Journal of Advances in Computer Research
Quarterly pISSN: 2345-606x eISSN: 2345-6078
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 5, No. 4, November 2014), Pages: 67-78
www.jacr.iausari.ac.ir

Presenting a New Text-Independent Speaker Verification System Based on Multi Model GMM

Mohammad Mosleh[1]✉, Faraz Forootan[2], Najmeh Hosseinpour[3]

1)Young Researchers and Elite club, Dezfoul Branch ,Islamic Azad University, Dezfoul, Iran
2) Department of Computer Engineering, Dezfoul Branch, Islamic Azad University, Dezfoul, Iran
3) Young Researchers and Elite club, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran
mosleh@iaud.ac.ir; forootan@iaud.ac.ir; hosseinpour@dums.ac.ir

Received: 2014/02/14; Accepted: 2014/04/20

Abstract

Speaker verification is the process of accepting or rejecting claimed identity in terms of its sound features. A speaker verification system can be used for numerous security systems, including bank account accessing, getting to security points, criminology and etc. When a speaker verification system wants to check the identity of individuals remotely, it confronts problems such as noise effect on speech signal and also identity falsification with speech synthesis. In this system, we have proposed a new speaker verification system based on Multi Model GMM, called SV-MMGMM, in which all speakers are divided into seven different age groups, and then an isolated GMM model for each group is created; instead of one model for all speakers. In order to evaluate, the proposed method has been compared with several speaker verification systems based on Naive, SVM, Random Forest, Ensemble and basic GMM. Experimental results show that the proposed method has so better efficiency than others.

Keywords: *biometric attributes, speaker verification, Gaussian Mixture Model (GMM), Support Vector Machine (SVM), Decision Trees (DT), Ensemble Classifiers*

1. Introduction

Nowadays, the importance of biometric attributes as human identity is no secret for anyone. Almost, all modern security systems based on one or more biometric attributes can be designed and built. Some attributes such as fingerprint, face, iris and speech are considered as the most famous and important parameters. Because of the simple implementation, lower hardware costs as well as the ability to execute real-time, the biometric systems based on speech have a special place. Speaker Recognition is referred to the process of identifying a specific speaker (*Speaker Identification*) or an authenticated identity claim (*Speaker Verification*), among a group of speakers[1]. Speaker recognition has many practical applications. For example, controlled access to secured services and locations, including bank accounts, restricted databases and buildings can be mentioned. In general, there are two main categories in speaker recognition: *text-dependent*—which requires the speaker to read a set of pre-defined keywords or sentences having the same text, *text-independent*— this method does not rely on a specific text being spoken. In this paper, we pay attention to the text-independent speaker verification systems. In general, any speaker verification system

has two main phases including feature extraction and classification. The feature extraction process generates a set of characteristic parameters of a signal that can be used for classifying the signal. The commonly used method in signal processing is Fourier transform (FT) which decomposes the signal into its frequency components[2]. One disadvantage of FT is that it only has frequency resolution and without time resolution. It is not a good method for analyzing non-stationary and non-periodic signals, such as speech signals. Wavelets are another approach to overcome to FT difficulty[3]. Linear Prediction Coding (LPC) is an alternative spectrum estimation method to FT that has very good intuitive interpretation both time and frequency domains [4]. One of the well-known features to make parameterized speech signal is Mel Frequency Cepestral Coefficients (MFCC)[4, 5]. Since such features are inspired from the human auditory system, they can be applied well for speech recognition applications. In [6], S.Nemati and M.E Basiri used an optimized subset of MFCC and LPC coefficients for speaker verification problem via ACO optimization algorithm.

Signal classification is another stage in speaker verification systems. In[7], a classifier based on Vector Quantization (VQ) was applied for text-independent speaker verification system. Support Vector Machine (SVM) is alternative approach which has properly been used in the pattern recognition fields, well [8]. Since SVM has a good generalization capability; it has been utilized in both speech and speaker recognition [9-11]. At 2010, M.A Lacerda et al. offered a Radial Basis Function (RBF) classifier, which is a particular type of Artificial Neural Network (ANN) for speaker verification[12]. In [13], Hidden Markov Model (HMM) is discussed for text-dependent speaker verification. One of the best statistical classifiers is Gaussian Mixture Model (GMM) which has high ability for modeling the dynamic patterns[14]. In [15], P.Kenny et al. presented a corpus-based approach for speaker verification, in which, maximum likelihood II criteria are used to train a large scale generative model of speaker and session variability. A.Larcher et al. in 2013, proposed a speaker recognition engine based on GMM for mobile devices[16]. Also, in [17], a speaker verification system based on training of the GMM with diagonal covariances- under a large margin criterion- has been proposed. Due to high potential of GMM method, it is applied in combination with many other methods. C. H You et al. proposed an speaker verification system in terms of fusion SVM-GMM[18]. Also, an SV system based on ANN-GMM was presented by B. Xiang and T. Berger[19].

In this paper, by inspiring from advantages and avoiding from deficiencies of earlier methods, we try to propose an efficient method which is able to perform speaker verification operation well. For this purpose, first, we divide all speakers into seven groups, in terms of their age category. Then, by following the preprocessing and feature extraction operations, a GMM model is made for each group. Finally, speaker verification operation is performed by voting.

The paper organization is as following: in Sec. 2, some types of utilized classification methods are presented. The SV system based on the proposed method is introduced in Sec. 3. In Sec. 4, experimental results are given and finally, concluding statements are presented in Sec. 5.

2. Background

In this section, the Gaussian Mixture Model, Support Vector Machine, Decision Trees and Ensemble Classifiers are introduced.

2.1. Gaussian Mixture Model

Gaussian Mixture Model was widely used for problems like classification, especially for speaker modeling in text independent speaker verification systems [14]. Gaussian Mixture Models are linear mixtures of multivariate Gaussian density sequences, generally used for estimating the complicated probable density sequence. Fig.1 shows the GMM model structure. Since GMM can estimate density distribution, its precision and accuracy are essential and important.

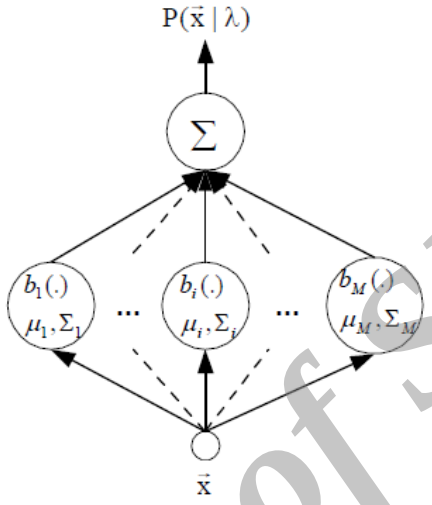

Figure 1. The structure of a GMM model

A Gaussian mixed density, according to Eq. 1, can be written as the linear addition of *M* parameters of Gaussian density.

$$P\left(\vec{x} \mid \lambda\right) = \sum_{i=1}^{M} P_i b_i\left(\vec{x}\right) \tag{1}$$

where $\vec{\mathbf{x}}$ is a next *D*-dimension random vector, $p_i\ (i=1,...,M)$ are mixed weights, and $b_i\ (i=1,...,M)$ are density parameters. Each density parameter is a Gaussian function which is written below:

$$b_i\left(\vec{x}\right) = \frac{1}{(2\pi)^{D/2}\ |\ \Sigma i\ |^{1/2}} exp\left[\frac{-1}{2}\left(\vec{x} - \overrightarrow{\mu_i}\right)' \sum_{i}^{-1}\left(\vec{x} - \overrightarrow{\mu_i}\right)\right] \tag{2}$$

Gaussian Mixture Model can be defined by means of vectors, covariance matrices, and mixed densities of all density functions, as the following:

$$\lambda = \left\{ p_i, \vec{\mu}_i, \sum_i \right\} \quad for \quad i = 1, 2, ..., M \tag{3}$$

The mixed weights of p_i should meet the following conditions:

$$\sum_{i=1}^{M} p_i = 1 \tag{4}$$

In recognizing speaker's identity, each speaker is shown by a GMM, related to a man or woman model, λ.

2.2. Support Vector Machine

Support Vector Machine is a powerful binary classifier which has attracted attentions in recent years [20]. By using an optimized algorithm, this classifier acquires samples which form class borders. These samplings are called *support vectors*. In the other words, in this method, a number of teaching points, closest to decision making, can be taken as a subset for defining the decision making borders. These are considered as support vectors. In Figure 2, two classes and their support vectors are shown.

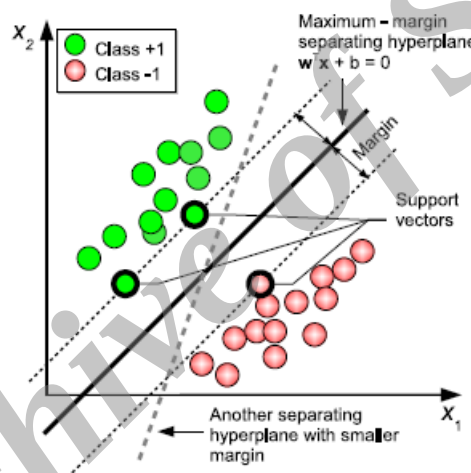

Figure 2. Support vectors of the two classes (distance between classes, support vectors)

Assume that we have *n* teaching samples, $\{\langle x_i \rangle, \langle \mathbf{y_i} \rangle\}$, in which $\langle x_i \rangle = (x_{i1}, x_{i2}, ..., x_{il})$ is feature vector of *m* dimension and $\langle y_i \rangle \in \{-1, +1\}$ is the related label of x_i. The objective of the classifier is finding an optimized hyper plane which can separate these two classes. A hyper plane in feature space can be defined as follows:

$$w.x + b = 0 \quad w \in R^m \tag{5}$$

where *w* is an m-dimension vertical vector on the separating plane, $|b|/||W||$ is the vertical distance between hyper plane and orientations origin, $\|W\|$ is Gaussian norm, and *b* is the bias amount. Assume that $d_+(d_-)$ is the shortest distance between the separating hyper planes and positive (negative) samples. We denote the hyper plane distance as $d_+ + d_-$. The hyper plane is calculated in a way that can maximize the

shortest distance of the teaching samples between the two classes. This hyper plane is called optimized hyper plane and should meet the following conditions:

$$w.x_i + b \geq +1 \quad for \ \ y_i = +1 \tag{6}$$

$$w.x_i + b \leq -1 \quad for \ \ y_i = -1 \tag{7}$$

The above conditions can be summarized in the following relation:

$$y_i (w.x_i + b) \geq 1 \quad \forall i \tag{8}$$

Now, we take the points for which their equality versus the non-equation (6) is true. These points are located on the hyper plane $H_1 : x_i.w+b=1$. The vertical distance between this hyper plane and the orientations origin equals to $|b|/||W||$. Similarly, the points for which the Eq. (7) is true are located on the hyper page $H_2 : w.x+b=-1$. The vertical distance of this hyper plane to the origin equals to $|1-b|/\|w\|$ and the distance of hyper page, equals to $2/\|w\|$. The hyper planes H_1 and H_2 are parallel with each other and no teaching sample is located between them. Then, we can find a couple of hyper planes which can increase the distance, by minimizing $||w||$ to its maximum. The above mentioned way can be summarized as follows:

$$min \frac{1}{2} |w|^2, \quad y_i (w.x_i + b) - 1 \geq 0 \quad i = 1,...,L \tag{9}$$

Solving the optimization problem in Eq. (8) is difficult. To simplify the problem by using unspecified Lagrange coefficients, it can be changed to the following: (λ_i are Lagrange coefficients)

$$\begin{aligned} &\max_{\lambda_1,...,\lambda_L} \left[-\frac{1}{2} \sum_{i=1}^{L} \sum_{j=1}^{L} \lambda_i y_i (x_i . x_j) y_j \lambda_j + \sum_{i=1}^{L} \lambda_i \right] \\ &\sum_{i=1}^{L} \lambda_i y_i = 0 \end{aligned} \tag{10}$$

After solving the above optimization problem and finding Lagrange coefficients, *w* is calculated by the following formula:

$$w = \sum_{i=1}^{L} \lambda_i y_i x_i \tag{11}$$

Each of the Lagrange coefficients is similar to one teaching sample. Those teaching samples with Lagrange coefficients larger than zero are called support vectors and are located on hyper plane H_1 or H_2. To find the decision border, all teaching samples aren't necessary, but we need just a limited number of them (i.e. support vectors). Having

found *w* by using the following relation, *b* is calculated versus different support vectors. Also, the final *b* is calculated by computing the mean value by adding up obtained *bs*.

$$\lambda_i [y_i (w.x_i + b) - 1] = 0 \quad i = 1, \ldots, L \tag{12}$$

The final classifier is obtained through the following formula:

$$f(x,w,b) = sgn(w.x + b) \tag{13}$$

2.3. Decision Trees

One of the efficient ways for classifying data is creating a decision tree. It is included in the most famous algorithms of deductive learning, which has been successfully used for different applications. It operates in a way in which the samples, from the root, grow downwards and finally reach the leaves knots. This feature poses a question related to the input example. In each internal knot, there are as many branches as answers to this question. Each one of the leaves on this tree indicates a class or a group. The reason for its naming as 'decision tree' is that the tree shows the decision making process for determining a group of input examples [21]. An educational example in the decision tree is classified as the following: it starts with the root; then the specified feature is tested by this knot and according to the feature in the example, it moves downwards along the branches. The process is repeated for the knots below the tree.

Decision trees are applied where they can be presented in a way in which they can offer a single response. This response can be considered as the name of a group or a class. They can be used in cases where the objective function possesses an output with inconsistent values. For example, we can use it in a question with 'yes' or 'no' response. A decision tree has the following features: It can be used for approximating inconsistent functions (classification). It is resistant to the noise of entered data. It is used for data of high volume and then is used in data detection.

We can use the tree as 'if-then' rules which can be easily understood. It can also be used in cases in which the examples lack all features. Most learning algorithms in decision tree act based on an avid top-down searching process in the available space of the trees. Its basic algorithm is named Concept Learning System (CLS) which was introduced in 1950. Then it was more comprehensively presented by Ross Quinlan in 1986, under the title Inducing Decision Trees (IDS). Later, a more complete algorithm was presented under the title C4.5 which removed some of the ID3 deficiencies. A practiced algorithm in cloning of this article is the algorithm C4.5 which is used for creating a decision tree to do the classification.

2.4. Ensemble Classifier and its Methods

Generally, in algorithms of monitoring training, the searching is done in an imaginative space to find a solution for a special problem. An ensemble classifier is a training monitored algorithm which combines different hypotheses to make a better one. Then, the ensemble classifier combines weak learners to create a strong learner. Fast algorithms, like decision trees, are also applied with ensemble classifiers. Observations show that various ensemble classifiers operate more efficiently. Then, different methods are proposed to make variation in the combining models. The famous method which can be mentioned is Bagging and Boosting[22]. In the Bagging method, the classifiers designed on different versions of data are combined together, and majority voting is performed among a single classifier decisions. This method is called *Bootstrap*

ensemble or *Bagging*, for short. Random Forest is one of the classifiers which use Bagging method. It contains several decision trees and its output is obtained through individual trees. To create a group of different decision trees which can be controlled, this method combines Bagging method with features, randomly. The high precision of the classifier is one of its advantages, while it can also work with a lot of inputs [23]. The second famous method, Boosting, can teach new samples to enhance training samples and then create changes in the ensemble classifier. This method is more precise in some cases, compared with the Bagging method. A problem with Boosting is the long training phase of training for samples. AdaBoost is one of the most famous methods of Boosting.

3. The proposed method

As already mentioned, up to now, many methods for speaker verification have been presented. The main purpose of them is increasing speaker verification accuracy. By studying these methods, it is inferred that one of the significant problems is applying a single model for all speakers. In this paper, a multi model speaker verification system in terms of age category is proposed. This system is able to increase verification accuracy in comparison with other methods. It should be mentioned that the GMM model is applied as basic classifier. Therefore, it is called speaker verification system based on Multi Model GMM (SV-MMGMM). In the following, the training and testing processes of the proposed system will be discussed in detail.

3.1. Training process

The training pseudo code of the proposed SV-MMGMM is shown in Figure 3. As can be seen, in the learning stage, instead of creating one reference model for all speakers, they are divided into seven different age categories, including less than 20 years, 20 to 30 years, 31 to 40 years, 41 to 50 years, 51 to 60 years, 61 to 70 years and finally more than 70 years. Then for speakers in each age category, one Gaussian Mixture Model (GMM) is created and saved.

```
Procedure SV-MMGMM Train (Inputs: Speakers' audio signals; Outputs: GMM_Models)
Begin
        Divide all speakers into seven separate age groups G1, G2…G7;
        GMM_Models= [];
        Matrix_Featurs=[];
        For i=1 to 7 do
        Begin
          For j=1 to number of train samples Gi do
          Begin
                SAudio=Gi[j];
                SFrame=Audio_Framing (SAudio);
                 Matrix_Featurs[i][j] =Feature_Extraction( SFrame );
          End;
        GMM_Model[i][:]=GMM_Train(Matrix_Featurs[i][:]);
        End;
End;
```

Figure 3. Training Pseudo code of the proposed SV-MMGMM method

3.2. Testing Process

The block diagram for testing process of the proposed method has been shown in Figure 4. As can be seen, at first, pre-processing and feature extracting operations are done on the claimed speaker signal. And then the likelihood measure of the speaker pattern is computed with seven saved reference models, instead of one reference model. In the following, the likelihood measures for rejecting or accepting are passed to the decision making module. While comparing the speaker pattern with reference models of several different age categories, we can confirm or reject the speaker's identity, more accurately and decisively.

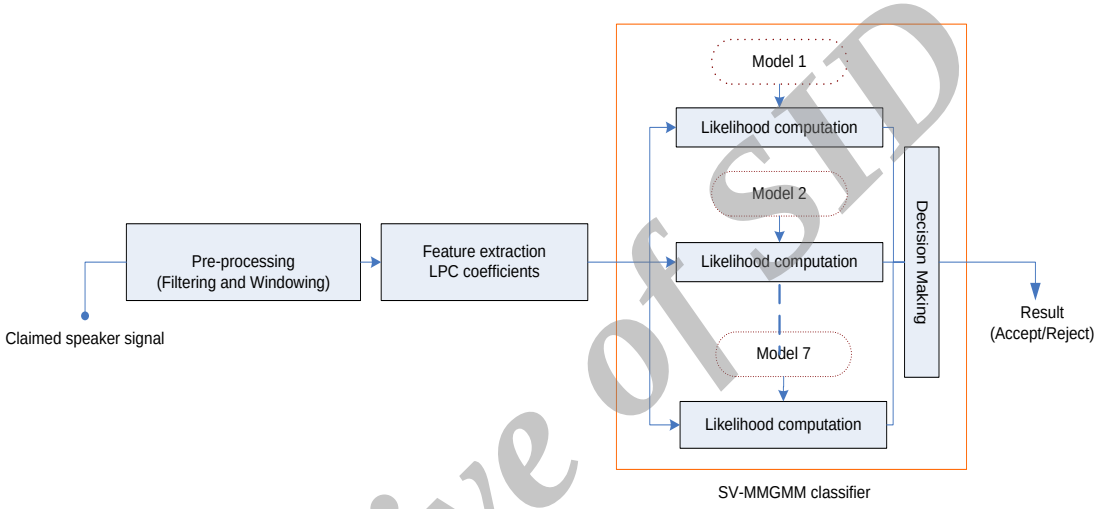

Figure 4. Testing block diagram of the proposed SV-MMGMM method

4. Experimental Results

In this paper, the FARSDAT database was used for simulations. This database contains words and phrases uttered by men and women. We have used phrases uttered by 135 speakers, including women and men (42 women, 93 men) in our simulations. Also, in the training phase, five different samples with lengths varying from 1 to 4 seconds were used for each speaker. Also, in the testing phase, we used two different learning samples with varying lengths. The simulations were carried out by means of MATLAB software. 39 LPC coefficients are used as feature measurement for feature extraction.
In order to evaluate the proposed method, at first, it is necessary to introduce confusion matrix and then express the evaluation criterions as follows:

		Actual Class	
		Acceptance	**Rejection**
Predicted Class	**Acceptance**	True Acceptance	False Acceptance
	Rejection	False Rejection	True Rejection

Figure 5. Simple confusion matrix

True Acceptance (TA): This parameter indicates the number of authentic speakers correctly detected as authentic ones.

False Acceptance (FA): This parameter indicates the number of fake speakers wrongly detected as authentic ones.
False Rejection (FR): This parameter indicates the number of authentic speakers wrongly detected as fake ones.
True Rejection (TR): This parameter indicates the number of fake speakers properly detected as fake ones.

Therefore, evaluation measures such as *Precision*, *Recall*, *F-measure* and *Accuracy* are defined as followings:

$$Precision = \frac{TA}{TA + FA} * 100 \quad (14)$$

$$Recall = \frac{TA}{TA + FR} * 100 \quad (15)$$

$$F - Measure = \frac{2\left(Precision * Recall\right)}{Precision + Recall} * 100 \quad (16)$$

$$Accuracy = \frac{TR + TA}{TA + FA + TR + FR} * 100 \quad (17)$$

As you know, the Precision measure is the fraction of retrieved instances that are relevant. However, the Recall measure is the fraction of relevant instances that are retrieved. A measure combining the Precision and Recall is their harmonic mean, the traditional F-measure[24].

We also used the measure *Detection Cost Function (DCF)* as one of the most important measures in the speakers verification systems, defined as [25]:

$$Min\ DCF = C_{miss}.FRR.P_{target} + C_{FA}.FAR.\left(1 - P_{target}\right) \quad (18)$$

where P_{target} is the prior probability of target test with $P_{target} = 0.01$. As previously mentioned, FAR and FRR are False Acceptance Rate and False Rejection Rate, respectively. And operating rate and the specific cost factors are $C_{miss} = 10$ and $C_{FA} = 1$, respectively.
The results obtained by evaluating the proposed SV-MMGMM have been shown in Table 1.

Table1. The obtained results of the proposed SV-MMGMM method

Evaluation Measures	Obtained Results (%)
TA	46
FA	3
TR	97
FR	54
Precision	93.8
Recall	32.1
F-Measure	47.9
Accuracy	71.5
Min DCF	8.3

In order to evaluate the efficiency of the proposed SV-MMGMM method, several speaker verification systems based on classifiers such as Bayesian, SVM, Decision Trees (C4.5), Ensemble (Random Forest, Bagging) and basic GMM have been implemented. The obtained results are presented in Table2.

Table2. Obtained results of a speaker verification system using different classifiers

Measures (%)	Classifiers					
	Naïve Bays	C4.5	SVM	Random Forest	Bagging	GMM
TA	91	37	46	54	41	43
FA	50	6	24	13	4	4
TR	50	94	76	87	96	96
FR	9	63	54	46	59	57
Precision	64.5	86	65.7	80.5	91.1	91.4
Recall	64.5	28.2	37.7	54	29.9	30.9
F-measure	64.5	42.5	47.9	64.6	45	46.1
Accuracy	70.5	65.5	61	70.5	68.5	69.5
Min DCF	50.4	12.2	29.1	17.4	98.6	9.6

Table 3 shows a comparison between the proposed method and several mentioned systems, from *F-Measure*, *Accuracy* and *Min DCF* measures viewpoints.

Table3. The comparison between the proposed SV-MMGMM and mentioned methods from F-Measure, Accuracy and DCF view points

Classifiers	Measures (%)		
	F-Measure	Accuracy	Min DCF
Naïve Bays	64.5	70.5	50.4
C4.5	42.5	65.5	12.2
Random Forest	64.6	70.5	17.4
Bagging	45	68.5	98.6
SVM	47.9	61	29.1
GMM	46.1	69.5	9.6
MMGMM	47.9	71.5	8.3

As can be seen from Table 3, the proposed SV- MMGMM system has the best accuracy (71.5%) and also the best DCF (8.3%) among other methods in which these two measures are very critical in the speaker verification systems. Such results are also

expected. As were expressed, the proposed method is able to create a parallel classifier based on multi model GMM in terms of age speakers. Since this method applies multiple GMM models instead of a single GMM model for all authorized speakers, it is capable to verify authentication of claimed speaker based on majority vote, well.

5. Conclusion

This paper presents a new text-independent speaker verification system based on dividing speakers into seven separate groups, called SV-MMGMM. Because the proposed method applies several GMM models instead of one GMM model, it is able to recognize people identification, well. In order to evaluate, the proposed SV-MMGMM method is compared with several speaker verification systems based on Naïve Bayes, C4.5, Fandom Forest, Bagging, SVM and GMM classifiers. The obtained results show that the SV-MMGMM method has better efficiency from Accuracy and Min DCF viewpoints. For future works, in the proposed architecture, applying improved GMM models (such as GMM/UBM) instead of basic GMM, as well as adding feature selection methods, can be efficiently improved.

6. References

[1] G. B. Varile, and A. Zampolli, *Survey of the state of the art in human language technology*: Cambridge University Press, 1997.

[2] L.-j. YANG, B.-h. ZHANG, and X.-z. YE, "Fast Fourier transform and its applications," *Opto-electronic Engineering*, pp. S1, 2004.

[3] C. S. Burrus, R. A. Gopinath, and H. Guo, *Introduction to wavelets and wavelet transforms: a primer*: Prentice Hall, New, 1997.

[4] T. Kinnunen, and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12-40, 2010.

[5] A. Maesa, F. Garzia, M. Scarpiniti *et al.*, "Text Independent Automatic Speaker Recognition System Using Mel-Frequency Cepstrum Coefficient and Gaussian Mixture Models," *Journal of Information Security*, vol. 3, no. 4, pp. 335-340, 2012.

[6] S. Nemati, and M. E. Basiri, "Text-independent speaker verification using ant colony optimization-based selected features," *Expert Systems with Applications*, vol. 38, no. 1, pp. 620-630, 2011.

[7] D. Burton, "Text-dependent speaker verification using vector quantization source coding," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, no. 2, pp. 133-143, 1987.

[8] K. Soman, and V. A. Loganathan. R, *"Machine Learning with SVM and other Kernel Methods"*: Prentice Hall of India, 2009.

[9] W. M. Campbell, J. P. Campbell, D. A. Reynolds *et al.*, "Support vector machines for speaker and language recognition," *Computer Speech and Language*, vol. 20, no. 2, pp. 210-229, 2006.

[10] S. Raghavan, G. Lazarou, and J. Picone, "Speaker verification using support vector machines," in Proceedings of the IEEE SoutheastCon, pp. 188-191, 2006.

[11] W. M. Campbell, J. P. Campbell, T. P. Gleason *et al.*, "Speaker verification using support vector machines and high-level features," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2085-2094, 2007.

[12] M. A. Lacerda, R. C. Guido, L. M. de Souza *et al.*, "A wavelet-based speaker verification algorithm," *International Journal of Wavelets, Multiresolution and Information Processing*, vol. 8, no. 06, pp. 905-912, 2010.

[13] D.-P. Munteanu, and S.-A. Toma, "Automatic speaker verification experiments using HMM," in 8th International Conference on Communications (COMM), pp. 107-110, 2010.

[14] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing,* vol. 10, no. 1, pp. 19-41, 2000.

[15] P. Kenny, G. Boulianne, P. Ouellet *et al.*, "Speaker and session variability in GMM-based speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on,* vol. 15, no. 4, pp. 1448-1460, 2007.

[16] A. Larcher, J.-F. Bonastre, and J. S. Mason, "Constrained temporal structure for text-dependent speaker verification," *Digital Signal Processing,* vol. 23, no. 6, pp. 1910-1917, 2013.

[17] R. Jourani, K. Daoudi, R. Andre Obrecht *et al.*, "Discriminative speaker recognition using Large Margin GMM," *Neural Computing and Applications,* vol. 22, no. 7-8, pp. 1329-1336, 2013.

[18] C. H. You, K. A. Lee, and H. Li, "GMM-SVM kernel with a Bhattacharyya-based distance for speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing,* , vol. 18, no. 6, pp. 1300-1312, 2010.

[19] B. Xiang, and T. Berger, "Efficient text-independent speaker verification with structural gaussian mixture models and neural network," *IEEE Transactions on Speech and Audio Processing,* , vol. 11, no. 5, pp. 447-456, 2003.

[20] V. N. Vapnik, *The nature of statistical learning theory*: Springer-Verlag New York Inc, 2000.

[21] S. Ahmed, F. Coenen, and P. Leng, "Tree-based partitioning of date for association rule mining," *Knowledge and information systems,* vol. 10, no. 3, pp. 315-331, 2006.

[22] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning,* vol. 40, no. 2, pp. 139-157, 2000.

[23] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse, "An empirical study of learning from imbalanced data using random forest." pp. 310-317, 2007.

[24] D. Powers, "Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation," *Journal of Machine Learning Technologies,* vol. 2, no. 1, pp. 37-63, 2011.

[25] D. A. Reynolds, and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing,,* vol. 3, no. 1, pp. 72-83, 1995.