Improving the Performance of Artificial Neural Networks via Instance Selection and Feature Dimensionality Reduction

Ali Abroudi, Mohammad Shokouhifar, and Fardad Farokhi

***Abstract*—This paper presents a hybrid approach with two phases for improving the performance of training artificial neural networks (ANNs) by selection of the most important instances for training, and then reduction the dimensionality of features. The ANNs which are applied in this paper for validation, are included Multi-Layer Perceptron (MLP) and Neuro-Fuzzy Network (NFN). In the first phase, the Modified Fast Condensed Nearest Neighbor (MFCNN) algorithm is used to construct the subset with instances very close to the decision boundary. It leads to achieve the instances more useful for training the network. And in the second phase, an Ant-based approach to the supervised reduction of feature dimensionality is introduced, aims to reduce the complexity, and improve the accuracy of learning the ANN. The main purpose of this method is to enhance the classification performance by improving the quality of the training set. Experimental results illustrated the efficiency of the proposed approach.**

***Index Terms*—Artificial neural networks, instance selection, feature selection, ant colony optimization, EDA, MFCNN.**

I. Introduction

The learning procedure has different steps including, constructing a Training Set (TS), training the network, testing the behavior of the network on the new instances. The first step consists of editing and condensing the data. The most important objective of any abstraction technique is to obtain a TS with the best performance in classification process. To build an optimal TS, we have to select the instances and features in an appropriate way to guarantee attaining to the best result.

The whole data is located in a matrix which its rows show the instances and the columns are the features. The instance selection schemes generally try to reduce the number of rows in entire dataset with no loss in the classification performance [1]. Non-parametric K-Nearest Neighbor (KNN) rule [2], predicts the label of a new instance by computing a similarity scale between the selected one and all other instances in TS. Condensed Nearest Neighbor rule (CNN) [3], was introduced for the nearest neighbor classification method and tried to build a condensed subset from the whole set of examples. The basic idea of this algorithm is that if an example is not correctly classified, this may occur because it is a boundary instance, and it should be considered in the selected condense set.

Fast Condensed Nearest Neighbor (FCNN) [4], starts by set S that is initialized with centroid of each class in TS. Then, until the TS is not completely classified with condensed set S, the algorithm selects for insertion a representative of the misclassified points of each Voronoi (Voronoi of point $P \in S$ is the set of all instances in TS that are closer to P than to any other instance in S) cell induced by the current subset. Edited Nearest Neighbor (ENN) removes noisy instances, as well as close border points, leaving smoother decision boundaries [5]. It edits out those prototypes that misclassified using KNN rule. Allknn [6] is based on the ENN method. This algorithm, for k=1 to m, marks as removable any prototype misclassified by its k NNs. When the algorithm iterates m times, it eliminates the prototypes flagged as removable.

The Patterns with Ordered Projections (POP) algorithm [7] consists of removing the instances that are not boundary ones. That is, the behavior of this algorithm is in opposite direction from that of Allknn and ENN. The Reduced Nearest Neighbor rule (RNN) produces the smallest subset which classifies all the learning prototypes correctly [8]. In Drop3 algorithm [9], we have the *associate* concept, so that the associate of the instance x_i , is the instance x_a which has x_i as nearest neighbor. This algorithm eliminates x_i if at least majority of its associates in TS would be classified correctly without x_i. Unlike most of instance selection methods that concentrate on improving the performance of KNN algorithm, in our previous work [10], we used the concept of FCNN rule aims to select the most efficient instances for training the ANNs.

Since most of the practical problems are written in high dimensional spaces, their features have to be either selected or combined into a fewer number. Feature selection approaches mostly are employed for reduction of the features which must be extracted. However, the feature combination methods are applied to improve the performance of training the ANN, which is the main concern in this paper. All feature selection algorithms according to subset evaluation strategies, can be categorized in *wrapper* and *filter* approaches. In filters, a classifier independent index is used for evaluation the gathered subsets; while the classification error is used in wrappers [11].

Typically feature selection algorithms which are based on Evolutionary Algorithms (EAs), perform better than the conventional techniques especially for high-dimensional data sets. In [12], a rough set approach based on Ant Colony Optimization (ACO) was proposed, which adopts mutual information based feature significance as heuristic information. In [13], an ACO was proposed, which in the each iteration, the classification accuracy is used for evaluation of feature subsets represented by ants. In the

Manuscript received December 30, 2012; revised April 15, 2013.

The authors are with the Electrical Engineering Department, Islamic Azad University of Central Tehran Branch, Tehran, Iran. (e-mail: a.abroudi@ieee.org; m_shokouhifar@ieee.org; f_farokhi@iauctb.ac.ir).

previous works, we proposed both filter and wrapper feature selection approaches based on ACO [14]-[17], and artificial bee colony (ABC) algorithm [18], [19].

Feature combination algorithms can be classified into two categories, depending on whether the problem is supervised or not. In unsupervised situations, Principal Component Analysis (PCA) [20] is widely used as feature dimensionality reduction algorithm. And when supervision is available, Fisher's Discriminant Analysis (FDA) [21] is mostly employed. In FDA, a projected class is located far from the others as possible, and the patterns are grouped around their means. This is a smart measure for class separation. FDA is a dimensionality reduction technique, not a classification method. It is mostly used as a preprocessing step before classification. Evolutionary Discriminant Analysis (EDA) [22], was introduced by Sierra and Echeverria as a modified version of FDA, which use a genetic algorithm to optimize the classification error.

In this paper, a modified FCNN (named MFCNN) is introduced for effective selection of most important instances for train the ANN. And then an Ant-based EDA (named AntEDA) is proposed in order to achieve the reduced feature space, and it has been shown that this approach is more powerful to correctly classify the patterns compare to the original ones.

II. The Used Artificial Neural Networks

A. MLP Network

MLP is a well-known feed-forward neural network that usually used for classification problems because of its good performance. A typical MLP consist of input layer which represents the input features of the problem, and an output layer which represents the possible outcomes, and one or more hidden layers between them.

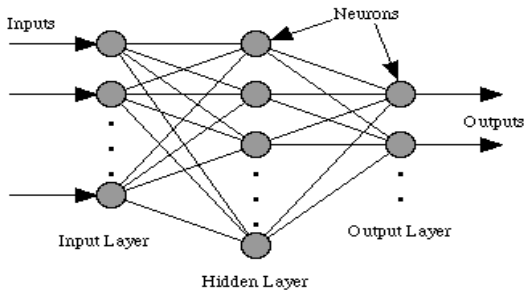

Fig. 1. The MLP structure: consist of an input layer, an output layer, and one or more hidden layers between them.

In this network, the neurons are fully connected to each other via connecting weights. Sigmoid activation function is applied to get the output of each neuron using its inputs, and Back Propagation algorithm is used to learn the weights.

B. Neuro-Fuzzy Network

The Neuro-Fuzzy Network (NFN) formally defined as combination of the Neural Networks and the Fuzzy Logic. In NFN, the membership functions are optimized iteratively according to the input-output pairs [23]. The NFN has the *Product* inference engine, *Singleton* Fuzzifier, *Center Average* Defuzzifier and *Gaussian* Membership function. The overall function on the network is shown in "Eq. (1)".

$$f(x)=\frac{\sum_{l=1}^{M}\bar{y}^{l}\left(\prod_{i=1}^{n}\exp\left[-\left(\frac{x_i-\bar{x}_i^{l}}{\sigma_i^{l}}\right)^2\right]\right)}{\sum_{l=1}^{M}\left(\prod_{i=1}^{n}\exp\left[-\left(\frac{x_i-\bar{x}_i^{l}}{\sigma_i^{l}}\right)^2\right]\right)} \quad (1)$$

where M is the number of the used rules, $\bar{y}^{l}$, $\bar{x}_i^{l}$ and σ_i^{l} are the parameters which should be optimized during the training process [23]. The algorithm has the following steps:

Step 1: Estimate network initial size and selecting $\bar{y}^{l}(0)$, $\bar{x}_i^{l}(0)$ and $\sigma_i^{l}(0)$ randomly.

Step 2: To map the input $x\in U\subset R^n$ to the output $f(x)\in V\subset R$, x should be passed through a product Gaussian operator to become Z^l, then Z^l are entered to a summation and weighted summation operators respectively to construct a and b. The mentioned operation is shown in "Eq. (2)" to "Eq. (4)". At the end, the output of the network is computed by $f(x)$ as "Eq. (5)".

$$Z^{l}=\left(\prod_{i=1}^{n}a_i^{l}\exp\left[-\left(\frac{x_i-\bar{x}_i^{l}}{\sigma_i^{l}}\right)^2\right]\right) \quad (2)$$

$$b=\sum_{l=1}^{M}Z^{l} \quad (3)$$

$$a=\sum_{l=1}^{M}\bar{y}^{l}Z^{l} \quad (4)$$

$$f(x)=a/b \quad (5)$$

Step 3: In this step, the parameters are updated to minimize the error e^p. The updated parameters $\bar{y}^{l}$, $\bar{x}_i^{l}$ and σ_i^{l} at the $(q+1)^{\text{th}}$ stage of training are computed as "Eq. (6)" to "Eq. (9)".

$$e^{p}=\frac{1}{2}[f(x_0^{p})-y_0^{p}]^2 \quad (6)$$

$$\bar{y}^{l}(q+1)=\bar{y}^{l}(q)-\eta\frac{f-y}{b}Z^{l} \quad (7)$$

$$\bar{x}_i^{l}(q+1)=\bar{x}_i^{l}(q)-\alpha\frac{f-y}{b}(\bar{y}^{l}(q)-f)Z^{l}\frac{2(x_{i0}^{p}-\bar{x}_{i(q)})}{\sigma_i^{l2}(q)} \quad (8)$$

$$\sigma_i^{l}(q+1)=\sigma_i^{l}(q)-\alpha\frac{f-y}{b}(\bar{y}^{l}(q)-f)Z^{l}\frac{2(x_{i0}^{p}-\bar{x}_i(q))^2}{\sigma_i^{l3}(q)} \quad (9)$$

Step 4: Returning to Step 2 with q=q+1, until e^p is less than a pre-determined number ε, or until the number of iterations equal to a pre-specified number.

Step 5: Returning to Step 2 with p=p+1; in other word, update parameters using the next instance.

III. Methodology

A. Instance Selection Using MFCNN Algorithm

In this sub-section, we describe the instance selection

method that we use for our hybrid model. Random selection of instances may not be the most adequate way for training the ANNs. The FCNN is one of the most efficient techniques for selecting instances, thus we perform some modifications on this method and name it: Modified FCNN (MFCNN).

Our proposal of MFCNN has two levels that are shown in Fig. 2. In level 1 of MFCNN, the centroids of the all classes in *TS* are determined and then transferred to condensed set *S* as initial instances. Then, during each iteration, for each instance in *S*, the Voronoi set is obtained. Then, those samples of Voronoi that have a different label from corresponding instance (we called this set as Voren) are picked and all class centroids of Vorenare calculated and eventually inserted to *S*.

```
Input: A dataset D;
Output: A double-condensed set SSof D;
Level 1:
S= ø;
        ΔS = Centroids(D);
        while (ΔS≠ø) {
S= S∪ΔS;
        ΔS= ø;
        for each (P∈S)
        ΔS= ΔS∪ {Centroids(P,Voren(P, S, D))};
        }
          return(S);
Level 2:
       SS= ø;
        ΔSS = Centroids(S);
        while (ΔSS≠ø) {
SS= SS ∪ΔSS;
        ΔSS= ø;
        for each (P∈SS)
        ΔSS= ΔSS∪ {rep(P,Voren(P, SS, S))};
        }
          return(SS);
```

Fig. 2. The MFCNN rule.

This level stops when no more instances can be added to *S*, in other words, when *TS* is correctly classified using condensed-set *S*. In level 2 of MFCNN, we try to find the smaller condensed set (double-condensed set) from the output of level 1. In this level, we apply the FCNN base rule on *S* set and finally the *SS* set is produced.

In level 2, first of all, the centroids of all classes in resultant S are determined and placed into the new condensed set *SS* (During each iteration, we used ΔS and ΔSS as temporary sets for *S* and *SS* sets respectively). And after that, the Voren set of each instance P∈*SS* is constructed, then the closest sample to *P* is selected and inserted to *SS* as representative. The stop criterion of this level is similar to previous level.

B. Feature Dimensionality Reduction Using Evolutionary Discriminant Analysis and ACO Algorithm

ACO has been introduced based on the foraging behavior of real ants [24]. Ants deposit some pheromone on ground when they walking, in order to mark some favorably paths that should be followed by the other ants. Pheromone will be evaporated soon, but in the short time it remained on the ground as trail of ants. The ants in option to choose between two paths, probability choose that path which has more amount of pheromone. We have been previously proposed some ACO-based feature selection and ranking approaches [14-17]. In this paper, we propose a new feature combination algorithm using ACO (named AntEDA), to achieve the most useful projected features for classification.

The used structure for performing AntEDA can be seen in Fig. 3. In this structure f_i is the i^{th} original feature, and $\overline{f_j}$ is the j^{th} projected feature, which is calculated by "Eq. (10)". The *m* and *n* (*n*<*m*) are respectively the number of all original features and the number of projected features. The mean of projected feature *j* for the instances belonging to class *k* is calculated as "Eq. (11)". Note that *i*=1,2,…,*m*, *j*=1,2,…,*n* and *k*=1,2,...,*c*, which *c* is the number of all classes within the given dataset. as "Eq. (5)".

$$\overline{f_j} = \sum_{i=1}^{m} f_i \, w_{ij} \tag{10}$$

$$\overline{M_j^k} = \sum_{i=1}^{m} M_i^k \, w_{ij} \tag{11}$$

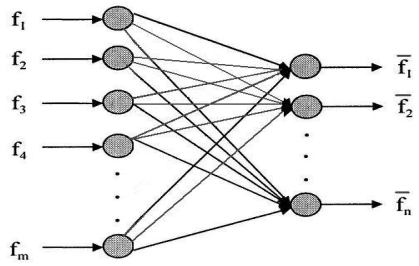

Fig. 3. The projection structure used for AntEDA : The left nodes are the original features, and the right nodes are the projected features, which be achieved with AntEDA algorithm .

The AntEDA classifies patterns as belonging to the class of the closest projected mean as "Eq. (12)". In this manner, the instance *x* belongs to the class that it closest to its mean.

$$class(x) = argmin_k\left(\sum_{j=1}^{n}(\overline{f_j} - \overline{M_j^k})^2\right) \tag{12}$$

We are searching for the optimal linear projections. To do that, we propose an ACO algorithm to optimize the weights of the structure. It leads to minimize the number of misclassified patterns, in order to achieve the projected features more useful than original ones. Note that, the total weights of the AntEDA structure are equal to *m*×*n*. Typically ACO has performed to discrete problems. The allowable values for each weight is considered from the set of {-1, -0.99, -0.98,…, 0.98, 0.99, 1}. The optimization process using ACO for obtaining the optimal weights is performed by generation a colony consists of *k* artificial ants to search through the possible weights of AntEDA structure. To construct a solution, each ant selects values for each weight among allowable values, with respect to probability rule as "Eq. (13)". The probability of k^{th} ant to select the i^{th} value for the j^{th} at time *t* is given as:

$$P_{ij}^{k}(t) = \frac{\rho_{ij}^{\alpha}(t)}{\sum_{l=1}^{T}\rho_{lj}^{\alpha}(t)} \tag{13}$$

where ρ_{ij} is the pheromone amount of i^{th} value for the j^{th} weight. *T* is the number of discreteness of the weights which is 201. After all ants construct their projections, they are evaluated by "Eq. (14)". The projection's cost is proportional to the number of misclassified patterns. In the other words, the

cost of a given projection is equal to the percentage of all misclassified patterns for that projection.

$$J = \frac{1}{N}\sum_X L(t(x), class(x)) \quad (14)$$

Which $t(x)$ is target output of pattern x . If $t(x)=class(x)$ then $L=0$, and otherwise $L=1$. And N is the number of all patterns within the dataset. After all ants traverse among the problem space and have constructed their projections, pheromone trails on each node (each value of each weight) will be updated. The change in the pheromone of each node is according to:

$$\Delta\rho_{ij} = -\lambda\rho_{ij}(t) + \begin{cases} Q & \text{for edges of the best ants} \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where $\rho_{ij}(t)$ is the pheromone amount on node (i,j) at this iteration. Note that, the node (i,j) means the i^{th} value of j^{th} weight. And λ $(0<\lambda<1)$ is the evaporation of pheromone trails for all nodes. Q is a user-defined constant that adjust the amount of pheromone deposition on the associated nodes of the best ants. During each iteration, the pheromone amount of all nodes must be bounded between two values. The overall steps to achieve the optimal weights using ACO algorithm can be summarized as follow:

• An initial random population of ants is generated. Each ant randomly selects a possible value for each weight, and constructs a projection.

• The following steps are repeated until the error rate is less than *max-iterations*, or a prescribed number of generations is reached.

1) The gathered projections are evaluated by the number of misclassified patterns according to "Eq. (14)".
2) The best ants are selected.
3) The pheromone trails are updated by "Eq. (15)": evaporation for all values, and deposition just for the values of best ants.
4) A new population is generated according to pheromone trails by the selection rule as "Eq. (13)".
5) A new projection becomes the current best projection when the error is reduced.

• The best projection found so far, is chosen as output.

Note that our approach classifies patterns by proximity to means in the projected space. Since this classification rule is not necessarily optimal, after performing AntEDA, a neural network must be used to take full advantage of the projected features. So after performing the proposed MFCNN instance selection algorithm and the AntEDA feature selection approach, MLP and NF networks are used for classification on achieved TS with useful features. The overall flowchart of proposed hybrid approach can be seen in Fig. 4.

IV. Experimental Results

To further verify the effectiveness of our method, 13 UCI classification data sets were used as listed in Table I. To validate the proposed approach, we applied our method on MLP and NFN. The training results by the randomly selected TS using all original features and instances (we called it RND), and the selection via proposed approach (HYBRID), and also MFCNN are reported in Table. II. Furthermore the number of features and instances extracted from each datasets (represented by F and I respectively) is included. pe.

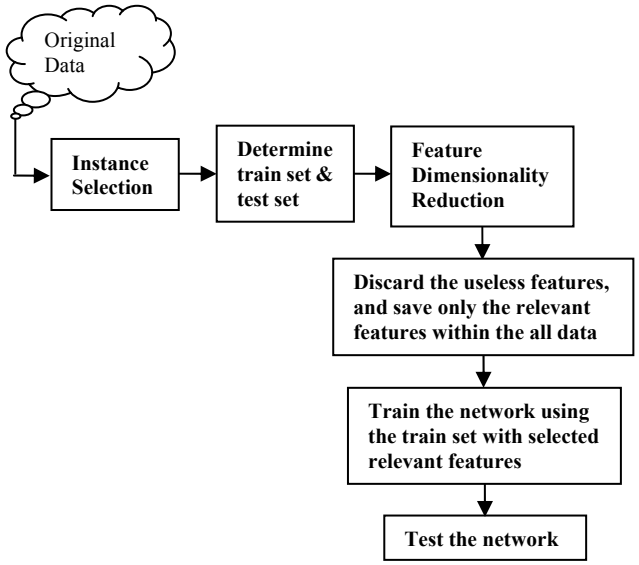

Fig. 4. Overall flowchart of proposed hybrid algorithm.

TABLE I: Datasets Details

Data base	Instances	Features	Classes
Heart	270	13	2
Letter	20000	16	26
Breast	286	9	2
Twonorm	7400	20	2
Wine	178	13	3
Sonar	208	60	2
Nursery	12960	8	3
Pima	768	8	2
Spambase	4597	57	2
Magic	19020	10	2
Ionosphere	351	33	2
Bupa	345	6	2
Chess	3196	36	2

As shown in this table, the performance of the Hybrid and MFCNN methods is always better than conventional random selection method. Furthermore, because of selection of effective features in Hybrid method, this method guarantees the same and even better classification accuracy than MFCNN approach. The results approve the power of the presented Hybrid method for training artificial intelligent networks with the most effective features and instances.

TABLE II: Comparison Between Obtained Results

Database	RND		MFCNN			HYBRID		
	MLP	NFN	I	MLP	NFN	MLP	NFN	F
Heart	90.2	83.25	105	95.34	89.21	92.3	87.64	6
Letter	98.36	98.12	4536	99.87	99.21	99.37	99.04	8
Breast	97.21	96.43	105	99.92	99.89	98.88	98.32	2
Twonorm	95.78	93.84	1043	99.02	98.31	98.92	98.12	7
Wine	99.85	99.55	61	100	99.99	100	99.95	6
Sonar	77.12	74.43	56	81.35	75.98	78.82	74.03	13
Nursery	96.32	93.02	2653	98.11	95.23	97.88	94.66	6
Pima	73.14	71.36	342	90.78	87.92	92.85	88.12	5
Spambase	89.41	87.52	987	91.89	90.54	93.78	91.54	21
Magic	84.74	81.68	6548	91.35	86.67	93.27	88.12	7
Ionosphere	88.73	87.01	83	90.65	88.99	89.64	88.25	13
Bupa	60.68	73.22	196	74.56	77.12	75.34	77.89	4
Chess	92.39	91.88	305	97.01	94.97	97.06	95.11	10

V. Conclusion

In this paper a new Hybrid approach based on instance selection and feature selection, was proposed for improving the performance of training ANNs. In FDA, two different criteria should be optimized: the Fisher's measure, and the classification error in the projected space. But in proposed AntEDA, the error is minimized directly without using any intermediate criteria by aid of ACO. The proposed Hybrid algorithm efficiently reduced the number of features and instances. As a result, it achieves better results than random selection method and even MFCNN approach that uses all of features for training the networks. As a future work, the algorithm will be applied for very large datasets.

References

[1] H. Liu and H. Motoda, "On issues of instance selection," *Data Mining Knowledge Discovery 6 (2)*, 2002, pp.115–130.

[2] G. Shakhnarovich, T. Darrel, P. Indyk (Eds.), *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*, MIT Press, Cambridge, MA, 2006.

[3] P. E. Hart, "The Condensed Nearest Neighbor Rule," *IEEE Trans. Information Theory*, 1968,vol. 14, no. 3, pp. 515-516.

[4] F. Angiulli, "Fast Condensed Nearest Neighbor Rule," in *Proc. 22nd International Conference on Machine Learning*, 2005, pp. 25-32.

[5] D. L. Wilson, "Asymptotic properties of nearest neighbor rules using edited data," *IEEE Trans. Syst. Man Cybern,* vol. 2, no.3, 1972, pp. 408–421.

[6] I. Tomek, "An experiment with the edited nearest-neighbor rule," *IEEE Trans. Syst. Man Cybern,* vol. 6, no. 6, 1976, pp. 448–452.

[7] J. C. Riquelme, J. S. Aguilar-Ruiz, and M. Toro, "Finding representative patterns with ordered projections," *Pattern Recognition 36*, 2003, pp.1009–1018.

[8] G. W. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Inf. Theory,* vol. 18, no.3, 1972, pp. 431–433.

[9] D. R. Wilson and T. R. Martinez, "Reduction techniques for instance-based learning algorithms," *Mach. Learn,* vol. *38,* 2000, pp. 257–286.

[10] A. Abroudi, F. Farokhi, and F. Zahedi, "Instance Selection for Training of Intelligent Networks based on Fast Condensed Nearest Neighbor Rule," in *Proc. 3rd International Conference on Information Science and Engineering (ICISE'11)*, 2011, pp. 3394–3398.

[11] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell,* 1997, vol. 97, pp. 273–324.

[12] Y. Chen, D. Miao, and R. Wang, "A rough set approach to feature selection based on ant colony optimization," *Pattern Recognition Letters*, 2010, vol. 31, pp. 226–233.

[13] R. K. Sivagaminathan and S. Ramakrishnan, "A hybrid approach for feature subset selection using neural networks and ant colony optimization," *Expert systems with applications*, 2007, vol. 33, pp. 49-60.

[14] M. Shokouhifar and F. Farokhi, "Feature Selection using Supervised Fuzzy C-Means Algorithm with Ant Colony Optimization," in *Proc. 3rd International Conference on Machine Vision(ICMV)*, December 2010, pp. 441-446.

[15] M. Shokouhifar and F. Farokhi, "Simultaneous Feature Ranking and Classification using Ant Colony Optimization applied in Multilayer Perceptron Neural Network," in *Proc. 3rd International Conference on Machine Vision (ICMV)*, December 2010, pp.507-512.

[16] M. Shokouhifar, "An Ant Colony Optimization Algorithm for Effective Feature Selection," in *Proc. 3rd International Conference on Machine Learning and Computing (ICMLC)*, February 2011, pp. 36-40.

[17] M. Shokouhifar, F. Farokhi, Amir Hossein Abdollahi, and Gholamhasan Sajedy Abkenar "Improved UTA Feature Selection using Ant Colony Optimization," in *Proc. 3rd International Conference on Machine Learning and Computing (ICMLC)*, February 2011, pp. 266-270.

[18] M. Shokouhifar and F. Farokhi, "An Artificial Bee Colony Optimization for Feature Subset Selection using Supervised Fuzzy C_Means Algorithm," in *Proc. 3rd International Conference on Information Security and Artificial Intelligent (ISAI)*, December 2010, pp. 427-432.

[19] M. Shokouhifar and Sh. Sabet, "A Hybrid Approach for Effective Feature Selection using Neural Networks and Artificial Bee Colony Optimization," in *Proc. 3rd International Conference on Machine Vision (ICMV)*, , December 2010, pp. 502-506.

[20] J. E. Jackson, *A User's Guide to Principal Components*. New York: Wiley, 1991.

[21] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Ann. Eugenics*, 1936, vol. 7, pp. 179–188.

[22] A. Sierra, A. Echeverría, "Evolutionary Discriminant Analysis," *IEEE Transctions on Evolutionary Computing*, 2006, vol. 10, no. 1, pp.81-92.

[23] L. X. Wang, *ACourse in Fuzzy Systems and Control. Englewood Cliffs*, NJ: Prentice-Hall, 1997.

[24] M. Dorigo, V. Maniezzo, A. Colorni, "The Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man Cybernet*, 1996, vol. 26-B, no. 1, pp. 29–41.

M. Shokouhifar was born in Dezfoul, Iran, in 1987. He received the MS degree in electronic engineering from Islamic Azad University of Central Tehran Branch, Tehran, Iran, in 2011. He currently is a MS student in electronic engineering in the field of analog circuit design, Shahid Beheshti University, Tehran, Iran. His research interests include: pattern recognition, image processing, fuzzy sets theory, analog and digital circuit design, wireless sensor networks, feature selection, and solving NP-complete problems by using evolutionary algorithms. He has more than 20 international journals and conference papers in the field of evolutionary algorithms. In 2010, he received the award of superior researcher from the Islamic Azad University of Central Tehran Branch. He is a student member of IEEE.