Data Missing Solution Using Rough Set Theory and Swarm Intelligence

Dr. Ahmed Tariq Sadiq
Computer Science Department
University of Technology
Baghdad, Iraq
drahmaed_tark@yahoo.com

Dr. Mehdi Gzar Duaimi[1], Samir Adil Shaker[2]
Computer Science Department
Baghdad University
Baghdad, Iraq
[1]mehdi_duaimi@scbaghdad.edu.iq
[2]samir.adel2000@yahoo.com

***Abstract*— This paper presents a hybrid approach for solving null values problem; it hybridizes rough set theory with intelligent swarm algorithm. The proposed approach is a supervised learning model. A large set of complete data called learning data is used to find the decision rule sets that then have been used in solving the incomplete data problem. The intelligent swarm algorithm is used for feature selection which represents bees algorithm as heuristic search algorithm combined with rough set theory as evaluation function. Also another feature selection algorithm called ID3 is presented, it works as statistical algorithm instead of intelligent algorithm. A comparison between those two approaches is made in their performance for null values estimation through working with rough set theory. The results obtained from most code sets show that Bees algorithm better than ID3 in decreasing the number of extracted rules without affecting the accuracy and increasing the accuracy ratio of null values estimation, especially when the number of null values is increasing.**

Keywords-component; Null Values Problem; Rough Set; Bees algorithm; ID3; Incomplete Databases.

I. Introduction

Most of the real world databases are characterized by an unavoidable problem of incompleteness, in terms of missing or erroneous values. A variety of different reasons result in introduction of incompleteness in the data. Examples include manual data entry procedures, incorrect measurements, equipment errors, and many others. Existence of errors, and in particular missing values, makes it often difficult to generate useful knowledge from data, since many of data analysis algorithms can work only with complete data [1].

Many methods has been used to solve incompleteness or data missing problem, case deletion, prediction rules, substitution, hot/cold deck imputation, linear regression, decision trees and multiple imputations [2].

The approach in this paper composed of two main parts; learning and testing. The input to learning stage is a set of complete data and a decision attribute, while the output is a set of decision rules leads to the given decision in the right side of the rule. The rule extraction methodology used, depends on rough set theory in writing decision rules, certain rules with confidence equals to one can be corresponded from rough set lower approximations, and possible rules with confidence greater than zero can be found from the boundary regions of rough sets.

The bees algorithm is a population-based search algorithm inspired by the natural foraging behavior of honey bees [3].

The proposed approach uses bees algorithm as intelligent swarm algorithm for attribute selection depending on evaluation function provided by rough set theory. Also, the paper presents ID3 decision tree induction algorithm as statistical algorithm for feature selection instead the intelligent algorithm to make comparison between them in performance through working with rough set theory.

The ID3 decision tree induction algorithm uses a selection function in selecting the root property or attribute of the tree or the current sub-tree. The selected attribute will be considered as the left hand side conditional attribute of the generated rules. Rules will be written using rough set theory and then another attribute using the evaluation function in Bees algorithm or the selection function of ID3 will be selected till the set of the selected attributes represents a reduct.

Reduct is the minimal subset of attributes that enables the same classification as the whole set of attributes. So, reduct has been used as a stopping condition for the proposed approach, in other word stopping the selection and writing rules operations while the set of left hand side attribute represents a reduct.

Section two of this paper shows the related works and section three explains rough set theory, while ID3 and bees algorithm are explained in section four. Section five presents the proposed approach and finally section six and seven shows the Experimental results and conclusions.

II. Related Works

Sadiq, Chawishly & Sulaka [4], presented a hybrid approach for solving null values problem, it hybridized rough set theory with Iterative Dichotomiser 3 (ID3) decision tree induction algorithm, Large set of complete

978-0-7695-4959-0/13 $25.00 © 2013 IEEE
DOI 10.1109/ACSAT.2012.29

data called learning data is used to find the decision rule sets that have been used in solving the incomplete data. Chiu, Wei & Lee [5], proposed a Compounded Particle swarm Optimization (CPSO) clustering approach for the missing value estimation. Normalization methods were first utilized to filter outliers and to prevent some attributes from dominating the clustering result. Then the K-means algorithm and reflex mechanism were combined with the standard Particle Swarm Optimization (PSO) clustering so that it can quickly converge to a reasonable good solution. An iteration-based filling-in value scheme was utilized to guide the searching of CPSO clustering for the optimal estimate values. Marwala [2], proposed a method that combines the use of multi-layer auto associative neural networks with genetic algorithms to approximate missing data. A genetic algorithm was used to estimate the missing values by optimizing an objective function. Also Marwala [2], presented a data imputation method, based on rough set computations. Characteristic relations were introduced that describe incompletely specified decision tables and then these were used for missing data estimation. It has been shown that the basic rough set idea of lower and upper approximations for incompletely specified decision tables may be defined in a variety of different ways. Gómez, Bello, Puris & García [6], proposed a new approach to Swarm Intelligence called Two-Step Swarm Intelligence. They Presented Ant Colony Optimization and Particle Swarm Optimization as searching algorithms combined with evaluation function based on rough set theory for feature selection problem. The feature selection is based on the reduct concept of the Rough Set Theory. Chen & Hsiao [7], presented a new method for estimating null values in relational database systems based on automatic clustering techniques. Chen & Huang [8], presented a new method to generate weighted fuzzy rules from relational database systems for estimating null values using genetic algorithms (GAs), where the attributes appearing in the antecedent part of the generated fuzzy rules have different weights. Chen & Yah [9], presented a fuzzy concept learning system (FCLS) algorithm to construct fuzzy decision trees from relational database systems and to generate fuzzy rules from the constructed fuzzy decision trees. Based on the generated fuzzy rules, they presented a method for estimating null values in the relational database system.

III. ROUGH SETS THEORY

Rough set theory was developed by Zdzislaw Pawlak, in the early 1980's. It deals with the classificatory analysis of data tables. The data can be acquired from measurements or from human experts. The main goal of the rough set analysis is to synthesize approximation of concepts from the acquired data [10].

The rough set approach seems to be of fundamental importance to artificial intelligent and cognitive sciences, especially in the areas of machine learning, knowledge acquisition, decision analysis, knowledge discovery from databases, expert systems, inductive reasoning and pattern recognition [11]. Rough sets can be used separately but usually they are used together with other methods such as fuzzy sets, statistic methods, genetic algorithm etc. The rough sets theory uses different approach to uncertainty. As well as fuzzy sets this theory is only part of the classic theory, not an alternative [12].

A. *Theory Fundamentals*

A data set is represented as a table, where each row represents an object. Every column represents an attribute that can be measured for each object; this table is called an information system. More formally, Let A = (U, A), where U is a non-empty finite set of objects called the universe and A is a non-empty finite set of attributes then with any $B \subseteq A$ there is associated an equivalence relation $IND_A(B)$:

$$IND_A(B)= f(x, x') \in U^2 \,|\, \forall a \in B \; a(x) = a(x') \tag{1}$$

$IND_A(B)$ is called the B-indiscernibility relation. If $(x, x') \in IND_A(B)$, then objects x and x' are indiscernible from each other by attributes from B. The equivalence classes of the B-indiscernibility relation are denoted [x]B [10].
Now approximations can be defined as follows:

$$B_*(X) = \{x \in [x]B : [x]B \subseteq X\} \tag{2}$$

$$B^*(X) = \{x \in [x]B : [x]B \cap X \neq Ø\} \tag{3}$$

Assigning to every subset X of the universe U two sets $B_*(X)$ and $B^*(X)$ called the B-lower and the B-upper approximation of X, respectively. The set

$$BN_B(X) = B^*(X) - B_*(X) \tag{4}$$

will be referred to as the B-boundary region of X. If the boundary region of X is the empty set, i.e., BNB(X) = ∅, then the set X is crisp (exact) with respect to B; in the opposite case, i.e., if BNB(X) ≠ ∅, the set X is rough (inexact) with respect to B [13].
Rough set can be also characterized numerically by the following coefficient

$$\alpha_B(X) = \frac{|B_*(X)|}{|B^*(X)|} \tag{5}$$

Called the accuracy of approximation, where |X| denotes the cardinality of X ≠ Ø. Obviously $0 \leq \alpha_B(X) \leq 1$. If $\alpha_B(X) = 1$, X is crisp with respect to B (X is precise with respect to B), and otherwise, if $\alpha_B(X) < 1$, X is rough with respect to B (X is vague with respect to B) [10].

B. Attributes Dependency

A set of attributes D depends totally on a set of attributes C, denoted C ⇒ D, if all values of attributes from D are uniquely determined by values of attributes from C.
Formally dependency can be defined in the following way. Let D and C be subsets of A. We will say that D depends on C in a degree k ($0 \leq k \leq 1$), denoted $C \Rightarrow_k D$, if

$$k = \quad (C, D) = \frac{|pos_c(D)|}{|U|} \tag{6}$$

Where

$$pos_c(D) = \bigcup_{x \in U/D} C*(X) \tag{7}$$

Called a positive region of the partition U/D with respect to C, is the set of all elements of U that can be uniquely classified to blocks of the partition U/D, by means of C:
If k = 1 we say that D depends totally on C, and if k < 1, we say that D depends partially (in a degree k) on C.
The coefficient k expresses the ratio of all elements of the universe, which can be properly classified to blocks of the partition U/D; employing attributes C and will be called the degree of the dependency [10].

C. Reduction of Attribute

Reduct is the minimal set of conditional attributes that observe the degree of dependency. In a different way we can say, that it is a subset of original conditional attributes that enables us to make the same decision [12].

$$\sigma(c,d)(B) = \frac{(\gamma(c,d) - \gamma(c-B,D))}{\gamma(c,d)} = 1 - \frac{\gamma(c-B,D))}{\gamma(c,d)} \tag{8}$$

Denoted by (B), if C and D are understood, where B is a subset of C:
If B is a reduct of C, then σ (C - B) = 0, i.e., removing any reduct complement from the set of conditional attributes enables to make decisions with certainty, whatsoever [10].

D. Decision Rules

Let I = (U , A ∪ {d}) be a decision system. Any implication of the form: [14].

$$(a_{i1} = v_1) \wedge \ldots \wedge (a_{im} = v_m) \Rightarrow (d = k) \tag{9}$$

Where, $a_{ij} \in$ A and $v_j \in Va_{ij}$, is called a decision rule for the k–th decision class. Any decision rule **r** of the above form can be characterized by the following parameters: [14]

- Length (**r**)= the number of descriptors in the premise of **r**;
- [**r**] = carrier of **r**, i.e., the set of objects satisfying the premise of **r**;
- Support (**r**)= number of objects satisfying the premise of **r**;
- Confidence (**r**) = the measure of truth of the decision rule,

Confidence (**r**) = card ([**r**] ∩ $CLASS_k$) / card ([**r**]) . . . (10)

The lower approximation corresponds to certain rules while the upper approximation to possible rules (rules with confidence greater than 0) for X. The B-lower approximation of X is the set of all objects, which can be with certainty classified to X using attributes from B. The set $U - B^*X$ is called the B-outside region of X and consists of those objects, which can be with certainty classified as not belonging to X using attributes from B. The set $BN_B(X) = B^*X - B_*X$ is called the B-boundary region of X thus consisting of those objects that on the basis of the attributes from B cannot be unambiguously classified into X [15].
In data mining, we are interested in searching for short, strong decision rules with high confidence [14].

IV. TWO DIFFERENT ALGORITHMS WORK WITH ROUGH SET THEORY

A. Bees algorithm

The bees algorithm is a population-based search algorithm inspired by the natural foraging behavior of honey bees first developed in 2005. In its basic version, the algorithm starts by scout bees being placed randomly in the search space. Then the fitnesses of the sites visited by the scout bees are evaluated and Bees that have the highest fitnesses are chosen as "selected bees" and sites visited by them are chosen for neighborhood search. Then, the algorithm conducts searches in the neighborhood of the selected sites, assigning more bees to search near to the best e sites. Searches in the neighborhood of the best e sites are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. The remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts, those that were the fittest representatives from a patch and those that have been sent out randomly. The algorithm performs a kind of neighborhood search combined with random search and can be used for both combinatorial and functional optimization [3]. Figure 1 shows the pseudo code of basic bees algorithm [16].

Bees algorithm has found many applications in engineering including training neural networks for pattern recognition, forming manufacturing cells, scheduling jobs for a production machine, finding multiple feasible solutions to a preliminary design problem, data clustering, optimising the design of mechanical components, sequencing flow shops, and tuning fuzzy control systems [17].

B. The ID3 Decision Tree Induction Algorithm

Interactive Dichotomizer 3 or ID3, uses a basic tree induction algorithm that assigns an attribute to a tree node based on how much information is gained from that node [18]. ID3 algorithm searches through the attributes of a dataset for the one that conveys the most information about the desired target. If the attribute classifies the target perfectly, then it will stop. Otherwise, the process is repeated recursively on the two or more subsets generated by setting this "best" attribute to their states [19].
Given a universe of messages, M={m1,m2, … ,mn} and a probability, p(mi), for the occurrence of each message, the expected information content of a message M is given by:

$$I[M] = \sum_{i=1}^{n} -p(m_i)\log_2(p(m_i)) \tag{11}$$

Assume a set of training instances, C. if we select property P, with n values, this will partition C into subsets, {C1, C2, . . . , Cn}. The expected information needed to complete the classification after selecting P:

$$E[p] = \sum_{i=1}^{n} (|C_i|/|C|) * I[C_i] \tag{12}$$

The gain from property P is computed by subtracting the expected information to complete the classification from the total information content:

$$\text{Gain}(P) = I[M] - E[P] \tag{13}$$

ID3 decision tree induction algorithm picks the property that provides the greatest information gain as the root of the current sub-tree [20].

V. PROPOSED APPROACH

The proposed approach hybridizes rough set theory with intelligent swarm algorithm for solving null values problem. This approach is a supervised learning model and large set of complete data called learning data is used to find the decision rule sets that have been used in solving the incomplete data problem. The intelligent swarm algorithm used for feature selection and represents bees algorithm as heuristic search algorithm combined with rough set theory as evaluation function. Also the paper uses id3 algorithm that works as statistical algorithm instead of intelligent algorithm and made comparison between them in their performance for null values estimation through working with rough set theory.
The proposed approach composed of many stages; coding, attributes selection, applying rough set concepts, rule generating algorithms and finally the estimation process, figure 2 show the framework of components of the proposed approach.
The next sections contain the definition of each step of the proposed approach.

A. Coding

Coding is the first stage of the system that applies on a completed learning database, from which the rules will be extracted. In this stage the data set is converted from its real values into codes equivalent to specified ranges, which its values belong to. We made four sets of coding which were different by the number of classes and ranges to show their effect on the system.

B. Feature selection

In feature Selection Function each attribute will considered as decision and the other attributes as conditions. When an attribute is selected as a decision, the other attributes enters the selection function. Two different feature selection algorithms were used, they are discussed bellow.

1) Intelligent algorithm

The proposed approach uses bees algorithm as heuristic search algorithm combined with evaluation function provided by rough set theory in discovering the most effective attribute in classifying the universe of learning data to the given decision by selecting the attribute that provide highest dependency about the decision. The dependency function of rough set theory computes the importance of each condition attribute by calculate its dependency with the decision attribute, the best attribute which gives highest dependency ratio.

2) Statistical algorithm

In order to measure the performance of intelligent algorithm, we used ID3 decision tree induction algorithm as statistical algorithm for feature selection instead the intelligent algorithm. The ID3 decision tree induction algorithm uses a selection function in selecting the root property or attribute of the tree or the current sub-tree. This selection function think of each attribute of an instance as contributing a certain amount of information to its classification. ID3 measures the information gained by making each attribute the root of the current sub-tree, Then It picks the attribute that provides the greatest information gain.

the selected attribute will be considered as the left hand side conditional attribute of the generated rules, rough set approximations are found in the next stage and rules are written in that level. Then, the next attribute that provides the highest dependency in bees algorithm or greatest information gain in ID3 algorithm after adding it to the previous selected attribute, will be selected and added to the set of conditional attributes and another set of decision rules will be obtained, and so on.

C. Apply rough set theory

Applying rough set theory is the next stage in the system that find the lower and upper approximation sets necessary to write the decision rules. The main operations of this stage are:

1) Rule generation

Rule generation is the core of the system. The rules in the algorithm are extracted from the lower approximations (rules of confidence equal to one) and the boundary regions (rules of confidence less than one). The extracted rules from rule generation are saved in a decision rules sets to be used in the estimation process. Eight decision rule sets have been written while using four different sets of coding in the first stage of the approach for each feature selection method. The rule generation continues selecting a left hand side attributes using one of feature selection algorithms and writing decision rules using rough set approximations till the set of the conditional attributes on the left hand side represents a reduct for the current decision.

2) Reduction

Reduct is the minimal subset of attributes that enables the same classification of elements of the universe as the whole set of attributes. In other words, attributes that do not belong to a reduct are superfluous with regard to classification of elements of the universe. So, reduct has been used in the proposed approach in writing a stopping condition to prevent adding the unuseful attributes to the set of left hand side conditional attributes of the rules. Reduct is happen when the ratio of calculating the dependency of the selected conditional attributes with the decision to the dependency of all conditional attributes with the same decision equal one. The main goal of applying this reduction on the system is to make the extracted rule sets minimal as possible without decreasing the accuracy of the estimated values founded by testing the rules.

D. Estimation

The rules of the decision rule sets are applied on the testing data which is the set of data kept for testing and evaluating the proposed system by changing some observed values with null values, estimating these values using the extracted decision rules, and finally comparing the estimated values with the observed values. The result from the operations above is a set of complete data without missingness.

Figure 3 shows an algorithm that illustrates the main behavior of the training process using bees algorithm and rough set theory.

VI. Experimental results

A weather information database has been chosen to apply the proposed approach on, the information needed to built such a database have been taken from general directorate of meteorology and seismology/Baghdad as a set of Microsoft Excel sheets containing information about Baghdad city weather since january, 1990 till 2011 except (2001, 2002, 2003 and 2004) because they are missing from the source.

In order to separate the data used in learning level from this used in testing, The data from 1991 to 2010 have been considered as training data in learning process after converting it to Microsoft Access database, while the information of 1990 and 2011 have been used in testing level.

The database consist of a table of 192 tuples and thirteen attributes named real info, the attributes are; year, month, dust_storm for dust storm, suspend_dust for suspend dust, rising_dust for rising dust, rain, sun_shine for sun shine, dry_temp for dry temperature, max_temp for maximum temperature, min_temp for minimum temperature, humidity, evaporation and pressure.

Visual basic.net is the programming language used in implementing the proposed approach. The system interface seems like an ordinary form used for entering weather information data, but this form differs by its ability of distinguishing the missing fields and solving them, even if there are multiple missing fields. Figure 4 shows system interface.

When the user fills out the text fields by data and clicks on (save or edit) data button, the system starts checking all the filled fields one by one, if there is no problem in the data a new tuple will be (entered or update if exist), else, if any noisy or unacceptable data are entered, a message will appear to the user showing the number of missing or unacceptable values and asking for estimating these fields now or cancel estimation at this time to check them by the user. The operation of estimating the values of missing fields gives the user the option of selecting the used algorithm in the estimating process.

In order to assess the system and compare the two algorithms, three parameters are calculated; rule set size, rule set complexity and Accuracy of estimation.

Before examining these parameters, it’s worth indicating that some of decision attributes are not depending on available condition attributes in a sufficiently manner, in other words, some attributes are good as condition attributes but aren’t good as decision attributes, therefore, we used dependency function provided by rough set theory to select the best six decision attributes and the result show that (dry temperature, max temperature, min temperature, humidity, evaporation and pressure) are the best of all attributes. Also we used 100 nectar positions as an input in bees algorithm.

Rule set size is the number of rules belonging to the rule set. As an evaluation made using rule set size parameter, figure 5 shows the overall rule sets size generated by each algorithm using the best set of coding.

Rule set complexity is the number of conditional attributes on the left-hand side of the rules (rule length). The overall complexity of the rule sets generated by each algorithm for the decisions is calculated. Figure 6 represents the complexities of the rules generated by each algorithm using the best set of coding.

Finally the accuracy of estimation is the ratio of nulls (missing fields) that successfully estimated by the rules to the all missing values. This parameter is calculated by; observing the values of a number of fields in the dataset specialized for testing, making these fields of data missed randomly, calculating the values estimated for them by the system, comparing the real data observed with the estimated data and finally calculating the ratio of the values correctly estimated (accuracy of estimation). This test has been done for 24 records of testing data in six levels using the best set of coding (see figure 7):

1) *Calculating the accuracy of estimation when a single null found randomly within each record of testing data: In ID3 the 91.6% of the tested null values were approximately estimated and 87.5% of them were absolutely true estimated. Bees algorithm increased those ratios by making 91.6% of testing null values as absolutely true estimated, while 95.9% of null values were approximately estimated.*

2) *Calculating the accuracy of estimation when two nulls found randomly within each record of testing data: In this case 77.08% of the null values were absolutely true estimated by ID3, and 91.6% of the tested null values were approximately estimated, while 92.7% of tested values were approximately estimated by Bees and 78.08% of them were absolutely true estimated.*

3) *Calculating the accuracy of estimation when three nulls found randomly within each record of testing data: In this case 87.8% of null values were approximately estimated by ID3 and 72.22% of them were absolutely true estimated, in other case bees algorithm also increases the absolutely true estimation to 73.61% and the approximate estimation were 84.8% of the tested null values.*

4) *Calculating the the accuracy of estimation when four nulls found randomly within each record of testing data: By using ID3, 78.2% of null values were approximately estimated and 67.7% of them were absolutely true estimated, in other case bees algorithm made 70.83% of null values as absolutely true estimated while 82.3% of null values were approximately estimated.*

5) *Calculating the accuracy of estimation when five nulls found randomly within each record of testing data: In this case 66.66% of the null values were absolutely true estimated by ID3, and 75.9% of the tested null values were approximately estimated, while 78.4% of nulls values were approximately estimated by Bees and 69.16% of them were absolutely true estimated.*

6) *Finally, calculating the accuracy of estimation when six nulls found randomly within each record of testing data: By using ID3 the 73.7% of tested null values were approximately estimated and 63.19% of them were absolutely true estimated, in other case bees algorithm made 65.27% of testing null values as absolutely true estimated while 75.7% of null values were approximately estimated.*

VII. CONCLUSIONS

The rules, resulted from the system were evaluated and the following points are concluded:

1) *Classifying data with large number of features during the training process is time consuming task that doesn't produce consistent results.*

2) *Bees algorithm reduces the number of selecting features (reduce the complexity) by selecting only the most relevant to separating the different groups in a data collection.*

3) *Bees algorithm was better than ID3 in most of coding sets by restricting the total number of rules without affecting the accuracy, in the best code set, Bees algorithm decreased the rule sets size from 4816 to 4808.*

4) *Rule set complexity or the number of conditional features on the left-hand side of the rules is the same for both algorithms using the best code set.*

5) *In the case of one null value occurrence within each record of testing data using the best code set, the proposed system that used bees algorithm had a higher performance than ID3 by estimating 95.9% of the null values as an approximate estimation and 91.6% of them as a true estimation. while ID3 algorithm decreased those ratios to 91.6% and 87.5% respectively .*

6) *By increasing the number of null values occurred within each record of testing data, the accuracy rate of estimation is decreased. In the case of two null values occurrence within each record of testing data using the best code set, the proposed system also had a higher performance than ID3. It made 92.7% of the null values as approximately estimated while 78.08% of them as absolutely true estimated. ID3 algorithm decreased those ratios to 91.6% and 77.08% respectively.*

7) *When the number of null values occurred within each record of testing data increased to three using the best code set, the proposed system also had a higher performance than ID3 by estimating 84.8% of the null values as an approximate estimation and 73.61% of them as*

a true estimation while ID3 algorithm made those ratios 87.8% and 72.22% respectively.

8) All coding sets show the bees algorithm is better than ID3 in accuracy of null value estimation when the number of null values is increasing in each record of the table.

9) The results from most of coding sets show that bees algorithm is better than ID3 algorithm in accuracy of estimation for all number of null values.

10) Bees algorithm more efficient than Id3 through working with rough set in extracting minimal rule sets because it decreases the rule sets size. Also, it increases the Accuracy of estimation.

REFERENCES

[1] A. Farhangfar, L. Kurgan, & W. Pedrycz, “Experimental analysis of methods for imputation of missing values in databases”, University of Alberta, Electrical and Computer Engineering Department, Edmonton, AB, Canada, T6G 2V4, 1999.

[2] Tshilidzi, Marwala. “Computational Intelligence for Missing Data Imputation, Estimation, and Management: Knowledge Optimization Techniques”, Information Science Reference, IGI Global, 2009.

[3] F. Aghazadeh, & M. R. Meybodi, ”Learning Bees Algorithm For optimization”, International Conference on Information and Intelligent Computing, IPCSIT vol. 18, IACSIT Press, Singapore, 2011.

[4] A. T. Sadiq, S. A. Chawishly, & N. J. Sulaka, ”Solving Null Values Problem Using Intelligent Methods”, Journal of Advanced Computer Science and Technology Research Vol.2, No.2, Jun. 2012, pp. 91-103.

[5] H. Chiu, T. Wei, & H. Lee,”A Novel Approach for Missing Data Processing based on Compounded PSO Clustering”, WSEAS Transactions on Information Science and Applications, Vol. 6 ,No. 4, Apr. 2009.

[6] Y. Gómez, R. Bello, A. Puris, & M. M. García, “Two Step Swarm Intelligence to Solve the Feature Selection Problem”, Journal of Universal Computer Science, Vol. 14, No. 15, 2008, pp. 2582-2596.

[7] S. Chen, & H. Hsiao, “A New Method to Estimate Null Values in Relational Database Systems Based On Automatic Clustering Techniques”, Information Sciences: an international journal vol. 169, Elsevier, 2005, pp. 47–69.

[8] S. Chen, & C. Huang, “Generating Weighted Fuzzy Rules From Relational Database Systems for Estimating Null Values Using Genetic Algorithms”, IEEE Transactions on Fuzzy Systems, Vol. 11, No. 4, Aug. 2003.

[9] S. Chen, & M. Yeh, “Generating Fuzzy Rules From Relational Database Systems For Estimating Null Values”, Cybernetics and Systems: An International Journal, vol. 28, Taylor & Francis, 1997, pp. 695-723.

[10] J. Komorowski[1], L. Polkowski1, & A. Skowron[2], “Rough sets: A tutorial”, [(1)]Polish-Japanese Institute of Information Technology Koszykowa 86, 02-008 Warszawa, Poland, [(2)]Institute of Mathematics, Warsaw University of Technology Pl. Politechniki 1, 00-665 Warszawa, Poland, 2006.

[11] Pawlak, Zdzisław. “Rough Sets”, Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, ul. Bałtycka 5, 44 100 Gliwice, Poland, 2004.

[12] P. Jurka, & F. Zbořil, “Using Rough Sets in Data Mining”, Doctoral Degree Programme, Dept. of Intelligent Systems, FIT, BUT, 2005.

[13] Z. Pawlak, “Rough sets Theoretical aspects of reasoning about data”. Dordrecht Kluwer Academic Publishers, 1991.

[14] J. Bazan, N. H. Son, A. Skowron, & M. Szczuka, “A View on Rough Set Concept Approximations”, international conference on rough sets, fuzzy sets, data mining and granular computing (RSFDGrC), Springer-Verlag Berlin Heidelberg 2003, pp. 181-188.

[15] H. K. Tripathy, B. K. Tripathy, & P. K. Das, “An Intelligent Approach of Rough Set in Knowledge Discovery Databases ”, International Journal of Electrical and Electronics Engineering ,Vol. 2, No. 5, 2008, pp. 334-337.

[16] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, & M. Zaidi, "The Bees Algorithm–A Novel Tool for Complex Optimization Problems", In Proceedings of the Intelligent Production Machines and Systems (IPROMS), Conference, 2006, pp. 454-461.

[17] D.T. Pham, & E. Koc, ” Design of a Two-dimensional Recursive Filter Using the Bees Algorithm”, International Journal of Automation and Computing Vol. 7, No. 3, Aug. 2010, pp. 399-402.

[18] M. W. BERRY, & M. BROWNE, ” LECTURE NOTES IN DATA MINING”, World Scientific Publishing Co. Pte. Ltd, 2006.

[19] A. Colin, "Building Decision Trees with the ID3 Algorithm", Dr. Dobb's Journal,Jun.1,1996, URL: www.ddj.com/architect/184409907.

[20] G. F. Luger, “Artificial Intelligent, Structures and Strategies for Complex Problem Solving”, Pearson Education, Fourth Edition, 2002.

1. Initialise population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //Forming new population.
4. Select sites for neighbourhood search.
5. Recruit bees for selected sites (more bees for best e sites) and evaluate fitnesses.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitnesses.
8. End While.

Fig.1: Pseudo code of the basic bees algorithm

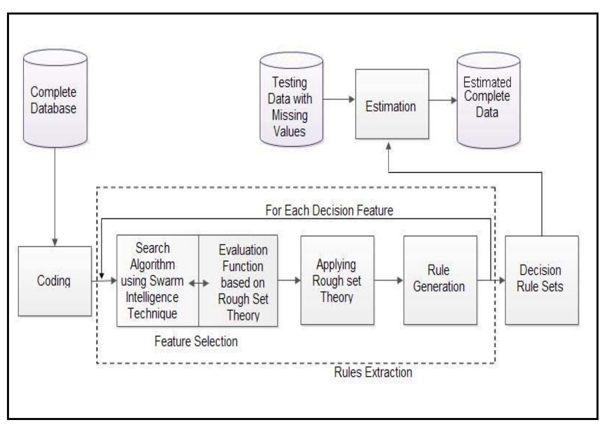

Fig.2: Framework of the Data missing Solution Approach

Input: (decision features list, complete information system, number of bees (codition features), number of nectar positions)
Output: rule sets for estimating missing values

Begin
- For each decision feature in decision features list do
- Initially send randomly one bee only to each nectar position
- Repeat
- Compute fitness for each nectar position using rough set function and select the best one.
- Add the bee of selected position that got highest amount of nectar to the best bees list (feature selection list).
- Send all of the best bees from the selected nectar position to all other positions (neighborhood search).
- Compute fitness for each nectar position using rough set function and select the best one
- Retrieve from all positions only the bees that get less amounts of nectar than the better one in selected nectar position to search randomly in the next step for another bee that can get largest amount of nectar.
- Send one bee (random search) to all positions that (contain number of bees as the same as the number of best bees in the best bees list or less) and (the amount of their nectar that gotten by number of available bees must not equal to the amount that get by all bees .
- Until the amount of nectar that gotten by the best bees in best bees list equal to the amount that gotten by all bees(reduct state).
- Initialize the set of chosen attributes B.
- For i = 0 to length of selected features(bees) -1
- Add selected feature of i to B
  Find the equivalence classes of the d-indiscernibility relation denoted [x]d
- For each class X = [x]d
- find the rough set approximations
- $B_*(X) = \{x \in [x]B : [x]B \subseteq X\}$
- $B^*(X) = \{x \in [x]B : [x]B \cap X \neq \emptyset\}$
- And the boundary region $B_{NB}(X)=B^*(X)-B_*(X)$
- Write decision rule sets using the approximation sets
- end For
- end For
- end for

End

Fig.3: Bees algorithm with rough set theory

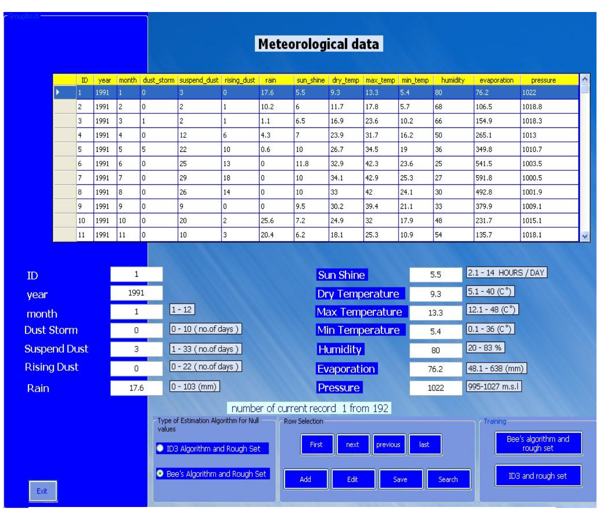

Fig.4: System Interface

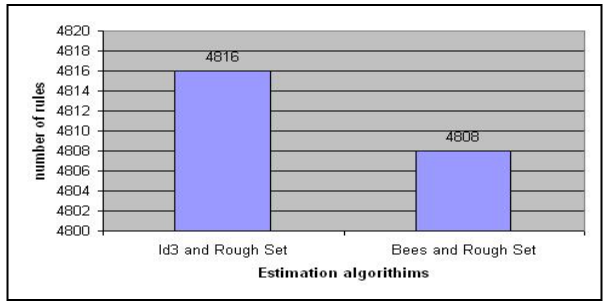

Fig.5: Rule Set Size

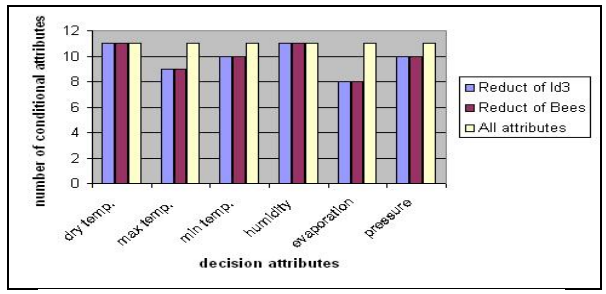

Fig.6: Rule Set complexity

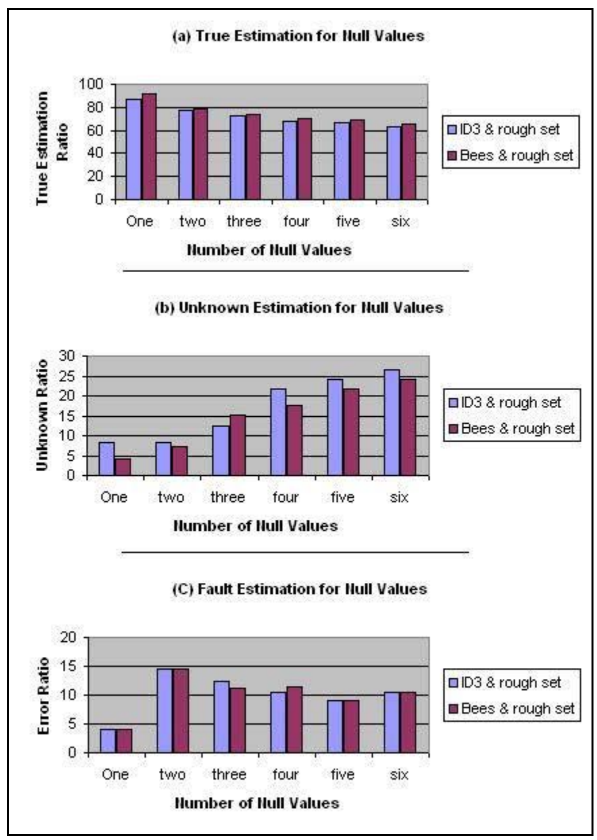

Fig.7: Accuracy of estimation