Highly Efficient Dimension Reduction for Text-Independent Speaker Verification Based on Relieff Algorithm and Support Vector Machines

Abdolreza Rashno[1], Hossein SadeghianNejad[2] and Abed Heshmati[3]

[1]Department of Engineering, Lorestan University
[2]Department of Computer Engineering,
Applied Science and Technology University, Yasuj
[3]Department of Computer Engineering and Information Technology,
Payame Noor University, I.R of IRAN
ar.rashno@gmail.com

Abstract

Automatic speaker verification (ASV) systems are among the biometric systems used in security and telephone-based remote control applications. Recent years have witnessed an increasing trend in research on such systems. These systems usually use high dimension feature vectors and therefore involve high complexity. However, there is a general belief that many of the features used in such systems are irrelevant and redundant. So far, many methods for feature dimension reduction in these systems have been proposed, most of which are wrapper-based and thus computationally expensive since system performance is used for feature subset evaluation. This involves system training and performance evaluation for each feature subset, which is a time consuming task. In this paper, we propose a feature selection approach based on Relieff algorithm for ASV systems using support vector machine (SVM) classifiers. This method is wrapper-based but makes use of Relieff weights in order to have a lower using of system performance. Thus this method has lower complexity compared to other wrapper-based methods, can lead to 69% feature dimension reduction and has a 1.25% of Equal Error Rate (EER) for the best case that appeared in RBF kernel of SVM. The proposed method has been compared with Genetic Algorithm (GA) and Ant Colony Optimization (ACO) methods for feature selection task. Results show that the EER, number of selected features and time complexity of the proposed method is lower than these methods for different kernels of SVM.

***Keywords**: Automatic speaker verification, Feature selection, Support vector machine, Relieff, Ant colony optimization, Genetic algorithm*

1. Introduction

Automatic Speaker recognition (ASR) systems refer to systems that recognize person from her/his voice. Such systems have been proposed since some decades ago. ASR generally divides into two stages: automatic speaker identification (ASI) and automatic speaker verification (ASV). ASI is a task of recognizing who is speaking from a set of known speakers or voices and the speaker ID is returned. ASV is a task of accepting or rejecting the claimed target speaker based on his/her voice and comparison of a score with a threshold [1]. ASV is an easier task compared to ASI. Speaker verification systems are divided into text-dependent and text-independent applications. In text-dependent ASV, speakers should pronounce the same text for train and test phases while in text-independent, there is no constraint on what the speakers should pronounce during the train/test phase. Thus text-independent speaker verification requires no restriction on the type of input speech. The design of text-independent ASV systems is usually more complex and their performance lower than text-dependent ones [2]. There are many applications for ASV systems such as bank security, telephone transactions, remote access control, forensic applications, remote network access etc. [3]. Generally, Features of each speech signal divide into low level and high level features. Low level features are easy to extract, independent to text and language of speaker, suitable for real time application, and affected by noise and mismatch conditions. On the other hand high level features are robust against channel effects and noise and difficult to extract [4]. Low level features are very important and useful and used recently in ASV applications. Some of low level features are: Mel Frequency Cepestral Coefficients (MFCC) [5], Linear Predictive Cepstral Coefficients (LPCCs) [6], Perceptual Linear Prediction (PLP) Coefficients [7] and so on.

Many of feature elements extracted from speech signals are redundant and irrelevant. A good feature vector should possess these properties: easy to extract from speech signal, occur frequently and naturally in speech, robust against noise and distortion, have large between-speaker variability and small within speaker variability, not be affected by the speaker's health or long-term variation in voice and difficult to mimic [4]. A key issue is that the size of the feature vector should be kept as small as possible since most of the classification methods used for ASV, such as Support Vector Machines (SVM) and Gaussian Mixture Models (GMM) involve high computational costs when feature vectors move towards higher dimensionalities. Feature selection is a method that finds and removes redundant and irrelevant feature components. As a result, low time complexity and high system accuracy can be achieved. In fact, the main purpose of feature selection is to reduce the number of feature components used in classification while maintaining acceptable classification accuracy and acceptable equal error rate (EER). The result is a subset of original features with a much lower number of elements per vector, compared to the original set. Many feature selection approaches have been proposed for different applications such as face recognition [8], data mining and pattern recognition [9, 10], text categorization [11] and so on. Feature selection has also been applied to speaker recognition systems. Examples include L plus–R minus feature selection algorithm proposed for text-dependent speaker verification [12], dynamic programming-based feature selection used in ASV systems [13], information gain and gain ratio based feature selection used for ASV systems [14]. In addition, genetic algorithm was used for feature selection in speaker recognition [15], in HMM-based speaker verification [16] and feature selection in text-independent speaker identification based on GMMs [17]. Feature selection based on ant colony optimization (ACO) for ASV systems has also been proposed recently [18].

In this paper, we have proposed a feature selection approach based on relieff algorithm for text-independent speaker verification using SVM. This method is a wrapper-based but uses the relieff weights for features to have a lower complexity than other wrapper methods. Thus, this method is much faster than other wrapper method used in ASV systems such as GA, ACO. We implemented the proposed method on TIMIT data and compared the results with GA and ACO. The results have shown that this method always has a lower EER, lower feature number and lower complexity than GA- and ACO-based methods for different kernels of SVM.

The rest of this paper is organized as follows: Section 2 describes the structure and components of generic ASV systems based on SVM. Feature selection is explained in more details in Section 3. The algorithm and structure of the proposed system is described in Section 4. Experimental setup and results are described in Sections 5 and 6 respectively.

Section 7 is devoted to discussion on the complexity of our proposed algorithm and its comparison with GA and ACO. Finally, the conclusion and future works are discussed in Section 8.

2. SVM-based Speaker Verification System

ASV systems composed of several main phases. A block diagram of a SVM-based ASV system is shown in Figure 1. The main blocks of these systems are Feature extraction, SVM Training, similarity measure and decision making. A feature selection block may be added after the feature extraction step, as used in our approach and shown as a shaded block in Figure 1. This step is described in more details in Section 3.

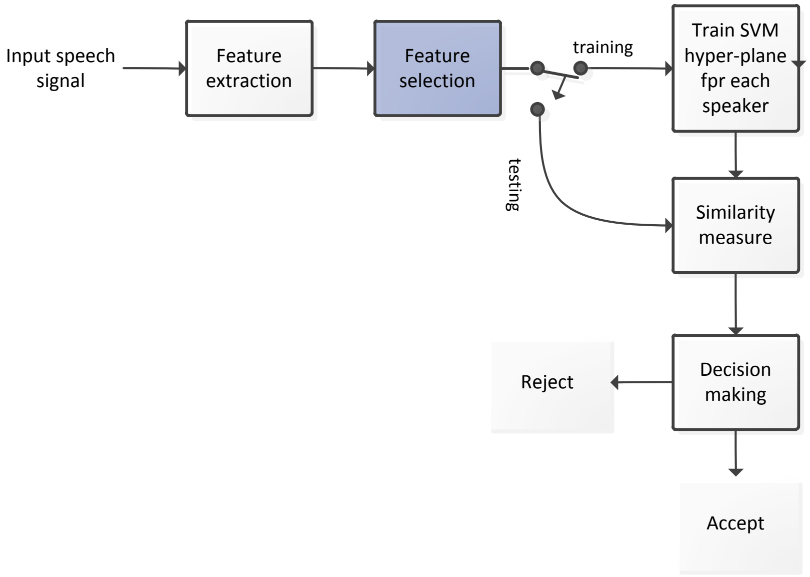


Figure 1. Block Diagram of a SVM-based ASV System

2.1. Feature Extraction

The speech signal is first analyzed by short-term analysis methods. This block is usually called feature extraction or front-end processing. Speech signal is the main input for the ASV system and contains much information about speakers. The objective of feature extraction is to convert speech signal to feature vectors. In fact, the main purpose of feature extraction is to extract features from speech with high variability between speakers. The choice of features in ASV is a primary concern, since if the feature set does not yield sufficient information, good estimate for system parameters would be almost impossible [19]. Most commonly used feature sets such as MFCC and LPCC have been particularly popular for ASV systems in recent years [20]. The recognition performance can be increased by adding dynamic features. This is often realized by looking at the derivation of each static parameter over time. These parameters are called delta parameters. However, delta parameters can be used as a simplified way of exploiting intera-frame dependencies and the dynamic behavior of signals [21].

2.2. SVM Training for Speakers

Support vector machines are among the powerful machine learning methods based on statistical learning theory [22, 23]. SVMs are widely used in pattern recognition problems because of their good generalization ability compared with traditional classification methods. SVM was originally proposed for binary classification problems but it can also be generalized for multi-class applications. Suppose that the training samples are$\left\{(x_i, y_i) : x_i \in R^d,\ y_i \in \{-1, +1\}\right\}_{i=1}^{N}$, where $x_i$an input vector with dimension d is and y_i is its class label. The problem is to find a hyper-plane to separate instances from two classes. If the input data are not linearly separable, we can't find this hyper-plane. Thus, in SVM, input data are mapped into input higher dimensional space using map functions in order to create linear separability between instances [24]. This problem can be viewed as a two-class pattern recognition problem with this hyper-plane:

$$f(x) = w^T x + b. \tag{1}$$

Where w is a d-dimension vector and b is a bias.

If the input data are not linearly separable, a map function, $\emptyset(x)$, is used to map the input space into a higher dimension space and its hyper plane is:

$$f(x) = w^T \emptyset(x) + b \tag{2}$$

Finding the optimal separating hyper-plane can be viewed as solving the following optimization problem:

$$Minimize \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \tag{3}$$

Subject to $y_i(w^T \emptyset(x) + b) \geq 1 - \xi_i \quad for\ i = 1, \ldots, N$

Where C is the parameter that determines the trade off between the maximization of the margin and minimization of the classification error. ξ_i is a positive slack variable that shows acceptable error for x_i. [25].

The objective of SVM is to determine the optimal weight w and optimal bias b that separates the positive and negative training data with maximum margin and leads to the best generalization. This hyper-plane is called an optimal separating hyper-plane. Using Lagrange multiplier techniques it leads to solving this optimization by dual optimization problem [26].

$$Maximize \quad F(\propto) = \sum_{i=1}^{N} \propto_i - \frac{1}{2}\sum_{i,j=1}^{N} \propto_i \propto_j y_i y_j K\left(x_i, x_j\right) \tag{4}$$

Subject to $\sum_{i=1}^{N} \propto_i y_i = 0 \qquad 0 \leq \propto_i \leq C$

$K(x_i, x_j)$ is a kernel function that is defined as:

$$K\left(x_i, x_j\right) = \emptyset(x_i)^T \emptyset\left(x_j\right) \tag{5}$$

Some popular kernels for SVM are:

Polynomial: $$K(x_i, x_j) = [(x_i, x_j) + 1]^d \tag{6}$$

Radial basis function (RBF): $$K(x_i, x_j) = exp\{-\frac{|x_i - x_j|^2}{\sigma^2}\} \quad (7)$$

Sigmoid function: $$K(x_i, x_j) = tanh[v(x_i.x_j) + c] \quad (8)$$

The training phase in SVM is formulated as a quadratic programming optimization procedure where the number of variables is equal to the number of training data and global optimal solution can be optioned. After solving this quadratic programming problem for each instance, $\propto_i$ is computed for it. Support vectors are the instances that their $\propto_i$s are more than 0 and these instances have a nearest distance with the decision boundary (hyper-plane). Using the support vectors, the decision boundary is given by:

$$f(x) = \sum_{i \in S} \propto_i y_i K(x_i, x) + b \quad (9)$$

where S is a set of support vectors and the weight vector w is given by:

$$w = \sum_{i \in S} \propto_i y_i \emptyset(x_i) \quad (10)$$

and the margin is given by:

$$\delta = \frac{1}{\|w\|} = \frac{1}{\sqrt{\sum_{i,j \in S} \propto_i \propto_j y_i y_j K(x_i, x_j)}} \quad (11)$$

In this work, we train a SVM for each speaker. Then the hyper-plane is used for each speaker that separates it from other speakers. In verification task, we have a claim for speaker *x* and in order to accept or reject this claim, SVM of the claimed speaker is used. Thus, for *n* speakers, *n* hyper-planes are trained. For each SVM, feature vectors of the main speaker create one side of training data and feature vectors of imposter speakers are the other side of training data. Since the amount of training data for the imposter speakers is very high, we apply k-means clustering algorithm to these vectors and use the cluster centroids instead of the main vectors. Thus, the overall number of training vactors for the imposter speakers is reduced and the SVM will have lower complexity and will converge faster.

3. Feature Selection

Let *X* be the set of original features with cardinality *n*; the continuous feature selection problem refers to the assignment of weight w_i to feature *i*, where the weight indicates the importance of this feature. The binary feature selection problem refers to the assignment of binary weights to features [27]. Feature selection is a procedure of selecting *m* features from *n* features that $m \leq n$ and it is a discrete optimization problem [28].The whole search space for the optimization contains all possible subsets of features, meaning that its size is:

$$\sum_{s=0}^{n} \binom{n}{s} = \binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{n} = 2^n \quad (12)$$

Where *n* is the dimensionality (the number of features) and *s* is the size of the current feature subset [28].

By removing irrelevant and redundant features from the main features of data, performance of learning models is improved by: alleviating the effect of the dimensionality, speeding up learning process, enhancing generalization capability, improving model interpretability and so on [28, 18, 27]. Generally, feature selection methods divide into filter and wrapper methods. Wrapper methods use a search algorithm to search the space of possible features and evaluate each feature subset by running and training a model on the subset. They use the classification accuracy as a measure for evaluation of the selected feature subset. Filters are similar to Wrappers in the search approach, but instead of evaluating the model, use filter metrics including class separability, error probability, inter-class distance, probabilistic distance, entropy, consistency and correlation [29]. In wrapper methods, for each subset of feature, classification model (SVM, ANN and so on) must be trained and evaluate this subset using classification accuracy. Therefore, wrapper methods are computationally expensive and have a risk of overfitting to the model. However, filter methods do not need to train a system for each subset and feature subset evaluation is not dependent on classification model. Thus, filter methods have low complexity, low overfitting and better generalization in comparison to wrapper method. Meanwhile, wrapper methods usually have a higher accuracy than filter methods.

The basic components in filter methods are the feature search method and the feature selection criterion. Wrapper approaches consist of a search component, a learning algorithm and a feature evaluation criterion [30]. The most popular feature selection methods are Las Vegas filter [31], Las Vegas wrapper [32], Las Vegas incremental [33], Relief [34], sequential floating forward search [35], FOCUS genetic algorithm (GA) [36], simulated annealing (SA), particle swarm optimization (PSO) [37] and ant colony optimization (ACO) [9, 10, 30, 38].

3.1. Genetic Algorithm for Feature Selection

GA is a randomized heuristic search technique based on biological evolution strategies, introduced by Holland in 1975. GA is usually applied in complex optimization problems where candidate solutions are represented by individuals (or chromosomes) in a large population. Initial solutions usually generated randomly and next generations are created by individuals, among current ones, that have higher fitness, in each generation. If we have an individual that satisfies our constraint, the algorithm is stopped and this individual is a solution of the problem. GA has been applied to our problem of interest in several approaches, the most recent of which are feature selection and feature weighting. The purpose of feature selection is to find an optimal binary vector with the smallest number of 1s such that the classifier performance is maximized. Each bit corresponds to one feature where '1' or '0' means that the feature is selected or dropped respectively [39]. The second approach assigns numerical weights to features instead of binary select or drop [40]. GA has also recently been applied to feature selection in speaker recognition systems. Examples include application of genetic algorithm to HMM-based speaker verification for feature selection and weighting where optimal set of features created leading to decrease in EER [16]; application of genetic algorithm for feature selection in text-independent GMM-based speaker identification reducing the number of features to 24 and increasing the recognition rate by 5% [17]; genetic algorithm-based feature weighting used as an intermediate step towards feature selection and applied to speaker recognition by vector quantization finding an optimal set of weights for a 38-dimensional feature set [41].

3.2. Ant Colony Optimization for Feature Selection

Dorigo and colleagues introduced ACO in 1990 as an iterative, probabilistic meta-heuristic method for the solution of hard combinatorial optimization problems. ACO is a system based on agents, which simulate the natural behavior of ants, consisting of mechanisms of adaptation and cooperation [42]. The ability of ants to find shortest paths is achieved by their depositing of pheromone when they travel. Each ant probabilistically prefers to follow a direction that has more pheromone and the pheromone decays over time. Given that over time, the shortest paths will have more pheromone and higher chance for selection by ants. This path will be reinforced and the other paths' pheromone diminished until all ants follow the same path. In this way, the shortest path is achieved and system converges to a single solution [38]. The first application of ACO algorithm was the ant system (AS) [43] and then several improvements of the AS have been devised and applied to many applications [44, 45, 46].

ACO can be reformulated to solve feature selection problem. The main idea of ACO is to find a path with minimum cost in graph. Here, nodes in the graph represent features and the edges between nodes denote the choice of the next feature, i.e. selecting edges means that the corresponding feature is selected [47]. ACO starts to search for the optimal feature subset with the ant's traverse through the graph until a minimum number of nodes are visited and traversal stop criterion is satisfied. The graph is fully connected to allow any feature to be selected in the next stages. Based on this reformulation of the graph representation, the

Pheromone update rule and transition rule of standard ACO algorithm can be used. In this case, pheromone and heuristic value are not associated with edges. Instead, each feature has its own pheromone value and heuristic value because edges do not affect the optimum path but the features affect it. The probability that ant *k* selects feature *i* at time step *t* is:

$$P_i^k(t) = \begin{cases} \dfrac{|\tau_i(t)|^\gamma |\eta_i|^\delta}{\sum_{u\in J^k} |\tau_u(t)|^\gamma |\eta_u|^\delta} & if\ i \in J^k \\ 0 & otherwise \end{cases} \tag{13}$$

where J^k is the set of features that are allowed to be added to the partial solution if they are not visited so far. $\tau_i(t)$ and $\eta_i$are the pheromone value and heuristic desirability associated with feature i respectively. γ and δ are two parameters that determine the importance of the pheromone value and the heuristic information respectively [48]. After each ant has completed its tour, the solution is generated and then allowing each ant to deposit pheromone on the features that are part of its tour. The amount of pheromone deposited by ant k on feature *i* in step *t* is:

$$\Delta\tau_i{}^k(t) = \begin{cases} \phi.\mathrm{H}(S^k(t)) + \dfrac{\psi.(n - |S^k(t)|)}{n} & if\ i \in S^k(t) \\ 0 & otherwise \end{cases} \tag{14}$$

where $S^k(t)$ is a feature subset found by ant *k* at iteration *t* and $|S^k(t)|$ is its length. $\mathrm{H}(S^k(t))$ is the classifier performance of subset $S^k(t)$, n is the number of all features, ϕ and ψ are parameters that control importance of classifier performance and feature subset length respectively [18,11]. After all ants have completed their solutions, the pheromone trails are updated by the following relation:

$$\tau_i(t+1) = (1-\rho).\tau_i(t) + \sum_{k=1}^{m} \Delta\tau_i{}^k(t) + \Delta \tag{15}$$

where ρ is an evaporation rate constant, m is the number of ants and g is the best ant in previous iteration. This relationship means that all the ants can update the pheromone and the one with the best solution deposits additional pheromone on nodes. This causes the search of ants to stay around the optimal solution in next iterations [11].

4. Proposed System

4.1. Relief

Relief is a filter based feature weighting approach proposed by Kira and Rendell in 1992. This method is an individual evaluation method since it evaluates each feature independent of the other features and assigns the weight for each feature. Weights of features are interpreted as importance of features. The main idea of this algorithm is that if we have M instances, for each instance, from the random subset m ($\mathrm{m} \leq \mathrm{M}$), calculate the nearest instance from the same class (nearest hit $\mathrm{x_h}$) and the nearest instance from the opposite class (nearest miss $\mathrm{x_m}$). If the nearest hit is different from the selected sample in feature i, this is counted as an undesirable property of this feature. Hence, the weight of this feature is decreased. In contrast, if the nearest miss is different from the selected sample in feature i , it is counted as a desirable property of this feature. Hence, the weight of this feature is increased [30]. The update rule in each step is:

$$w[i] = w[i] - diff(i, x_k, x_h)/m + diff(i, x_k, x_m)/m \tag{16}$$

where x_k is a randomly selected instance and $diff(i, x_k, x_h)$ is a difference of feature values i of two instances $x_k \; and \; x_h$. This algorithm assigns a weight between 1 and -1 to each feature, where a higher weight means a high relevancy.

Relief was designed for two-class problems and is quite sensitive to noise. A more realistic variant of relief is its extension, called relieff that is more appropriate for noisy data and can be used for evaluating the feature quality in multi-class problems [30]. Relieff selects the instance randomly and finds m nearest instances from each class in each iteration instead of select one nearest miss and nearest hit. If labels of m instances are same as the label of the selected instance, they are nearest miss and otherwise they are nearest hit. The update rule for relieff in each step is as follows.

For nearest hits:

$$w[i] = w[i] - diff(i, x_k, x_h)/m * n \tag{17}$$

For nearest misses:

$$w[i] = w[i] + \frac{p_y}{1 - p_{yk}} * diff(i, x_k, x_h)/m * n \tag{18}$$

Where $\frac{p_y}{1-p_{yk}}$ is a percentage of class y to all data except class yk (the class of randomly selected instance).

4.2. Proposed Feature Selection Algorithm

In ASV systems, feature selection stage is placed after feature extraction stage. In feature extraction stage, speaker’s speech is converted to a set of feature vectors where each vector corresponds to one frame of speech. We applied our method to select the optimal feature subset for ASV systems using the following algorithm:

1. Apply k-means clustering to each speaker's data (codewords are used instead of all data of that speaker)
2. Initialize: set W[i]=0 for all features
3. For m iterations do 4-9
4. Randomly select a speaker (i) and randomly select a codeword from this speaker
5. From all speakers: find k nearest instances to the instances selected in stage 4 and save them in NEARESTS list
6. Select an instance from instances found in stage 5 and delete it from NEARESTS list
7. If the selected instance in stage 6 belongs to speaker i, update *w* for all features using (17)
8. Else, if the selected instance in stage 6 does not belong to speaker *i* update *w* for all features by (18)
9. If no instance remained in NEARESTS list go to 10 else go to 6
10. Sort all features by their weights and save in SORTFEATURES
11. Create *n* feature subsets: subset 1 consists of the feature with the highest weight (first element in SORTFEATURES); subset 2 consists of two features with highest weights (element 1 and 2 from SORTFEATURES)… subset n consists of all features (all elements in SORTFEATURES).
12. Train ASV system with all subsets and compute EER for each subset.
13. Compute evaluation function using (19) for all subsets created in 11
14. Find maximum evaluation function amounts computed in 13. The feature set that belongs to this state is an optimal feature set
15. Take optimal feature set as the input of ASV system.

In relieff, the number of instances is selected randomly for feature update. Since the number of all instances in each speaker is too high, we apply k-means clustering to each speaker data in order to compact it. In each cluster, the use of center of clusters instead of all data is very important and useful since all data may not be used for features update. K-means clustering is used, in order to have a good coverage of all data. After k-means clustering, all features are updated using the selected codeword from k-means. Finally, EER of the candidate subsets and their cardinalities are used to compute the evaluation function in each feature subset. This evaluation function is defined as:

$$E(FN, EER) = \alpha * exp^{-\frac{FN}{N}} + \beta * exp^{-EER} \tag{19}$$

where FN is the feature cardinality of the selected feature subset, N is the number of all features, α and β are the parameters that control the effect of feature size and EER respectively. For example, if β is much higher than α, this means that the algorithm will lead to a feature subset with lower EER.

Note that if we have N features, the number of all candidate feature subsets is N, since we sort features by their weights and each subset is created by adding the next feature with the highest weight from the remaining features to the previous one. Thus, the first subset consists of one feature with the highest weight, second subset contains the first two features with the highest weights … and the last subset consists of all features.

After computing the evaluation function for all candidate feature subsets, the feature subset with the maximum evaluation function is found and selected as the optimal feature subset. The block diagram of the proposed system is showed in Figure 2.

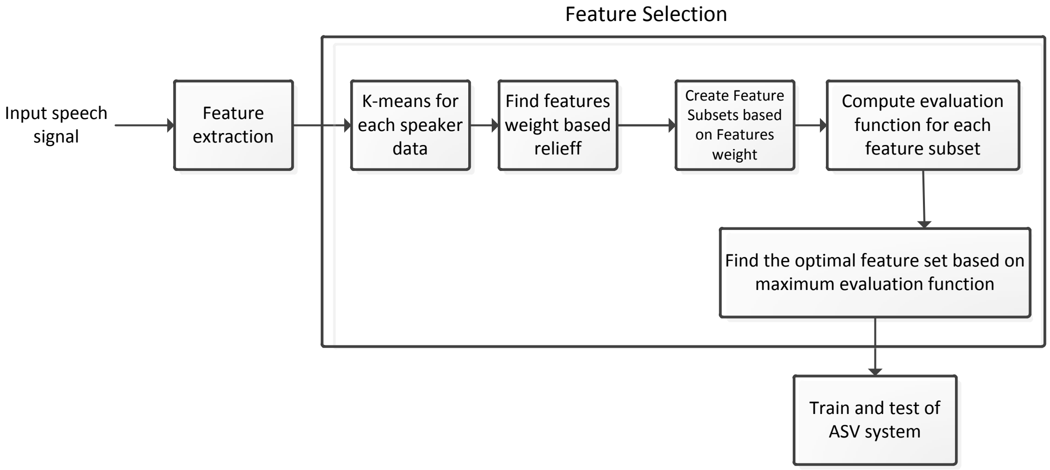

Feature Selection
Input speech signal
Feature extraction
K-means for each speaker data
Find features weight based relieff
Create Feature Subsets based on Features weight
Compute evaluation function for each feature subset
Find the optimal feature set based on maximum evaluation function
Train and test of ASV system

Figure 2. Block Diagram of the ASV System using the Proposed Feature Selection Method

5. Experimental Setup

To show the utility of the proposed feature selection algorithm a series of experiments is conducted. We implement our algorithm for feature selection and GA based feature selection on a machine with 2.26 GHz CPU and 4GB of RAM and windows 7. The following sections describe dataset, features and parameter settings used in this paper.

5.1. TIMIT Dataset

In this paper we have used TIMIT speech corpus [49]. TIMIT contains 6300 sentences spoken by 630 speakers, including 438 males and 192 females. It includes two sections: train and test. Both sections have eight dialect regions labeled from DR1 to DR8. Each speaker has uttered 10 sentences including 2 sentences labeled as SA, 5 sentences labeled as SX, 3 sentences labeled as SI and each sentence has an approximate duration of 3 seconds. The speech signals in TIMIT are recorded in a quiet environment with a sampling frequency of 16 kHz. In our experiments, we use 100 speakers including 72 males and 28 females randomly. For each speaker we use 6 sentences for training and 4 sentences for test. In addition, since the verification task needs test data from the main and imposter speakers, we randomly select sentences from other speakers as test data for imposter speakers.

5.2. Feature Vectors

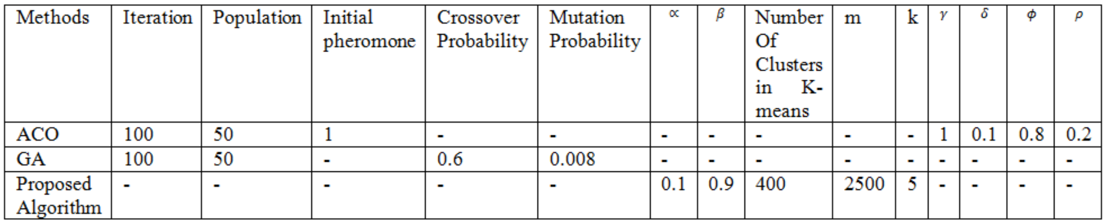
The first step in speech processing is to extract appropriate features from the available numerical data obtained from the speaker voice. A successful technique used to achieve this goal is Mel Frequency Cepstral analysis leading to feature components called MFCC [50, 51]. In our experiments, MFCCs are used. First, speech is pre-emphasized with a factor of 0.97

and then segmented into frames using a 20ms frame length at 10ms frame shift. A Hamming window is then applied to each frame and FFT converts the short-time time domain signal into its frequency domain representation. A set of 26 overlapped triangular filters uniformly spread over the mel frequency axis are then applied to the spectrum and discrete cosine transform (DCT) used to converted the log energy output of filters to a vector of 12 mel cepstral (MFCC) features. Frame log energy is also added to this vector the first and second time derivation of MFCC vector components found and appended to the vector. This leads to a feature vector per frame of length 39, used in our experiments.

5.3. Parameter Settings

In these experiments, various parameter values were tested for ACO, GA and the proposed algorithm in feature selection task. According to our experiments, the highest performance in each method is achieved by setting the parameters to values shown in Table 1.

Table 1. Parameter Settings for ACO, GA and the Proposed Algorithm



Methods	Iteration	Population	Initial pheromone	Crossover Probability	Mutation Probability	$\propto$	β	Number Of Clusters in K-means	m	k	γ	δ	ϕ	ρ
ACO	100	50	1	-	-	-	-	-	-	-	1	0.1	0.8	0.2
GA	100	50	-	0.6	0.008	-	-	-	-	-	-	-	-	-
Proposed Algorithm	-	-	-	-	-	0.1	0.9	400	2500	5	-	-	-	-

6. Experimental Results

In a speaker verification system, two types of errors can occur, namely false rejection error and false acceptance error, represented by P_{fa} *and* P_{fr} respectively. A false rejection error happens when a valid identity (speaker) claim is rejected and a false acceptance error consists of accepting an identity claim from an impostor speaker. Both types of error depend on the threshold T used in the decision making process. If the threshold is set to low values, the system tends to accept more identity claims and the false acceptance error will be high. On the contrary, if the threshold is set to high values, the system rejects more claims and false rejection error will be high [3]. Setting the threshold is a trade-off between the two types of errors to represent the performance of a system and it is shown by plotting P_{fr} as a function of P_{fa}. This curve is decreasing. Furthermore, it has become a standard to plot the error curve on a normal deviate scale, the curve is known as the detection error trade-offs (DETs) curve [52]. In our experiments, we use EER for evaluating system performance. In a DET curve, EER corresponds to the operating point where $P_{fa} = P_{fr}$and it corresponds to the intersection of the DET curve with the first bisector curve.

The main objective of all systems based on feature selection is the reduction of feature dimension without decreasing performance. In this paper, an ASV system with feature reduction based on relieff algorithm is proposed and implemented by SVM classifier. After applying k-means clustering to each speaker's data, optimal weight is found by the algorithm and evaluation functions computed for each candidate feature subset. In the proposed system with n features, we have n candidate feature subsets to be selected as the optimal subset. Figure 3 shows the EERs of various subsets in different kernels of SVM. The red marker is a subset that has an evaluation function with maximum value, thus this point is selected as an optimal feature subset.

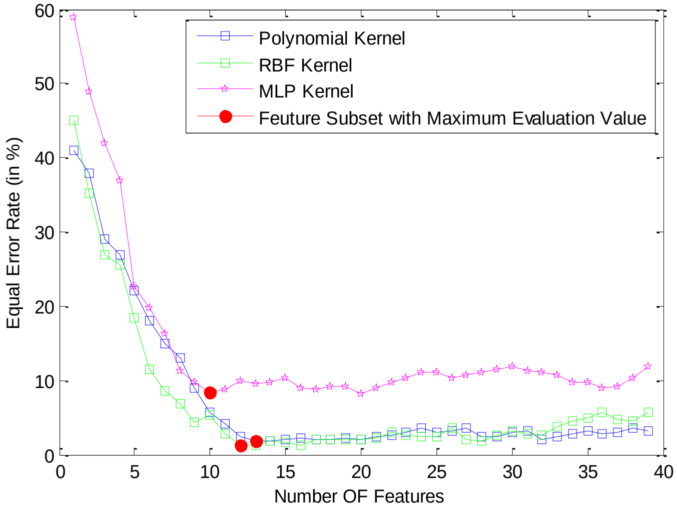


Figure 3. Feature Subsets and their EERs Obtained by Relief

The DET curves for the selected points in Figure 3 (red points) are shown in Figures 4, 5 and 6. These figures show the performances of ACO-based, GA-based and the proposed system based on relieff. Also, Figures 4, 5 and 6 represent the DET curves for the proposed method with the SVM using radial basis function (RBF), multi-layer perceptron (MLP) and polynomial kernels respectively. As seen, the proposed system has a considerably lower EER in comparison to the other systems, especially in its best performance, which is in the RBF kernel case.

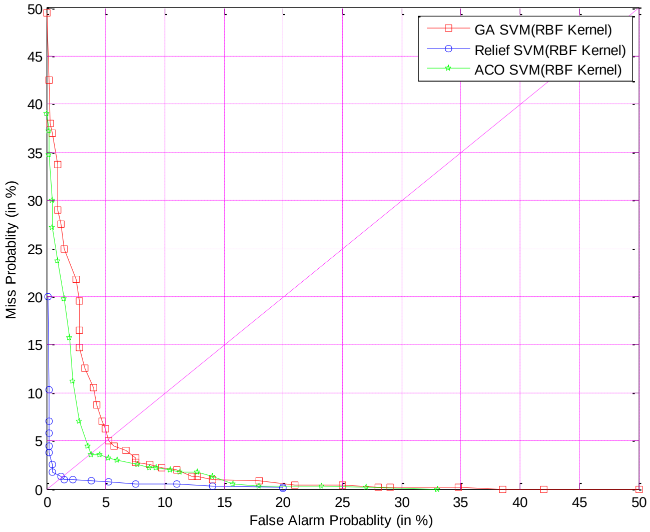


Figure 4. DET Curves for ACO-SVM, GA-SVM and Relief_SVM with RBF Kernel used in SVM

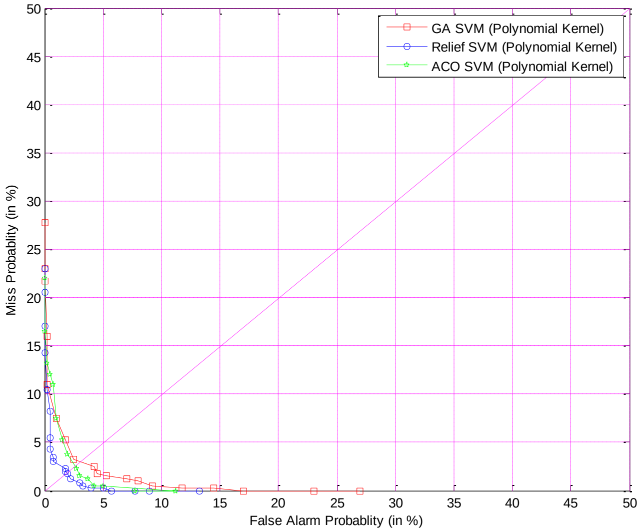


Figure 5. DET Curves for ACO-SVM, GA-SVM and Relief_SVM with Polynomial Kernel used in SVM

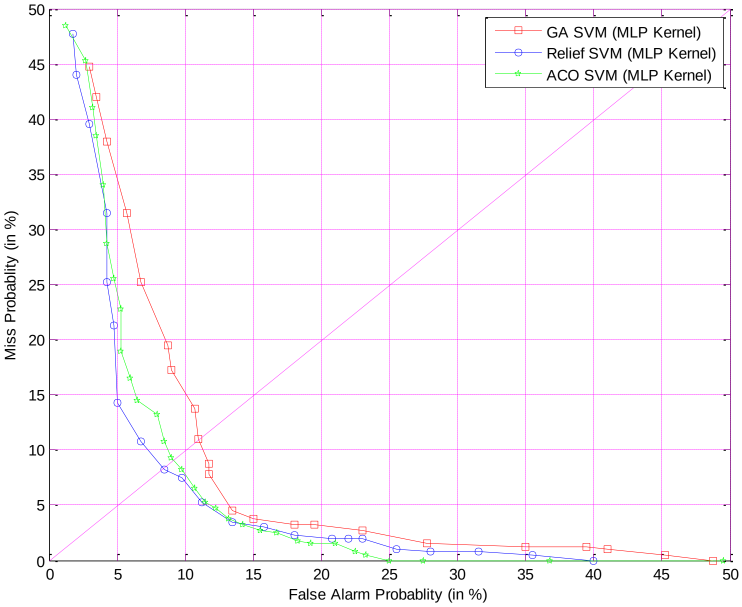
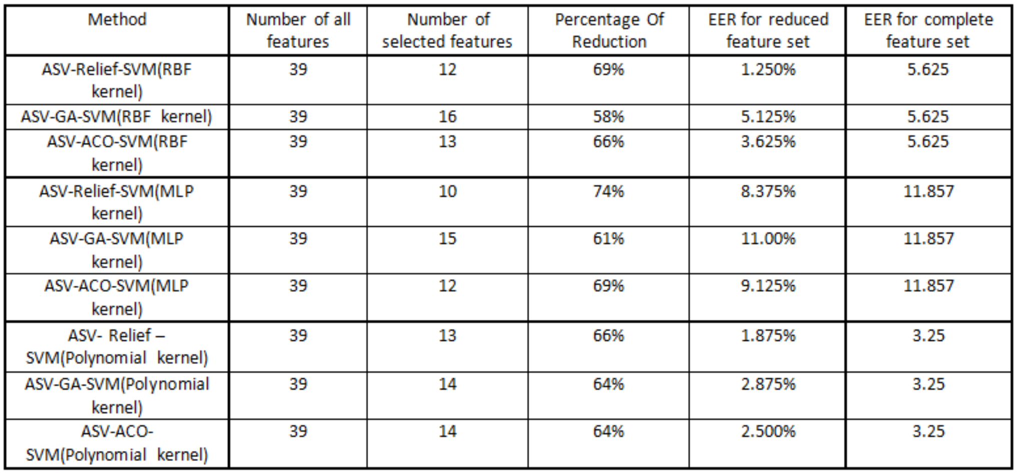


Figure 6. DET Curves for ACO-SVM, GA-SVM and Relief_SVM with MLP Kernel used in SVM

Note that all the mentioned methods reduce the feature vector size. Table 2 shows the EERs and the number of selected features by ACO, GA and proposed algorithm. This table shows that the EER and number of selected features by relieff are lower than other methods in all kernels of SVM.

Table 2. Equal Error Rates, the Number of Reduced Features and Percentage of Reduction in each Method for Different Kernels



Method	Number of all features	Number of selected features	Percentage Of Reduction	EER for reduced feature set	EER for complete feature set
ASV-Relief-SVM(RBF kernel)	39	12	69%	1.250%	5.625
ASV-GA-SVM(RBF kernel)	39	16	58%	5.125%	5.625
ASV-ACO-SVM(RBF kernel)	39	13	66%	3.625%	5.625
ASV-Relief-SVM(MLP kernel)	39	10	74%	8.375%	11.857
ASV-GA-SVM(MLP kernel)	39	15	61%	11.00%	11.857
ASV-ACO-SVM(MLP kernel)	39	12	69%	9.125%	11.857
ASV- Relief – SVM(Polynomial kernel)	39	13	66%	1.875%	3.25
ASV-GA-SVM(Polynomial kernel)	39	14	64%	2.875%	3.25
ASV-ACO-SVM(Polynomial kernel)	39	14	64%	2.500%	3.25

7. Complexity of the Proposed Algorithm

To find out the effectiveness of the proposed algorithm its complexity should be compared with other methods. Generally, feature selection based wrapper methods are time consuming tasks and involve high computational cost since the performance of a learning algorithm is used to evaluate the goodness of selected feature subsets. In these experiments, in order to compute the system performance, SVM hyper-planes must be trained and estimated for each speaker in each step which is a high complexity task. Thus, in each wrapper-based feature selection task, the number of times that classification performance is used is very important and affects the complexity of algorithm. To solve this problem, we added a filter method, relieff, to the algorithm and assigned weights to the features in this step. If the number of features is *n*, the number of feature subsets would be *n* too and the algorithm evaluates the classification performance *n* times. Since there is'nt any rule and equation for evaluate the evaluation algorithm such as GA and ACO but we consider the maximum number of iteration that viewd in this experiment. In GA, if the number of features is *n*, the number of generations GenNum and the number of individuals IndNum, the maximum number of times that classification performance is used in feature selection algorithm would be GenNum*IndNum, because in each generation, classification performance should be computed for all individuals. In ACO, if the number of iteration is IterNum, the number of ants AntNum and the number of features *n*, in each iteration, each ant at first step selects one feature randomly from *n* features and then at the second step selects second best feature from the remaining *n-1* features based on its classification performance and its pheromone. At the third step, it selects the third feature among *n*-2 remaining features and so on. Thus, In worst case, each ant in each iteration needs a $(n-1)+(n-2)+\cdots+1=\frac{n(n-1)}{2}$ evaluations to find its path from the start point to the final point. However, some thresholds are defined, such as the one for EER and the max number of times allowed for search per ant, that prevent the ants to explore all features in each iteration. The best case is when each and every ant explores just one feature to achieve its threshold and its order is 1. The worst case is that each and every ant explores $\frac{n(n-1)}{2}$ steps before reaching the threshold. Finally the average case is when each ant explores half of the features before reaching the threshold and order of this case is $(n-1)+(n-2)+\cdots+\frac{n}{2}=\frac{n(3n-2)}{8}$. Thus the maximum number of iteration for all ants in average case is

$\mathrm{GenNum} \times \mathrm{AntNum} \times \frac{n(3n-2)}{8}$. These relationships show that the iteration number of the proposed algorithm is much lower than both GA and ACO. Finally, it is clear that GA has a lower complexity order than ACO in feature selection task but ACO has a better result in comparison to GA (Table 2).

8. Conclusion and Future Work

In this paper, we proposed a feature selection approach for SVM-based automatic speaker verification. We used MFCC and energy features with first and second derivations as speech feature vectors. In the proposed approach, we used relieff algorithm to assign weights to features. Features weights were then used to construct feature subsets and EERs feature subsets used in the evaluation function. The algorithm uses the intrinsic properties of speaker data to compute feature weights and uses EERs of feature subsets to find the optimal one. Thus, this approach is a mixture of filter and wrapper methods for feature selection. This approach is compared with two popular population-based wrapper feature selection methods, namely ACO and GA. Results show that the proposed approach leads to lower EER and lower computational overhead as well as finding a shorter feature subset than the other methods.

For future work, the performance of the proposed approach can be evaluated by taking into account other classifiers such as GMM, in ASV systems. Other feature selection methods can be improved and applied to ASV systems. In addition, intrinsic property of data such as relieff weights can be used in swarm intelligence techniques such as ACO, GA and Particle swarm optimization (PSO) algorithms in ASV systems to converge more quickly. Finally, feature selection with filter method-based relieff algorithm can be proposed in ASV systems to achieve lower complexity systems.

References

[1] S. Nemati, R. Boostani and M. D. Jazi, “A novel text-independent speaker verification system using ant colony optimization algorithm”, In ICISP2008 LNCS, vol. 5099, Berlin, Heidelberg: Springer-Verlag, **(2008)**, pp. 421–429.

[2] D. A. Reynolds, “An Overview of Automatic Speaker Recognition Technology”, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) vol. 4, **(2002)**, pp. 4072-4075.

[3] F. Bimbot, J. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garc, D. Petrovska-Delacretaz and D. A. Reynolds, “A Tutorial on Text-Independent Speaker Verification”, EURASIP Journal on Applied Signal Processing, vol. 4, **(2004)**, pp. 430–451.

[4] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to super vectors”, Speech Communication, vol. 52, **(2010)**, pp. 12-40.

[5] S. Davis and P. Mermelstein, “Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences”, IEEE Trans on Acoustics, Speech, Signal Process, vol. 28, **(1980)**, pp. 357–366.

[6] L. R. Rabiner and R. W. Schafer, “Introduction to Digital Speech Processing”, Foundations and Trends in Signal Processing, **(2007)**, (chapter 6).

[7] H. Hermansky, “Perceptual linear prediction (PLP) analysis for speech”, The Journal of the Acoustical Society of America, vol. 87, **(1990)**, pp. 1738–1752.

[8] S. Gundimada, V. K. Asari and N. Gudur, “Face recognition in multi-sensor images based on a novel modular feature selection technique”, Information Fusion, vol. 11, **(2010)**, pp. 124-132.

[9] H. R. Kanan, K. Faez and M. Hosseinzadeh, “Face recognition system using ant colony optimization-based selected features”, In Proceedings of the first IEEE symposium on computational intelligence in security and defense applications, CISDA **(2007)**, pp. 57–62, USA: IEEE Press.

[10] R. Jensen, “Combining rough and fuzzy sets for feature selection”, Ph.D. thesis, University of Edinburgh **(2005)**.

[11] S. Nemati, M. Basiri, N. Ghasem-Aghaee and M. HosseinzadehAghdam, “A novel ACO–GA hybrid algorithm for feature selectionin protein function prediction”, Expert Systems with Applications, vol. 36, **(2009)**, pp. 12086–12094.

[12] M. Pandit and J. Kittkr, “Feature selection for a DTW-based speaker verification system”, Acoustics, Speech and Signal Processing, vol. 2, **(1998)**, pp. 769-772.

[13] A. Cohen and Y. Zigel, “On feature selection for speaker verification”, In Proceedings of COST 275 workshop on the advent of biometrics on the Internet, **(2002)**.

[14] T. Ganchev, P. Zervas, N. Fakotakis and G. Kokkinakis, “Benchmarking feature selection techniques on the speaker verification task”, In Fifth international symposium on communication systems, networks and digital signal processing, CSNDSP’06, **(2006)**, pp. 314–318.

[15] M. Zamalloa, G. Bordel, L. Rodriguez and M. Penagarikano, “Feature Selection Based on Genetic Algorithms for Speaker Recognition”, Speaker and Language Recognition Workshop, **(2006)**, IEEE Odyssey, pp. 557–564.

[16] D. Charlet and D. Jouvet, “Optimizing feature set for speaker verification”, Pattern Recognition Letters, vol. 18, **(1997)**, pp. 873-879.

[17] M. Demirekler and A. Haydar, “Feature Selection Using a Genetics-Based Algorithm and its Application to Speaker Identification”, Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 1, **(1999)**, Proceedings, pp. 329-332.

[18] S. Nemati and M. Basiri, “Text-independent speaker verification using ant colony optimization-basedselected features”, Expert Systems with Applications, vol. 38, **(2011)**, pp. 620–630.

[19] M. E. Basiri, N. Ghasem-Aghaee and M. H. Aghdam, “Using ant colony optimization-based selected features for predicting post-synaptic activity in proteins”, EvoLNCS, LNCS vol. 4973, Heidelberg: Springer-Verlag, **(2008)**, pp. 12–23.

[20] L. Cheung-chi, “GMM-based speaker recognition for mobile embedded systems”, Ph.D. thesis, University of Hong Kong, **(2004)**.

[21] J. Holmes and W. Holmes, “Speech Synthesis and Recognition”, 2nd ed., **(2001)**, (chapter 10).

[22] V. Vapnik, “The Nature of Statistical Learning Theory”, 2nd ed., Springer-Verlag, London, UK, **(1995)** (Chapter 5).

[23] V. Vapnik, “Statistical Learning Theory”, John Wiley & Sons, NewYork, N.Y, **(1998)**, (Chapter 12).

[24] C. J. C. Burges, “Geometry and Invariance in Kernel Based Method”, Advance in Kernel Method-Support Vector Learning, MIT Press, Cambridge, MA, **(1999)**, pp. 86–116.

[25] S. Katagiri and S. Abe, “Incremental training of support vector machines using hyper spheres”, Pattern Recognition Letters, vol. 27, **(2006)**, pp. 1495–1507.

[26] R. Andrzej, “Nonlinear Optimization, Princeton University Press, **(2006)** (Chapter 4).

[27] L. Molina, L. Belanche and A. Nebot, “Feature Selection Algorithms: A Survey and Experimental Evaluation”, IEEE International Conference, **(2002)**, pp. 306-313.

[28] D. Mladenic ́, “Feature selection for dimensionality reduction, Subspace, latent structure and feature selection, statistical and optimization, perspectives workshop”, Lecture Notes in Computer Science, vol. 3940, Springer. **(2006)**, pp. 84–102.

[29] H. C. Peng, F. Long and C. Ding, “Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, **(2005)**, pp. 1226–1238.

[30] H. Liu and H.Motoda, “Computational methods of feature selection”, Ebook, **(2008)** (chapter 9).

[31] H. Liu and R. Setiono, “A Probabilistic Approach to Feature Selection: a Filter Solution”, The 13th International Conference on Machine Learning, **(1996)**, pp. 319–327.

[32] H. Liu and R. Setiono, “Feature Selection and Classification: a Probabilistic Wrapper Approach”, The 9th International Conference on Industrial and Engineering Applications of AI and ES, **(1996)**, pp. 129–135.

[33] H. Liu and R. Setiono, “Scalable Feature Selection for Large Sized Databases”, The 4th World Congress on Expert System, **(1998)**, pp. 68–75.

[34] K. Kira and L. A. Rendell, “A Practical Approach to Feature Selection”, The 9th International Conference on Machine Learning, **(1992)**, pp. 249–256.

[35] P. Pudil, J. Novovicová and J. Kittler, “Floating Search Methods in Feature Selection”, Pattern Recognition Letters, vol. 15, no. 11, **(1994)**, pp. 1119–1125.

[36] W. Siedlecki and J. Sklansky, “A note on genetic algorithms for large-scale feature selection”, Pattern Recognition Letters, vol. 10, no. 5, **(1989)**, pp. 335–347.

[37] X. Wang, J. Yang, X. Teng, W. Xia and R. Jensen, “Feature selection based on rough sets and particle swarm optimization”, Pattern Recognition Letters, vol. 28, no. 4, **(2007)**, pp. 459–471.

[38] A. A. Ani, “Ant colony optimization for feature subset selection”, Transaction on Engineering, Computing and Technology, vol. 4, **(2005)**, pp. 3–35.

[39] J. Yang and V. Honavar, “Feature subset selection using a genetic algorithm”, IEEE Intelligent Systems, vol. 13, **(1998)**, pp. 44–49.

[40] W. F. Punch, E. D. Goodman, L. C. S. M. Pei, P. Hovland and R. Enbody, “Further research on feature selection and classification using genetic algorithms”, In Proceedings international conference on genetic algorithms, **(1993)**, pp.557–564.

[41] M. Zamalloa, G. Bordel, L. Rodriguez and M. Penagarikano, “Feature Selection Based on Genetic Algorithms for Speaker Recognition”, Speaker and Language Recognition Workshop, **(2006)** IEEE, pp. 1 – 8.

[42] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey”, Theoretical Computer Science, **(2005)**, pp. 243–278.

[43] M. Dorigo and G. D. Caro, “Ant colony optimization: A new meta-heuristic”, In Proceedings of the congress on evolutionary computing, **(1999)**.

[44] L. M. Gambardella and M. Dorigo, “Ant-Q: A reinforcement learning approach to the TSP”, In Proceedings of the twelfth international conference on machine Learning, **(1995)**, pp. 252–260.

[45] L. M. Gambardella and M. Dorigo, “Solving symmetric and asymmetric TSPs by ant colonies”, In Proceedings of IEEE international conference on evolutionary computation, **(1996)**, pp. 622–627.

[46] T. Stützle and H. H. Hoos, “MAX–MIN ant system and local search for the traveling salesman problem”, In Proceedings of IEEE international conference on evolutionary computation, **(1997)**, pp. 309–314.

[47] C. L. Huang, “ACO-based hybrid classification system with feature subset selection and model parameters optimization”, Neurocomputing, vol. 73, **(2009)**, pp. 438–448.

[48] H. RashidyKanan and K. Faez, “An improved feature selection method based on ant colony optimization (ACO) evaluated on face recognition system”, Applied Mathematics and Computation, vol. 205, **(2008)**, pp. 716–725.

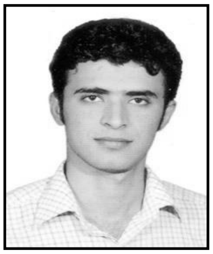
[49] J. Garofolo, *et al.*, “DARPATIMIT acoustic–phonetic continuous speech corpus CDROM”, National Institute of Standards and Technology, **(1990)**.

[50] S. Furui, “Cepstral Analysis Technique for Automatic Speaker Verification”, IEEE Transactions on Acoustic, Speech and Signal Processing, vol. 29, **(1981)**, pp. 254-272.

[51] M. Seltzer, “SPHINX III Signal Front End Specification”, CMU Speech Group, **(1999)**.

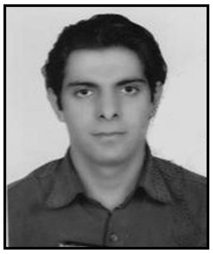
[52] A. Martin, G. Doddington, T. Kamm, M. Ordowski and Przybocki, “The DET curve in assessment of detection task performance”, European Conference on Speech Communication and Technology, (Eurospeech ’97), vol. 4, **(1997)**, pp. 1895–1898.

Authors



Abdolreza Rashno

Graduated from TMU University, Tehran, Iran, post-graduation from Shahid Chamran University, Ahvaz, Iran. He has been working at PNU University of Alashtar, Lorestan, Iran since 2 years as an instructor in the Department of Computer Engineering, research interest includes Signal Processing, Image and Video processing, Multimedia Systems Design.



Hossein SadeghianNejad

Graduated from TMU University, Tehran, Iran, post-graduation from Yasuj University, Yasuj, Iran. He has been working at Applied Science and Technology University of Yasuj, Iran as an instructor in the Department of Computer Engineering, research interest includes Signal Processing , Image and Video processing, Multimedia Systems Design

Abed Heshmati

Graduated from TMU University, Tehran, Iran, post-graduation from Payame Noor University, Hamedan, Iran. He has been working at Payame Noor University of Iran as an instructor in the Department of Computer Engineering and Information Technology, research interest includes Signal Processing , Image and Video processing, Multimedia Systems Design.