ACO Based Feature Subset Selection for Multiple *k*-Nearest Neighbor Classifiers

Shailendra Kumar Shrivastava

Information Technology Department
Samrat Ashok Technological Institute Vidisha, (M.P.) India

Pradeep Mewada

Information Technology Department
Samrat Ashok Technological Institute Vidisha, (M.P.) India

Abstract- The k-nearest neighbor (k-NN) is one of the most popular algorithms used for classification in various fields of pattern recognition & data mining problems. In k-nearest neighbor classification, the result of a new instance query is classified based on the majority of k-nearest neighbors. Recently researchers have begun paying attention to combining a set of individual k-NN classifiers, each using a different subset of features, with the hope of improving the overall classification accuracy. In this paper we proposed Ant Colony Optimization (ACO) based feature subset selection for multiple k-nearest neighbor classifiers. The ACO is an iterative meta-heuristic search technique, which inspired by the foraging food behavior of real ant colonies. In ACO, real ants become artificial ants with the particular abilities such as distance determination & tour memory. The solution is constructed in a probabilistic way based on pheromone model in the form of numerical values. The concept of this approach is selecting the best possible subsets of feature from the original set with the help of ACO and combines the outputs from multiple k-NN classifiers. The experimental results show that this proposed method improves the average classification accuracy of k-NN classifier.

Keywords- Machine Learning; k-Nearest Neighbor; Feature Subset Selection; Ant Colony Optimization.

I. INTRODUCTION

Machine Learning is a subfield of artificial intelligence that is concerned with the design and development of algorithms and techniques that allow computers to "learn". This is a broad field which includes Supervised Learning, Unsupervised Learning, Semi-supervised Learning, and Reinforcement Learning [1]. Classification is one of the important techniques used in the field of Machine Learning. Data classification is the categorization of data for its most effective and efficient use. In supervised learning, we have a training set of data & for each record of this set, the respective class to which it belongs is also known. Using the training set, the classification process attempts to generate the descriptions of the classes & these descriptions help to classify the unknown records.

The *k*-nearest neighbor (*k*-NN) is a supervised learning method, where a new pattern x is classified based on the majority of *k*-nearest neighbors [2]. There are three key elements of this approach, a set of labeled data, a distance metric to compute distance between elements, and the value of *k* (number of nearest neighbors). To classify an unlabeled data, the distance of this data to the labeled data is computed, its nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class-label of the data. The *k*-NN algorithm is very simple to understand and easy to implement. So it should be considered in seeking a solution to any classification problem. The classification process of *k*-NN is transparent, it is easy to implement, and can be very effective if an analysis of the neighbors is useful as explanation. The *k*-NN has many advantages over other methods. For example, it can learn from small set of example, and often gives competitive performance with more modern methods such as decision trees or neural network [3].

Recently, feature subset selection for multiple classifiers has been viewed as a new direction for the development of highly reliable pattern recognition systems. Preliminary results indicate that combination of several complementary classifiers with different subsets of feature leads to classifiers with improved performance. The reason of using the concept of feature subset selection is that, for any specific recognition problem, there are often numerous types of feature which could be used to represent and recognize patterns. Irrelevant or redundant features may reduce the performance of classifier. Feature selection attempts to select the minimally sized subset of features where the classification accuracy does not significantly decrease. The main purpose of feature selection is to preserve an individual classifier's accuracy, while feature subset selection aims to improve the combination performance of multiple classifiers [4].

In this paper we proposed Ant Colony Optimization (ACO) based feature subset selection for multiple *k*-nearest neighbor classifiers. The ACO is an iterative meta-heuristic search technique, which inspired by the foraging food behavior of real ant colonies. These have been successfully applied to a large number of difficult combinatorial problems such as the quadratic assignment and the traveling salesman problems [5], [6]. In ACO, real ants become artificial ants with the particular abilities such as distance determination & tour memory. The solution is constructed in a probabilistic way based on pheromone model in the form of numerical values [7]. The concept of this approach is selecting the best subsets of features from the original set with the help of ACO and combines the outputs from multiple *k*-NN classifiers. We apply this method on the selected dataset from UCI Machine Learning Repository [8]. We have evaluated and compared the performance of our proposed method with four different existing techniques namely MFS, D*k*NN, FC-MNNC and DFTS3/1NN. The experimental result indicates that this proposed method provides much better classification accuracy as compare to the existing techniques.

II. *K*-NEAREST NEIGHBOR ALGORITHM

The *k*-nearest-neighbor algorithm is a basic instance based learning method and widely used in similarity classification. The purpose of this algorithm is to classify a new dataset based on attributes and training [9]. The classifier does not use any model to fit and only based on memory. To classify a new pattern x, the *k*-NN classifiers find k nearest patterns in the training database, and uses the k pattern to determine the class of pattern x.

Given a training set *D* and a test object *x = (*x', *y')*, the algorithm computes the distance (or similarity) between *z* and the entire training object *(*x, *y)* which is belongs to *D* to determine it's nearest- neighbor list, *Dz*. (x is the data of training object, while *y* is its class. Likewise, x' is the data of the test object and *y*' is its class). Once the nearest-neighbor list is obtained, the test object is classified based on the majority class of its nearest neighbors [4]:

$$\text{Majority Voting: } y' = \text{argmax} \sum_{(xi,yi)} I(v=y_i) \qquad (1)$$

where v is a class label, y_i is the class label for the i^{th} nearest neighbors, and $I(v=y_i)$ is an indicator function that returns the value '1'if its argument is true and '0' otherwise.

Training algorithm:

Input: - D, the set of k training objects and test object z= (x', y').

Classification algorithm:

Process: -
1. Compute d(x', x), the distance between z and every object(x, y) belongs to D.
2. Select D_z (subset of D, the set of *k* closest training objects to z).

$$\text{Output: - } y' = \text{argmax} \sum_{(xi,yi)} I(v=y_i) \qquad (2)$$

There are several key issues that affect the performance of *k*-NN [9]. One is the choice of *k*. If k is too small, then the result can be sensitive to noise points. On the other hand, if *k* is too large, then the neighborhood may include too many points from other classes.

Another issue is the approach to combining the class labels. The simplest method is to take a majority vote, but this can be a problem if the nearest neighbors vary widely in their distance and the closer neighbors more reliably indicate the class of the object. A more sophisticated approach, which is usually much less sensitive to the choice of *k*, weights each object's vote by its distance, where the weight factor is often taken to be the reciprocal of the squared distance:

$$w_i = 1/d(\mathbf{x}', \mathbf{x}_i)^2 \qquad (3)$$

This amounts to replacing the last step of the *k*-NN algorithm with the following:

Distance-Weighted Voting:

$$y' = \text{argmax} \sum_{(xi,yi)} w_i * I(v=y_i) \qquad (4)$$

The choice of the distance measure is another important consideration. Although various measures can be used to compute the distance between two points [10], the most desirable distance measure is one for which a smaller distance between two objects implies a greater likelihood of having the same class. Thus, for example, if *k*-NN is being applied to classify documents, then it may be better to use the cosine measure rather than Euclidean distance. Some distance measure can also be affected by the high dimensionality of the data. A number of techniques have been developed for efficient computation of *k*- nearest neighbor distance that make use of the structure in the data to avoid having to compute distance to all the objects in the training set [11, 12].

III. ANT COLONY OPTIMIZATION

The Ant colony optimization (ACO) is an iterative meta- heuristic search technique, which inspired by the foraging food behavior of real ant colonies. Ant behavior was the inspiration for the meta-heuristic optimization techniques [13]. The basic concept of real ants is that, while walking between their colony and the food source, ants deposit pheromones along the paths they moved. A pheromone is an odorous substance, which is used as an indirect communication medium. When a source of food is found, ants lay some pheromone to mark the path. The quantity of the laid pheromone depends upon the distance, quantity and quality of the food source. While an isolated ant that moves at random detects a laid pheromone, it is very likely that it will decide to follow its path. This ant will itself lay a certain amount of pheromone, and hence enforce the pheromone trail of that specified path. Accordingly, the path that has been used by more ants will be more attractive to follow. In other words, the probability which an ant chooses a path increases with the number of ants that previously chose that path [5].

Marco Dorigo [6] was adopted this concept and proposed an artificial colony of ant's algorithm, which was called the Ant Colony Optimization (ACO). Marco Dorigo proposed four basic steps for ACO:

A. Probabilistic forward ants and solution construction:

Forward ants build a solution by choosing probabilistically the next node to move to among those in the neighborhood of the graph node on which they are located.

B. Deterministic backward ants and pheromone update:

The use of an explicit memory allows an ant to retrace the path it has followed while searching the destination node.

C. Pheromone updates based on solution quality:

In ACO the ants memorize the nodes they visited during the forward path, as well as the cost of the arcs traversed if the graph is weighted. They can therefore evaluate the cost of the solutions they generate and use this evaluation to modulate the amount of pheromone they deposit while in backward mode.

D. Pheromone evaporation:

In real ant colonies, pheromone intensity decreases over time because of evaporation. In ACO evaporation is simulated by applying an appropriately defined pheromone evaporation rule. For example, artificial pheromone decay can be set to a constant rate. Pheromone evaporation reduces the influence of the pheromones deposited in the early stages of the search, when artificial ants can build poor-quality solutions.

The ACO was originally applied to solve the classical travelling salesman problem [14] where it was shown to be an effective tool in finding good solutions. The ACO has also been successfully applied to other optimization problems including data mining, telecommunications networks, vehicle routing, etc [20, 21, and 22].

In order to solve an optimization problem, a number of artificial ants are used to iteratively construct solutions. In each iteration, an ant would deposit a certain amount of pheromone proportional to the quality of the solution. At each step, every ant computes a set of feasible expansions to its current partial solution and selects one of these depending on two factors: local heuristics and prior knowledge [23].

IV. EXISTING TECHNIQUES FOR MULTIPLE *K*-NN CLASSIFIERS

A. k-NN by Multiple Feature Subset(MFS)

Stephen D. Bays [3] has proposed a method of classification from multiple feature subsets (MFS) which combines multiple *k*-NN classifiers. In this method random subset of features were selected by sampling from original dataset. This method evaluated the performance of MFS using two different sampling functions: Sampling with replacement (MFS1) and sampling without replacement (MFS2). In MFS, all classifiers used unweighted Euclidean distance function.

MFS allows many desirable properties of the *k*-NN classifier in a multiple model framework. For example one of the primary advantage of *k*-NN classifier is its ability to incrementally add new data (or remove old data) without requiring retraining. Another useful property of *k*-NN classifier is its ability to predict directly from the training data without using intermediate structures. As a result, no matter, how many classifiers we combine in MFS, we require only the same memory as a single *k*-NN classifier.

Since *k*-NN is sensitive to the change of feature sets, the combination of *k*-NN based on different feature subsets may lead to a better performance. The experimental result indicates that MFS has a significance increases in the accuracy of the *k*-NN Classifier.

B. k-NN by Multiple Distance Function(Dk-NN)

Takahiro Yamada, Kyohei Yamashita [12] proposed a method to combine *k*-NN classifiers based on different distance functions with weights. The choice of distance function influences the bias of the *k*-NN classification. There are many distance functions that have been proposed for the *k*-NN classification [10]. The process of D*k*-NN method is simple First, it inputs several distance functions. Then, it uses each distance function to generate *k* nearest samples in the training data .Then it combines the all *k* nearest samples and determines the class of unknown object based on the simple voting. The experimental results of D*k*-NN method indicates that this method give much better classification accuracy as compare with MFS method.

C. k-NN by Feature Subset Clustering(FC-MNNC)

Li-Juan Wang, Xiao-Long Wang [16] has proposed a method FC-MNNC (Feature subset Clustering for Multiple Nearest Neighbor Classifiers) which was based on feature subset clustering for combining multiple *k*-NN to obtain better performance than that using a single *k*-NN. This method uses the concept of feature subset clustering which is used to improve the combination performance of multiple classifiers. Feature subset selection is similar to feature selection, both of which use a part of feature to classify unknown patterns. Feature subset clustering can be used to improve the combination performance of multiple classifiers, where the feature set is clustered into different feature subsets, not discarding the irrelevant features. Because feature selection cannot deal with multiple decisions, it only uses a part of features and discards the irrelevant features to classify. In FC-MNNC method, GA is used for clustering features to form different feature subsets according to the combination classification accuracy. The final decision is aggregated by majority voting rule [17], which is a simple and efficient technique. On the basis of our study it is concluded that the resultant classification accuracy of FC-MNNC outperforms the D*k*-NN method.

D. k-NN using Simultaneous Metaheuristic Feature Selection(DF-TS3/1NN)

Tabu search (TS) has been applied to the problem of feature selection by Zhang and Sun [18]. Starting from an initial solution, TS examines a set of feasible neighboring solutions and moves to the best admissible neighbor, even if this causes the objective function to deteriorate. To avoid cycling, solutions that were recently explored are declared forbidden or Tabu for a number of iterations. The Tabu status of a solution is overridden when certain aspiration criteria are satisfied [18].

Muhammad Atif Tahir and Jim Smith [15] proposed the DF-TS3/1NN (simultaneous metaheuristic feature selection) technique which combines multiple *k*-NN classifiers, each using a different distance function, and potentially a different set of features (feature vector). These feature vectors are determined for each distance metric simultaneously using Tabu Search to minimize the ensemble error rate [18]. M. Kudo [19] was proposed and compares different algorithms that use the concept of feature selection for pattern classification. In the training phase of DF-TS3/1NN method, during each iteration random neighbors are generated for each distance metric using *k*-NN classifier. m_1 best neighbors are selected from *n* neighbors during each iteration. m_n solutions are then evaluated using simple voting scheme. Thus, the feedback from the simple voting scheme allows TS to iteratively search for feature vectors that improves the classification accuracy. By using n distance functions, n feature vectors are obtained using TS in the training phase. In the testing phase, multiple *k*-NN classifiers are combined using n distance functions and n different feature vectors. The experimental results of DF-TS3/1NN method indicates that this method outperforms the above three existing *k*-NN techniques.

V. PROPOSED ACO BASED MULTIPLE *K*-NEAREST NEIGHBOR CLASSIFIERS

A new ensemble technique is proposed in this paper to improve the performance of *k*-nearest-neighbor (*k*-NN) classifier. This approach combines multiple *k*-NN classifiers, where each classifier uses a different subset of features. These subsets of feature are selected through ACO based search procedure. ACO is basically a meta-heuristic search technique, which is inspired by the foraging food behavior of real ant colonies. This method has been successfully applied to a large number of difficult combinatorial problems such as the traveling salesman problems [5]. In ACO, real ants become artificial ants with the particular abilities such as distance

determination & tour memory. The solution is constructed in a probabilistic way based on pheromone model in the form of numerical values.
The main steps of proposed ACO algorithm for Feature Subset Selection using multiple *k*-NN are as follows:

Step *1*: Feature Subset Generation

Initialize the input parameter values and pheromone trail value. At the beginning of the search process, a constant amount of pheromone is assigned to the entire path;

Set $\tau_{ij}=1$, $(i, j) \in A$

$\Delta\tau^k = 0$

where, τ_{ij} is the artificial pheromone trail ('*i*' and '*j*' are two neighboring nodes);

```
for j=1 to n_a
{
  generate the initial candidate feature subsets;
}
```

Step 2: Feature subset Evaluation

The candidate feature subsets are evaluated to measure the goodness of the produced features;

```
for j=1 to n_a
{
    Evaluate the generated feature subset
}
end
{
    Estimate the error rate;
}
```

Step 3: Feature subset Update

```
for j=1 to k
{
  Update the pheromone value using best k ants;
  τij ←— τij + Δτ^k
}
```

Step 4: Pheromone evaporates

```
{
  The pheromone trails are evaporated after k^th ant has
  moved to a next node;
  τij ←— (1-ρ) τij
  where, ρ is a constant, 1-ρ represents evaporation in
  pheromone trail;
}
```

Step 5 : Applying *k*-Nearest Neighbor classification

```
{
  Majority Voting: y'= argmax
}
end.
```

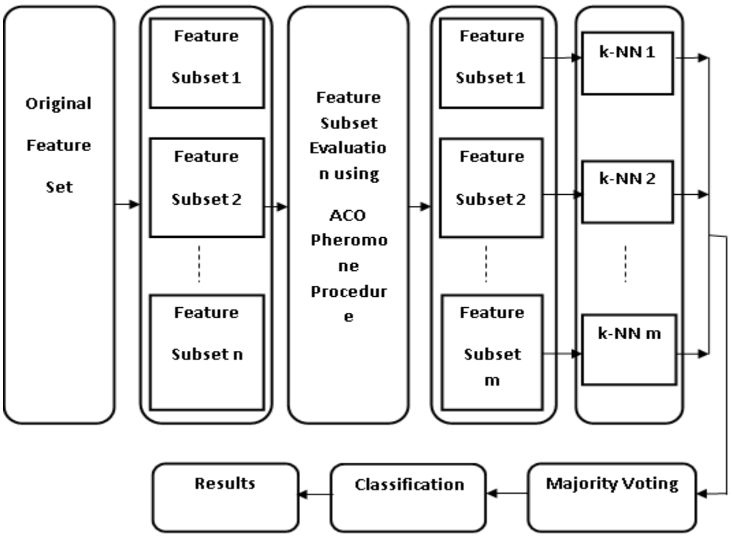


Figure 1. Architecture of the proposed ACO based *k*-NN method.

VI. EXPERIMENTS AND RESULTS

To evaluate the performance of our method, we select the Abalone dataset from UCI Machine Learning Repository [8]. The dataset is comes from an original (non-machine-learning) study: Warwick J Nash, Tracy L Sellers "The Population Biology of Abalone in Tasmania" (1994). In our experiment we have used seven continuous attributes of this dataset. We have evaluated and compared the results of our proposed method along with four different existing techniques namely Multiple Feature Subset (MFS) [3], Distance Weighted *k*-NN (D*k*-NN) [12], Feature Subset Clustering Using Multiple Nearest Neighbor Classifier (FC-MNNC) [16], Simultaneous Metaheuristic Feature Selection (DF-TS3/1NN) [15]. The results are compared in the form of classification accuracy. We have also compared the mean absolute error, root mean absolute error, relative absolute error and root relative absolute error of our proposed method with the four existing methods. The table given below shows the experimental results of our proposed method compared with four existing methods.

Table 1. Performance Comparison with Existing method

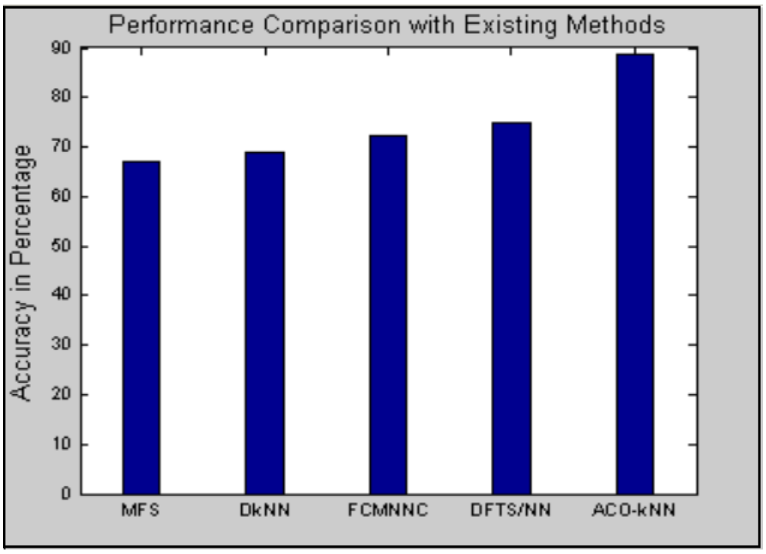
Algorithm/ Technique	Mean Absolute Error	Root-Mean Absolute Error	Relative Absolute Error	Root Relative Absolute Error	Average Accuracy In %
MFS	5.3	2.3	5.3	2.3	67.1
D*k*-NN	5.3	2.3	5.1	2.2	68.9
FC-MNNC	5.2	2.2	5.0	2.2	72.3
DF-TS3/1NN	5.1	2.2	5.0	2.2	75.0
Proposed Method	10.1	3.2	9.0	3.0	88.7

MFS- Multiple Feature Subset, D*k*-NN- Distance Weighted *k*-Nearest Neighbor FC-MNNC-Feature subset clustering

for Multiple *k*-Nearest Neighbor Classifiers, DF-TS3/1NN- Simultaneous metaheuristic feature selection.

The experimental results of our proposed method along with the four existing methods are shown in table1. These results are evaluated on a fixed threshold value (the generating value from which the classifier start searching of nearest neighbor). We can also change the value of threshold; if we change the value of threshold the results will be changed nevertheless the proposed method obtained best classification accuracy on different value of threshold among the four existing methods with a little loss in error rate, this is due to increase in the value of *k*.

Graph.1 shows the performance comparison of our proposed method with existing techniques



MFS- Multiple Feature Subset, D*k*-NN- Distance Weighted *k*-Nearest Neighbor FC-MNNC-Feature subset clustering

for Multiple *k*-Nearest Neighbor Classifiers, DF-TS3/1NN- Simultaneous metaheuristic feature selection.

VII. CONCLUSION

In this paper we proposed a new method for classification in which the feature subset selection is performed by ACO and the classification results is obtained by majority voting of multiple *k*-NN classifiers. We have evaluated and compared the performance of our proposed method with four different ensemble techniques namely MFS, D*k*-NN, FC-MNNC and DFTS3/1NN. The MFS method, which combines *k*-NN classifiers using random feature subsets, gives the classification accuracy as 67.1% on selected abalone dataset. The D*k*-NN method uses the concept of weighted distance function for multiple *k*-NN classifiers. The classification accuracy of D*k*-NN method is 68.9%. The FC-MNNC method based on feature subset clustering for combining multiple *k*-NN classifiers to obtain better performance than that using a single *k*-NN. The classification accuracy of FC-MNNC method is 72.3%. The DF-TS3/1NN method combines multiple *k*-NN classifiers, each using a different distance function, and potentially a different set of features (feature vector). The classification accuracy of this method is 75.0%.

We proposed a new method in which the feature subset selection is performed by ACO based search procedure and the results is generated by majority voting of multiple *k*-NN classifiers. This proposed method gives maximum classification accuracy as 88.7%. On the basis of experimental results, it is concluded that the proposed method obtained best classification results among the four existing techniques. Our future research will be concerned to apply this proposed ACO based feature subset selection method on other classifier like SVM.

REFERENCES

[1] A. K. Pujari "Data mining techniques" University Press: 42-66, February- 2001.
[2] Tom M. Mitchell "machine learning" McGraw-Hill Science/ Engineering/ Math: 230-245, March 1, 1997.
[3] Stephen D. Bay "Combining nearest neighbor classifiers through multiple feature subsets" Proceeding 17th Intl. Conf. on Machine Learning: 191-209,1998.
[4] Pradeep Mewada, Shailendra K. Shrivastava "Review of combining multiple *k*-nearest neighbor classifiers" International Journal of Computational Intelligence Research & Applications(IJCIRA),4(2): 187-191 July-December 2010.
[5] M. Dorigo and T. Stützle, "Ant colony optimization," MIT Press, 2004.
[6] M. Dorigo, V. Maniezzo, and A. Colorni. Ant Systems: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26: 29-41, 1996.
[7] M. Sadeghzadeh and M.H. Shenasa, Correlation Based Feature Selection Using Evolutionary Programming, Proceedings of the Twelfth IASTED International Conference on Artificial Intelligence and Soft Computing, Palma de Mallorca, Spain: 185-190, 2008.

[8] C. Blake, E. Keogh, and C. J .Merz "UCI Repository of machine learning databases" University of California, Irvine, Department of information and computer science, 1998.
[9] X. Wu et al. "Top 10 algorithms in data mining" Knowledge information Springer-Verlag London Limited: 22-24, 2007.
[10] D. R. Wilson and T. R. Martinez "Improved heterogeneous distance functions" Journal of Artificial Intelligence Research, 6(1):1–34, 1997.
[11] M. A. Tahir, A. Bouridane, F. Kurugollu, and A. Amira "Feature Selection using Tabu Search for Improving the Classification Rate of Prostate Needle Biopsies" Proceedings of the 17th International Conference on Pattern Recognition, IEEE-ICPR-2004.
[12] Takahiro Yamada, Kyohei Yamashita, Naohiro Ishii "Text Classification by Combining Different Distance Functions with Weights" Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing- SNPD, 2006.
[13] Hamidreza Rashidy Kanan, Karim Faezand Mehdi Hosseinzadeh "Face Recognition System Using Ant Colony Optimization-Based Selected Features" Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Security and Defense Applications 1-4244-0700-1CISDA- 2007.
[14] Marco Dorigo, Luca Maria Gambardella, "Ant colonies for the travelling salesman problem", BioSystems, 43, pp. 73–81, 1997.
[15] Muhammad Atif Tahir and Jim Smith "Creating diverse nearest-neighbour ensembles using simultaneous metaheuristic feature selection" Article in press- pattern recognition letters journal homepage: www.elsevier.com, 2010.
[16] Li-Juan Wang, Xiao-Long Wang, Qing-Cai Chen "GA-based Feature Subset Clustering for Combination of Multiple Nearest Neighbor Classifiers" Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, 18-21 August 2005.
[17] L. Louisa Lam and C Y. Suen "Application of majority voting to pattern recognition: An analysis of its behavior and performance", IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 27, no. 5, pp.553-567, 1997.
[18] Zhang, H., Sun, G. "Feature selection using tabu search method" Pattern Recognition 35, 701–711, 2002.
[19] M. Kudo and J. Sklansky "Comparison of algorithms that select features for pattern classifiers" Pattern Recognition: 25–41, 2000.
[20] Shengyuan Yan, Jingling Zhang, Shuaiqi Wang and Yu Chen "Ant Colony Algorithm based Controls' Arrangement Optimization" 978-1-4244-5848-IEEE, 2010.
[21] Abdelwadood Moh'd Mesleh and Ghassan Kanaan "Support Vector Machine Text Classification System: Using Ant Colony Optimization Based Feature Subset Selection 978-1-4244-2116-9IEEE, 2008.
[22] A. Yahya, "On the complexity of the initial stages of Arabic text processing," First Great Lakes Computer Science Conference; Kalamazoo, Michigan, USA, October, 1989.
[23] R. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," Expert Systems with Applications, vol. 33, 49-60, 2007.