Comparative Study on Bio-inspired Approach for Soil Classification

K. Sumangala[1] and G. Nithya[2]
[1]Assistant Professor, [2]Research Scholar
Department of Computer Application,
Vellalar College for Women, Erode, Tamil Nadu

ABSTRACT

Ant miner is a data mining algorithm based on Ant Colony Optimization. Ant miner algorithms are mainly for discovery rule for optimization. Ant miner+ algorithm uses MAX-MIN ant system for discover rules in the database. Soil classification deals with the systematic categorization of soils based on distinguished characteristics as well as criteria. In this paper, Ant miner and Ant miner+ algorithm were applied to both training and soil dataset to obtain classification rules and found that Ant miner+ performs better than Ant miner.

General Terms

Classification, Soil, Supervised Learning

Keywords

Ant Colony Optimization, Ant miner, Ant miner+.

1. INTRODUCTION

Data mining is a gifted and relatively new technology and it is defined as a process of discovering hidden valuable and useful facts by analyzing huge amounts of data storing in databases or data warehouse by means of different techniques such as machine learning, Artificial Intelligence(AI) and statistics. Machine Learning, a branch of Artificial Intelligence and the types of learning are supervised learning, unsupervised learning, semi-supervised learning and Reinforcement learning. One of the supervised learning is classification. Application of data mining includes retail industry, telecommunication industry, biological data analysis, intrusion detection and aerospace to take advantages over their competitors. Large-scale organizations apply various data mining techniques on their data, to extract useful information and patterns. Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [7]. Some of the data mining tasks are clustering, classification, regression and dependence modeling.

Swarm intelligence (SI) is an Artificial Intelligence (AI) technique and it is a collective behaviour of trustworthy, decentralized, self-organized systems [3]. One of the swarm intelligence techniques is Ant Colony Optimization (ACO). Ant Colony Optimization is an meta heuristic algorithm inspired in the cooperative foraging behavior of ants to find and exploit the food source that is nearest to nest. ACO is based on supportive search paradigm that can be applicable to the solution of combinatorial optimization problem.

Ants communicate with each other by means of an indirect form of communication mediated by pheromone [3] [9]. ACO can also be used for the classification task of data mining. The first ACO algorithm for discovering classification rules was Ant miner and it was proposed by Parpinelli, Lopes and Freitas [11]. Ant-based search is more lithe and robust than conventional approaches. Ant miner uses a heuristic value based on entropy measure.

2. CLASSIFICATION

Classification is a supervised learning. Given a set of predefined categorical classes, determine to which of these classes a specific data item belongs [2][6]. The task of supervised classification is learning to expect the class membership of test cases given labeled training cases is a familiar machine learning problem. Classification is unsupervised learning, where training cases are also unlabeled. This type of classification, related to clustering, is often very useful in cautious data analysis, where one has few preconceptions about what structures new data may hold. Some of the classification algorithms are neural networks, k-nearest neighbor classifiers, Support vector machines, Rule based classifier and Fuzzy classification. Examples of classification application include image and pattern recognition, medical diagnosis, loan approval, detecting faults in industry applications and classifying financial market trends.

3. SWARM INTELLIGENCE

Swarm intelligence is a division of evolutionary computation, which is the application of methods motivated by the natural world to hard problems in artificial intelligence. Swarm intelligence was introduced by Gerardo Beni and Jing Wang in 1989 [3] and is the collective behaviour of decentralized, self-organized systems, natural or artificial. SI is mainly based on reliable, robustness and simplicity.

The four bases of self-organization are positive feedback, negative feedback (for counter-balance and stabilization), amplification of fluctuations (randomness, errors, random walks) and multiple interactions [3]. Examples are ant colonies, bird flocking, animal herding, bacterial growth, termites and fish schooling. Applications of SI are Ant-based routing, Crowd simulation. There are two popular swarm-inspired methods in the computational intelligence area Ant Colony Optimization and Particle Swarm Intelligence (PSO). ACO was enthused by the behavior of ants and has lots of applications in discrete optimization problems. PSO is a population based stochastic optimization inspired by the behaviour of flocks of birds and schools of fish [3].

4. ANT COLNY OPTIMIZATION

Ant Colony Optimization (ACO), which was introduced in the early 1990s as a novel technique to solve hard combinatorial optimization problems [9]. ACO is an optimization algorithm inspired in the collective foraging behavior of ants to find and exploit the food source that is nearest to the nest. ACO is based on cooperative search paradigm that is applicable to the solution of combinatorial optimization problem. Stigmergy, a type of indirect contact mediated by variation of the environment. Interactions among individuals or between individuals and the surroundings is based on the use of chemicals formed by the ants called pheromone [3]**.** The different types of Ant Colony models are Ant system for Traveling Salesman Problem (TSP), Ant miner, Ant quantity, Ant-cycle, Max-Min ACS, Ant System (AS) ranking model and Ant density. The [11] design of an ACO algorithm implies the specification of the following aspects:

- a suitable materialization of the problem, which allows the ants to incrementally build solutions through the use of a probabilistic transition rule, based on the amount of pheromone in the trail and on a local problem-dependent heuristic.
- a method to insist on the creation of valid solutions, i.e., solutions that are legal permissible in the real-world circumstances analogous to the problem definition.
- Problem-dependent heuristic function (*η*) is used to measure the feature of items that can be added to the current partial solution.
- a statute for pheromone updating, which specifies how to fine-tune the pheromone trail (τ)
- a probabilistic transition rule based on the assessment of the heuristic function (η) and on the contents of the pheromone trail (τ) that is used to iteratively to make a solution.

5. ANT MINER

Ant colony based data miner called Ant miner mainly for discover rules in database. Ant miner uses separate-and-conquer approach Each artificial path constructed by an ant represents a candidate classification rule of the form [12][14]:

IF <term1 AND term2 AND ….> THEN <class>

Each term is a triple <*attribute, operator, value*>, where *value* is one of the values belonging to the domain of *attribute* [11].

The rule antecedent (IF part) contains a set of conditions, usually connected by a logical conjunction operator (AND). Each rule condition as a term, so that the rule antecedent is a logical conjunction of terms in the form. The rule consequent (THEN part) specifies the class predicted for cases whose predictor attributes satisfy all the conditions specified in the rule antecedent [11].

Figure 1 represents an overview of Ant miner algorithm. In Ant-Miner, an artificial ant follows three procedures they are rule construction, rule pruning and pheromone updating to provoke a rule from a contemporary training dataset [12]. The artificial ant starts with an empty rule (no attribute terms in rule antecedent), and iteratively adds term to its current partial rule based on the local problem-dependent heuristic function involving information gain and positive feedback involving artificial pheromone level equation (2) & (3) are given below.

When an ant completes its rule and the amount of pheromone in each trail is updated, another ants start to construct its rule, using the new amounts of pheromone to guide its search [14]. The process is continual for at most a predefined number of ants. Then all cases correctly covered by the discovered rule are removed from the training set and iteration is started. Ant miner algorithm is called again and again to find a rule in the condensed training set. When the search for the rule in the training set is less than max_uncovered_cases the search for rule stops or when the value of the ant equal to the number of ants, a user specified threshold value. The discovered rules are stored in an ordered rule list.

```
Training set=all training cases;
While (No .of cases in the training set >max_uncovered_cases)
  i=0;
  REPEAT
      i=i+1;
      Ant_i incrementally constructs a classification rule;
      Prune the just constructed rule;
      Update the pheromone of the trail followed by Ant_i;
  UNTIL (i ≥ No_of_Ants)
      Select the best rule among all constructed rules;
      Remove the cases correctly covered by the selected
          rule from the training set;
END WHILE
```

Figure 1: Overview of Ant miner

5.1 Pheromone Initialization

Initially, for all attributes *i* and their possible values *j*, a preliminary amount of pheromone is deposited [4]. Pheromone deposited at every path is inversely proportional to the total number of values of all attributes, and is given by the following equation:

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

where *a* is the total number of attributes, b_i is the number of values in the domain of attribute *i*.

5.2 Rule construction

Each rule in Ant-Miner contains a condition part as the antecedent and a predicted class [4] [5]. The condition part is a combination (blend) of attribute-operator-value tuples [10]. The probability P_{ij}, that this condition is added to the current partial rule that the ant is constructing is given by the following equation:

$$\mathrm{P}_{ij}(t) = \frac{\tau_{ij}(t).\eta_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\tau_{ij}(t).\eta_{ij,}\forall i\varepsilon I} \qquad (2)$$

Where η_{ij} is the value of problem-dependent heuristic function for term_{ij} equation (4). The higher the value of η_{ij} the more appropriate for classification the term_{ij} is and so the higher its probability of being chosen. τ_{ij} is the amount of pheromone currently available.

5.3 Heuristic Function

The heuristic function (η) is based on the amount of information related with the attribute *i* and the amount of information is given by equation,

$$InfoT_{ij} = -\sum_{w=1}^{k}\left[\frac{freqT_{ij}^{w}}{\left|T_{ij}\right|}\right]*\log_2\left[\frac{freqT_{ij}^{w}}{\left|T_{ij}\right|}\right] \qquad (3)$$

Where k is the number of classes, $|T_{ij}|$ is the total number of cases in partition T_{ij} (partition containing the cases where attribute Ai has value V_{ij}), $freqT_{ij}^{w}$ is the number of cases in partition Tij with class w, a is the total number of attributes, and b_i is the number of values in the domain of attribute i. The higher the value of $InfoT_{ij}$, the less likely that the ant will choose $term_{ij}$ to add to its partial rule.

In Ant miner, the heuristic value is taken to be an information theoretic measure for the quality of the term to be added to the rule [4] [5]. The quality here is measured in terms of the entropy for preferring this term to the others, and is given by the following equations:

$$\eta_{ij} = \frac{\log_2(k) - InfoT_{ij}}{\sum_{i}^{a}\sum_{j}^{b_i}\log_2(k) - InfoT_{ij}} \qquad (4)$$

5.4 Rule Pruning

Immediately after the ant completes the construction of a rule, rule pruning is undertaken to increase the lucidity, accuracy of the rule and to avoid overfitting to noisy training data [1]. After the pruning step, the rule may be assigned a different predicted class based on the majority class in the cases covered by the rule antecedent. The rule pruning procedure iteratively removes the term whose removal will cause a maximum increase in the quality of the rule [1][4]. The quality of a rule is measured using the following equation:

$$Q = \text{Sensitivity} \times \text{Specificity} \qquad (5)$$

$$\text{Sensitivity} = \left(\frac{TruePos}{TruePos + falseNeg}\right)$$

$$\text{Specificity} = \left(\frac{TrueNeg}{FalsePos + TrueNeg}\right)$$

where TruePos is the number of cases roofed by the rule and having the same class as that predicted by the rule, FalsePos is the number of cases roofed by the rule and having a different class from that predicted by the rule, FalseNeg is the number of cases that are exposed by the rule, while having the class predicted by the rule, TrueNeg is the number of cases that are exposed by the rule which have a different class from the class predicted by the rule [4]. Sensitivity is the accuracy among positive instances and specificity is the accuracy among negative instances.

5.5 Pheromone Update Rule

When a rules is constructed by an ant and it is pruned, pheromone updating for a $term_{ij}$ performed based on the following equation

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) * Q\ \forall\ term_{ij}\, \varepsilon\, the\ rule \qquad (6)$$

To simulate the phenomenon of pheromone evaporation in real ant colony systems, the amount of pheromone associated with each $term_{ij}$ which does not occur in the constructed rule must be decreased. The cutback of pheromone of an unused term is performed by dividing the value of each τ_{ij} by the summation of all τ_{ij}.

6. ANT MINER+ ALGORITHM

Ant miner+, which applies *MAXMIN* Ant System (*MM*AS) and it use a greedier search than Ant System to results from these search space analysis of the combinatorial optimization problems and it is different from a normal ant system in three ways:

- After completion of every iteration only the best ant is allowed to add pheromone to its trail. This allows for a better exploitation of the best solution found.
- To avoid stagnation of the search, the range of possible pheromone trails is limited to an interval maximum and minimum [τ_{minn}, τ_{max}].
- Each trail is initialized with a pheromone value of τ_{max}, as such the algorithm achieves a higher exploration at the beginning of the algorithm.

```
construction graph
WHILE (not early stopping)
    Initialize heuristics, pheromones and probabilities of edges
    WHILE (not converged)
        create ants
        let ants run from source to sink
        evaporate pheromone on edges
        prune rule of best ant
        update path of best ant
        adjust pheromone levels if outside boundaries
        kill ants
        update probabilities of edges
    END
    extract rule
    flag data points covered by the extracted rule
END
evaluate performance on test set
```

Figure 2: Overview of Ant miner+

The Figure 2 represents the overview of Ant miner+ algorithm. Max-Min ant system approach additionally requires the pheromone levels to lie within a minimum and maximum interval. Convergence occurs when all the edges of one path have a higher pheromone level then the pheromone level of all others edges [9]. Next, the rule corresponding with the path is extracted and the training data covered by this rule is removed from the training set. The iterative process will be repeated until all ants visited the path.

Ant moving from one node to another is probabilistic based on the pheromone amount on the edge between the two nodes and the value calculated by a heuristic function equation (8) applied on the destination node. After a rule is constructed, it is evaluated against the training set and the pheromone on the rule edges is updated based on the quality of the rule equation (5). In each iteration the best rules generated are selected and added to the result rule set, the covered cases in the training set by the rules are removed and the pheromone in the construction graph is reset. The algorithm runs until all the ants finished their search.

6.1 Pheromone Initialization

Initially the maximum pheromone is deposited at each path,

$$\tau_{i,j}(t=0) = \tau_{\max} \qquad (7)$$

6.2 Heuristic Function

The heuristic function (η) is based on the amount of information associated with the attribute i and j is

$$\eta_{ij} = \left| \frac{T_{ij} \,\&CLASS = class_{ant}}{T_{ij}} \right| \qquad (8)$$

6.3 Pheromone Update Rule

When a rules are constructed by an ant and it is pruned, pheromone updating for a term$_{ij}$ is performed based on the equation (9) and quality (Q) of the rule is based on the equation(5). In [5] quality of the rule is based on the sum of confidence and coverage but in the paper sensitivity and specificity are used for calculating the quality of the rule.

$$\tau(v_{i,j,}v_{i+1})(t+1) = \rho.\tau(v_{i,j,}v_{i+1})(t) + \frac{Q}{10} \qquad (9)$$

7. EXPERIMENTAL RESULTS

The performance evaluations are evaluated using 4 training data set Iris, Glass, Wine, Soil and real soil dataset based on the accuracy and execution time. In the experiment training data sets are taken from the UCI repository. Irish dataset which contains 150 instances, 4 attribute and 3 classes (Iris Setosa, Iris Versicolour, Iris Virginica), wine dataset which contains 178 instances, 13 attribute and 3 classes, soil database contains 6435 instances, 36 attribute and 6 classes (red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil) and glass database contains 214 instances, 10 attribute and 7 classes.

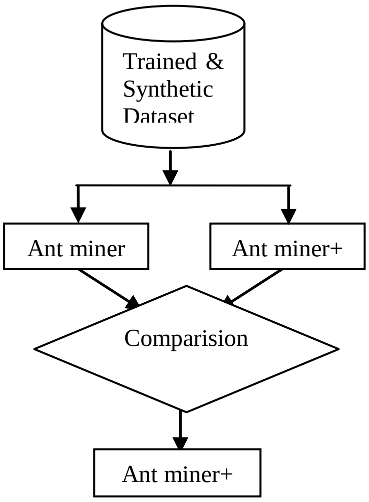

Figure 3: Overview

Real data for soil has been collected and preprocessed using the data mining approach and the data are converted into CSV format for further processing. Real soil dataset contains 125 instances, 2 attributes and 4 classes. The figure 3 shows the overview of the proposed study.

Ant miner and Ant miner+ algorithm has been implemented using MATLAB R2010a. Parameters for ant is set as 100, 1000 and accuracy, execution time were calculated for the two algorithms for 10 iteration. Accuracy and execution time for two algorithms applied to different data set are shown in Table1.

Table 1. Accuracy and Execution time for two algorithms

Algorithm	Data set	100 Ants		1000 Ants	
		Accuracy	Execution Time	Accuracy	Execution Time
Ant miner	Irish	47.4073	0.0513	47.8383	0.0526
	Soil	68.1603	0.3912	68.1603	0.4102
	Wine	42.4348	0.0632	43.5656	0.0673
	Glass	40.0830	0.0933	40.0830	0.1354
Ant miner+	Irish	56.2067	0.0307	56.2067	0.0317
	Soil	69.7457	0.3993	69.7458	0.4120
	Wine	82.0493	0.0511	82.0493	0.0531
	Glass	76.4157	0.0700	76.4157	0.0787

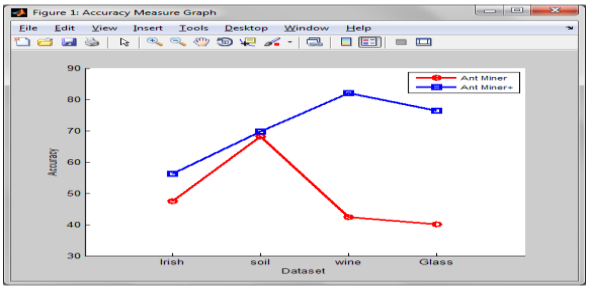

Figure 4: Accuracy for two algorithms on 4 Training Dataset

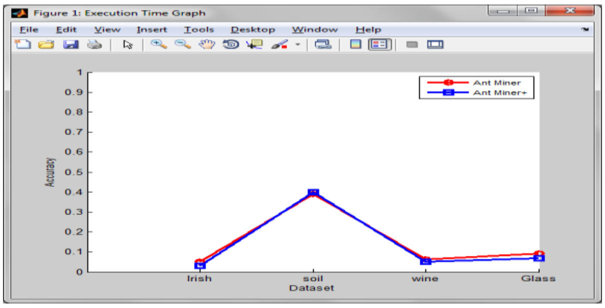

Figure 5: Execution Time for two algorithms on 4 Training Dataset

The Figure 4 and 5 represents Accuracy graph and Execution Time graph for Ant miner and Ant miner+ algorithm on Irish, Soil, Wine and Glass datasets.

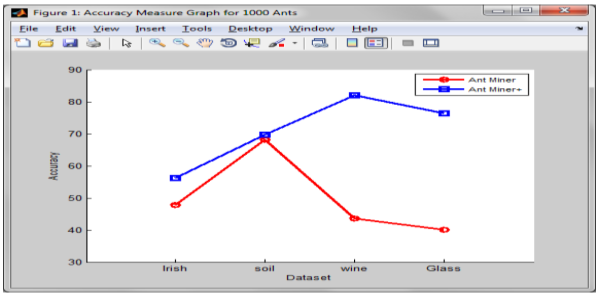

Figure 6: Accuracy graph for 1000 Ants on 4 Training Dataset

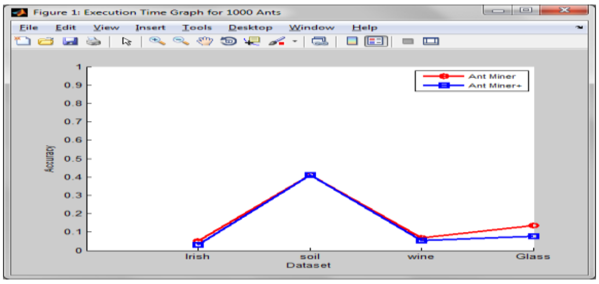

Figure 7: Execution Time graph for 1000 Ants on 4 Training Dataset

The Figure 6and 7 show the Accuracy and Execution Time of the algorithms with 1000 Ants. Both the algorithms were tested using various training datasets (Irish, Soil, Wine and Glass). Increasing of ants slightly improves the accuracy and execution time. Additionally real soil dataset have been implemented for both the algorithms are shown in Table 2.

Table 2: Accuracy and Execution time for two algorithms for real dataset

Algorithm	Data set	100 Ants		1000 Ants	
		Accuracy	**Execution Time**	**Accuracy**	**Execution Time**
Ant miner	Real Soil	40.7718	0.0600	40.7098	0.0738
Ant miner +	Real Soil	87.0115	0.0228	87.0115	0.0195

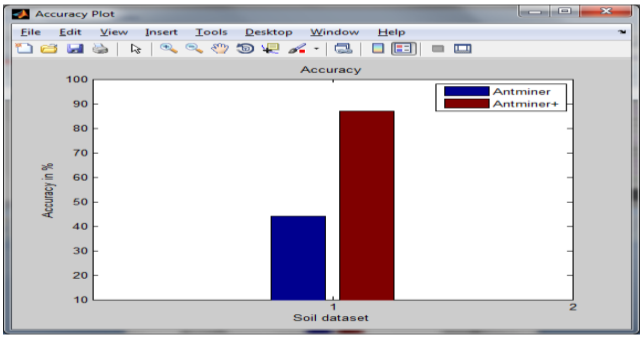

Figure 8: Accuracy for the real soil dataset

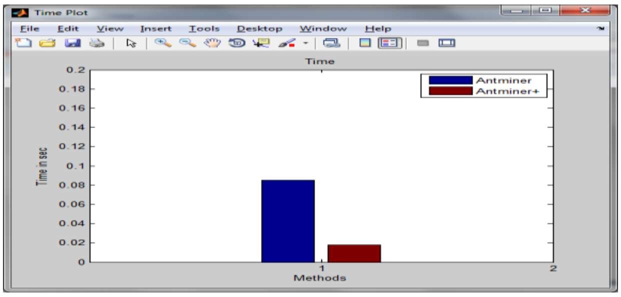

Figure 9: Execution Time for the real soil dataset

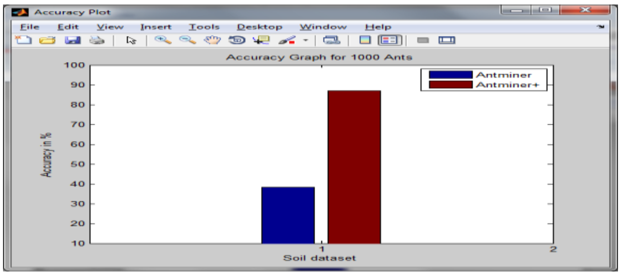

Figure 10: Accuracy for the real soil dataset-1000 Ants

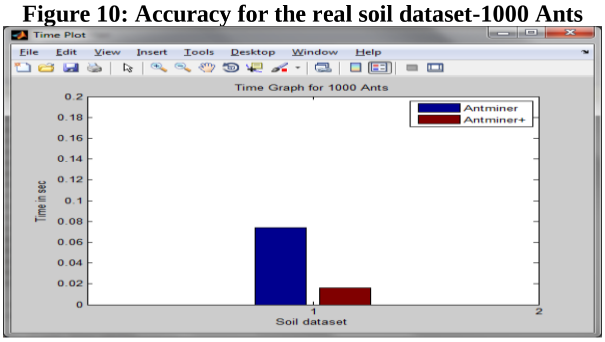

Figure 11: Execution Time for the real soil dataset-1000 Ants

The Figure 8 and 9 represents Accuracy graph and Execution Time graph for Ant miner and Ant miner+ for real soil dataset. The figure 10 and 11 represents Accuracy graph and Execution Time graph for 1000 Ants.

8. CONCLUSION AND FUTURE RESEARCH

In this research, Ant miner and Ant miner+ algorithms were experimented with different training datasets. The same have been applied for real dataset which were rendezvous in and around the erode district and preprocessed for this proposed work.

This proposed research classifies the real soil data set that helps to know the best soil based on the Potential of Hydrogen (PH) and the Electrical conductivity (EC) for fertilization and to improve the quality of the soil.

The experimental value shows that Ant miner+ performs better than Ant miner based on accuracy and execution time. When number of ants has been increased from 100 to 1000 the accuracy has been improved slightly so small number of ants yields better result. In future, the proposed work may be combined with Neural and Genetic algorithms and can also be applied for the unsupervised learning.

9. REFERENCES

[1] Adel Ardalan, Prof. Rahgozar, "Ant Miner: Ant Colony - Based Association Rule Miner", spring 2006.

[2] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. A. K. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art Classification algorithms for credit scoring", J. Oper. Res. Soc., vol.54, no. 6, pp. 627–635, 2003.

[3] Bonabeau, E., Dorigo, M., & Theraulaz, G., "Swarm Intelligence: From Natural to Artificial Systems", New York: Oxford University Press, 1999.

[4] Bo Liu, Hussein A. Abbass and Bob McKay, "Classification Rule Discovery with Ant Colony Optimization", IEEE Computational Intelligence Bulletin, February 2004, vol.3 No.1.

[5] David Martens, Manu De Backer, Raf Haesen, Student Member, IEEE, Jan Vanthienen, Monique Snoeck, and Bart Baesens, "Classification with Ant Colony Optimization", IEEE transactions on evolutionary computation, vol. 11, no. 5, October 2007.

[6] D. J. Hand and S. D. Jacka, Discrimination and classification, New York: Wiley, 1981.

[7] Jiawei Han and Micheline Kamber, "Data Mining Concepts and Techniques", Edition 2006 and ISBN: 978-1-55860-901-3.

[8] Loannis Michelakos, Nikolaos Mallios, Elpiniki Papageorgiou and Michael Vassilakopoulos, "Ant Colony Optimization and Data Mining: Techniques and Trends", 2010

[9] Marco Dorigo and Thomas Stutzle, "Ant Colony Optimization", Edition 2004 and ISBN-81-203-2684-9.

[10] Michael Goebel and Le Gruenwald, "A Survey of Data Mining and Knowledge Discovery Software Tools", June 1999

[11] Parepinelli, R. S., Lopes, H. S., & Freitas, A., "Data Mining with an Ant Colony Optimization Algorithm", IEEE transactions on evolutionary computation, vol. 6, no. 4, August 2002.

[12] Parepinelli, R. S., Lopes, H. S., & Freitas, A., "Data Mining with an Ant Colony Optimization Algorithm".

[13] Parepinelli, R. S., Lopes, H. S., & Freitas, A. (2002), "An Ant Colony Algorithm for Classification Rule Discovery".

[14] Parepinelli, R. S., Lopes, H. S., & Freitas, A, "An Ant Colony Based System for Data Mining: Applications to Medical Data".