International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

"A Comparative Study of Classification Rule Discovery with Ant Colony Optimization: AntMiner"

Bhawna Jyoti[a]*, Dr. Aman Kumar Sharma[b]

[a,b]*Computer Science Department, Himachal Pradesh University, Shimla, India*
[a]*Email: bhawnashkl@gmail.com*
[b]*Email: sharmaas1@gmail.com*

Abstract

Rule based classification is the fundamental and important task of data classification. To discover classification rules, ant colony optimization algorithms are successfully applied that follow a sequential covering approach to build a list of rules. AntMiner Rule Based Classification algorithms are inspired from self- organizing behaviour of ant colonies. In this paper, we presented a study on Ant Colony Optimization Algorithm, AntMiner, c_AntMiner, c_AntMiner2, c_AntMiner PB and conducted experiments to find predictive accuracy against well-known rule induction algorithms JRIP and PART and results shows that AntMiner and its variants shows comparable as well as better performance in some datasets taken in the experimental study.

Keywords: Ant Colony Optimization; AntMiner; Classification Rules; Rule induction; Sequential covering Strategy.

1. Introduction

Classification is the important task of human decision making which comprises of assigning an object to a predefined class according to its characteristics [5,6,7]. Ant-Miner and its derivatives have been successfully used for discovering classification rules in the domain of data mining [7,11,12]. Ant Colony Optimization is a metaheuristic technique proposed by Marco Dorigo in 1992 who pioneered the research in this area and different variants have appeared afterwards[4,5,6]. This heuristic technique has set up a basic platform which comprises a framework applicable to real life existing applications and its variants [1,2]. Such algorithms which are based on the characteristics of ant behavior are known as ant colony optimization algorithms [7].

----

* Corresponding author.

In data mining, classification algorithm are built and learning behaviour of these algorithm is analyzed by using statistical measurements to validate the hypothesis of classification algorithm [5,6,7]. A classification model is built by analyzing the training set is used to classify the value of the class attribute [13,14]. Ant Colony optimization algorithms are used to discover a list of *IF-THEN* classification rules [5]. The artificial ants create stochastic solution using construction procedure by building an optimal solution based on the iterative addition of components to partial solutions [10,11,12].

Ant-Miner is based on sequential covering strategy in which the discovery of a rule is considered as an independent search problem and the best rule is discovered by constructing a heuristic solution to the given problem [5,13,14,19,20]. Selection of class value is achieved during each repetitive cycle of ant which is directly proportional to the concentration of pheromone levels left behind by foraging behaviour of ant colony in search of food [1,2,3,4].

The Paper outlined is given as follows: Sec II will provide an overview of ant colony Optimization, AntMiner and its variants. Sec III describes comparative study of AntMiner algorithms with existing rule induction algorithms. Sec IV analysis about predictive accuracy of these algorithms on different datasets and Sec 5 discusses future research issues and conclusion.

2. Ant Colony Optimization

Real ants live together in colonies having population about 2 to 25 millions. An Ant releases scent chemicals called pheromone to communicate with each other. When foraging in their local environment, a swarm of ants communicate with each other by the pheromone trails [5,29]. When a food source is found by ant, it is marked by pheromone and also marking of the trails is done from source to destination. The concentration of the pheromone varies from the initial random foraging route and ants follow the route with higher concentration of pheromone [11,12]. The increasing number of ants going on the same route will increase the quantity of pheromone and it will become favorable and shortest path. There is no master ant overseeing the entire colony and such emergent behavior is very similar to self organized phenomena. Some ant species such as Army Ants show extraordinary regularity by following the route with an angle of 123° apart [5,11,12,25,29]. These algorithms are based on the (1)the problem containing heuristic information, if available (2)artificial pheromone trails that keeps on changing dynamically to enhance search space experience of ants as agents[5,26,28]. The application domains of ACO algorithms is ever increasing as it can be applied to any discrete optimization problem for which some solution construction mechanism can be formulated[12,14].

2.1 Basic Ideology Used in AntMiner

By collective behavior of ants in self organized group, this algorithm has the ability to find best solutions in large search space [2,3]. A candidate solution to a given problem is associated with each path followed by an ant. When an ant follows a path, the quality of given solution for a particular problem is directly proportional to the amount of pheromone deposited on foraging path and decision is made to choose a path by giving preference with a large amount of pheromone present on the path[16,17,18]. Ants incrementally construct the appropriate solution of the given problem by using probability transition rule function [5][29]which is given below:

$$P_{ij} = \frac{(\emptyset_{ij})^{\alpha}(d_{ij})^{\beta}}{\sum_{i,j=1}^{n}(\emptyset_{ij})^{\alpha}(d_{ij})^{\beta}}$$

(1)

Where in (1) equation, P_{ij} denotes the probability of ants to select the route from a particular node i to reach node j; $(\emptyset_{ij})^{\alpha}$ is the pheromone concentration on the route between i and j and $(d_{ij})^{\beta}$ is the desirability of the same route which is inversely proportional to s_{ij} which implies that shorter routes will be selected due to their shorter travelling time; α and β are influence parameters and their typical values are $\alpha \approx \beta \approx 2$. This formula reflect the fact that the ants will normally follow the path with higher pheromone concentrations [5,25,28,29].

The concentration of pheromone is changed with time due to evaporation [1,2,3]. It is also advantageous that path randomly chosen by the first ant will not become the preferred path as the other ants will attract due to concentration of pheromone present on the path [6,7,8,9,10]. For a constant rate γ of pheromone decay varies exponentially with time by the following function:

$$\emptyset(t) = \emptyset_0 e^{-\gamma t} \quad (2)$$

Where in equation(2), $\emptyset_0$ is the initial concentration of pheromone and t is the time and γ is the constant rate of evaporation[5,28,29] . For the unitary time increment, the pheromone update formula is :

$$\emptyset_{ij}^{t+1} = (1-\gamma)\emptyset_{ij}^{t} + \delta\emptyset_{ij}^{t} \quad (3)$$

Where in equation (3), $\gamma \varepsilon [0,1]$ is the rate of pheromone evaporation. The increment $\delta\emptyset_{ij}^{t}$ is the amount of pheromone deposited at time t along route *i* to *j* when an ant travels a distance L. Usually $\delta\emptyset_{ij}^{t} \alpha \frac{1}{L}$ and if there is no ant on the foraging path then deposit of pheromone will be zero[5,28,29]. A possible acceleration scheme is used to find the global best solution by using some bounds of pheromone concentration used [9,10,11,12].

Algorithmic Skeleton of Ant Colony Optimization

procedure antColony

Objective function $f(x_{ij})$ where *x=(*$x_1, x_2 ..., x_n$*)* and *(i,j) ε* {1,2,…n}

Define evaporation rate γ

while (*criterion)*

forloop *over all nodes*

new solutions generate

evaluations of solutions

better locations identification with pheromone

pheromone updation using $\emptyset_{ij}^{t+1} = (1-\gamma)\emptyset_{ij}^{t} + \delta\emptyset_{ij}^{t}$

end for

find the current best

end while

Output the best results

Ant colony optimization algorithms are used to discover classification rules from a given data set based on sequential covering approach by implementing the system Ant-Miner [28,30,31,32].

In an ACO algorithm, each ant incrementally construct the solution and in Ant-Miner, each ant is used for the creation of classification rules [5,6,10,13]. It consists of the form: IF<*term1* AND *term2*…> THEN<class> where each term describes value of attribute pair upon which consequent part is selected to predict the class value of discovered rule [5,29]. Firstly, ant starts with an empty rule list. All trails with the same amount of pheromone is initialized [5,10]. The current partial rule constructed by an ant and being added to the current rule [26,27,28] in the generated list of classification rules. The probability that a $term_{ij}$ is chosen to the current partial rule is given by the following equation:

$$P_{ij} = \frac{x_{ij}.\tau_{ij}}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b} x_{ij}.\tau_{ij}} \tag{4}$$

Where in equation (4), x_{ij} is the problem dependent heuristic function for term and τ_{ij} is the amount of pheromone associated with term[5,26,28,29]. Rule constructed by an ant is pruned to remove irrelevant terms and attribute interactions are ignored.

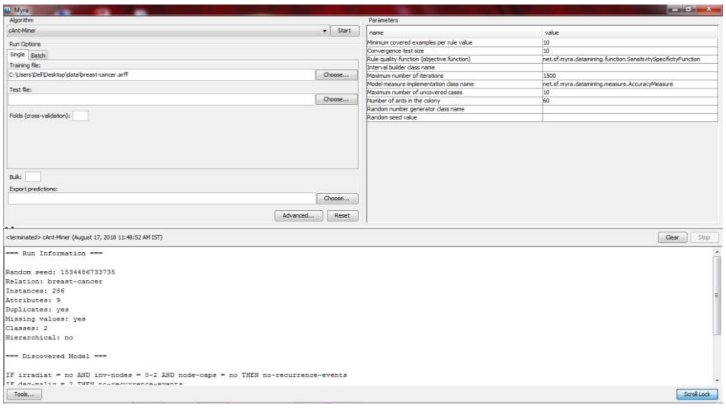
2.2 Variants of AntMiner

There are many variations and improvements in AntMiner rule induction algorithm like, Ant tree miner, c_AntMiner, c-AntMiner2, c-AntMiner-PB for the rule extraction [15,14,31].

Ant-Tree-Miner [32] is based on traditional top-down approach which uses divide-and-conquer strategy and a decision tree is created from the root node to leaf nodes iteratively. Firstly, an attribute presents the root node of tree and each branch is represented by particular attribute value. Data is divided into subsets associated with its corresponding branch and each attribute selection is repeated recursively until all instances present on a subset corresponds to the same class value or the number of instances is lesser than a particular defined minimum value [5,32]. A leaf node predicting the majority class value will be added to the tree and the recursive process of ant tree miner stops [5,17,31].

The c_Ant-Miner algorithm and its extensions like c_Ant-Miner2, c_Ant-MinerPB can deal with continuous attributes without implementing C4.5 algorithm as a discretization preprocessing step [5,15,32]. In the rule construction process threshold values are generated using binary discretization in the continuous attributes domain [30]. In the construction graph of Ant-Miner, nodes for each continuous attribute are added and connected to all other nodes. These nodes do not represent valid terms, as they do not have a relational operator and an associated value [26,27]. An ant selects a continuous attribute node which determines value of operator dynamically and node is added to the next term to the antecedent of a rule.

In AntMiner, the search is guided by quality of the individual rules and the best rule found is always given priority regardless of the fact that how it affects the list of rules but in c_AntMinerPB, focus is on the improvement of entire list quality rather than individual rule generated by ant [5,21,22,15].



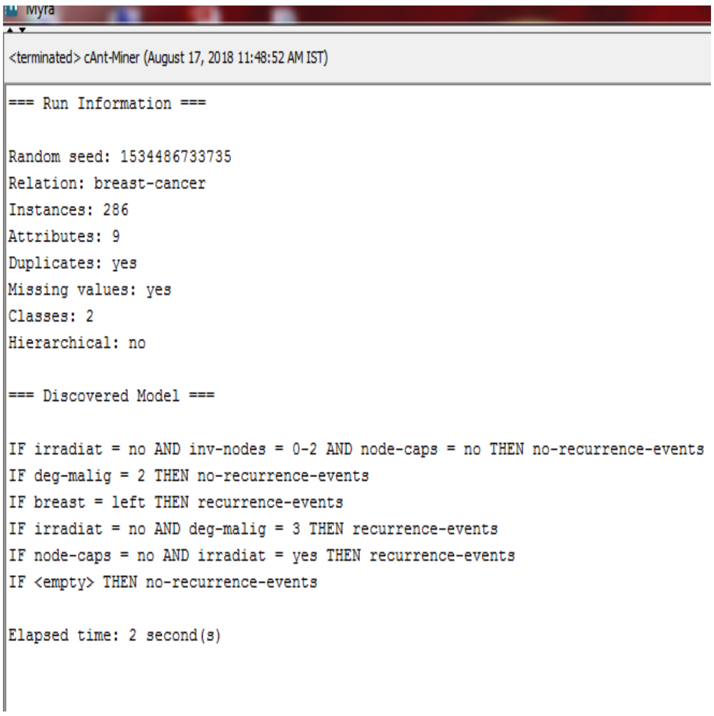


Figure 1: Snapshots of runtime environment of Ant Miner, c-AntMiner

3. Experimental Study

The performance of AntMiner is evaluated by using ten-fold cross validation after splitting the dataset into 10 stratified folds. In this strategy, all cases are used only once as testing dataset and (k-1) times as training dataset. The final accuracy rate is calculated by finding the average of the accuracy rate of the *k* iterations which are randomly generated considering all available cases. For extracting classification rules from data, GUI Ant-Miner tool is used. In it, the data input file is used which complies with the ARFF (Attribute-Relation File Format) of the Weka tool and file is standardized with the well-known Weka system which runs on virtually any operating system since it is written in Java. Myra is a open source framework for Ant Colony Optimization based on Java language and offers a specialized data mining layer to support implementation of Ant-Miner, c_Ant-Miner, Ant-Tree-Miner [17]. Datasets are taken from UCI Repository of Machine Learning [34]. The main characteristics of the datasets used in the experiments are shown in table 1. Parameters taken for experimental study:

- Convergence test size :10
- Maximum iterations covered by ant :1500
- Maximum size number of uncovered branches: 5
- Number of ants in colony :60
- Value of Max-Min Evaporation factor :0.90
- Minimum number of examples per rule :10
- % of maximum number of uncovered classes: 1.0

Some of the datasets contain only nominal attributes while others contain combination of nominal and continuous attributes.

Table 1: Description of Dataset used in Experiments

Dataset	Instances	Nominal Attributes	Continuous Attributes	Classes	Hierarchal values	Duplicate values	Missing values
weather nominal	14	4	0	2	No	no	No
weather numeric	14	2	2	2	No	no	No
Soyabean	683	35	0	19	No	yes	yes
Labor	57	8	8	2	No	no	yes
iris2d	150	2	2	3	No	yes	No
Glass	214	0	9	7	No	yes	No

Six data sets are taken for experimental study and predictive accuracy of the generated rules is recorded using different random seeds by implementing AntMiner, Ant-tree-miner, c- AntMiner, c-AntMiner2,c-AntMiner-PB, JRIP and Part (Weka based induction algorithms). Experimental results are discussed in Table 2. The results values of algorithms that are shown in bold describes high predictive accuracy rate for that particular data set value as shown in Table 2.

Table 2: A Comparative Study of Predictive Accuracy (average$\pm$ standard deviation) different AntMiner Algorithms with JRIP and PART Algorithms

Data set	Ant miner	ant tree miner	c -ant miner	c-ant miner2	c-ant miner pb	j rip	part
weather(nominal)	65.0 $\pm$ 13.017	65.0$\pm$13.017	67.5$\pm$8.331	68.33 $\pm$6.558	69.0 $\pm$4.926	**70.0 $\pm$0.367**	66.7 $\pm$0.436
weather(numeric)	Na	**67.5 $\pm$ 4.576**	66.4 $\pm$ 4.283	65.6 $\pm$4.036	66.1 $\pm$3.775	50.0$\pm$0.55	61.5 $\pm$0.551
Soyabean	68.37 $\pm$3.465	66.6 $\pm$0.0	88.2 $\pm$1.525	88.4 $\pm$0.809	87.8$\pm$0.79	92.2 $\pm$0.011	**92.4 $\pm$0.015**
Labor	Na	86.3$\pm$4.880	**88.0 $\pm$4.394**	87.6 $\pm$3.687	75.3 $\pm$4.560	77.1$\pm$0.22	78.9 $\pm$0.288
iris 2d	Na	84.2$\pm$6.099	94.0 $\pm$1.197	94.0 $\pm$1.846	**94.6 $\pm$2.177**	94.2 $\pm$0.050	94.2 $\pm$0.0445
Glass	Na	69.9 $\pm$1.787	66.7 $\pm$2.815	62.1 $\pm$3.249	**71.2 $\pm$2.976**	64.5 $\pm$0.115	66.5 $\pm$0.044

4. Analysis

As Table 2 indicates that by comparing AntMiner algorithms with existing rule induction algorithms JRIP and PART, it is seen that AntMiner is unable to support continuous attributes in case of weather numeric, labor, iris2d and glass datasets and predictive accuracy rate is higher: JRIP in weather (nominal), cAntMiner in weather (numeric), PART in soyabean, cAntMiner in labor, cAntMiner PB in iris 2D and tree Miner in glass dataset.

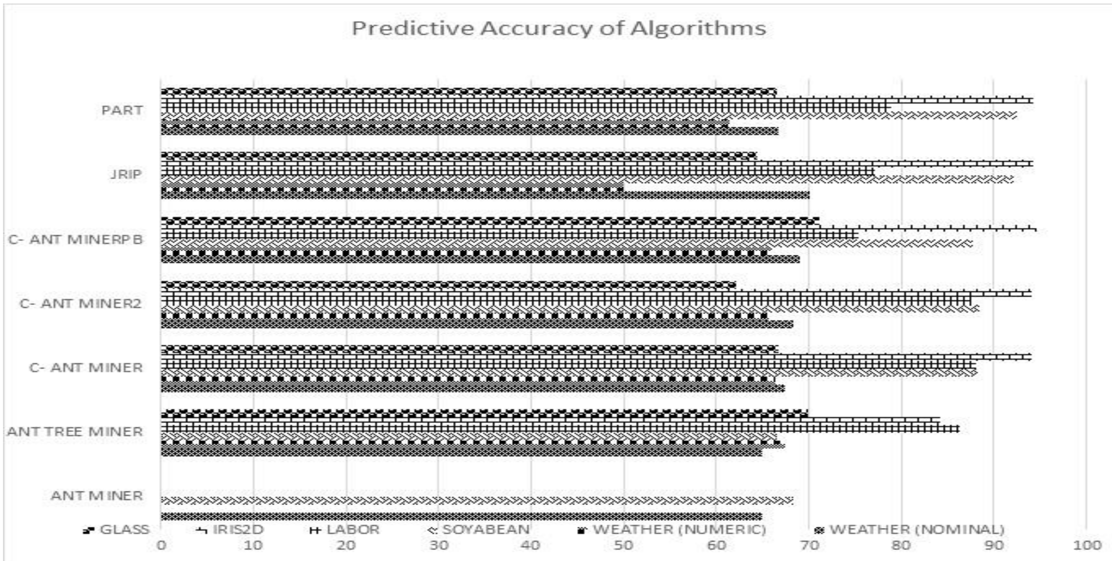


Figure 2: Predictive Accuracy of AntMiner algorithms with existing rule induction algorithms JRIP and PART.

5. Conclusion and Future work

We discussed discovery of classification rules by using AntMiner and its variations. The outcome of a discovered rule is affected by previously discovered rules depends on the pheromone matrix and quality function used by ant colony algorithm. We conducted experiments involving six publicly available data sets, and compared the results against PART and JRIP. Ant Miner and its variations show good predictive accuracy for classification rule discovery in data mining. A heuristic solution of given problem is represented by generating a complete list of rules created by single iteration of ant. An optimal solution of given heuristic problem can be obtained by evaluating global pruning procedure of the entire rule list of discovered rules rather than pruning each rule individually in the search space as a future research direction.

References

[1]. G. Chennupati. "Eant-miner: an ensemble ant-miner to improve the aco classification", arXiv preprint arXiv, pp. 2014 .

[2]. S. P. Rami, H. M. Panchal. "Comparative Analysis of Variations of Ant-Miner by Varying Input Parameters", International Journal of Computer Applications, vol. 60, issue 3, 2012.

[3]. X. Y. Zhong, Z. Lei, Z, Yu-Fang. "A classification rule mining method using hybrid genetic algorithms" in IEEE Region 10 Conference TENCON, 2004, pp. 207-210.

[4]. B. Suri, S. Singhal."Implementing ant colony optimization for test case selection and prioritization," International journal on computer science and engineering, vol. 3, issue 5, pp. 1924-32, May 2011.

[5]. F.E. Otero, A. A. Freitas, C.G. Johnson. A new sequential covering strategy for inducing classification rules with ant colony algorithms. IEEE Transactions on Evolutionary Computation. vol.17, issue 1, pp. 64-76. Feb 2012.

[6]. R. S. Parpinelli, H.S. Lopes, A. A. Freitas. "Data mining with an ant colony optimization algorithm," IEEE transactions on evolutionary computation. vol.6, issue. 4, pp. 321-32, Aug 2002.

[7]. M. Dorigo, T. Stutzle. "Ant colony optimization: overview and recent advances," in Handbook of metaheuristics Springer, pp. 311-351, 2019.

[8]. R. Seidlova, J. Pozivil. "Implementation of Ant Colony Algorithms in Matlab," Institute of Chemical Technology, Department of Computing and Control Engineering, 2012.

[9]. S. Katiyar, N. Ibraheem, A. Q. Ansari. "Ant colony optimization: a tutorial review," in Proceedings of the National Conference on Advances in Power and Control, Manav Rachna International University, Faridabad, Haryana, India, Aug 2015,vol. 574.

[10]. H. Li, C. P. Lam. "Software Test Data Generation using Ant Colony Optimization," in International conference on computational intelligence, 2004,pp. 1-4.

[11]. S. Roy, S. S. Chaudhuri. "Bio-inspired ant algorithms: A review," International Journal of Modern Education and Computer Science. vol. 5, issue. 4, May 2013.

[12]. B. C. Mohan, R. Baskaran. "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," Expert Systems with Applications, vol. 39, issue. 4, pp. 4618-27, Mar 2012.

[13]. J. Brookhouse, F.E. Otero. "Discovering regression rules with ant colony optimization," in

Proceedings of the Companion Publication of the 2015 ACOGEC, 2015, pp. 1005-1012. ACM.

[14]. R. S. Parpinelli, H. S. Lopes, A. A. Freitas. “An ant colony based system for data mining: applications to medical data,” in Proceedings of the 3rd ACOGEC, 2001, pp. 791-797.

[15]. M. Medland, F. E. Otero, A. A. Freitas. “Improving the cAnt-Miner PB classification algorithm,” in international Conference on Swarm Intelligence, Sep 2012, pp. 73-84. Springer.

[16]. D. Martens, D. Backer, R. Haesen, J. Vanthienen, M. Snoeck, B. Baesens. “Classification with ant colony optimization,” IEEE Transactions on Evolutionary Computation. vol. 11, issue. 5, Oct 2007, pp. 651-65.

[17]. F. E. Otero. “MYRA: a Java ant colony optimization framework for classification algorithms,” in proceedings of the Genetic and Evolutionary Computation Conference Companion, Jul 2017, pp. 1247-1254. ACM.

[18]. K. M. Salama, A. M. Abdelbar, F. E. Otero, A. A. Freitas. “Utilizing multiple pheromones in an ant-based algorithm for continuous-attribute classification rule discovery,” Applied Soft Computing, Jan 2013, vol. 13, issue 1, pp. 667-75.

[19]. Otero, A. A. Freitas. “Improving the interpretability of classification rules discovered by an ant colony algorithm,” in Proceedings of the 15th annual conference on GEC, ACM, 2013, pp. 73-80.

[20]. A. Helal, F. E. Otero. “A mixed-attribute approach in ant-miner classification rule discovery algorithm,” in Proceedings of the Genetic and Evolutionary Computation Conference, 2016, pp. 13-20, ACM.

[21]. J. Brookhouse, F. E. Otero. “Using an Ant Colony Optimization Algorithm for Monotonic Regression Rule Discovery,” in Proceedings of the GEC, Jul 2016, pp. 437-444. ACM.

[22]. A. Amuthan, K. D. Thilak. “Improved Ant Colony Algorithms for Eliminating Stagnation and Local Optimum Problem—A Survey,” in proceedings of ICTACC, Apr 2017, pp. 97-101, IEEE.

[23]. M. Dorigo, V. Maniezzo, A. Colorni. “Ant system: optimization by a colony of cooperating agents,” IEEE Transactions on Systems, man and cybernetics, Part B: Cybernetics. Feb 1996, vol. 26, issue. 1, pp. 129-41.

[24]. A. Helal, F. E. Otero. “Automatic design of ant-miner mixed attributes for classification rule discovery,” in Proceedings of the Genetic and Evolutionary Computation Conference, Jul 2017, pp. 433-440. ACM.

[25]. A. Helal, J. Brookhouse, F. E. Otero. “Archive-Based Pheromone Model for Discovering Regression Rules with Ant Colony Optimization,” IEEE Congress on Evolutionary Computation (CEC), Jul 2018, pp. 1-7.

[26]. W. J. Jiang, Y. H. Xu, Y. S. Xu. “A novel data mining algorithm based on ant colony system,” in International Conference on Machine Learning and Cybernetics, Aug 2005, vol. 3, pp. 1919-1923, IEEE.

[27]. H. N. Al-Behadili, K. R. Ku-Mahamud, R. Sagban. “Rule pruning techniques in the ant-miner classification algorithm and its variants: A review,” in IEEE Symposium on ISCAIE, Apr 2018, pp. 78-84, IEEE.

[28]. V. K. Panchal, P. Singh, A. Narula, A. Mishra. “Review on ant miners,” in World Congress on Nature & Biologically Inspired Computing, Dec 2009, pp. 1641-1644, IEEE.

[29]. Yang XS. "Nature-Inspired Metaheuristic Algorithms," Second Edition. Luniver Press, 2010.

[30]. Shah C, Jivani AG. "Comparison of data mining classification algorithms for breast cancer prediction," in Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Jul 2013, pp. 1-4. IEEE.

[31]. A. Colorni, M. Dorigo, F. Maffioli, V. Maniezzo, G. I. Righini, M. Trubian. "Heuristics from nature for hard combinatorial optimization problems," International Transactions in Operational Research. 1996, vol. 3, issue. 1, pp. 1-21.

[32]. K. M. Salama, A. M. Abdelbar, F. E. Otero. "Investigating evaluation measures in ant colony algorithms for learning decision tree classifiers," in IEEE symposium series on computational intelligence, 2015, pp. 1146-1153.

[33]. S. H. Ripon, "Rule Induction and Prediction of Chronic Kidney Disease Using Boosting Classifiers, Ant-Miner and J48 Decision Tree," International Conference on ECCE, Feb 2019, pp. 1-6, IEEE.

[34]. http://www.ics.uci.edu/~mlearn/MLRepository.html