Neural Computing and Applications manuscript No.
(will be inserted by the editor)

Speaker Recognition with Hybrid Features from a Deep Belief Network

Hazrat Ali · Son N. Tran · Emmanouil Benetos · Artur S. d'Avila Garcez

Received: date / Accepted: date

Abstract Learning representation from audio data has shown advantages over the hand-crafted features such as Mel Frequency Cepstral Coefficients (MFCC) in many audio applications. In most of the representation learning approaches, the connectionist systems have been used to learn and extract latent features from the fixed length data. In this paper, we propose an approach to combine the learned features and the MFCC features for speaker recognition task, which can be applied to audio scripts of different length. In particular, we study the use of features from different levels of Deep Belief Network for quantizing the audio data into vectors of audio-word counts. These vectors represent the audio scripts of different length that make them easier to train a classifier. We show in the experiment that the audio-word count vectors generated from mixture of DBN features at different layers give better performance than the MFCC features. We also can achieve further improvement by combining the audio-word count vector and the MFCC features.

Hazrat Ali is grateful for funding from the Erasmus Mundus Strong Ties grant. Emmanouil Benetos is supported by the UK AHRC-funded project 'Digital Music Lab - Analysing Big Music Data', grant no. AH/L01016X/1. Hazrat and Son have equal contribution to the paper.

Hazrat Ali
Department of Electrical Engineering
COMSATS Institute of Information Technology
University Road, Tobe Camp, Abbottabad 22060, KPK, Pakistan
E-mail: hazratali@ciit.net.pk

Son N. Tran, Emmanouil Benetos, Artur S. d'Avila Garcez
Department of Computer Science
City University London
Northampton Square, London EC1V 0HB, United Kingdom
Emmanouil Benetos
School of Electronic Engineering and Computer Science
Queen Mary University of London, United Kingdom

Keywords Deep Belief Networks · Deep Learning · Mel-Frequency Cepstral Coefficients

1 Introduction

Human speech essentially contains different kind of information besides the dominant linguistic information. Given the diverse range of vocal tract structure, speech data from every individual has its own speaker-specific information. Extraction of the speaker-specific information from the complex speech data is trivial for human beings but a challenging task for computers, though several techniques have been developed to extract this information effectively for speaker recognition applications.
Traditionally, discriminative models have been in use for learning speech data. Recently, generative models are getting popular as more computational resources are available now. Generative models can be useful by learning many more parameters if much more structures are present in the input vectors than the labels [1]. Typically, generative models are more popular for use with unlabeled data.
The use of Mel-frequency Cepstral Coefficients (MFCC) have been extremely popular for speech processing applications such as speech recognition and speaker recognition [1]. However, to the best of our knowledge, they have not been combined with unsupervised features to be evaluated on speaker recognition task. We propose a hybrid features model by combining the unsupervised features learned through a deep belief network with the classical MFCCs and then train a supervised classification model to perform speaker classification. We report our results on speaker recognition task for MFCC features, for Deep Belief Network (DBN) features and for hybrid features.
The remaining of this paper is organized as follows; In Section 2, we summarize the previous work as reported on the use of unsupervised models for speech data in general and speaker recognition in particular. In Section 3, we provide a discussion on the pipeline of our approach and the relevant technologies and how we use them. This will cover the concepts of MFCCs, Restricted Boltzmann Machines, Deep Belief Networks, the contrastive divergence algorithm used for training them and the supervised Support Vector Machine classification. We provide more details on our experiment and the model configuration in Section 4. We then conclude the paper in Section 5.

2 Related Work

While most of the work for using deep learning models target the speech recognition task, the task of speaker recognition has gained very little atten-

[1] A useful survey is presented by Kinnunen et al [2] on the use of MFCCs and other features such as super vectors for speaker recognition

tion [2]. On their work for phone recognition, Mohamed et al [1] suggest a more powerful alternative to Gaussian Mixture Models (GMMs) for relating Hidden Markov Models (HMM) states to feature vectors. They use a feed-forward neural network with multiple layers, for which features vectors are given as input to produce posterior probabilities of HMM states as output. They pre-train the network in layer-wise fashion (one layer at a time) as a generative model of the window of speech coefficients and achieve better performance on phone recognition. We do not use back propagation for supervised classification and rather combine Support Vector Machine (SVM) with the unsupervised model to perform the classification on the speaker recognition task. Lee et al [4] use convolutional deep belief networks (cvDBN) for unsupervised features learning and presented the results for speaker classification on TIMIT test set [5]. In some of their configurations, Lee et al [4] combine scores from different classifiers to improve classification accuracy, using a linear combination constant selected through cross validation run. This, however, is different than what we are proposing here in two ways; firstly, we do not use cvDBN which requires fixed length data in the input; secondly, we are not using decision level fusion but rather propose fusion at the feature level by combining the features from the different models.

For speaker recognition task, a first attempt on the use of RBMs has been reported by [6]. They use a single RBM training and apply the model to a speaker verification task. They model pairs of i-vectors using Gaussian-Bernoulli RBMs. The results obtained by [6] though did not outperform the then state-of-the-art systems, yet opened new directions of research by placing an interesting question at the end whether *i-vector extraction can be performed with Boltzmann Machines in a non-linear way such that the phonetic context is represented by this characterization.* A more recent work has been reported by Ghahabi & Hernando [7] proposing the application of deep belief network for i-vector based speaker verification [3]. Their experimentation is based on network with only one hidden layer as they train a single RBM (with 512 hidden units) with i-vectors. They train the system using selected speech files from NIST 2004 and NIST 2005 SRE corpora. They use 400-dimensional i-vectors and compare results with the baseline cosine distance classifier. The error rates suggest that modeling the i-vectors with RBM outperforms both conventional neural networks as well as the baseline cosine distance classifier. These developments suggest that learning speaker-specific characteristics from speech data with unsupervised model offers great potential. In the following section, we briefly describe the pipeline of our approach which include (a) the concept of MFCC (b) unsupervised feature learning with RBMs and DBNs (c) the use of k-means algorithm to build the audio codebook and (d) the classifier for recognition tasks.

[2] Besides the work reviewed in this section, a more recent work has been reported lately in [3] which presents a deep neural network approach for speaker recognition task.

[3] The i-vector is a recently developed features set for representation of speech data in low dimension [8] and has attracted the machine learning community through the NIST i-vector challenge [9, 10].

3 Methodology

3.1 Speaker Recognition Pipeline

In this section, we describe the pipeline of our approach, as shown in Figure 1. We convert an audio script into a vector of word counts and then combine it with the MFCC features of the audio script to form the input for the classifier.

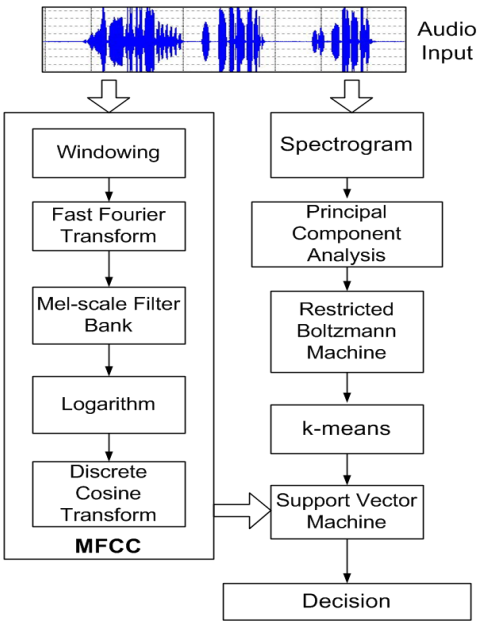


Fig. 1: Flow Chart

In order to build the vocabulary of the audio words we start with converting the audio script into spectrogram. We consider the spectrogram of a script as a 2-D matrix with time $\times$ frequency dimensions. We reduce the dimensions of the data for further processing steps by applying principal component analysis (PCA) to linearly transform the frequencies to lower dimension space. We retain only 80 components out of the 256 dimensions of the PCA transformed data. After that we learn the latent features of the audio data using DBN/RBM where each input sample is the PCA-transformed frequencies at a time slot. We then use k-means to build a codebook from the DBN features and quantize the features into audio words. At the end, the audio script will be represented in a bag of words, i.e. a vector of audio word counts.

For the MFCC features, we follow the standard processing steps as will be explained in the next section.

3.2 MFCC

The Mel Frequency Cepstral Coefficients (MFCCs) have been widely used for speech and speaker recognition applications [11]. The MFCC features capture the power spectral information for a single frame. However, speech signal contains information that can be represented by learning the dynamics. This information is captured by the delta and delta-delta coefficients which are the time derivatives of the original MFCC features. We calculate a total of 36 features per frame, including the delta and delta-delta features. The delta and delta-delta coefficients are also referred to be the differential and acceleration coefficients.

3.3 Feature Learning

Representation learning has emerged as trending topic recently following the success of deep learning in many machine learning applications [12,13,1,14]. In this paper, we are interested in learning and combining features using Deep Belief Networks (DBN). The DBN is built up by stacking several RBMs, one on the top of another. RBM itslef is a two-layer connectionist system with no connection between units in the same layer. A binary RBM is the basic type of RBM having binary input and binary hidden units in which the state of the model is characterized by an energy function:

$$E(\mathbf{v}, \mathbf{h}|\theta) = -\sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij} v_i h_j - \sum_{i=1}^{I} b_i v_i - \sum_{j=1}^{J} c_j h_j \tag{1}$$

where v denotes the state of visible layer, h denotes the state of hidden layer. The set of parameters $\theta = (\mathbf{w}, \mathbf{b}, \mathbf{c})$ denote the set of connection weights between visible unit i and hidden unit j as w_{ij} and the biases of the visible and hidden layers. I, J are the number of visible and hidden units, respectively.

For an input v, the probability distribution assigned to the visible-hidden units pair is defined by;

$$p(v, h) = \frac{e^{-E(v,h)}}{Z} \tag{2}$$

where Z is the normalization term (partition function) defined by $Z = \sum_{v,h} e^{-E(v,h)}$

Due to the restricted connections, the inference of RBMs is tractable given the state of a layer. For example given the state of visible layer one can sample the state of hidden layer from the conditional distribution $p(h_j|v, \theta)$:

$$p(h_j = 1|v, \theta) = \sigma(c_j + \sum_{i=1}^{I} w_{ij} v_i) \tag{3}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is a sigmoid function. Similarly, sampling the state of visible layer given the state of hidden layer can be done through the conditional

distribution $p(v_i|h,\theta)$ is given by:

$$p(v_i = 1|h,\theta) = \sigma(b_i + \sum_{j=1}^{J} w_{ij}h_j) \tag{4}$$

In many cases, the input data is continuous rather than binary and therefore the Gaussian-Bernoulli RBMs should be used to represent the data distribution. In Gaussian-Bernoulli RBMs the energy function is represented as:

$$E(v,h|\theta) = \sum_{i=1}^{I} \frac{(v_i - b_i)^2}{2} - \sum_{i=1}^{I}\sum_{j=1}^{J} w_{ij}v_i h_j - \sum_{j=1}^{J} c_j h_j \tag{5}$$

Since the hidden layer is still binary we can use the conditional distribution in equation (3) to sample its state. For the visible layer, the conditional distribution $p(v_i|h,\theta)$ is given by:

$$p(v_i|h,\theta) = \mathcal{N}(b_i + \sum_{j=1}^{J} w_{ij}h_i, 1) \tag{6}$$

where $\mathcal{N}(\mu,\nu)$ represents Gaussian distribution with mean μ and variance ν. Normally the means and variances can be learned through standard algorithms, it would be more efficient to normalize data to zeros means and unit variance and use (6) for inference. For some distributions, RBMs might not achieve representation as efficient as can be obtained with unrestricted Boltzmann machines. However, with enough number of hidden units, any discrete distribution can be represented with RBMs [15,13]. Besides, under certain circumstances, the addition of hidden units with proper weights and bias helps to improve the log-likelihood.
Taking the gradient of the log likelihood *log* $p(\mathbf{v};\theta)$, the weights for the RBM can then be updated as follows;

$$\Delta w_{kj} = \mathsf{E}_{data}(v_k h_j) - \mathsf{E}_{model}(v_k h_j) \tag{7}$$

where $\mathsf{E}_{data}(v_k h_j)$ represents the expectation observed in the training set, sampling h_j given v_k, and $\mathsf{E}_{model}(v_k h_j)$ is the expectation from the model. The challenge arises due to the fact that computation of $\mathsf{E}_{model}(v_k h_j)$ is difficult. This challenge was addressed by the contrastive divergence (CD). The CD algorithm is an approach to approximate the gradient for determining the expected value. $(\mathbf{v_1}.\mathbf{h_1})$ is a sample from the model roughly estimating $\mathsf{E}_{model}(v_k h_j)$. This approximation of $\mathsf{E}_{model}(v_k h_j)$ by using $(\mathbf{v_1}.\mathbf{h_1})$ is referred as CD-1, as described by Hinton [16].
In our work on using DBNs for learning speech data, we use Gaussian input units and binary hidden units for the RBMs for the first layer and binary units for the second one.

3.4 Bag of Audio Words

In order to convert the data with different length into vectors of the same dimensionality, we use an approach similar to bag of words (BoW) in document processing. The BoW approach creates a vector of the size of a vocabulary for each document. The vector represents either the appearance/occurrence of a word in the document as a single element. BoW approach is also very popularly used in the vision applications that contain continuous features. Here, the continuous features are quantized into a set of groups where each group can be seen as a representation of a visual word. Similarly, we apply the idea to audio data. We assume that at each time slot, the set of frequencies represent an audio word. In this paper, we use the k-means algorithm, a simple yet efficient algorithm for quantization. We also use the DBNs features as mentioned earlier as the representation of the audio data to learn the codebook (the audio vocabulary).

3.5 Classification

The final step in our pipeline is classification. Since the main focus of the paper is learning and combining features we use the same classifier for evaluation. In particular, we use the Support Vector Machine (SVM) as our classifier. SVM is a popular kernel based discriminative classification algorithm [4]. For multi-class problems, the one-vs-all approach is adapted to achieve classification. For SVM, we use the libSVM library [18], which can handle the problem of multi-class data. We train the SVM with the Gaussian RBF kernel, which for two data points x_i and x_j is given as;

$$K(x_i, x_j) = exp(\gamma(\|x_i - x_j\|)^2) \tag{8}$$

where γ is the hyper-parameter, selected through validation set.

4 Experiment

4.1 Experiment Preparation

We use the Urdu dataset [5]. The Urdu dataset is a dataset of Urdu (the national language of Pakistan) and comprises of audio data from ten speakers including both male and female, native and non-native speakers. There are 250 files for each speaker and each file is a recording of a distinct isolated word [19] [6].

We randomly divide our data into train, validation and test sets with a ratio of $2:1:1$ and observe the results for MFCCs as well as for features learned

[4] A useful tutorial on SVM is available from Burges [17]

[5] The dataset can be requested via email.

[6] Previous experimentations with this dataset for speech recognition applications have been reported by [20, 21].

with the RBM framework. For the MFCC experiment, we also calculate delta and delta-delta coefficients giving a total of 36 coefficients for each file.

We perform our experimentation on a single CPU of 3.0 GHz clock frequency with RAM capacity of 16 GB. A single run of an experiment for one setting takes approximately 2 hours. For the first layer we use the Gaussian-Bernoulli RBMs. We normalize the data so that it has zero mean and unit variance. The output of the first RBM is used as input for the second RBM which has binary visible and hidden units. Both of the RBMs are trained within 100 epochs and the learning rate (0.01 was selected), sparsity hyper-parameters are selected using validation set. We also use the validation set to select the SVM parameters for classification.

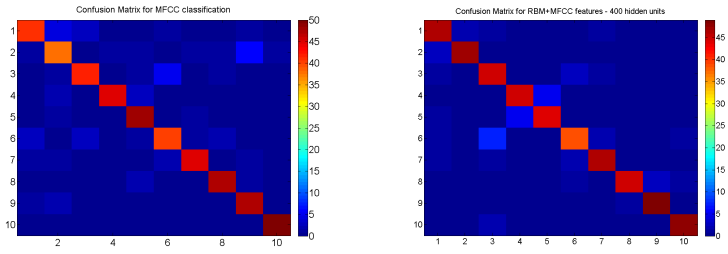
We train the DBN with the data for different configurations. After the DBNs have been trained, we use the state of the top hidden layer given the data as input as the features to build a codebook using k-means algorithm. We test the codebook with different sizes of $10, 50, 100, 400, 600$, which is also the dimension of the word-count vector representing an audio script. We use these word-count vectors to learn the SVM classifier for speaker recognition task.

4.2 Experimental Results

Table 1: Test set accuracy for speaker classification

Approach	Accuracy (%)
MFCC	88.6
Audio words (DBN-1)	90.40
Hybrid (DBN-1 + MFCC)	91.40
Audio words (DBN-2)	72.20
Hybrid (DBN-2 + MFCC)	87.00
Audio words (DBN-1 + DBN-2)	90.60
Hybrid (DBN-1 + DBN-2 + MFCC)	92.60

In representation learning, the dimension is important since it not only decides the size of the output features but also the efficiency of the learning. Normally the larger hidden layer may produce more generalized features however it would create more computation overhead. In the experiment, we test different type of features. The MFCC features attempt to eliminate information from speech data that is not relevant for recognition purposes, thus providing input representation of modest size. DBN on the other hand make use of less-processed input data. Instead, it learns the latent features from the PCA-transformed spectrogram. The advantage of DBN is it can learn useful representation as we can see from the results in Table 1. Here, taking the advantage of layer-wise learning in DBN we also combine different features in different layers.



(a) Confusion Matrix - MFCC Features only

(b) Confusion Matrix - Hybrid (DBN-1 + MFCC) Features

Fig. 2: A visualization of confusion matrix for accuracies achieved from MFCC features and from hybrid (DBN-1 + MFCC) features. Horizontal axis shows actual labels for speakers. Vertical axis shows predicted labels for speakers.

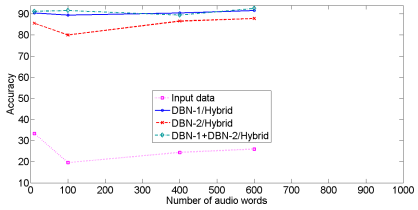
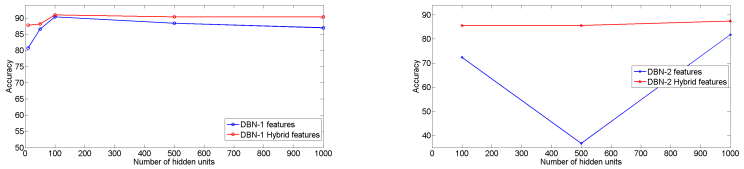


Fig. 3: Classification performance on number of audio words from DBN-1 and DBN-2. (% accuracy on vertical axis)



(a) DBN-1

(b) DBN-2

Fig. 4: Classification performance on dimensions of learned features from DBN-1 and DBN-2. (% accuracy on vertical axis)

Let us denote DBN-1 and DBN-2 as the features from the first and the second RBMs in deep network. The results in Table 1 show that the audio words built from features in first layer of DBNs (DBN-1) outperform the MFCC features. The classification performance even achieve improvement when we combine the audio words generated from DBN-1 features with the MFCC. For example, Figure 2a and 2b show the confusion matrix of accuracies from MFCC and from DBN-1+MFCC features. However, when an extra layer is

used in the DBN, the features are not good enough to build audio words and generalize the classifier. It seems that the DBN-1 features generalize better than the DBN-2 counterpart because the expansion of the feature's dimension in first layer make it more difficult to learn in the second one. This effect also can be seen when applying the convolutional DBNs on audio data [4], but in this experiment it is more severe. As the results show, combining DBN-1 and DBN-2 features does not show any improvement. However, it is interesting that combining them with MFCC can give better results than the DBN-1 features. In Figure 3, we show the results over different codebook sizes learned from DBN-1, DBN-2, and the combination DBN-1+DBN-2 features. One can see that, when combining DBN-1+DBN-2 features with MFCCs, the classification accuracies with codebook size 10, 100, 600 the DBN-1+DBN-2 features seems to achieve better performance than the DBN-1 features.

In representation learning, the dimension is important since it not only decides the size of the output features but also the efficiency of the learning. Normally the larger hidden layer may produce more generalized features however it would create more the computational overhead.

5 Conclusion

The goal behind this work is to show that we can get better performance by taking benefit of discriminative information retained by the traditional MFCCs and the features learnt by the unsupervised model. The experimental results have shown that using hybrid features is promising and improves performance on speaker classification task. We did not aim to show the performance on a variety of data but rather investigated the usefulness of the proposed technique by exploiting the Urdu dataset and obtaining empirical results. As speaker recognition is a language independent task, the proposed framework can be extended for speech data of other languages. We believe that this work will encourage other researchers to adopt the approach for similar tasks and on much bigger datasets for which we were most often restricted due to the available computational resources. Similarly, with large-scale data and GPU sufficient resources, the framework can be employed for data of much bigger size and reduced computational time.

Acknowledgements The authors would like to thank Nasir Ahmad, University of Engineering and Technology Peshawar Pakistan and Tillman Weyde, City University London for their useful feedback during this work.

References

1. A. R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.
2. 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, Jan. 2010.

3. F. Richardson, D. Reynolds, and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, October 2015.
4. H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems*, Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, Eds. NIPS, 2009, pp. 1096–1104.
5. J. Garofolo, L. Lamel, W. Fisher, J. Fiscus, D. Pallett, N. Dahlgren, and V. Zue, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1
6. M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, "First attempt of boltzmann machines for speaker verification," in *Odyssey 2012, The Speaker and Language Recognition Workshop*. ACM, June 2012, pp. 1064–1071.
7. O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1700–1704.
8. N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.
9. "NIST i-vector Machine Learning Challenge," 2014, accessed: December 2014. [Online]. Available: https://ivectorchallenge.nist.gov/
10. H. Ali, A. d'Avila Garcez, S. Tran, X. Zhou, and K. Iqbal, "Unimodal late fusion for NIST i-vector challenge on speaker detection," *Electronics Letters*, vol. 50, no. 15, pp. 1098–1100, July 2014.
11. S. Molau, M. Pitz, R. Schluter, and H. Ney, "Computing mel-frequency cepstral coefficients on the power spectrum," in *Proceedings of 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2001, pp. 73–76.
12. G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
13. N. Le Roux and Y. Bengio, "Representational power of restricted boltzmann machines and deep belief networks," *Neural Comput.*, vol. 20, no. 6, pp. 1631–1649, Jun. 2008.
14. L. Deng and D. Yu, *DEEP LEARNING: Methods and Applications*. NOW Publishers, January 2014.
15. Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," University of California at Santa Cruz, Santa Cruz, CA, USA, Tech. Rep., 1994.
16. G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
17. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.
18. C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.
19. H. Ali, N. Ahmad, K. M. Yahya, and O. Farooq, "A medium vocabulary urdu isolated words balanced corpus for automatic speech recognition," in *2012 International Conference on Electronics Computer Technology (ICECT 2012)*, 2012, pp. 473–476.
20. H. Ali, N. Ahmad, X. Zhou, M. Ali, and A. Manjotho, "Linear discriminant analysis based approach for automatic speech recognition of urdu isolated words," in *Communication Technologies, Information Security and Sustainable Development*, ser. Communications in Computer and Information Science. Springer International Publishing, 2014, vol. 414, pp. 24–34.
21. H. Ali, N. Ahmad, X. Zhou, K. Iqbal, and S. M. Ali, "DWT features performance analysis for automatic speech recognition of urdu," *SpringerPlus*, vol. 3, no. 1, p. 204, 2014.