DEVELOP SPEAKER IDENTIFICATION SYSTEM

Mohamed M. Abdelmotaleb, Mohamed Mostafa ,

Mohamed Maged, Nada Hamdy Tawfi, *Norhan Ahmed*

***Abstract*—Speaker identification technologies are widely applied in voice authentication, security and surveillance, electronic voice eavesdropping, and identity verification. In the speaker identification process, extracting discriminative and salient features from speaker utterances is an important task to accurately identify speakers. Various features for speaker identification have been recently proposed by researchers. Most studies on speaker identification have utilized short-time features, such as perceptual linear predictive (PLP)coefficients and Mel frequency cepstral coefficients (MFCC), due to their capability to capture the repetitive nature and efficiency of signals**.

I. INTRODUCTION

Automatic speaker identification (ASI) is the process of extracting the identity of a speaker by using a machine from a group of familiar speech signals. Speech signals are powerful media of communication that always convey rich and useful information, such as emotion, gender, accent, and other unique characteristics of a speaker. These unique characteristics enable researchers to distinguish among speakers when calls are conducted over phones although the speakers. Speaker utterances are trained with machine learning algorithms from the collected dataset, and then speakers are identified using the test utterances. Speaker identification involves the identification of a speaker utterance from a group of trained speaker utterances. Then, the speaker with a high probability of test utterance is identified as the speaker. Alternatively, speaker verification involves the process of determining whether a speaker of a test utterance belongs to a group of speakers through binary classification. Speaker recognition has become an area of intense research due to its wide range of applications, including forensic voice verification to detect suspects by government law enforcement agencies, access control to different services, such as telephone network services, voice dialing, computer access control , mobile banking, and mobile shopping . Furthermore, speaker identification systems are extensively used to improve security , automatic speaker labeling of recorded meetings and personalized caller identification using intelligent answering machines. The major challenge in speaker identification is the extraction of discriminative features from speech signals that can elicit improved performance from classification algorithms. In this regard, many studies have proposed different feature-engineering techniques.so in this paper try to discuss to create a model of speaker identification by using neural network classifier(as shown in figure 1)

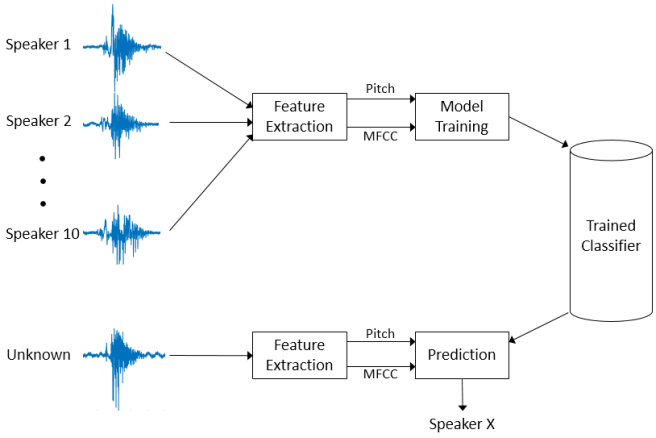

Figure 1: model of speaker identification

1) Data: the data is divided to training and test data where Learning the process of extracting phonetic characteristics from a speaker that has already been recorded or saved as a sample, storing them in a database, and familiarizing the system with the characteristics of the speaker's voice
and for testing: the process of comparing questionable sound and phonetic characteristics from a speaker recognition database.

2) Feature engineering: it is responsible for extracting discriminative features from speaker utterances and transforming these features into a numeric master feature vector. This vector is then used by a machine learning or deep learning classifier to quickly learn the classification rules and develop a classification model.
Here is some features which we are interested

a) Mel frequency cepstral coefficients (MFCC)(as shown in figure 2):

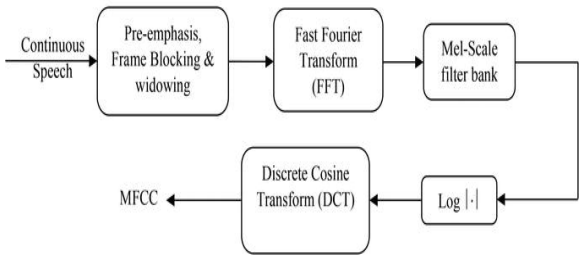

Figure 2: MFCC block daigram

As the main point to understand about speech is that the sounds generated by a human are filtered by the shape of the vocal tract which determines what sound comes out. Using MFCC we can determine the shape accurately, this Should give us an accurate representation of the phoneme being produced. The shape of the vocal tract manifests itself in the envelope of the short time power spectrum, and the job of MFCCs is to accurately represent this envelope.

Steps

1) Pre-processing:

it is very critical phase in the systems where background-noise or silence is completely undesirable. we use pre-emphasis and silence removal techniques. The pre-emphasis method increases the strength of high frequencies of speech signal, while the low frequencies remain in their original condition in order to improve the signal-to-noise ratio. Pre-emphasis works by enhancing the high-frequency energy through applying high-pass filter (FIR) (as shown in equ 1)

$H(z) = 1 - \alpha z^{-1}, \alpha = [1, -0.97]$ (equ 1)

2) MFCC feature extraction:

1. Frame the signal into short frames taking in consideration overlapping (into 20-40ms frames.) Each frame will act as a training sample. In each frame we will perform speech detection, pitch and MFCC. (As shown in figure3)

To get number of frames used equ 2 and 3 Total frames = Number of Samples Frame step × Sample (equ2)
N = Frame length × Sample Rate (equ3)

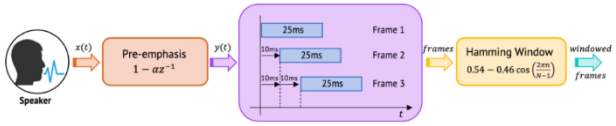

Figure 3: framing of speaker

2. For each frame calculate the periodogram estimate of the power spectrum.

3. Apply the mel filterbank to the power spectra,sum the energy in each filter, As the frequencies get higher our filters get wider as we become less concerned about variations ,also Mel scale is a scale that relates the rceived frequency of a tone to the actual measured frequency. It scales the frequency in order to match more closely what the human ear can hear (humans are better at identifying small changes in speech at lower frequencies). This scale has been derived from sets of experiments on human subjects. (as shown in figure 4).

ie: Mel(f)=1127 log$(1 + \frac{f}{700})$

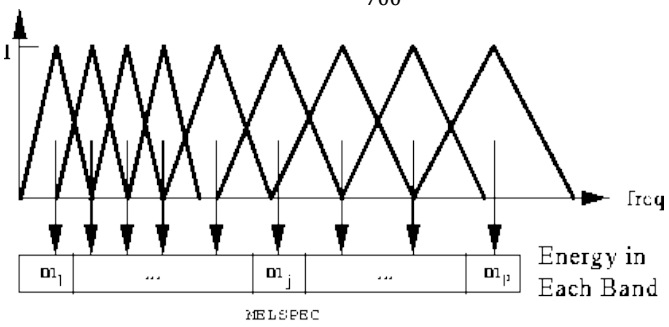

Figure 4:mal filters

4. take the logarithm of all filterbank energies: this is motivated by human hearing: as we don't hear loudness on linear scale.

5. Take the DCT of the log filterbank energies: As filterbanks are all overlapping, the filterbank energies are quite correlated with each other. The DCT decorrelates the energies. But by keeping the 12 coefficients from total DCT coefficients. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance.

3) Pitch:

Pitch is frequency domain analysis of voice data and it represents the fundamental frequency for the given speech, we take a short time frame (window) and do the FFT of that portion, then go to next time frame and do the same thing. Thus we get a frequency response depending on time. This is important in our case, because we want to extract frequencies when speaker is talking and that should be variable frequency depending on time. Pitch analysis of voice data (as shown in figure 5):

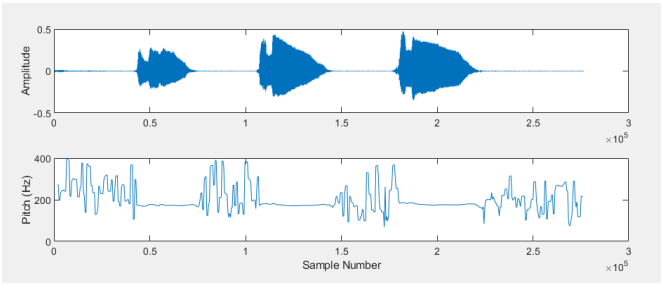

Figure 5: pitch analysis

---

4) Linear prediction coefficients (LPC):

It imitates the human vocal tract and gives robust speech feature. It evaluates the speech signal by approximating the formants, getting rid of its effects from the speech signal and estimate the concentration as it is used to obtain the filter coefficients equivalent to the vocal tract by reducing the mean square error in between the input speech and estimated speech. Linear prediction analysis of speech signal forecasts any given speech sample at a specific period as a linear weighted aggregation of preceding samples. The linear predictive model of speech creation is given for next equations (4, 5):

$$\hat{s}(n)=\sum_{k=1}^{p} a_k s(n-k) \quad \text{equ (4)}$$

where $\hat{s}$ is the predicted saample,
s is the speech saample,
p os the predictor coeddicients

$$e(n) = s(n) - \hat{s}(n) \qquad \text{equ (5)}$$

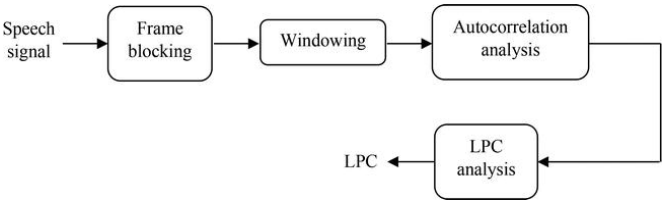

Figure 6: LPC block diagram

Also for cepstral coefficients derived from LPC calculated spectral envelope. Where LPCC are the coefficients of the Fourier transform illustration of the logarithmic magnitude spectrum of LPC (as shown in figure 6).Cepstral analysis is able to symbolize speech Waveforms and characteristics with a limited size of features where coefficient is calculated from equ 4

$$C_m = a_m + \sum_{k=1}^{m-1} \left[\frac{k}{m}\right] c_k a_{m-k} \quad \textit{equ (4)}$$

Where a_m is the linear prediction coefficient, C_m is the cepstral coefficient.
And last thing in this step is to scale the features for better convergence as shown in equ (5).

$$z = \frac{x-\mu}{\delta} \quad \text{Equ (5)}$$

Where μ= Mean
δ =standard deviation

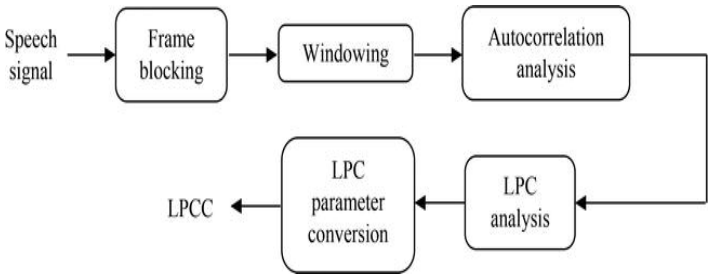

Figure 7: LPCC block diagram

Note: classification performance relies on the quality of a feature set. In deep learning and machine learning, extracting discriminative feature sets is an important task to obtain reasonable classification performance. If the extracted features correlate well with the class, then classification will be easy and accurate

3) Classification algorithm:

The construction of a classification model is frequently the fastest step in the speaker identification task because feature engineering .As it is important part of process to identify the person from the given data the neural network and deep learning plays in important role on. It is responsible for extracting discriminative features from speaker utterances and transforming these features into a numeric master feature vector. This vector is then used by a machine learning or deep learning classifier to quickly learn the classification rules and develop a classification model.

1) K-nearest neighbours(KNN):

It measures the distance between a query scenario and a set of scenarios in the data set using Euclidian distance as shown in equation 6

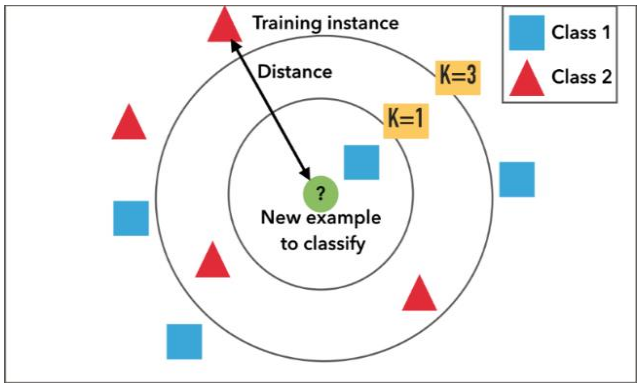

Figure 8:KNN algorithm

$$dist(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad \text{equ(6)}$$

Advantages of KNN

1. Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

2. KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function

3. No Training Period: KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.

Disadvantages of KNN

1. Does not work well with large dataset: In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.

2. Does not work well with high dimensions: The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.

3. Need feature scaling: We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.

4. Sensitive to noisy data, missing values and outliers: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

To implement the KNN algorithm to the model

1)Define the value of K, number of nearest neighbor that we are going to pick (for optimum k =square root (number samples)).

2)Add the distance and the index of the example to an ordered collection.

3)Sort the ordered collection of distances and indices

from smallest to largest (in ascending order) by the distances.

4) Pick the first K entries from the sorted collection.

5) Get the user in the selected K entries.

6) Return the most frequent of the selected user.

2)Extra-Trees algorithm(Extremely Randomized Trees):

It is an ensemble machine learning algorithm. It works by creating a large number of unpruned decision trees from the training dataset. Predictions are made by averaging the prediction of the decision trees in the case of regression or using majority voting in the case of classification. there are three main hyper parameters to tune in the algorithm

1)they are the number of decision trees in the ensemble

2)the number of input features to randomly select and consider for each split point

3)and the minimum number of samples required in a node to create a new split point.

It has two parameters: K, the number of attributes randomly selected at each node and n_{min}, the minimum sample size for splitting a node. It is used several times with the original learning sample to generate an ensemble model.

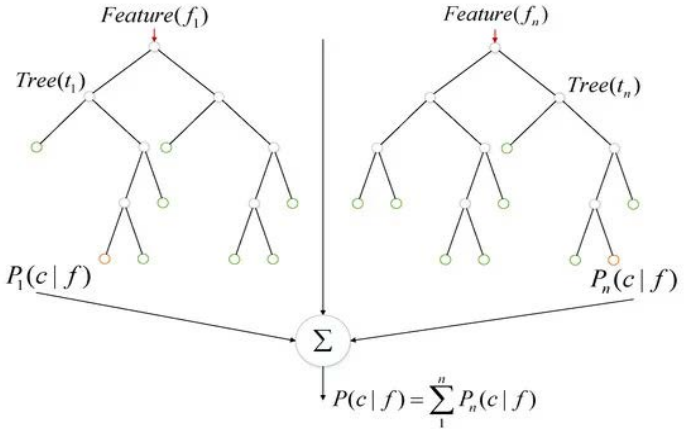

Figure 9:Extra tree algorithm

Here how to algorithm works

Trees_node(M)
Input signal: the local learning subset M corresponding to the node
Output signal: a tree [a< ac] or nothing

- If **Tree(S)** is TRUE then return nothing;
- Otherwise select K attributes {a1,...,aK} among all non constant (in S) candidate attributes;
- Draw K trees $\{s_1,...,s_K\}$, where s_i = Random_split(S, a_i), ∀i = 1,..., K;
- Return a tree s_* such that Score(s_*, S) = $\max_{i=1,...,K}$ Score(s_i, S).

Random_split(S, a**)**
Inputs: a subset S and an attribute a
Output: a split

- Let a_{max}S and a_{min}S denote the maximal and minimal value of a in S;
- Draw a random cut-point acuniformly in [a_{max}S, a_{min}S];
- Return the tree [a< ac].

Tree(S)
Input: a subset S
Output: a Boolean

- If |S| < n_{min}, then return TRUE;
- If all attributes are constant in S, then return TRUE;
- If the output is constant in S, then return TRUE;
- Otherwise, return FALSE.

3) Support vector clustering(SVC):

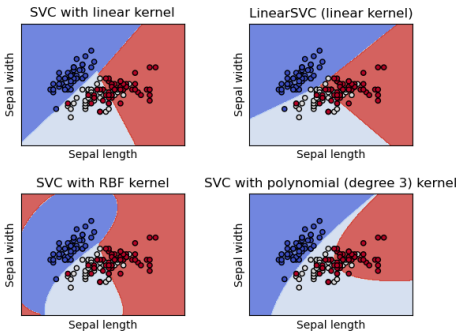

Figure 9

They are a set of supervised learning methods used for classification, regression and outliers detection.

The advantages of support vector machines are:

1)Effective in high dimensional spaces.
2)Still effective in cases where number of dimensions is greater than the number of samples.
3)Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

Disadvantages:

1)SVM algorithm is not suitable for large data sets.

2)SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.

3)In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.

4)Multi-layer Perceptron (MLP)

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

Multilayer perceptrons are often applied to supervised learning problems3: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the

model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error (RMSE).

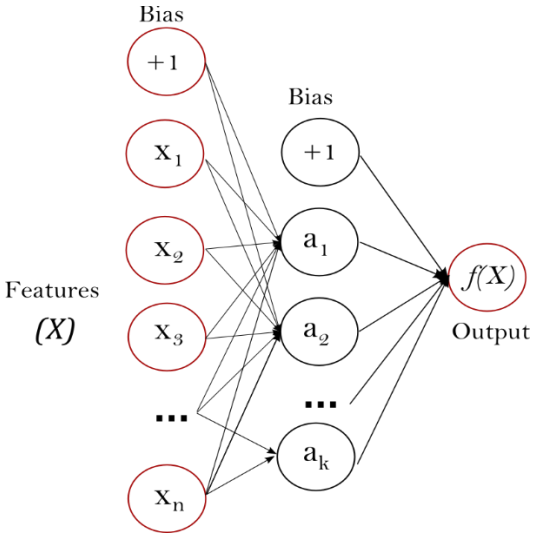

Figure 10

The advantages of Multi-layer Perceptron are:

- Capability to learn non-linear models.
- Capability to learn models in real-time

The disadvantages of Multi-layer Perceptron (MLP) include:

- MLP with hidden layers have a non-convex loss function where there exists more than one local minimum. Therefore different random weight initializations can lead to different validation accuracy.
- MLP requires tuning a number of hyperparameters such as the number of hidden neurons, layers, and iterations.
- MLP is sensitive to feature scaling.

5) *Gaussian NB algorithm*

Naive Bayes Algorithm is one of the popular classification machine learning algorithms that helps to classify the data based upon the conditional probability values computation. It implements the Bayes theorem for the computation and used class levels represented as feature values or vectors of predictors for classification. Naive Bayes Algorithm is a fast algorithm for classification problems.

This algorithm is a good fit for real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for the large data set.

It helps to calculate the posterior probability P(c|x) using the prior probability of class P(c), the prior probability of predictor P(x) and the probability of predictor given class, also called as likelihood P(x|c).

how a Gaussian Naive Bayes (GNB) classifier works

At every data point, the z-score distance between that point and each class-mean is calculated, namely the distance from the class mean divided by the standard deviation of that class(as shown in figure 11).Thus, we see that the Gaussian Naive Bayes has a slightly different approach and can be used efficiently.

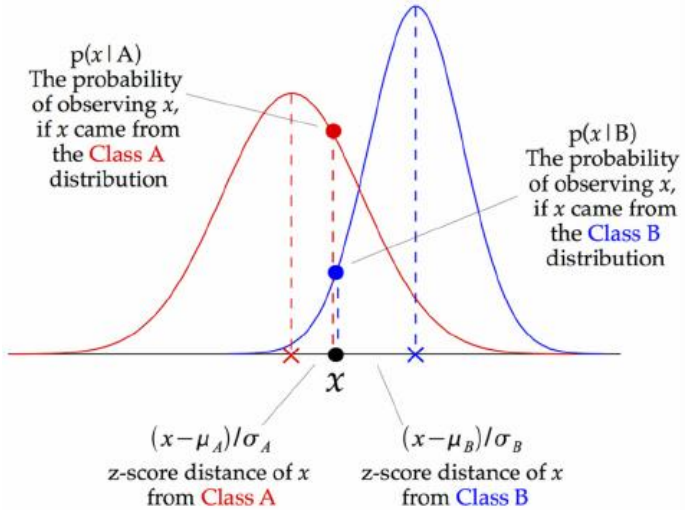

Figure 11

For our project : we use KNN algorithm and here is a summary of our steps

This project has two parts:

- Training Module: voice data were collected from our collogues and processed to store them for training purpose.
- Testing Module: speaker was identified with the help of trained classifier.

Part 1: Training Module:

In training module, all speakers voice data were recorded to extract features from their voice

For each frame we need to detect if it a speech or non-speech data. If it is speech then we will do further calculation, otherwise not. For detecting speech steps are-

- Calculate average power of frame.
- If average value it higher than a threshold value, then its speech data.

This method is called short time average power algorithm. Then if the frame is speech data we calculate its pitch and MFCC. Then we calculate pitch and MFCC for this frame. By using matlab functions

Part 2: Testing Module:

In testing module we will take speaker voice to detect the speaker. We will use previously stored speech data in training module and apply machine learning. These steps below for testing module are almost exactly same as training module-

For classification KNN Algorithm Implementation:

After framing and calculating MFCC and pitch, we have got good number of frames as test data, each of them will be tested using KNN. Then we will take the most frequent result out of all test frame.

After we understand the model how To improve the a ccuracy of our model

1) Window size of the frame
2) Type of window (like hamming, hanning , black man)
2) Overlapping: to decrease the amount of data we lose during framing
3) For used classifier optimization the parameters used to classify like k parameter :its optimum value =$\sqrt{number\ samples}$

Conclusion

In this paper we studied the MFCC features for speaker identification along with LPCC and LMACC features. LMACC features represented the normalized averages of both, MFCC and LPCC, features combined. However, large feature vector size didn't converge towards the needed results, so in the experiments we used MFCC features along with pre-processing techniques in order to enhance and increase the quality of data collected from different speakers where each one used a different microphone to record the data. We proposed the KNN model when trained on the given data and it met the requirements. The overall accuracy was approximately 75% – 80%. In the future, we intend to use other classification techniques in order to enhance the proposed algorithm and apply the MFCC, LPCC and LMACC feature vector.

Refernces

[1] Rabiner, L. R. and Juang, B.-H. (1993). Fundamentals of speech recognition.

[2] Elbarougy, R. and Algrbaa, H.A., 2017. Adoption Speaker Recognition System using Mel-frequency Cepstral Coefficients. Journal of Computer Science Approaches, 4 (1).

[3] Jahangir, R., Teh, Y.W., Memon, N.A., Mujtaba, G., Zareei, M., Ishtiaq, U., Akhtar, M.Z. and Ali, I., 2020. Text-Independent Speaker Identification Through Feature Fusion and Deep Neural Network. IEEE Access, 8, pp.32187-32202.

[4] J. Naik and G. Doddington, ``Evaluation of a high performance speaker verication system for access control," in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Mar. 2005, pp. 23922395.

[5] M. G. Gomar, ``System and method for speaker recognition on mobile devices," Google Patents 9 042 867, Mar. 26, 2015.

[6] M. Faundez-Zanuy, M. Hagmüller, and G. Kubin, ``Speaker identication security improvement by means of speech watermarking," Pattern Recog- nit., vol. 40, no. 11, pp. 30273034, Nov. 2007.

[7] B. M. Arons, Interactively Skimming Recorded Speech. Cambridge, MA, USA: Massachusetts Institute of Technology, 1994.

[8] C. Schmandt and B. Arons, ``A conversational telephone messaging system," IEEE Trans. Consum. Electron., vols. CE30, no. 3, pp. 2124, Aug. 1984.

[9] A. Maurya, D. Kumar, and R. K. Agarwal, ``Speaker recognition for hindi speech signal usingMFCC-GMMapproach," Procedia Comput. Sci., vol. 125, pp. 880887, Jan. 2018.

[10] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, ``Speaker verication using adapted Gaussian mixture Models," Digit. Signal Process., vol. 10, nos. 13, pp. 1941, Jan. 2000.

[11] D. A. Reynolds, ``Speaker identication and verication using Gaussian mixture speaker models," Speech Commun., vol. 17, nos. 12, pp. 91108, Aug. 1995.

[12] W.-C. Chen, C.-T. Hsieh, and C.-H. Hsu, ``Robust speaker identication system based on two-stage vector quantization," J. Sci. Eng., vol. 11, no. 4, pp. 357366, 2008.

[13] D. B. A. Mezghani, S. Z. Boujelbene, and N. Ellouze, ``Evaluation of SVM kernels and conventional machine learning algorithms for speaker identication," Int. J. Hybrid Inf. Technol., vol. 3, pp. 2334, Jul. 2010.

[14] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, ``Librispeech: An ASR corpus based on public domain audio books," in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Apr. 2015, pp. 52065210.

[15] M. A. Islam, W. A. Jassim, N. S. Cheok, and M. S. A. Zilany, ``A robust speaker identication system using the responses from a model of the auditory periphery," PLoS ONE, vol. 11, no. 7, Jul. 2016, Art. no. e0158520.

[16] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep Learning. Cambridge, MA, USA: MIT, 2016.

[17] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, ``Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," Expert Syst. Appl., vol. 105, pp. 233261, Sep. 2018.

[18] R. Karthik, D. Tyagi, A. Raut, S. Saxena, and R. Kumar M, ``Implementation of neural network and feature extraction to classify ECG signals," 2018, arXiv:1802.06288. [Online]. Available: http://arxiv.org/abs/1802.06288

[19] J. S. Lim and A. V. Oppenheim, ``Enhancement and bandwidth compression of noisy speech," Proc. IEEE, vol. 67, no. 12, pp. 15861604, Dec. 1979.

[20] Z. Wu and Z. Cao, ``Improved MFCC-based feature for robust speaker identication," Tinshhua Sci. Technol., vol. 10, no. 2, pp. 158161, Apr. 2005.