence of its attributes. To select a more precise rule set, [20] created a multilevel rule selection system. To further improve terms selection during rule development, [35] developed a coupling between Ant-Miner and Simulated Annealing methods. Protein function prediction is a systematic multi-class classification problem that Prabha and Balraj addressed with the Ant-miner method [32]. An enhancement to the Ant-Miner algorithm was proposed in [33], which involved allowing individual ants to choose the quality function on the fly before building the rule.

Antminer+ [43] is the first algorithm to utilize the highly efficient MMAS (Min-Max Ant System) for rule-based categorization. There are numerous ways in which Antminer+ diverges from AS-based categorization techniques. First, upper and lower limitations are introduced to the total amount of pheromones released. A powerful elitist method, such as iterations of the best Ant or the global-best Ant, is used to update the pheromones. The next thing to know is that Antminer+ uses a new heuristic function, a different pheromone update technique, and a self-adaptive mechanism to adjust the weight of the parameters. Finally, while all other algorithms use a fully linked network, Antminer+ specifies the environment as a directed acyclic graph. However, the Antminer+ has used a simple heuristic function that gradually decreases the computation cost. However, the accuracy of this function is less when compared to the original Ant-Miner.

There are various other studies that combine machine learning and deep learning approaches to image-based disease diagnosis. The idea is to use medical imaging to identify and diagnose the type and treatment for the diseases. In [40], an algorithm name AW-HARIS was proposed to

diagnose the abnormality in the human liver by using CT-scan images. In [8, 9], the image enhancement filters, and their modifications have been proposed. In [9] the researchers have proposed a hybrid approach by fusing fast guided and matched filters on fundus images for the identification of ophthalmological disease. In another study by [8], again medical image filter fusion techniques have been presented, namely the Jerman filter and curvelet transform for retinal blood segmentation representation. In other previous studies [3, 5] aimed at identifying medical diagnoses, the researchers combined a variety of deep learning algorithms with particle swarm optimization.

3 Research gaps

From the studies discussed above, we have identified the two most common research gaps. Our proposed ACI system will address the first three gaps in the subsequent section of this study. However, the last research gap can be agenda for future scope.

- The studies led to an overlapping class problem in which a sample being tested is closely matched with multiple rules with distinct classes.
- The methods are only capable of handling nominal attributes; in order to accommodate additional attributes, a conversion step is required during the pre-processing phase. In the context of the current system, this kind of conversion almost always results in the loss of information in the dataset.
- The effectiveness of ACO-based classification rule discovery algorithms has been evaluated with the use of benchmark datasets provided by UCI. Comparing them with other datasets gathered from real-world applications could provide more information on their behavior and performance.
- There are no consistent parameter settings between datasets and application domains. The examples of such parameters which influence the performance of the model are the Number of Ants, Number of Iterations, Maximum number of uncovered cases, Convergence test size, and Minimum number of covered examples per rule.

4 Proposed work

We have proposed a Fuzzy-ACI medical decision support system to identify a variety of disease datasets, including original and diagnostic breast cancer, diabetes, and heart disease. Fuzzy logic and ACI are used to create a set of fuzzy rules using the patient's dataset. The ACI system enhances the extraction of fuzzy rules by generating additional rules. The fuzzy inference system categorizes the test data according to these new rules. The classification precision of the system is tested using tenfold cross-validation. Within this section's subpart, we have also included a classification rule-generating procedure that we have suggested.

4.1 The ant system for ant colony improvement

The Ant System [11, 12] is the first algorithm to incorporate the stochastic path select technique into its design. Figure 1 shows how ants gather food by iteratively constructing solutions and adding Semiochemical (pheromone) to the pathways that correspond to these

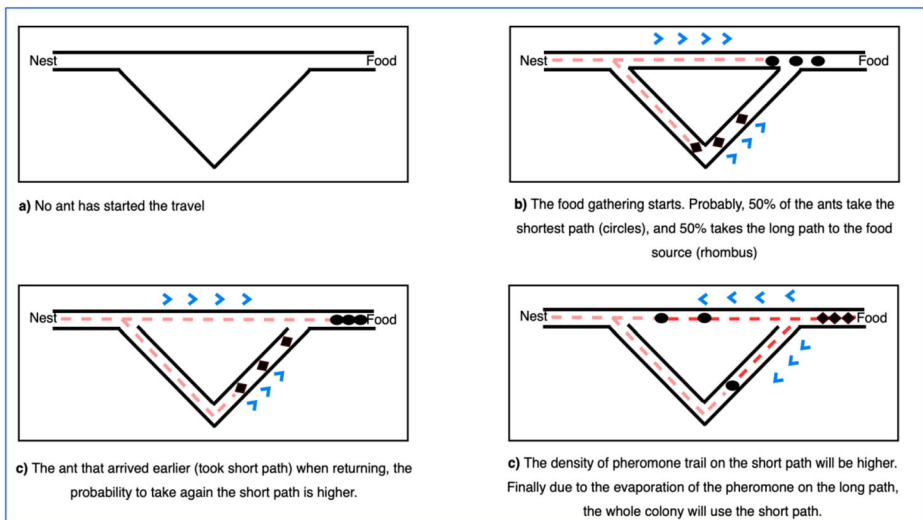


Fig. 1 Discovery of the shortest path by Ants

solutions. There are two parameters in the stochastic path selection technique: heuristic and semiochemical values.

4.1.1 Semiochemical value

It is determined by how many ants recently used the track. Initial value of whose is the same for the ants. The amount of pheromone is updated according to the equation mentioned below. The quality of the rule is directly proportional to the value of the Q .

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

where

Q the quality of the rule whose values range $0 < Q < 1$.

$\tau_{ij}(t)$ the amount of pheromone to $term_{ij}$ available for the Ant during its current trail.

4.1.2 Heuristic function

It is a quality criterion whose value decides which node to be added to the construction graph. The node represents a *term* that could be added to the ruleset. There are different heuristic functions available, for example, entropy-based heuristic function.

$$\text{Entropy}(T_{ij}) = -\sum_{w=1}^k \left(\frac{\text{Freq}T_{ij}^w}{|T_{ij}|} \right) \times \log_2 \left(\frac{\text{Freq}T_{ij}^w}{|T_{ij}|} \right) \tag{2}$$

where

K number of classes

$|T_{ij}|$ the total number of cases in partition T_{ij}

Note: The partition includes the cases in which attribute A_i has value V_{ij}

$\text{Freq}T_{ij}^w$ number of cases in the partition with class w

Ants employ heuristic behavior to select the trail with the highest Semiochemical density. After an ant reaches its destination, its journey is analyzed, and the Semiochemical value is suitably enlarged [15]. Evaporation eventually degrades the semiochemical composition of all traces. Subsequently, the Semiochemical levels of unfollowed trails steadily decline, decreasing the likelihood of the trail being picked by succeeding ants.

4.2 Fuzzy classification

Considering our problem, an m class classification problem which has n dimensional pattern space consisting of continuous variables. Also assumes that there is a total of V -real vectors with $a_x = \{a_{x1}, a_{x2}, a_{x3} \dots a_{xn}\}$ where $x = 1, 2, 3, \dots V$ are given as training pattern from m classes where $m \ll V$.

The value of every attribute for each pattern of the dataset ranges in the interval of $\{0,1\}$, since the pattern space belongs to $[0, 1]^n$. This can be illustrated by Eq. (2).

$$a_{xi} \in [0, 1], \text{ for } x = 1, 2, 3 \dots V \text{ and } i = 1, 2, 3 \dots n \quad (3)$$

The attributes have been normalized in a range of $\{0,1\}$ which will be explained later in the dataset description. The fuzzy IF-THEN used in our research is represented in the following form.

R_i : IF a_1 is A_{i1} , a_2 is $A_{i2} \dots a_n$ is A_{in} and, THEN class is C_i with $CF = CF_i$

where

R_i fuzzy IF-THEN rule,
 $A_{i1}, A_{i2} \dots A_{in}$ antecedent fuzzy sets on the unit interval $[0, 1]$
 C_i consequent class whose values belong to m -class
 CF_i grade of certainty of rule R_i .

Although there are different membership functions available for the fuzzy set, we have picked the most widely used membership function named the triangular membership function. In this, the domain of each attribute is partitioned into different values such as small, medium-small, medium, medium-large, and large. In order to show the high performance of the fuzzy classifier system, we use simple specifications in simulated experiments. We have used a simple, regardless of whether or not each antecedent fuzzy set's membership function is custom-tailored. On the other hand, with our fuzzy classifier system, we are free to employ any membership function that has been specifically crafted for the pattern classification issue at hand. The following steps are involved in computing the certainty grade of each fuzzy IF-THEN rule created by ACO (Fig. 2).

Step 1: Determine the degree to which each training pattern $a_x = \{a_{x1}, a_{x2}, a_{x3} \dots a_{xn}\}$ is compatible with one with the fuzzy IF-THEN rule R_i by using Eq. (4):

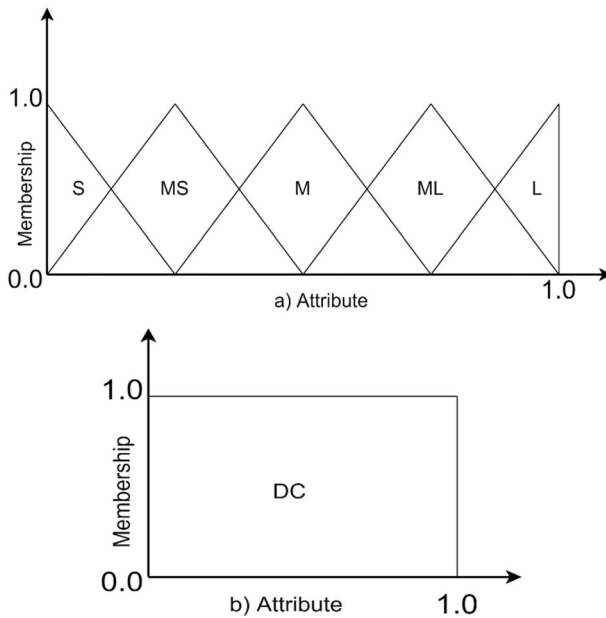


Fig. 2 Antecedent fuzzy-sets

$$\mu_i(a_x) = \mu_{i1}(a_{x1}) \times \dots \times \mu_{in}(a_{xn}), \quad x = 1, 2, \dots, c \tag{4}$$

where $\mu_{ij}(a_{xj})$ is the membership function of j^{th} attribute of x^{th} pattern and c denote the total number of patterns.

Step 2: Calculate the relative sum of compatibility grades of the training patterns for each class with the fuzzy IF-THEN rule R_i

$$\beta_{Class\ h}(R_i) = \sum_{a_x \in Class\ h} \mu_i(a_x) / N_{Class\ h}, \quad h = 1, 2, \dots, c \tag{5}$$

where $\beta_{Class\ h}(R_i)$ is the sum of the compatibility grades of the training patterns in *Class h* with the fuzzy IF-THEN rule R_i and $N_{class\ h}$ is the number of training patterns whose corresponding class is *Class h*.

Step 3: The grade of certainty j CF is determined as follows:

$$CF_i = \left(\beta_{Class\ h_i} \hat{\wedge}(R_i) - \bar{\beta} \right) / \sum_{h=1}^c \beta_{Class\ h}(R_i) \tag{6}$$

where

$$\bar{\beta} = \sum_{h \neq h_i} \beta_{\text{Class } h}(R_i) / (c-1) \quad (7)$$

For each given combination of antecedent fuzzy sets, we may now provide an associated certainty grade. Our fuzzy classifier system has been tasked with the responsibility of producing combinations of antecedent fuzzy sets in order to produce a rule set S that has a high level of classification ability. When a rule set S is given, an input pattern $a_x = \{a_{x1}, a_{x2}, a_{x3} \dots a_{xn}\}$ is classified by a single winner rule R_i in S , which is determined as follows:

$$\mu_i(a_x) \cdot CF_i = \max\{\mu_i(a_x) \cdot CF_i | R_i\} \quad (8)$$

In other words, the winning rule has the highest possible sum of compatibility and certainty grade CF_i .

4.3 Generating classification rules

The following are the steps involve in generating the classification rules.

Step 1: Initialization

Initially, the discovered rule is set to empty, whereas training set contains all the training instances.

$$DR = \{\}, TS = \{\text{all training instances}\}$$

where

DR Discovered Rule set

TS Training set

Step 2: Semiochemical (pheromone) value initialization

In the starting, the same pheromone value for each is set by using the eq. (9) below, which gives a fair chance to every Ant in rule discovery.

$$\tau_{i,j}(t=0) = \frac{1}{\sum_{i=1}^a b_i} \quad (9)$$

where

a total number of attributes.

bi number of values in the domain of attribute i.

Step 3: Construction of the rule

Rules are taught differently in each iteration. There are no found rules at the start, and all training instances are the same. One rule at a time, the starting Ant builds the rule ‘R_j.’ This is done at random. The ants use the maximum change parameter to adjust the rule ‘R_j.’ The first Ant constructs a rule R_j randomly by adding one term at a time. Also, the element value of the heuristic function is computed.

Step 4: Rule Updation

The Ant creates a rule on the first iteration ($t = 0$), which it later amends (t_1). The maximum change value defines how many times an ant can change the rule during an iteration (t_1). Each Ant alters τ_{ij} based on the probability stated in the following equation:

$$P_{ij} = \frac{\tau_{ij}(t), \eta_{ij}}{\sum_i^a \sum_j^{b_i} \tau_{ij}(t), \eta_{jj'}, \forall i \in I} \quad (10)$$

$\tau_{ij}(t)$ is the heuristic value of the Semiochemical that is currently available on the path between attribute ‘I’ and antecedent fuzzy set ‘j’. The entire number of antecedent fuzzy sets for attributes is given by b_i . In contrast, the set of attributes that Ant has not yet utilized is given by I . A semiochemical is allocated to each of the ant pathways based on the quality of the amended regulations. In areas with high concentrations of Semiochemicals, ants tend to congregate in or around those areas.

Step 5: Calculate the quality of the rule

After a rule has been created, its quality has to be calculated so that a decision can be taken about whether or not it will be included in the final discovered rule set $DS\{\}$. The quality of the rule can be checked by using the equation below.

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{TN + FP} \quad (11)$$

Step 6: Semiochemical update rule

When the rule’s precision prior to updating improves, semiochemical updates are implemented. Formulas used in the process include these:

$$\Delta Q = Q_i^{\text{After Updation}} - Q_i^{\text{Before Updation}} \quad (12)$$

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ii}(t) \left(\Delta_i^Q \cdot C \right)$$

‘ ΔQ ’ is a way to compare the attributes of a rule before and after an update, and the parameter ‘C’ is used to control the Semiochemical’s reaction to ‘ Δ_i^Q ’.

Step 7: Terminating conditions

The user can adjust this condition based on how the model is doing. An example of one such condition is the total number of iterations. Once a stopping condition is reached, then the updated rules can be added to the discovered set DS $\{R_1, R_2, R_3 \dots R_n\}$.

5 Experimental setup

The proposed system is constructed on four publicly available disease diagnoses datasets by using java programming language and a machine learning tool name WEKA [45]. The dataset's instances are divided into ten folds using the k-fold cross-validation technique to ensure that all instances are trained and evaluated. In addition, two of the four datasets have been evaluated against a different test set. Finally, the results are compared with other closely related approaches (such as J48, SVM (Support Vector Machine), AntMiner1, cAntMiner (Continuous AntMiner), and cAntMiner_{PB} (PittsburgContinuousAntMiner)) using two performance evaluation matrices which includes classification accuracy and execution time. For simplicity, we have used the WEKA tool to run J48 and SVM models. However, we have used JVM to run Fuzzy-ACI and J48, cAntMiner, cAntMiner_{PB}. The operating system used to run all the models was MACOS Ventura 16.0 Beta 6.

This section is further divided into different sub-sections to study the experimental setup in more detail, including dataset description and parameter settings.

5.1 Dataset description

The performance of the ACI was evaluated by using four datasets, namely, WDBC (Wisconsin Diagnostic Breast Cancer), WPBC (Wisconsin Prognostic Breast Cancer), PID (Pima Indian Diabetes), and Heart-Disease. The first three datasets have been downloaded from the UCI (University of California at Irvin) machine learning dataset repository [1], whereas the last dataset has been downloaded from Kaggle (A Machine Learning and Data Science repository) [17]. These datasets are publicly available to download from the reference link below. The datasets obtained from UCI have a *.data* extension, but our proposed system requires them to be in *arff* (attribute relation file format); hence WEKA is used to convert them. As identified in the research gaps, the models created in previous studies employed nominal variables; however, in our proposed study, continuous variables were used. However, our fuzzy-ACI can also operate with nominal variables as well. The more enhanced view of the dataset characteristics is shown in Table 2. In the UCI dataset, there is a requirement for data processing as it may contain some fields which are not necessary to be included for generating classification rules, such as PatientID and Gender. No such data pre-processing is needed in the case of the Heart-Disease dataset. The Min-Max normalization method is used to normalize the dataset between 0.0 and 1.0, as indicated in the following equation:

Table 2 Dataset description

S.No.	Dataset name	Total No. of attribute	Total of instances	Number of classes
01	WDBC	31	569	2
02	WPBC	34	198	2
03	PID	9	768	2
04	Heart Disease	14	1025	2

Table 3 Number of parameters and their value used in ACI

S.No.	Parameter Name	Value
01	Number of Ants	60
02	Number of Iterations	10
03	Maximum number of uncovered cases	5
04	Convergence test size	10
05	Minimum number of covered examples per rule	5

$$\text{Normalize } (X) = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (13)$$

where,

X_{\min} and X_{\max} represent the lowest and highest values of the linguistic value ‘X,’ respectively, as well as the membership function (X).

5.2 Parameters settings

Every model operates with a set of parameters whose values must be optimized to generate more precise results. Similarly, when comparing it to other models, these parameter values must be the same so that the findings are more transparent and freer of bias. When comparing the outcomes of our proposed study (ACI) to those of other models, we used the following parameter value pair (Table 3).

6 Results and discussions

Taking one dataset at a time, the performance of the proposed Fuzzy-ACI system is compared with other rule-based classification models, including J48, SVM, AntMiner1, cAntMiner, and cAntMiner_{PB}. Classification accuracy and execution time are calculated for all the datasets. It is discovered that the suggested approach outperforms all existing methodologies for all datasets. Our proposed Fuzzy-ACI approach has been showing improved results. The results are generated by using the parameter values according to Table 3. The tables that follow provide a more detailed representation of the findings.

Table 4 Performance comparison of the fuzzy and non-fuzzy rule-based classifier on the Pima Indian Diabetes dataset

S.No	Algorithm	Classification accuracy	Execution time (approx.)
1	J48	73.00	0.08
2	SVM	76.93	0.11
3	AntMiner1	79.56	124.58
4	cAntMiner	79.30	38.98
5	cAntMiner _{PB}	83.46	35.34
6	Fuzzy-ACI	85.42	123.48

Table 5 Performance comparison of the fuzzy and non-fuzzy rule-based classifier on Wisconsin Prognostic Breast Cancer

S.No	Algorithm	Classification accuracy	Execution time (approx.)
1	J48	73.73	0.13
2	SVM	76.26	0.06
3	AntMiner1	84.34	6.40
4	cAntMiner	83.84	0.38
5	cAntMiner _{PB}	90.91	15.89
6	Fuzzy-ACI	95.91	29.31

The classification accuracy and execution time of different algorithms have been compared individually for every dataset in each table. Table 4 demonstrates that the suggested Fuzzy-ACI method achieves the highest classification accuracy of 85.42 on (PID) Pima Indian Diabetes, compared to other models on the same dataset, including J48 (73.00), SVM (76.93), cAntMiner (79.30), and cAntMinerPB (83.46). Our proposed fuzzy-ACI model takes 123.48 seconds to run, which is longer than any other model. On the other hand, J48 ran in 0.08 seconds, which was the fastest. According to Table 5, the recommended Fuzzy-ACI approach has a classification accuracy of 95.91, while the lowest accuracy is achieved by J48 (73.73) on WPBC (Wisconsin Prognostic Breast Cancer). To run its algorithm, SVM requires the smallest amount of time (0.06 seconds), whereas the Fuzzy-ACI system requires the most time (29.31 seconds). Table 6 displays a comparison of the performance of the findings acquired from the WDBC (Wisconsin Diagnostic Breast Cancer) dataset. Our suggested model has the best classification accuracy, 99.65, while the J48 classification model has the lowest, 93.22. The classification accuracy of other models is, SVM (95.07), AntMiner1 (97.89), cAntMiner (97.36), and cAntMiner_{PB} (98.59). The execution time of SVM is the lowest, 0.23 sec, while AntMiner1 got the highest execution time of 53.61 seconds. In Table 7, our suggested model, Fuzzy-ACI, has the best classification accuracy, 99.61, while the SVM classification model has the lowest, 83.70, on the Heart-Disease dataset. The time taken to build the J48 model is the smallest, 0.06 sec, while the execution time of Fuzzy-ACI is the highest, 32.89 seconds. In contrast to previous assessment metrics on WDBC, WPBC, and PID, such as ten cross folds, we tested the classification accuracy of Fuzzy-ACI and other models on the Heart-Disease dataset using a separate test file (Table 8). It can be seen from this table that Fuzzy-ACI achieves the maximum classification accuracy of 85.79, while AntMiner1 earned the lowest classification accuracy of 79.38. On the other hand, the execution time is lowest for J48 and highest in cAntMiner, 38.25 seconds.

Table 6 Performance comparison of the fuzzy and non-fuzzy rule-based classifier on Wisconsin Diagnostic Breast Cancer

S.No	Algorithm	Classification accuracy	Execution time (approx.)
1	J48	93.22	0.31
2	SVM	95.07	0.23
3	AntMiner1	97.89	53.61
4	cAntMiner	97.36	12.59
5	cAntMiner _{PB}	98.59	20.67
6	Fuzzy-ACI	99.65	18.68

Table 7 Performance comparison of the fuzzy and non-fuzzy rule-based classifier on Heart-Disease (ten cross fold)

S.No	Algorithm	Classification accuracy	Execution time (approx.)
1	J48	98.04	0.06
2	SVM	83.70	0.11
3	AntMiner1	87.90	5.47
4	cAntMiner	88.29	8.46
5	cAntMiner _{PB}	91.21	126.78
6	Fuzzy-ACI	99.61	72.60

The results have shown that in most of the datasets under study, our model often takes longer to execute than competing models do. This is due to the additional time required for the fuzzy inference engine to transform the crisp set into the fuzzy set. This is not the case with other algorithms, as no fuzzification techniques were used. We have also seen from the results tables that the execution time of two machine learning rule-based classification algorithms, J48 and SVM, are very less compared to other models because these models are trained and tested using WEKA software, where most machine learning algorithms are prebuilt, and the tool is optimized according to them. But no such library is present for AntMiners models. We used a command line interface to run our proposed model on which other system resources play their role in influencing the runtime. Lastly, the accuracy and the execution time depend on the parameter settings on which the models depend. We have already discussed these parameter settings in section 5.2 of this research. Figure 3 is included to assist in better visualizing the findings in terms of the accuracy achieved by different models on every dataset.

7 Limitations

Given the fact that our suggested system has reached a significant amount of classification accuracy, it does suffer from some shortcomings.

7.1 Parameter settings

There is a need to find the optimal values for the parameter as these settings can be used to make a balance between the exploration and exploitation phase. One such example of the parameter is No_Of_Ant; the larger the value of this parameter, the more will be the classification rules which results in an increase in model training time.

Table 8 Performance comparison of the fuzzy and non-fuzzy rule-based classifier on Heart-Disease (test set)

S.No	Algorithm	Classification accuracy	Execution time (approx.)
1	J48	85.79	0.09
2	SVM	80.50	0.22
3	AntMiner1	79.38	12.23
4	cAntMiner	81.34	38.25
5	cAntMiner _{PB}	81.06	7.68
6	Fuzzy-ACI	85.06	32.89

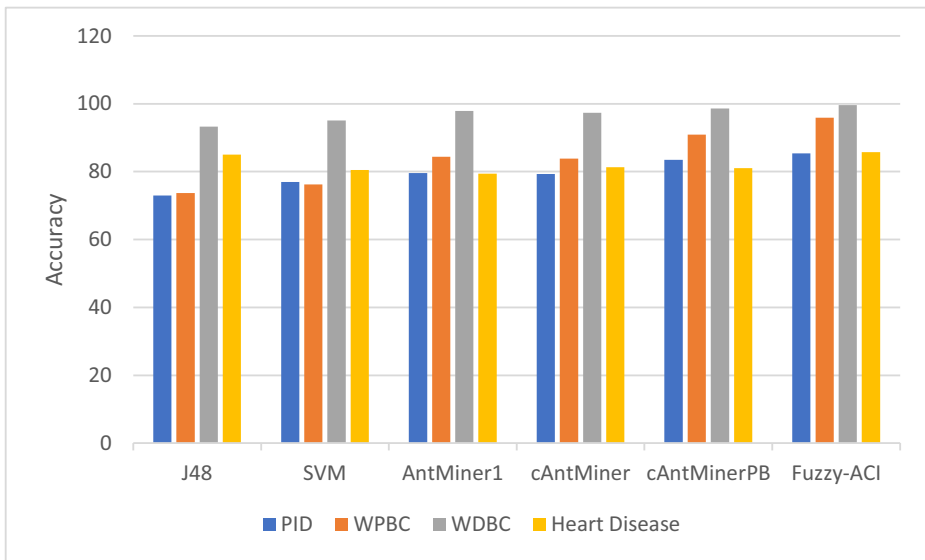


Fig. 3 Comparison of models' accuracy across four datasets

7.2 Termination conditions

A model's determination of the terminating condition is crucial. Our system seeks a solution that satisfies a predetermined termination condition. If one of these conditions is met, the wisest course of action is to return then the best solution. However, attaining the terminating condition too quickly may result in a suboptimal solution, whilst searching beyond the terminating condition will provide no significant results.

7.3 Rule pruning

The classification rules generated by our system depend on the number of features in the dataset. If a dataset has more features, then the rules generated by the system would be long, which reduces the comprehensibility of the rule. Consequently, the rules are difficult to understand and interpret, which also affects the training time.

7.4 Execution time

The suggested Fuzzy-ACI system has a longer training time than previous rule-based classification systems.

8 Conclusion and future work

This research proposes a disease diagnosis medical decision assistant system, ACI. It is a rule-based classification system that is built by the introduction of fuzzy logic with an ant colony optimization technique. The artificial ants generated potential fuzzy search criteria, and

stochasticity the method encourages ants to develop more precise rules. The fuzzy inference engine draws the conclusion about the testing pattern based on an enhanced set of rules. The performance of the system is tested on ten-fold cross-validation, and by supplying a separate test file, the suggested system obtains the highest classification accuracy across all datasets. The results show that our proposed Fuzzy-ACI system outperforms several well-known rule-based classification methods available such as J48, SVM, AntMiner1, cAntMiner, and cAntMiner_{PB}. The proposed algorithms can be evaluated further using standard medical datasets and used to diagnose real-world hospital data. This system finds its application in areas like the medical, banking, financial, and education sector. Given that the classification accuracy of the diagnosis of the disease is more significant than the model's execution time, this work focuses primarily on maximizing the accuracy. Thus, the planned works can be expanded by considering the model's execution time. In addition, a comprehensive examination of optimum parameter settings, such as self-adaptive controls, is required for rule-based classification systems.

Acknowledgments There was no particular grant from a governmental, commercial, or non-profit funding agency for this study.

Declarations

Competing interests All the authors of this study have stated that they do not have any relationships, either personal or financial, with any other individuals who would in any way be able to influence the results of this research.

References

1. Aha D (1987) UC Irvine machine learning repository, [Online]. Available: <http://archive.ics.uci.edu/ml/index.php>. Accessed 7 May 2022
2. Al-Behadili HNK, Sagban R, Ku-Mahamud KR (2020) Adaptive parameter control strategy for ant-miner classification algorithm. *Indonesian Journal of Electrical Engineering and Informatics (IJEI)* 8(1):149–162
3. Ali S, El-Sappagh S, Ali F, Imran M, Abuhmed T (2022) Multitask deep learning for cost-effective prediction of patient's length of stay and readmission state using multimodal physical activity sensory data. *IEEE J Biomed Health Inform*
4. Baig AR, Shahzad W (2012) A correlation-based ant miner for classification rule discovery. *Neural Comput Appl* 21(2):219–235
5. Basak H, Kundu R, Singh PK, Ijaz MF, Woźniak M, Sarkar R (2022) A union of deep learning and swarm-based optimization for 3D human action recognition. *Sci Rep* 12(1):1–17
6. Chan A, Freitas A (2005) A new classification-rule pruning procedure for an ant colony algorithm, in *International Conference on Artificial Evolution (Evolution Artificielle)*, pp. 25–36
7. Chan A, Freitas AA (2006) A new ant colony algorithm for multi-label classification with applications in bioinformatics, in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 27–34
8. Dash S, Verma S, Khan MS, Wozniak M, Shafi J, Ijaz MF (2021) A hybrid method to enhance thick and thin vessels for blood vessel segmentation. *Diagnostics* 11(11):2017
9. Dash S, Verma S, Bevinakoppa S, Wozniak M, Shafi J, Ijaz MF (2022) Guidance image-based enhanced matched filter with modified thresholding for blood vessel extraction. *Symmetry (Basel)* 14(2):194
10. Dorigo M, Stützle T (2019) Ant colony optimization: overview and recent advances. *Handbook of metaheuristics*:311–351
11. Dorigo M, Maniezzo V, Colomi A (1991) Positive feedback as a search strategy
12. Dorigo M, Maniezzo V, Colomi A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 26(1):29–41

13. El-Rashidy N et al (2022) Sepsis prediction in intensive care unit based on genetic feature optimization and stacked deep ensemble learning. *Neural Comput Appl* 34(5):3603–3632
14. El-Sappagh S, Saleh H, Ali F, Amer E, Abuhmed T (Sep. 2022) Two-stage deep learning model for Alzheimer's disease detection and prediction of the mild cognitive impairment time. *Neural Comput Appl* 34(17):14487–14509. <https://doi.org/10.1007/s00521-022-07263-9>
15. Ganji MF, Abadeh MS (2011) A fuzzy classification system based on ant Colony optimization for diabetes disease diagnosis. *Expert Syst Appl* 38(12):14650–14659
16. He J, Long D, Chen C (2007) An improved ant-based classifier for intrusion detection. *Third International Conference on Natural Computation (ICNC 2007)* 4:819–823
17. Heart Disease Dataset [Online] Available: <https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset>. Accessed 7 May 2022
18. Jin P, Zhu Y, Hu K, Li S (2006) Classification rule mining based on ant colony optimization algorithm, in *Intelligent Control and Automation*, Springer, pp 654–663
19. Kumar Y, Koul A, Singla R, Ijaz MF (2022) Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda. *J Ambient Intell Humaniz Comput*:1–28
20. Liang S-C, Lee Y-C, Lee P-C (2011) The application of ant colony optimization to the classification rule problem, in 2011 IEEE International Conference on Granular Computing, pp 390–392
21. Liang Z, Sun J, Lin Q, Du Z, Chen J, Ming Z (2016) A novel multiple rule sets data classification algorithm based on ant colony algorithm. *Appl Soft Comput* 38:1000–1011
22. Liu B, Abbass HA, McKay B (2002) Density-based heuristic for rule discovery with ant-miner, in *The 6th Australia-Japan joint workshop on intelligent and evolutionary system*, vol. 184
23. Liu B, Abbas HA, McKay B (2003) Classification rule discovery with ant colony optimization, in *IEEE/WIC International Conference On Intelligent Agent Technology. IAT 2003*, pp 83–88
24. Mandal M, Singh PK, Ijaz MF, Shafi J, Sarkar R (2021) A tri-stage wrapper-filter feature selection framework for disease classification. *Sensors* 21(16):5571
25. Nasir HJA, Ku-Mahamud KR (2010) Grid load balancing using ant colony optimization, in 2010 Second International Conference on Computer and Network Technology, pp 207–211
26. Nasir HJA, Ku-Mahamud KR, Din AM (2010) Load balancing using enhanced ant algorithm in grid computing, in 2010 Second International Conference on Computational Intelligence, Modelling and Simulation, pp 160–165
27. Nozaki K, Ishibuchi H, Tanaka H (1996) Adaptive fuzzy rule-based classification systems. *IEEE Trans Fuzzy Syst* 4(3):238–250
28. Otero FEB, Freitas AA, Johnson CG (2008) cAnt-Miner: an ant colony classification algorithm to cope with continuous attributes, in *International Conference on Ant Colony Optimization and Swarm Intelligence*, pp 48–59
29. Otero FEB, Freitas AA, Johnson CG (2009) Handling continuous attributes in ant colony classification algorithms, in 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp 225–231
30. Otero FEB, Freitas AA, Johnson CG (2012) Inducing decision trees with an ant colony optimization algorithm. *Appl Soft Comput* 12(11):3615–3626
31. Parpinelli RS, Lopes HS, Freitas AA (2002) Data mining with an ant colony optimization algorithm. *IEEE Trans Evol Comput* 6(4):321–332. <https://doi.org/10.1109/TEVC.2002.802452>
32. Prabha GM, Balraj E (2014) A HM ant miner using evolutionary algorithm. *Int J Innov Res Sci Eng Technol* 3(3):1687–1692
33. Rajiplawala S, Singh DK (2014) Enhanced cAntMiner PB algorithm for induction of classification rules using ant Colony approach. *IOSR J Comput Eng* 16(3):63–72
34. Sagban R, Ku-Mahamud KR, Abu Bakar MS (2015) ACOustic: a nature-inspired exploration indicator for ant colony optimization. *The Scientific World Journal* 2015:1–11
35. Saian R, Ku-Mahamud KR (2011) Hybrid ant colony optimization and simulated annealing for rule induction, in 2011 UKSim 5th European Symposium on Computer Modeling and Simulation, pp 70–75
36. Salama KM, Abdelbar AM (2010) Extensions to the Ant-Miner classification rule discovery algorithm, in *International Conference on Swarm Intelligence*, pp 167–178
37. Salama KM, Otero FEB (2013) Using a unified measure function for heuristics, discretization, and rule quality evaluation in Ant-Miner," in 2013 IEEE Congress on Evolutionary Computation, pp 900–907
38. Shahzad W, Baig AR (2010) Compatibility as a heuristic for construction of rules by artificial ants. *Journal of Circuits, Systems, and Computers* 19(01):297–306
39. Smaldon J, Freitas AA (2006) A new version of the ant-miner algorithm discovering unordered rule sets, in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp 43–50
40. Srinivasu PN, Ahmed S, Alhumam A, Kumar AB, Ijaz MF (2021) An AW-HARIS based automated segmentation of human liver using CT images. *Comput Mater Contin* 69(3):3303–3319

41. Thangavel K, Jaganathan P (2007) Rule mining algorithm with a new ant colony optimization algorithm. in International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) 2: 135–140
42. Tripathy S, Hota S, Satapathy P (2013) MTACO-miner: modified threshold ant colony optimization miner for classification rule mining, *Emerging Reserch in Computing, Information, Communication and Application*, no. August 2013, pp 1–6
43. Verbeke W, Martens D, Mues C, Baesens B (2011) Building comprehensible customer churn prediction models with advanced rule induction techniques. *Expert Syst Appl* 38(3):2354–2364
44. Vulli A, Srinivasu PN, Sashank MSK, Shafi J, Choi J, Ijaz MF (2022) Fine-tuned DenseNet-169 for breast cancer metastasis prediction using FastAI and 1-cycle policy. *Sensors* 22(8):2988
45. Weka 3: Machine Learning Software in Java (1993). [Online] Available: <https://www.cs.waikato.ac.nz/~ml/weka/>. Accessed 11 May 2022
46. Wu H, Sun K A simple heuristic for classification with ant-miner using a population. in 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, 2012 1:239–244
47. Young M (2002) *Technical writer's handbook*. University Science Books
48. Zadeh LA (1965) Fuzzy sets. *Inf Control* 8(3):338–353

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.