

## Predicting Missing Attribute Values Using k-Means Clustering

Nambiraj Suguna and Keppana Gowder Thanushkodi  
Department of Computer Science and Engineering,  
Akshaya College of Engineering and Technology,  
Coimbatore, Tamil Nadu, India

---

**Abstract: Problem statement:** Predicting the value for missing attributes is an important data preprocessing problem in data mining and knowledge discovery tasks. Several methods have been proposed to treat missing data and the one used more frequently is deleting instances containing at least one missing value of a feature. When the dataset has minimum number of missing attribute values then we can neglect the instances. But if it is high, deleting those instances may neglect the essential information. Some methods, such as assigning an average value to the missing attribute, assigning the most common values make good use of all the available data. However the assigned value may not come from the information which the data originally derived from, thus noise is brought to the data.

**Approach:** In this study, k-means clustering is proposed for predicting missing attribute values. The performance of the proposed approach is analyzed with nine different methods. The overall analysis shows that the k-means clustering can predict the missing attribute values better than other methods. After assigning the missing attributes, the feature selection is performed with Bees Colony Optimization (BCO) and the improved Genetic KNN is applied for finding the classification performance as discussed in our previous study. **Results:** The performance is analyzed with four different medical datasets; Dermatology, Cleveland Heart, Lung Cancer and Wisconsin. For all the datasets, the proposed k-means based missing attribute prediction achieves higher accuracy of 94.60 %, 90.45 %, 87.51 % and 95.70 % respectively. **Conclusion:** The greater classification accuracy shows the superior performance of the k-means based missing attribute value prediction.

**Key words:** Bees Colony Optimization (BCO), K-Nearest Neighbor (KNN), missing attributes, Most Common Attribute Value (MCAV), Event-Covering Method (EC), genetic algorithm, k-means clustering, clustering algorithm, onlooker bee, Artificial Bee Colony (ABC)

---

### INTRODUCTION

Missing attribute values are variables without observation or questions without answers. Even a small amount of data can cause serious problems may leading to wrong conclusions. There are several techniques to assign the values for missing items, but no one is absolutely better than the others. Different situations require different solutions; the only really good solution to the missing data problem is not to have any. Grzymala-Busse and Hu (2001) nine approaches on filling in the missing attribute values were introduced, such as selecting the most common attribute value, assigning all possible values of the attribute restricted to the given concept, ignoring examples with unknown attribute values, treating missing attribute values as special values, event-covering method and so on. Experiments on ten data sets were conducted to

compare the performances. Grzymala-Busse *et al.* (2001) a closest fit approach was proposed to compare the vectors of all the attribute pairs from a preterm birth data set and assign the value from the most similar pair to the missing value. In a more recent effort (Grzymala-Busse, 2005) four interpretations on the meanings of missing attribute values such as “lost” values and “do not care” values are discussed. Different approaches from rough sets theory are demonstrated on selecting values for the individual interpreted meanings.

Grzymala-Busse and Hu (2001) performed computational studies on the medical data, where unknown values of the attributes were replaced using probabilistic techniques. Recently, Greco *et al.* (1999) used a specific definition of the discernibility relation to analyze unknown attribute values for multicriteria decision problems. Stefanowski and Tsoukias (2001) presented two different semantics for incomplete

---

**Corresponding Author:** N. Suguna, Department of Computer Science and Engineering,  
Akshaya College of Engineering and Technology, Coimbatore, Tamil Nadu, India

information “missing values” and “absent values” were discussed also; they introduced two generalizations of the rough set theory to handle these situations. Nakata and Sakai (2005) the author examined methods of valued tolerance relations. They proposed a correctness criterion to the extension of the conventional methods which is based on rough sets for handling missing values.

In real-world data set, missing attribute values are very common. This may happen at the time of data collection, redundant diagnose tests, unknown data and so on. A common approach is that discarding all data containing the missing values can't fully preserve the characteristics of the original data. Before assigning the values for missing attributes, we must understand the background knowledge and its context will be helpful for finding the best approach for handling missing values. Several approaches on how to deal with the missing attribute values have been proposed in the ancient years. Here are the nine different approaches as discussed in (Grzymala-Busse and Hu, 2001).

**Most Common Attribute Value (MCAV):** It is one of the simplest methods to deal with missing attribute values. The value of the attribute that occurs most often is selected to be the value for all the unknown values of the attribute.

**Concept Most Common Attribute Value (CMCAV):** The most common attribute value method does not pay any attention to the relationship between attributes and a decision. The concept most common attribute value method is a restriction of the first method to the concept, i.e., to all examples with the same value of the decision as an example with missing attributes value. This time the value of the attribute, which occurs the most common within the concept is selected to be the value for all the unknown values of the attribute. This method is also called maximum relative frequency method, or maximum conditional probability method (given concept).

**C4.5:** This method is based on entropy and splitting the example with missing attributes values to all concepts.

**Method of Assigning All Possible Values of the Attribute (APV):** In this method, an example with a missing attribute value is replaced by a set of new examples, in which the missing attribute value is replaced by all possible values of the attribute. If we have some examples with more than one unknown attribute value, we will do our substitution for one attribute first and then do the substitution for the next

attribute until all unknown attribute values are replaced by new known attribute values.

**Method of Assigning All Possible Values of the Attribute Restricted to the Given Concept (APVRC):** The method of assigning all possible values of the attribute is not related with a concept. This method is a restriction of the method of assigning all possible values of the attribute to the concept, indicated by an example with a missing attribute value.

**Method of Ignoring Examples with Unknown Attribute Values (IGNORE):** This method is the simplest: just ignore the examples which have at least one unknown attribute value and then use the rest of the table as input to the successive learning process.

**Event-Covering Method (EC):** This method is also a probabilistic approach to fill in the unknown attribute values. By event-covering we mean covering or selecting a subset of statistically interdependent events in the outcome space of variable-pairs, disregarding whether or not the variables are statistically independent.

**A Special LEM2 Algorithm (LEM2):** A special version of LEM2 that works for unknown attribute values omits the examples with unknown attribute values when building the block for that attribute. Then, a set of rules is induced by using the original LEM2 method.

**Method of Treating Missing Attribute Values as Special Values (SPECIAL):** In this method, we deal with the unknown attribute values using a totally different approach: rather than trying to find some known attribute value as its value, we treat “unknown” itself as a new value for the attributes that contain missing values and treat it in the same way as other values.

In this study, k-means clustering is proposed for assigning missing attribute values. The core idea is based assigning all possible values. Each new value is assigned and the dataset is clustered using k-means. And the cluster is validated to check whether the instance having missing value is placed in correct cluster, if so, the assigned value is marked as permanent. Otherwise the next value will be assigned. If it doesn't fit with any possible value then the best fit value is assigned to that missing attribute. If an instance having more than one missing attributes values, then all the possible combinations are checked. Once the values has been assigned for all the missing attributes, then the

feature selection is performed with Bees Colony Optimization (BCO) as discussed in (Suguna and Thanushkodi, 2010a) and the improved Genetic KNN (Suguna and Thanushkodi, 2010b) is applied for finding the classification performance. The rest of the study is organized as: the following text describes the existing approaches to be compared, followed by the proposed k-means clustering approach for assigning values for missing attributes. The experiments are conducted on different datasets from medical domain and the results are presented and the study is concluded with the discussion about the performance of our proposed method.

## MATERIALS AND METHODS

**K-means clustering for missing attribute value prediction:** One of the most popular clustering techniques is the k-means clustering algorithm (Pavan *et al.*, 2010; Jaradat *et al.*, 2009). Starting from a random partitioning, the algorithm repeatedly (i) computes the current cluster centers (i.e. the average vector of each cluster in data space) and (ii) reassigns each data item to the cluster whose centre is closest to it. It terminates when no more reassignments take place. By this means, the intra-cluster variance, that is, the sum of squares of the differences between data items and their associated cluster centers is locally minimized. k-means' strength is its runtime, which is linear in the number of data elements and its ease of implementation. However, the algorithm tends to get stuck in suboptimal solutions (dependent on the initial partitioning and the data ordering) and it works well only for spherically shaped clusters. It requires the number of clusters to be provided or to be determined (semi-) automatically. In our experiments, the cluster number is kept equal to the number of classes.

1. Choose a number of clusters  $k$
2. Initialize cluster centers  $\mu_1, \dots, \mu_k$ 
  - a. Could pick  $k$  data points and set cluster centers to these points
  - b. Or could randomly assign points to clusters and take means of clusters
3. For each data point, compute the cluster center it is closest to (using some distance measure) and assign the data point to this cluster
4. Re-compute cluster centers (mean of data points in cluster)
5. Stop when there are no new re-assignments

From the original dataset the instances having missing attributes are separated from the dataset. Now

we have two different datasets denoted as F and M. the dataset F contains the instances which have all the attribute values filled. The M dataset contains all the instances having missing attributes. Then the instances from M are taken one by one and the missing attributes are filled with their possible values. Then the assigned instance is added with the dataset F. Now the k-means clustering is applied to the dataset F, from the resultant clusters, the newly added instance is validated that whether it is been clustered in the correct class or not. If it is in the correct cluster, then the assigned value is made as permanent then the procedure is continued with the next instance in the M dataset. If it is in the wrong cluster then the next possible value will be assigned and compared till we found the value which put the instance in the correct cluster. At the end of each clustering step the quality of the cluster is measured with entropy value. There are many different quality measures and the performance and relative ranking of different clustering algorithms can vary substantially depending on which measure is used. However, if one clustering algorithm performs better than other clustering algorithms on many of these measures, then we can have some confidence that it is truly the best clustering algorithm for the situation being evaluated.

We use entropy as a measure of quality of the clusters (with the caveat that the best entropy is obtained when each cluster contains exactly one data point). Let CS be a clustering solution. For each cluster, the class distribution of the data is calculated first, i.e., for cluster  $j$  we compute  $p_{ij}$ , the "probability" that a member of cluster  $j$  belongs to class  $i$ . Then using this class distribution, the entropy of each cluster  $j$  is calculated using the standard formula:

$$E_j = -\sum_i p_{ij} \log(p_{ij})$$

where the sum is taken over all classes. The total entropy for a set of clusters is calculated as the sum of the entropies of each cluster weighted by the size of each cluster:

$$E_{CS} = \sum_{j=1}^m \frac{n_j * E_j}{n}$$

Where:

- $n_j$  = The size of cluster  $j$
- $m$  = The number of clusters
- $n$  = The total number of data points

From the second iteration, the current entropy value is compared with the previous one; if the current

entropy is less than the previous one then the presently assigned value is considered as best fit for that missing attribute, otherwise the previously assigned value is considered as best fit for that missing attribute. If we can't find the correct cluster till the end then the best fit value is assigned for the missing attribute. Suppose an instance having more than one missing attribute values and then all the possible combinations are checked. After assigning the missing attributes, the feature selection is performed with Bees Colony Optimization (BCO) and the improved Genetic KNN is applied for finding the classification performance.

**Bee Colony Based Reduct (BeeRSAR):** Nature is inspiring researchers to develop models for solving their problems. Optimization is an instance field in which these models are frequently developed and applied. Genetic algorithm simulating natural selection and genetic operators, Particle Swarm Optimization algorithm simulating flock of birds and school of fishes, Artificial Immune System simulating the cell masses of immune system, ACO algorithm simulating foraging behaviour of ants and Artificial Bee Colony algorithm simulating foraging behaviour of honeybees are typical examples of nature inspired optimization algorithms.

Artificial Bee Colony (ABC) algorithm for real parameter optimization, is a recently introduced optimization algorithm and simulates the foraging behaviour of bee colony for unconstrained optimization problems (Karaboga and Basturk, 2008). For solving constrained optimization problems, a constraint handling method was incorporated with the algorithm (Srichandum and Rujiranyong, 2010).

Algorithm Bee Colony Optimization Algorithms:

- Initialize the food source positions
- Each employed bee produces a new food source in her food source site and exploits the better source
- Each onlooker bee selects a source depending on the quality of her solution, produces a new food source in selected source site and exploits the better source
- Determine the source to be abandoned and allocate its employed bee as scout for searching new food sources.
- Memorize the best food source found so far
- Repeat steps 2-5 until the stopping criterion is met

In a real bee colony, there are some tasks performed by specialized individuals. These specialized bees try to maximize the nectar amount stored in the hive by performing efficient division of labour and self-organization. The minimal model of swarm-intelligent

forage selection in a honey bee colony, that ABC algorithm adopts, consists of three kinds of bees: employed bees, onlooker bees and scout bees. Half of the colony comprises employed bees and the other half includes the onlooker bees. Employed bees are responsible from exploiting the nectar sources explored before and giving information to the other waiting bees (onlooker bees) in the hive about the quality of the food source site which they are exploiting. Onlooker bees wait in the hive and decide a food source to exploit depending on the information shared by the employed bees. Scouts randomly search the environment in order to find a new food source depending on an internal motivation or possible external clues or randomly. Main steps of the ABC algorithm simulating these behaviours are given in the above algorithm, this procedure can be implemented for feature reduction, let the bees select the feature subsets at random and calculate their fitness and finds the best one at each iteration. This procedure is repeated for number of iterations to find the optimal subset.

In first step of the algorithm, the employed bee produces the feature subset in random. Consider a conditional feature set C contains N features. Then 'p' number of bees has been chosen as the population size. From this population half of the bees are considered as employed bee and the remaining is considered as onlooker bee. For each employed bee N random numbers are generated between 1 and N and assigned to them. From these random numbers the feature subset is constructed by performing round operation and then extracts only the unique numbers from the set. For example, consider the random numbers:

{1.45, 1.76, 3.33, 1.01}, where N = 4

First we perform round operation, then the set is modified as:

{1 1 3 1}

from the above result extract the unique numbers alone, as {1 3} represent the feature subset. ie., the 1st and 3rd feature values alone. In the second step of the algorithm, for each employed bee, whose total number equals to the half of the number of food sources, a new source is produced by:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj})$$

Where:

$\phi_{ij}$  = A uniformly distributed real random number within the range [-1,1]

- k = The index of the solution chosen randomly from the colony ( $k = \text{int}(\text{rand} * N) + 1$ ),  $j = 1, \dots, D$
- D = The dimension of the problem

After producing  $v_i$ , this new solution is compared to  $x_i$  solution and the employed bee exploits the better source. In the third step of the algorithm, an onlooker bee chooses a food source with the probability and produces a new source in selected food source site. As for employed bee, the better source is decided to be exploited. The indiscernibility relation is calculated for each feature subset as objective value ( $f_i$ ). This value has to be maximized. From this objective value the fitness value is calculated for each bee as given in the following equation:

$$\text{fit}_i = \begin{cases} 1 / (1 + f_i) & \text{if } f_i \geq 0 \\ 1 + \text{abs}(f_i) & \text{otherwise} \end{cases}$$

The probability is calculated by means of fitness value using the following equation:

$$P_i = \frac{\text{fit}_i}{\sum_{j=1}^N \text{fit}_j}$$

where  $\text{fit}_i$  is the fitness of the solution  $x_i$ . After all onlookers are distributed to the sources, sources are checked whether they are to be abandoned. If the number of cycles that a source cannot be improved is greater than a predetermined limit, the source is considered to be exhausted. The employed bee associated with the exhausted source becomes a scout and makes a random search in problem domain by the following equation:

$$x_{ij} = x_j^{\min} + (x_j^{\max} - x_j^{\min}) * \text{rand}$$

The pseudocode of our proposed method is given as:

**Algorithm:** Bee Colony based Reduct Algorithm:

ROUGHBEE (C,D)

C, the set of all conditional features;

D, the set of decision features.

- (1) Select the initial parameter values for BCO
- (2) Initialize the population ( $x_i$ )
- (3) Calculate the objective and fitness value
- (4) Find the optimum feature subset as global.
- (5) do
  - a. Produce new feature subset ( $v_i$ )
  - b. Apply the greedy selection between  $x_i$  and  $v_i$

- c. Calculate the fitness and probability values
  - d. Produce the solutions for onlookers
  - e. Apply the greedy selection for onlookers
  - f. Determine the abandoned solution and scouts
  - g. Calculate the cycle best feature subset
  - h. Memorize the best optimum feature subset
- (6) repeat // for maximum number of cycles

The following parameters we have used in our proposed method:

The population size (number of bees)	10
The dimension of the population	N
Lower bound	1
Upper bound	N
Maximum number of iterations	1000
The number of runs	3

**Improved KNN classification based on genetic algorithm:**

In pattern recognition field, KNN is one of the most important non-parameter algorithms and it's a supervised learning algorithm (Eskandarinia *et al.*, 2010; Lee *et al.*, 2011; Saaid *et al.*, 2009). The classification rules are generated by the training samples themselves without any additional data. KNN classification algorithm predicts the test samples category according to the K training samples which are the nearest neighbors to the test sample and judge it to that category which has the largest category probability. The process of KNN algorithm to classify sample X is:

- Suppose that there are j training categories as  $C_1, C_2, \dots, C_j$  and the sum of the training samples is N after feature reduction, they becomes m-dimension feature vector
- Make sample X to be the same feature vector form ( $X_1, X_2, \dots, X_m$ ) as all training samples
- Calculate the similarities between all training samples and X. Taking the  $i^{\text{th}}$  sample,  $d_i$  ( $d_{i1}, d_{i2}, \dots, d_{im}$ ) as an example, the similarity  $\text{SIM}(X, d_i)$  is as following:

$$\text{SIM}(X, d_i) = \frac{\sum_{j=1}^m X_j \cdot d_{ij}}{\sqrt{\left(\sum_{j=1}^m X_j\right)^2} \sqrt{\left(\sum_{j=1}^m d_{ij}\right)^2}}$$

- Choose k samples which are larger from N similarities of  $\text{SIM}(X, d_i)$ , ( $i=1, 2, \dots, N$ ) and treat them as a KNN collection of X. Then, calculate the probability of X belong to each category respectively with the following formula:

$$P(X, C_j) = \sum_d \text{SIM}(X, d_i) \cdot y(d_i, C_j)$$

where,  $y(d_i, C_j)$  is a category attribute function, which satisfied:

$$y(d_i, C_j) = \begin{cases} 1, & d_i \in C_j \\ 0, & d_i \notin C_j \end{cases}$$

- Judge sample  $X$  to be the category which has the largest  $P(X, C_j)$

In this study, Genetic Algorithm (GA) is combined with K-Nearest Neighbor (KNN) algorithm called as Genetic KNN (GKNN) to overcome the limitations of traditional KNN. In traditional KNN algorithm, initially the distance between all the test and training samples has been calculated and the k-neighbors with greater distances are taken for classification. In our proposed method, by GA, k-number of samples is going to be chosen for each iteration and the classification accuracy is calculated as fitness. The highest accuracy is recorded each time. Thus, it does not require calculating the similarities between all samples and no need to bother about weight of the category. Genetic Algorithm (GA) is randomized search and optimization techniques guided by the principles of evolution and natural genetics, having a large amount of implicit parallelism. GA perform search in complex, large and multimodal landscapes and provide near-optimal solutions for objective or fitness function of an optimization problem (Asfaw and Saiedi, 2011; Mahi and Izabatene, 2011; Mosavi, 2011; Matondang and Jambak, 2010; Nazif and Lee, 2010; Alfred, 2010; Sarabian and Lee, 2010; Yedjour *et al.*, 2010).

In GA, the parameters of the search space are encoded in the form of strings (called chromosomes). A collection of such strings is called a population. Initially, a random population is created, which represents different points in the search space. An objective and fitness function is associated with each string that represents the degree of goodness of the string. Based on the principle of survival of the fittest, a few of the strings are selected and each is assigned a number of copies that go into the mating pool. Biologically inspired operators like cross-over and mutation are applied on these strings to yield a new generation of strings. The process of selection, crossover and mutation continues for a fixed number of generations or till a termination condition is satisfied.

GA have applications in fields as diverse as VLSI design, image processing, neural networks, machine learning and job shop scheduling.

**String representation** - Here the chromosomes are encoded with real numbers; the number of genes in each chromosome represents the samples in the training set. Each gene will have 5 digits for vector index and k number of genes. For example, if  $k=5$ , a sample chromosome may look like as follows:

00100 10010 00256 01875 00098

Here, the 00098 represents, the 98th instance and the second gene say that the 1875 instance in the training sample. Once the initial population is generated now we are ready to apply genetic operators. With these k neighbors, the distance between each sample in the testing set is calculated and the accuracy is stored as the fitness values of this chromosome.

**Reproduction (selection):** The selection process selects chromosomes from the mating pool directed by the survival of the fittest concept of natural genetic systems. In the proportional selection strategy adopted in this article, a chromosome is assigned a number of copies, which is proportional to its fitness in the population, that go into the mating pool for further genetic operations. Roulette wheel selection is one common technique that implements the proportional selection strategy.

**Crossover:** Crossover is a probabilistic process that exchanges information between two parent chromosomes for generating two child chromosomes. In this study, single point crossover with a fixed crossover probability of  $p_c$  is used. For chromosomes of length  $l$ , a random integer, called the crossover point, is generated in the range  $[1, l-1]$ . The portions of the chromosomes lying to the right of the crossover point are exchanged to produce two offspring.

**Mutation:** Each chromosome undergoes mutation with a fixed probability  $p_m$ . For binary representation of chromosomes, a bit position (or gene) is mutated by simply flipping its value. Since we are considering real numbers in this study, a random position is chosen in the chromosome and replace by a random number between 0-9.

After the genetic operators are applied, the local maximum fitness value is calculated and compared with

global maximum. If the local maximum is greater than the global maximum then the global maximum is assigned with the local maximum and the next iteration is continued with the new population. The cluster points will be repositioned corresponding to the chromosome having global maximum. Otherwise, the next iteration is continued with the same old population. This process is repeated for N number of iterations. From the following section, it is shown that our refinement algorithm improves the cluster quality. The algorithm is given as.

**Algorithm:** Genetic k- Nearest Neighbor Algorithm:

1. Choose k number of samples from the training set to generate initial population (p1).
2. Calculate the distance between training samples in each chromosome and testing samples, as fitness value.
3. Choose the chromosome with highest fitness value store it as global maximum (Gmax).
  - a. For i = 1 to L do
    - i. Perform reproduction
    - ii. Apply the crossover operator.
    - iii. Perform mutation and get the new population. (p2)
    - iv. Calculate the local maximum (Lmax).
    - v. If  $G_{max} < L_{max}$  then
      - a.  $G_{max} = L_{max}$ ;
      - b.  $p1 = p2$ ;
  - b. Repeat
4. Output: The chromosome which obtains Gmax has the optimum K-neighbors and the corresponding labels are the classification results

## RESULTS AND DISCUSSION

The performance of the reduct approaches discussed in this study has been tested with 4 different medical datasets, downloaded from UCI machine learning data repository. Table 1 shows the details about the datasets used in this study.

The advantage of our proposed approach is, it doesn't check all the possible values for all the instances. It may assign at first time also, once it is correctly clustered then no need to check with the remaining possible values. Thus the runtime complexity can be enormously reduced.

Table 1: Datasets used for predicting missing attribute values

Dataset name	Total number of instances	Total number of features	Total number of missing attributes
Dermatology	366	34	64
Cleveland Heart	300	13	41
Lung Cancer	32	56	37
Wisconsin	699	09	121

Table 2: Reduced feature sets

Dataset name	Total number of features	Reduced feature Set (BeeRSAR)
Dermatology	34	7
Cleveland heart	13	6
Lung cancer	56	4
Wisconsin	09	4

Once the values are predicted for missing attributes, then the reduced feature set is received from a novel method based on Rough set theory hybrid with Bee Colony Optimization (BCO) as we have discussed in our earlier work. Table 2 shows the reduced feature sets.

Then the Genetic Algorithm (GA) is combined with k-Nearest Neighbour (KNN) algorithm called as Genetic KNN (GKNN) classifier is employed to analyze the classification performance. Table 3 shows the comparison of classification accuracy of our proposed approach with the existing methods. It is clearly shown that k-means clustering approach can predict the missing attributes better than any other existing approaches.

## CONCLUSION

Missing attribute values are very common the real-world dataset. Several methods have been proposed to predict these missing attribute values, but we can't say that they can predict well than the others. In this study, we have proposed a novel approach for predicting missing attribute values using simple k-means clustering. The missing attributes are assigned with one possible value each time and the dataset is clustered using k-means to check whether the instance is clustered in the correct class, if so then the assigned value is made as permanent. Otherwise the clustering is performed with the next possible value. If we found that no one possible value put the instance in the correct cluster then the best fit value is assigned for that missing attribute based on entropy measurement. This novel approach is implemented for x number of medical dataset with missing attribute values. After prediction, the reduced feature set is constructed using Rough set theory hybrid with BCO and the classification performance is studied with Genetic-KNN classifier. The results shows that k-means clustering can predict the missing attribute values better than any other approaches.

Table 3: Performance analysis of preprocessing approaches

	K	Dermatology	Cleveland Heart	Lung Cancer	Wisconsin
k-means	5	93.86 ± 0.12	85.56 ± 0.63	86.69 ± 0.81	95.70 ± 0.53
	10	93.65 ± 0.53	90.45 ± 0.61	84.69 ± 0.58	86.39 ± 0.47
	15	94.60 ± 0.26	87.87 ± 0.19	87.51 ± 0.49	94.28 ± 0.78
	20	93.80 ± 0.65	88.96 ± 0.85	86.78 ± 0.35	91.38 ± 0.84
MCAV	5	84.70 ± 0.74	84.89 ± 0.52	85.32 ± 0.34	86.39 ± 0.42
	10	85.10 ± 0.39	85.03 ± 0.69	85.98 ± 0.36	86.80 ± 0.36
	15	85.52 ± 0.95	85.04 ± 0.49	86.70 ± 0.59	87.93 ± 0.63
	20	86.53 ± 0.92	85.15 ± 0.82	86.82 ± 0.32	88.15 ± 0.34
CMCAV	5	87.26 ± 0.39	85.78 ± 0.22	87.35 ± 0.14	88.48 ± 0.24
	10	87.66 ± 0.17	85.96 ± 0.27	88.11 ± 0.32	88.92 ± 0.53
	15	88.17 ± 0.18	86.26 ± 0.23	88.25 ± 0.29	89.02 ± 0.19
	20	88.71 ± 0.41	86.88 ± 0.12	88.38 ± 0.26	89.18 ± 0.17
C4.5	5	88.99 ± 0.78	86.89 ± 0.13	89.16 ± 0.24	89.36 ± 0.12
	10	89.14 ± 0.61	87.14 ± 0.24	89.55 ± 0.88	89.39 ± 0.43
	15	89.37 ± 0.66	89.14 ± 0.39	89.58 ± 0.25	89.46 ± 0.77
	20	89.96 ± 0.61	89.85 ± 0.53	89.86 ± 0.83	89.82 ± 0.86
APV	5	79.57 ± 0.23	80.61 ± 0.62	78.90 ± 0.12	82.70 ± 0.50
	10	81.49 ± 0.98	80.67 ± 0.41	79.88 ± 0.88	82.73 ± 0.13
	15	82.06 ± 0.91	81.72 ± 0.75	81.44 ± 0.68	83.84 ± 0.66
	20	82.48 ± 0.79	82.33 ± 0.34	81.73 ± 0.11	85.06 ± 0.36
APVRC	5	78.89 ± 0.21	77.71 ± 0.17	77.76 ± 0.79	81.12 ± 0.18
	10	79.43 ± 0.15	77.72 ± 0.76	77.81 ± 0.78	81.73 ± 0.55
	15	79.47 ± 0.54	78.52 ± 0.18	78.43 ± 0.36	81.82 ± 0.34
	20	79.48 ± 0.41	80.42 ± 0.99	78.74 ± 0.78	81.94 ± 0.45
IGNORE	5	76.03 ± 0.27	74.07 ± 0.31	75.07 ± 0.54	78.60 ± 0.26
	10	76.07 ± 0.72	74.62 ± 0.16	75.08 ± 0.23	79.13 ± 0.74
	15	76.72 ± 0.71	76.53 ± 0.15	75.25 ± 0.58	79.53 ± 0.37
	20	77.68 ± 0.82	76.99 ± 0.60	77.59 ± 0.27	79.63 ± 0.31
EC	5	74.86 ± 0.57	72.49 ± 0.25	73.95 ± 0.62	75.81 ± 0.33
	10	74.87 ± 0.56	72.80 ± 0.29	74.06 ± 0.13	76.89 ± 0.71
	15	75.63 ± 0.28	73.00 ± 0.79	74.76 ± 0.27	78.10 ± 0.50
	20	75.95 ± 0.14	73.22 ± 0.72	74.89 ± 0.40	78.20 ± 0.86
LEM2	5	83.03 ± 0.81	82.37 ± 0.39	82.45 ± 0.45	85.18 ± 0.73
	10	83.06 ± 0.22	82.49 ± 0.91	83.71 ± 0.15	85.68 ± 0.88
	15	83.30 ± 0.23	84.26 ± 0.14	83.75 ± 0.76	85.79 ± 0.17
	20	84.35 ± 0.50	84.52 ± 0.12	84.08 ± 0.32	85.93 ± 0.11
SPECIAL	5	70.32 ± 0.52	70.70 ± 0.16	70.05 ± 0.49	72.47 ± 0.11
	10	71.57 ± 0.89	70.99 ± 0.96	70.99 ± 0.83	72.82 ± 0.79
	15	74.10 ± 0.43	71.53 ± 0.06	71.71 ± 0.47	74.98 ± 0.57
	20	74.20 ± 0.43	72.09 ± 0.53	71.94 ± 0.26	75.48 ± 0.53

**REFERENCES**

Alfred, R., 2010. Summarizing relational data using semi-supervised genetic algorithm-based clustering techniques. *J. Comput. Sci.*, 6: 775-784. DOI: 10.3844/jcssp.2010.775.784

Asfaw, T.D. and S. Saiedi, 2011. Optimal short-term cascade reservoirs operation using genetic algorithm. *Asian J. Applied Sci.*, 4: 297-305.

Eskandarinia, A., H. Nazarpour, M. Teimouri and M.Z. Ahmadi, 2010. Comparison of neural network and K-nearest neighbor methods in daily flow forecasting. *J. Applied Sci.*, 10: 1006-1010. DOI: 10.3923/jas.2010.1006.1010

Grzymala-Busse, J.W., W.J. Grzymala-Busse and L.K. Goodwin, 2001. Coping with missing attribute values based on closest fit in preterm birth data: A rough set approach. *Comput. Intell.*, 17: 425-434. DOI:10.1111/0824-7935.00155

Greco, S., B. Matarazzo and R. Slowinski, 1999. Handling missing values in rough set analysis of multi-attribute and multi-criteria decision problems. *Proceedings of the 7th International Workshop on New Directions in Rough Sets, Data Mining and Granular-Soft Computing, (PSFDGrC'99)*, Springer-Verlag London, UK, pp: 146-157.

Grzymala-Busse, J.W. and M. Hu, 2001. A comparison of several approaches to missing attribute values in data mining. *Lect. Notes Comput. Sci.*, 2005: 378-385. DOI: 10.1007/3-540-45554-X\_46

Grzymala-Busse, J.W., 2005. Incomplete data and generalization of indiscernibility relation, definability and approximations. *Lect. Notes Comput. Sci.*, 3641: 244-253. DOI: 10.1007/11548669\_26



- Jaradat, A., R. Salleh and A. Abid, 2009. Imitating K-means to enhance data selection. *J. Applied Sci.*, 9: 3569-3574. ISSN: 1812-5654
- Karaboga, D. and B. Basturk, 2008. On the performance of Artificial Bee Colony (ABC) algorithm. *Applied Soft Comput.*, 8: 687-697. DOI: 10.1016/j.asoc.2007.05.007
- Lee, L.H., C.H. Wan, T.F. Yong and H.M. Kok, 2011. A review of nearest neighbor-support vector machines hybrid classification models. *Journal of Applied Science*, 10: 1841-1858. ISSN: 1812-5654 <http://docsdrive.com/pdfs/ansinet/jas/2010/1841-1858.pdf>
- Mahi, H. and H.F. Izabatene, 2011. Segmentation of satellite imagery using RBF neural network and genetic algorithm. *Asian J. Applied Sci.*, 4: 186-194.
- Matondang, M.Z. and M.I. Jambak, 2010. Soft computing in optimizing assembly lines balancing. *J. Comput. Sci.*, 6: 141-162. DOI: 10.3844/jcssp.2010.141.162
- Mosavi, M.R., 2011. Applying genetic algorithm to fast and precise selection of GPS satellites. *Asian J. Applied Sci.*, 4: 229-237. DOI: 10.3923/ajaps.2011.229.237
- Nakata, M. and H. Sakai, 2005. Rough sets handling missing values probabilistically interpreted. *Lect. Notes Comput. Sci.*, 3641: 325-334. DOI: 10.1007/11548669\_34
- Nazif, H. and L.S. Lee, 2010. Optimized crossover genetic algorithm for vehicle routing problem with time windows. *Am. J. Applied Sci.*, 7: 95-101. DOI: 10.3844/ajassp.2010.95.101
- Pavan, K.K., A.A. Rao, A.V.D. Rao and G.R. Sridhar, 2010. Single pass seed selection algorithm for k-means. *J. Comput. Sci.*, 6: 60-66. DOI: 10.3844/jcssp.2010.60.66
- Saaïd, M.F.M., Z. Ibrahim, M. Khalid and N.H. Sarmin, 2009. DNA code word design for DNA computing with real-time polymerase chain reaction. *J. Comput. Sci.*, 5: 1-10. DOI: 10.3844/jcssp.2009.1.10
- Sarabian, M. and L.V. Lee, 2010. A modified partially mapped multicrossover genetic algorithm for two-dimensional bin packing problem. *J. Math. Stat.*, 6: 157-162. DOI: 10.3844/jmssp.2010.157.162
- Srichandum, S. and T. Rujiranyong, 2010. Production scheduling for dispatching ready mixed concrete trucks using bee colony optimization. *Am. J. Eng. Applied Sci.*, 3: 7-14. DOI: 10.3844/ajeassp.2010.7.14
- Stefanowski, J. and A. Tsoukias, 2001. Incomplete information tables and rough classification. *Comput. Intell.*, 17: 545-566. DOI:10.1111/0824-7935.00162
- Suguna, N. and K. Thanushkodi, 2010a. A novel rough set reduct algorithm for medical domain based on bee colony optimization. *J. Comput.*, 2: 49-54.
- Suguna, N. and Thanushkodi K, 2010b. An improved k-nearest neighbor classification using genetic algorithm. *Int. J. Comput. Sci. Issues*, 7: 18-21.
- Yedjour, D., H. Yedjour and A. Benyettou, 2010. Combining quine mc-cluskey and genetic algorithms for extracting rules from trained neural networks. *Asian J. Applied Sci.*, 4: 72-80. DOI: 10.3923/ajaps.2011.72.80