Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2023.00036

Research Article

Effect of number and position of frames in speaker age estimation

Mohammed Muntaz OSMAN[1,*], Osman BÜYÜK[1], Ali TANGEL[1]

[1]*Faculty of Engineering, Department of Electronics and Communication Engineering, Kocaeli University, Kocaeli, 41001, Türkiye*

ARTICLE INFO

Article history
Received: 20 April 2021
Revised: 12 August 2021
Accepted: 01 October 2021

Keywords:
Filter Banks; Frame Position; Mean Absolute Error; Regression; Speaker Age; Utterance Length

ABSTRACT

With the invention of powerful processing devices as well as lucrative capabilities in the first two decades of the 21st century, machine learning algorithms will soon be able to predict speaker age with higher accuracy or much lower error rate. It is an age-old quest for the human society to profile individuals remotely which basically includes age. Speaker age estimation has been treated in quite few perspectives. However, most of these approaches fail to show the effect of utterance length, aka number of frames on speaker age estimation. We present a detailed analysis on the effect of number of frames and position of frames for speaker age estimation using four magnitude-based and one phase-based spectral feature sets. The optimal speech duration for this objective is investigated. In addition, the mismatch between the training and test utterance duration is explored. The magnitude-based features are mainly derived from filter bank analysis. After the filter-bank analysis, an i-vector is generated for each utterance. Least squares support vector regression (LSSVR) is employed for speaker age estimation. In the experiments, the aGender database which consists of utterances from four age groups of German speakers is used. Increasing number of frames in the training and test increases the age estimation accuracy. This can be associated with the notion that more data helps the estimation process. Concerning position, the frames located at the centre of utterances tend to offer better results for both genders. The backend algorithms offer the best performance when the utterance length of training and test sets are equal for longer speech segments, otherwise training with medium length utterances and testing with longer ones offers better estimation performance especially for the female dataset.

Cite this article as: Osman MM, Büyük O, Tangel A. Effect of number and position of frames in speaker age estimation. Sigma J Eng Nat Sci 2023;41(2):243–255.

***Corresponding author.**
*E-mail address: mohammedmuntaz@yahoo.com

This paper was recommended for publication in revised form by Regional Editor Osman Nuri Uçan

Published by Yıldız Technical University Press, İstanbul, Turkey

INTRODUCTION

The intent of associating speech duration with performance of speaker age estimation came to our attention based on how the length of a conversation helps people to recognize the person they are talking to over a telephone. It takes several seconds or a couple of minutes in phone calls to recognize people from unknown phone numbers. It might be extremely hard to recognize our loved ones only by the first hello sound. But it gets easier and easier as the conversation goes on. In fact it also changes from person to person. Some can recognize quickly and others may not be able to find out even after the end of their conversation. Therefore recognition capability depends not only on the length of speech but also does on the training of the preceptor as well. By the same token speaker age estimation can also depend on length of utterances and training or adaptation of preceptors. Several studies have dealt with varieties of feature extraction techniques, classification and estimation schemes for speaker age estimation. For instance Buket D. Barkana and Jingcheng Zhou proposed pitch-range (PR) based feature set for age and gender classification in their article published in May 2015 [1]. PR achieved accuracies of 92.86%, 83.61%, 83.02%, 73.58%, 67.35%, and 34.33% for middle-aged female, senior female, children, middle-aged male, young female and senior male speakers respectively. Hence PR is the best to classify middle-aged female speakers whereas it failed to classify even half of the available senior male speakers correctly.

It is plausible to believe that listeners older than at least 20 years can judge speaker age at accuracy levels much better than chance. Actually, speaker age estimation relies on numerous perceptual features, including pitch, speech rate, loudness and voice quality [2]. In addition, a significant number of other factors may influence age perception. These can be related to (1) the speaker, e.g. gender, physiological condition and language spoken, (2) the listener, e.g. age, culture and motivation, (3) the speech sample, e.g. stimulus type (such as read or spontaneous speech) and length, and (4) the task, e.g. whether it involves classifying speakers into two or more age groups or making an exact estimation of age. The numerous studies with listening experiments are difficult to compare because of differences in subjects, speech material and method. Owing to the probable influence of these factors, there is no single answer to the question of how accurate listeners' judgements of speaker age actually are. Furthermore, although most studies carried out so far have found pitch and speech rate to be the most important perceptual cues to speaker age, some recent studies have suggested that spectral qualities may also be important.

Mel frequency cepstral coefficient (MFCC) has been repeatedly used with several back end schemes for speaker age estimation either in the form of regression or classification. MFCC gives the effective results in clean environment but it drops the results in noisy environment [3]. MFCC achieved 88.57% accuracy to identify speakers in noise mismatch condition with neural network classifier, in a noisy environment. The modified group delay function has been used in conjunction with the standard MFCC-based feature for speech recognition and improved phoneme recognition accuracy by 2% absolute over the best baseline MFCC-based system [4].

A study conducted on speaker age estimation using i-vectors briefly mentioned that speech duration, environment, recording device and channel conditions are some of the technical factors which influence the estimation accuracy of speaker age [5]. In other words, in a typical practical scenario, the quality of the available speech signal and the recording conditions are not controlled and the duration of the speech signal may vary from a few seconds to several hours. We have used the aGender database which consists of utterances as short as below half a second and as long as a little over 10 seconds [6].

Cepstral trajectories corresponding to lower (3-14 Hz) modulation frequencies provide best discrimination [7]. Accordingly modulation cepstrum achieved 50.2% overall accuracy for 7 age classes. This fact hinted the possible front-end options which can contribute for speaker age estimation positively.

Muller, Christian, Frank Wittig, and Jorg Baus argue that context and user diversity needs to be accommodated in order to come over the challenge of universal usability in their 2003 article for European conference on speech communication and technology [8]. They indicated that unlike linguistic features acoustic and prosodic features can be extracted relatively easily before the actual speech recognition process. In [9], the role of language variations in speakers for age estimation is investigated. Accordingly, a multilingual speaker age estimation study conducted on 6 languages widely spoken in South Africa mean absolute error (MAE) ranging from 7.7 to 12.8 years for same languages predictors is obtained. On the other hand the cross-language predictor offered an MAE value of 14.5 years. This clearly shows that the best predictions are obtained when training and test utterances are drawn from the same language. On the contrary to this a study conducted on human listeners showed that being able to speak someone's language does not help to predict his/her age [9].

Recently deep neural networks (DNN) have been giving impressive performances in numerous speech processing applications. To mention some of their achievements in speaker age estimation; a mean absolute error (MAE) of 4.9 is achieved by applying x-vector neural network architecture on national institute for standard and technology (NIST) SRE08 dataset for training and NIST SRE10 for evaluation [10]. This architecture uses a series of time delay layers (TDNN) followed by a temporal pooling layer which summarizes the feature sequence into a single fixed dimension embedding. The embedding is fed into a series of feed-forward layers to predict the age value. The x-vector alone outperformed the i-vector baseline by 14%. In addition combining both the i-vector and x-vector improved the i-vector baseline result by 9%. Recurrent neural networks (RNNs) are used in temporal information or sequences and hence suitable for speech processing applications unlike the convolutional neural networks (CNNs) which are widely used in image and pattern recognition. A typical RNN; long short-term memory (LSTM-RNN) is proposed for speaker age estimation and are able to deal with short utterances (from 3 to 10 s) [11]. LSTM-RNN can be easily deployed in a real-time architecture and has been tested using data from NIST speaker recognition evaluation 2008 and 2010 data sets. It has been compared to state-of-the-art i-vector systems and achieved from 18% to 28% relative improvement in terms of mean absolute error.

(Büyük and Arslan, 2018) contributed an article on how the combined effect of long term and short term features ımproves the performance of speaker age classification [12]. Gaussian mixture model (GMM) combined with DNN offered the best standalone performance with 74.22% accuracy and a 4% increase is achieved bringing short term and long term features together, which exhibited 77.5% accuracy. These authors proposed feed forward neural networks for age identification in another study which offered 74% classification accuracy [13]. An investigation of multilingual speaker age classification has been made with DNN and other two classifiers [14].

Recently, age dependent insensitive loss has been used to estimate speaker age and short duration speech data has been employed for speaker profiling [15-16]. The former study reported improvements in the mean absolute error (MAE) value ranging 3.1% to 5.2% using the NIST SRE 10 database as an evaluation set. And the later achieved MAE values of 5.2 years, and 5.6 years for male and female speakers respectively. Regression algorithms have also been proposed for emerging fast surrogate models in order to mitigate the high CPU computational cost due to massive simulations which is regarded as the major bottle neck in electromagnetic (EM) antenna design [17]. In a much anticipatory study to DNN surrogates, fully connected regression model (FCRM) based on Bayesian optimization is proposed for accurate modelling of frequency selective surfaces [18]. DNN based regression models have shown impressive performances with the increase of hardware processing speed recently. In this regard, a fully-connected regression model which combines a DNN surrogate with automated architecture and hyper-parameter determination using Bayesian optimization is proposed for improved modeling in microwave structures [19].

In a vast majority of studies speaker age estimation has not been dealt in utterance length perspective emphatically. Although we do not have control on the nature of incoming speech during a real time application, we can design a robust estimation algorithm which accommodates speech length diversity. Therefore, we decided to address this gap in our study. Accordingly the aim of this study is:

1) to prove that increasing the number of frames has a positive impact on speaker age estimation performance and to investigate on training-test utterance length mismatch effects. This helps to develop and implement a robust system for speaker age estimation.
2) to find out which set of frames contribute better performances; the frames found at the beginning, centre or end of utterances.

Universal back ground model (UBM) is used before i-vector extraction to determine the universal supervector [20]. Using the universal supervector we defined a total variability matrix (TV) which compensate the session and space variability [21]. The factor analysis and concepts related to i-vectors are well explained in a lecture note organized by academic members in Hong Kong university [22]. Five feature extraction schemes are employed as a frontend approach. Four of these being magnitude spectral features which only differ in filter banks and the remaining is a phase spectral feature called modified group delay (MODGD)[23].

The remaining part of this study is organized as follows: section 2 presents front end analysis and some feature extraction schemes applied in our experiments, section 3 focuses on the regression techniques used in our setup where a brief discussion of radial basis functions (RBF) and least squares support vector regression are made. Section 4 presents the experimental setup, and results are presented in section 5 and a discussion about outcomes follow in the same section. Finally a concluding remark is made in section 6. Resources and reference materials are listed at the end of the article.

FRONT-END ANALYSIS AND FEATURE EXTRACTION

The front-end analysis begins with accessing each audio file from its location and ends with generating a set of real numbers called features. Figure 1 below shows the complete diagram starting from speech production at the articulatory system which consists of lung, larynx and vocal tract. Once the speech is produced it is recorded. The recording device as well as the communication medium affects the performance of speaker age estimation. There is a missing block though which basically applies a pre-emphasis filter on each entire utterance before cutting it in to pieces using framing windows. The pre-emphasis filter is applied for the following purposes:

1) to amplify high frequency components
2) to balance the frequency spectrum
3) to avoid numerical problems during discrete Fourier transform (DFT) operations and
4) to improve the signal to noise ratio (SNR) of speech utterances [24].

Given a discrete speech sequence $x[n]$ accessed using a Matlab command, { [*x*, *fs*]= *audiored*(*wavFilePath*); }, the pre-emphasis is defined as:

$$y[n] = x[n] - \alpha x[n-1] \tag{1}$$

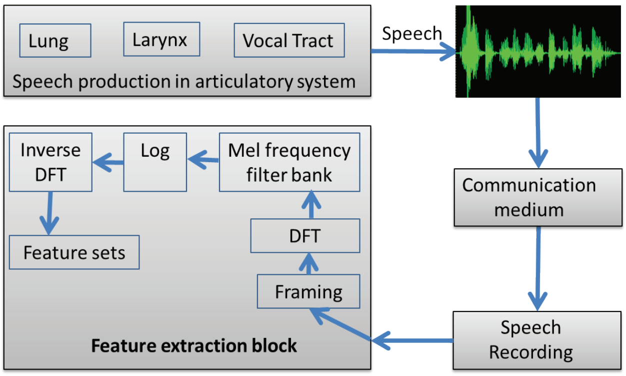


Figure 1. Speech production to feature extraction cycle.

$y[n]$ is processed in the last and bigger block set shown in Figure 1 above. The first thing to do after pre-emphasis is framing to split the longer speech segment in to pieces to get relatively static frames using a hamming window of length 20 milliseconds and 10 milliseconds overlap given in equation (2) [25]. Following windowing the frequency domain representation of each frame is determined using DFT since spectral features depend on spectrums rather than time domain amplitudes.

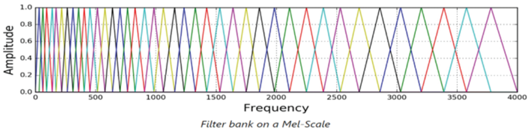
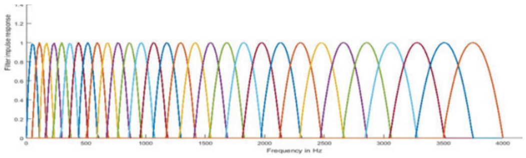
The 3rd step after framing and DFT Inside the feature extraction block shown in Figure 1 above is employing filter banks. The filter banks can be designed in a variety of ways based on their shape, number and spacing. The four magnitude-based feature sets are mel frequency cepstral coefficient (MFCC), parabolic filter mel frequency cepstral coefficient (PFMFCC) [26], rectangular filter cepstral coefficient (RFCC) and linear frequency cepstral coefficient (LFCC). The shape of these feature sets are displayed in Table 1 shown below with their impulse response filter function shown at the left of the table. RFCC offered an impressive performance for an experiment aimed at detecting replay or spoofing attack [27]. MFCC and PFMFCC use mel scale to split the range of frequencies between the minimum and the maximum while LFCC and RFCC use linear scales in our experiments.

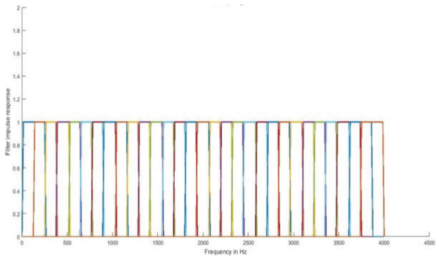
$$\mathrm{w[n]} = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{L-1}\right), & 0 \le n \le L-1 \\ 0, & otherwise \end{cases} \qquad \textbf{(2)}$$

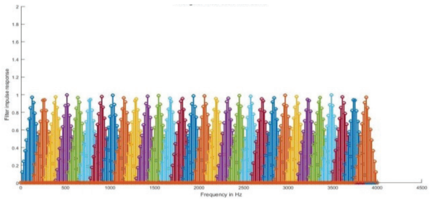
The next step after splitting the entire speech segment in to pieces is calculating the DFT of each frame. Frames are assumed to be more static than the entire utterance given their short duration 20 milliseconds, it is reasonable to believe so.

$$X[j,k] = \sum_{n=0}^{N-1} y[n]w[n-j]\,e^{-j\frac{2\pi kn}{N}} \;, \quad 0 \le k \le L-1 \; and \;\; j = 1,2,\dots,M \qquad (3)$$

Table 1. List of filter banks in magnitude spectral feature extractions

Features	**Filter banks used**
MFCC: $H[i,k] = \begin{cases} \frac{k-f_{i-1}}{f_i - f_{i-1}} \;, & f_{i-1} \le k \le f_i \\ \frac{f_{i+1}-k}{f_{i+1}-f_i} \;, & f_i \le k \le f_{i+1} \\ 0 \;, & otherwise \end{cases}$	
PFMFCC: $H[i,k] = \begin{cases} -(\frac{k-f_i}{f_{i+1}-f_i})^2 + 1, & f_{i-1} \le k \le f_{i+1} \\ 0, & otherwise \end{cases}$	

RFCC:
We used trapezoid functions
`fft_matris(i,j) = trapmf(fft_fr(j),[F_mel(i),F mel(i),...`
`F_mel(i+2),F_mel(i+2)]);`
	

LFCC: Functions are the same as MFCC but the spacing is linear rather than mel scale as in MFCC.	

The DFT of the j^{th} frame is $X[j,k]$ where M is the number of frames in a certain utterance which varies according to the duration of the speech, N is the DFT point and $w[n-j]$ is the j^{th} framing hamming window having equal length L for all j given by equation (2). The shortest speech segment is with $M = 92$ and the longest is $M = 1077$ frames long for the female dataset where as $M = 111$ and $M = 1055$ frames long respectively for the male dataset. The average length of the children dataset is 269 frames long.

Once we peak our filter banks H[i,k] from Table 1 above, we apply them on DFT of each frame X[j,k] as defined by equation (4) below. The effect is summed over the lower L_i and upper U_i frequencies of each particular filter bank i.

$$MF[i] = \frac{1}{A_i}\sum_{k=L_i}^{U_i}|H[i,k]\boldsymbol{X[j,k]}| \quad , \ i = 1,2,3,\ldots,30 \tag{4}$$

$$A_i = \sum_{k=L_i}^{U_i}|H[i,k]|^2 \tag{5}$$

$$\boldsymbol{X[j,k]} = \frac{1}{N}|X[j,k]|^2 \quad , \ power\ spectrum\ of\ each\ frame \tag{6}$$

Calculating the M static features using cepstral transform which in turn applies logarithm on $MF[i]$ values uses either DCT or inverse DFT for eventual generation of these features. We use equations (7), (8), and (9) to determine the static, dynamic and acceleration features respectively. The dynamic and acceleration features are commonly called as delta and double delta respectively.

$$MFCC[m] = \frac{1}{R}\sum_{i=1}^{R}\log\ MF[i])\,cos\left[\frac{2\pi}{R}\left(i+\frac{1}{2}\right)m\right] \ ,m = 1,2,3,\ldots,M \tag{7}$$

$$delta[t] = \frac{\sum_{n=1}^{Q} n(MFCC_{t+n} - MFCC_{t-n})}{2\sum_{n=1}^{Q} n^2} \ , \ Q = 2 \tag{8}$$

$$double\_delta[t] = \frac{\sum_{n=1}^{Q} n(delta_{t+n} - delta_{t-n})}{2\sum_{n=1}^{Q} n^2} \ , \ Q = 2 \tag{9}$$

where R is the total number of band bass filter banks used which is 30, t is an index used to identify adjacent frames whereas $t + Q$ and $t - Q$ are indexes of the farthest neighbour frames involved in calculation of the t^{th} frame dynamic and acceleration features. R could be a parameter of interest for further study.

In addition to the magnitude-based spectral features shown in Table 1 above, our study included several investigations on a phase-based spectral feature called modified group delay (MODGD) for speaker age estimation using LSSVR. These feature sets are extracted from the phase component of the DFT of speech frames as defined in (10). MODGD is the negative rate of change of the phase spectrum $\theta(\omega)$ with respect to frequency ω [4] [23].

$$\tau(\omega) = -\frac{d(\theta(\omega))}{d\omega} \tag{10}$$

$\theta(\omega)$ is taken from X(jω) written in its magnitude and phase components using polar representation as $|X(j\omega)|e^{j\theta(\omega)}$. This feature set is used for speaker age estimation with LSSVR in our study for the first time.

i-Vector

After the acoustic features are extracted for each frame i-vectors are determined for the entire speech segment of each utterance. For this computation UBM is trained using our development set and a universal mean super vector $\boldsymbol{m}$ is obtained and a total variability matrix T which compensates the space and session variability is generated using the Microsoft speaker recognition research (MSR)[28] identity matlab toolbox [29]. Finally the i-vectors ω_i for the i_{th} utterance consisting of acoustic feature frames X_i discussed in Table 1 above are determined as shown below. We carried out two tier feature extraction operation; first the acoustic features X_i and then the i-vectors ω_i [30]. The tables and figures in section 5 display the performance of our regression model which is presented in section 3 below on i-vector sequences generated from (MFCC, PFMFCC, RFCC, FCC and MODGD) feature sets. While the acoustic features are a matrix of N rows by M columns where N represents number of features in each frame and M denotes the number of frames in each utterance after removing non speech frames, i-vectors are column vectors consisting of fixed number of sequences. We used 200 vector sizes in our experiments and 256 for universal super vector in UBM training

$$\begin{gathered} X_i = \boldsymbol{m} + T\omega_i \\ \omega_i = (X_i - \boldsymbol{m})T^{-1} \end{gathered} \tag{11}$$

REGRESSION

Regression is a supervised learning problem where there is an input, X, an output, Y, and the task is to learn the mapping from the input to the output. The approach in machine learning is that we assume a model: $Y = f(X)$ defined using a set of parameters θ where $f(.)$ is the model and Y is a number in regression and is a class code (e.g., 0/1) in the case of classification. $f(.)$ is the regression function or in classification, it is the discriminant function separating the instances of different classes. Machine learning algorithms optimize the parameters, θ, such that the approximation error is minimized, that is, our estimates are as close as possible to the actual values given in training sets [31].

Multiple regression process incorporates multiple features or covariates to determine the outcome. The choice of this regression technique depends on the relationship between each feature and the outcome variable too. As a particular input sample involves vectors or matrix a multivariate regression is carried out using matrix computation.

The multivariate regression model can be approached in the following ways:

Let us assume the following notations for simplicity of our approach:

Input vector assumed to be $x \in \mathbb{R}^d$

Output value assumed to be $y \in \mathbb{R}$

Parameters $\beta = (\boldsymbol{\beta_0}\ , \ \boldsymbol{\beta_1}, \boldsymbol{\beta_2}, \boldsymbol{\beta_3}, \ldots., \ \boldsymbol{\beta_d})^T \in \mathbb{R}^{d+1}$

Then we set up the model as

$$f(x) = \boldsymbol{\beta_0} + \textstyle\sum_{j=1}^{d} \boldsymbol{\beta_j x_j} \ = \ x^t\beta \tag{12}$$

Given the training data $D = \ \{(x_{\boldsymbol{i}}, y_{\boldsymbol{i}})\}_{i=1}^{N}$ the least square cost or loss $L(\beta)$ is defined as

$$L(\beta) = \textstyle\sum_{i=1}^{N}(y_{\boldsymbol{i}} - f_i(x))^2 = \sum_{i=1}^{N}(y_{\boldsymbol{i}} - \boldsymbol{x_i^t}\beta)^2 \ \ = \ \|\boldsymbol{Y} - \boldsymbol{X}\beta\|^2 \tag{13}$$

Here

$$\boldsymbol{X} = \begin{Bmatrix} \boldsymbol{x_1^t} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \boldsymbol{x_N^t} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{1}, & \boldsymbol{x_{1,1}}, & \boldsymbol{x_{1,2}}, & \ldots., & \boldsymbol{x_{1,d}} \\ \boldsymbol{1}, & \boldsymbol{x_{2,1}}, & \boldsymbol{x_{2,2}}, & \ldots., & \boldsymbol{x_{2,d}} \\ & & \ldots\ldots\ldots\ldots\ldots. & & \\ \boldsymbol{1}, & \boldsymbol{x_{j,1}}, & \boldsymbol{x_{j,2}}, & \ldots., & \boldsymbol{x_{j,d}} \\ & & \ldots\ldots\ldots\ldots\ldots. & & \\ \boldsymbol{1}, & \boldsymbol{x_{N,1}}, & \boldsymbol{x_{N,2}}, & \ldots., & \boldsymbol{x_{N,d}} \end{Bmatrix} \ , \ \ \boldsymbol{Y} = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_N \end{Bmatrix}$$

$$\boldsymbol{x_j^t} \ = \{\boldsymbol{1}, \ \boldsymbol{x_{j,1}}, \ \boldsymbol{x_{j,2}}, \ \ldots., \ \boldsymbol{x_{j,d}}\}$$

In addition, N and d are the number of samples in our training set and number of features in each sample respectively. To find a minimum loss an optimization technique is applied and optimal parameters β_j that could lead to a linear model are obtained. For this purpose we need to take partial derivative of the loss function in (12) with respect to β.

$$0_d^t = \frac{\partial L(\beta)}{\partial \beta} \ \ = -2(\boldsymbol{Y} - \boldsymbol{X}\beta)^{\boldsymbol{T}}\boldsymbol{X} \ \leftrightarrow \ 0_d^t \ = \ \boldsymbol{X^T X}\beta - \ \boldsymbol{X^T Y} \tag{14}$$

$$\beta = \ (\boldsymbol{X^T X})^{\boldsymbol{-1}}\boldsymbol{X^T Y} \tag{15}$$

Non-linear model is required when a linear model does not fit in available data to replace the independent variables x_j in equation (12) above with a non-linear function $\varphi(\boldsymbol{x_j}) \in \mathbb{R}^k$ and these functions are named as non-linear features here after. The new estimation function $f(x)$ is expressed in a similar fashion in equation (16) below. Nonlinear regression provides the most flexible curve-fitting functionality. However it can take considerable effort to choose the nonlinear function that creates the best fit for the particular shape of the curve.

$$f(x) = \textstyle\sum_{j=1}^{k} \varphi(\boldsymbol{x_j})\boldsymbol{\beta_j} \ = \ \varphi(x)^T\beta \tag{16}$$

The expression for optimal parameter β remains the same as shown in equation (15) above except the independent variables x_j^T are replaced by non-linear features $\varphi(x_j^T)$. These features depend on the choice of the model attempt to apply. Selected non-linear models will briefly be discussed in this section. Therefore, $\boldsymbol{X} = \{\varphi(x_1), \varphi(x_2), \varphi(x_3), \ldots.. \varphi(x_i), \ldots\ldots \varphi(x_i)\}^T$ for non-linear models or kernels.

One of the most widely used non-linear models in acoustic modeling is the radial basis function (RBF) which is briefly discussed below. Some kernel functions are shown in Table 2 below.

Table 2. List of some kernel functions for regression models

Kernel functions	Description
Linear	$\varphi(x, \omega) = x^T\omega$
Polynomial :	$\varphi(x, \omega) = (\eta + \ x^T\omega)^d$
Radial basis function:	$\varphi(x, \omega) = e^{-\frac{\|x-\omega\|^2}{2\sigma^2}}$
Splines :	$\mathrm{f(x)} = \sum_{\mathrm{j=1}}^{\mathrm{m+k+1}} \beta_{\mathrm{j}}\mathrm{g_j(x)}$, where k is polynomial order, and m is number of polynomial kernel function $g_j(x)$. The approximation f(x)is a fitting functions while β_j's are coefficients.
Wavelets:	$W_{ij} = \varphi_j(x_i)$ where, $x_i = \frac{i}{n}, i = 1, 2, 3, \ldots\ldots.., n$
String – kernel :	These kernels measure the similarity of pairs of strings [32]. For instance, assuming *str*1 and *str*2 as two strings the kernel measure *φ*(*str*1, *str*2*)* would provide a higher value for higher similarity between *str*1 and *str*2.

Radial Basis Function (RBF)

Neural Networks are very powerful models for classification tasks. But we used them for regression in our study to develop the least square support vector regression (LSSVR). We used our training dataset and we projected the training trend into the test set to make predictions. Regression has been discussed earlier at the beginning of this section and has many applications in wide range of areas including in finance, physics, medicine, meteorology, biology and many others. Radial basis function (RBF) is a neural network architecture commonly used in non-linear regression as well as function approximation in addition to their popular application in classification. An RBF network is a 2-layer network apart from the output layer. We have an input that is fully connected to a hidden layer. The output of the hidden layer is taken to perform a weighted sum to get our final output. Hence, its architecture is not deep. Unlike the neurons in conventional neural networks and deep neural networks (DNN), the neurons in RBF networks contain Gaussian RBF. And hence the Gaussian RBFs are used as the activation functions.

The figure above shows some Gaussian densities with different parameters and their combined effect. These Gaussian densities makeup radial basis function. As it can be clearly observed in the figure, the values of individual densities are bound to [0,1]. The resultant density depends on the means and variances of all the individual densities. The individual densities follow normal distribution whose mathematical expression for univariate and multivariate random variables is given by equation (17) and (18) respectively.

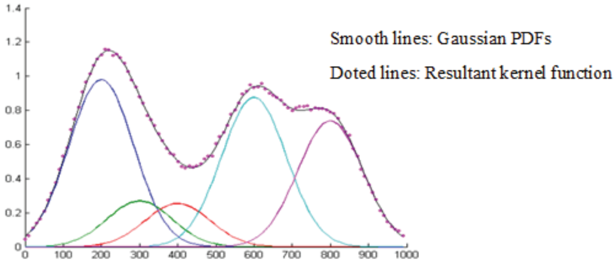


Figure 2. Gaussian probability density functions forming kernel functions.

$$N(x{:}\,\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \qquad (17)$$

For a multivariate dataset like ours equation (17) is rewritten as:

$$N(\boldsymbol{X}{:}\,\boldsymbol{\mu},\textstyle\sum) = \frac{e^{-\frac{1}{2}(X-\mu)^T\sum^{-1}(X-\mu)}}{\sqrt{(2\pi)^d|\Sigma|}} \qquad (18)$$

Here $\sqrt{(\boldsymbol{X}-\boldsymbol{\mu})^T\sum^{-1}(\boldsymbol{X}-\boldsymbol{\mu})}$ is the Mahalanobis distance and $|\Sigma|$ is the determinant of the covariance matrix of the dataset *X*.

The mean μ determines the center of the symmetrical graph where half of the whole dataset lays to the left of this vertical line and the other half remains to the right of the symmetrical vertical line representing (x = μ). In Figure 2, the Gaussians have different colors and are weighted differently. Taking the sum of all the probability densities gives a continuous function. The parameter which indicates the closeness of individual data sample is the variance σ^2 or in some literatures is the standard deviation σ which is the square root of the variance. Accordingly a large variance shows a wide variation between data samples therefore the resulting bell curve is shorter in height, flat and wide open. On the other hand, a small variance results in a long, steep in shape and indicates very close individual data samples.

Technically, the probability density function (pdf) described in equations (17) and (18) is used to determine the probability of observing an input x or X in multivariate case given that specific normal distribution. However the bell-curve properties of the Gaussian are more important than the fact that it represents a probability distribution for the application of radial basis function (RBF). It is logical to observe an inverse relation between the maximum of the probability density function, which occurs at (x = μ) and evaluated as $(\sqrt{2\pi\sigma^2})^{-1} = \frac{1}{\sigma\sqrt{2\pi}}$ since the total area covered by the bell curve is supposed to be unity. A linear combination of Gaussian density functions with a certain number of centers and a wide range of variances can be used to approximate any function.

The centers c_j for each kernel function $\varphi_j(.)$ of the RBF are determined using k-mean algorithms. The regression process begins after initializing the necessary variables and parameters. The input at the very beginning is a set of features for each sample speaker in our study which is given by $X^t = \{x_1, x_2, x_3, . . . , x_d\}$ where *d* is the dimension of the input or number of features representing each speaker. The approximation function which produces the estimate age $Y = \{\hat{y}_1, \hat{y}_2, \hat{y}_3, . . . , \hat{y}_N\}$ where *N* represents the number of utterances in the specified dataset, is given by:

$$F(x) = \textstyle\sum_{j=1}^{K} \omega_j \varphi_j(x\,,\ c_j) + b \qquad (19)$$

$$\varphi_j(x\,,\ c_j) = e^{-\frac{1}{2\sigma_j^2}(\|x-c_j\|)^2} \qquad (20)$$

The sum of the squared error is given by the cost formula shown below.

$$error = \textstyle\sum_{i=1}^{N}(y^{(i)} - F(x^{(i)}))^2 \qquad (21)$$

Now we apply optimization algorithms step by step to find optimal weight parameters ω_j and the bias *b*. For this purpose we take the partial derivative of the error function with respect to ω_j and bias *b* separately to compute optimal weights and optimal bias respectively.

$$\frac{\partial(error)}{\partial\omega_j} = \frac{\partial(error)}{\partial F}\frac{\partial F}{\partial\omega_j} = \frac{\partial}{\partial F}[\ \textstyle\sum_{i=1}^{N}\left(y^{(i)} - F\left(x^{(i)}\right)\right)^2 \cdot \frac{\partial}{\partial\omega_j}[\sum_{j=1}^{K}\omega_j\varphi_j(x\,,\ c_j) + b\,]$$

The new weights will be updated considering the error they have incurred in the previous iteration using the learning rate η. The result of the partial derivative is given by:

$$\nabla(error) = -(\textstyle\sum_{i=1}^{N}(y^{(i)} - F(x^{(i)}))) \cdot (\sum_{j=1}^{K}\varphi_j(x\,,\ c_j))$$

Then we deduce the updated weights are $\omega_j \leftarrow \omega_j + \eta(y^{(i)} - F(x^{(i)}))\varphi_j(x\,,\ c_j)$.

Similarly for the new bias parameter we take the partial derivative of the error function with respect to b. Algorithm 1 below procedurally implements the instructions depicted in Figure 3.

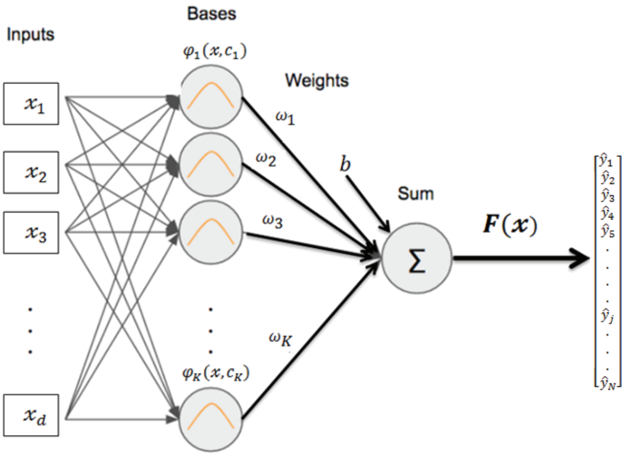


Figure 3. Radial basis functions and age estimation process for multivariate data.

$$\frac{\partial(error)}{\partial b} = \frac{\partial(error)}{\partial F}\frac{\partial F}{\partial b} = \frac{\partial}{\partial F}[\ \sum_{i=1}^{N}\left(y^{(i)} - F\left(x^{(i)}\right)\right)^2 \cdot \frac{\partial}{\partial b}[\sum_{j=1}^{K}\omega_j\varphi_j\left(x\ ,\ c_j\right) + b\ ]$$

$$\frac{\partial(error)}{\partial b} = (y^{(i)} - F\left(x^{(i)}\right) \text{ Giving } b \leftarrow b + \eta\left(y^{(i)} - F(x^{(i)})\right)$$

Algorithm 1: The process of training weight and bias parameters

```
Step. 1.    Define the radial basis function RBF:
            def rbf(x, c, s):
            return np.exp(-1 / (2 * s**2) * (x-c)**2)
Step. 2.    Define the approximation function using
            superposition of weighted radial basis functions (RBFs)
            def predict(self, X):
                y_pred = []
                for i in range(X.shape[0]):
                a = np.array([self.rbf(X[i], c, s) for c, s, in
                zip(self.centers, self.stds)])
                F = a.T.dot(self.w) + self.b
                y_pred.append(F)
            return np.array(y_pred)
Step. 3.    Compute the error subtracting values generated by approximation function from actual
            values
Step. 4.    Update the weights and bias parameters
Step. 5.    Continue the process until the error reaches
            a specified level or a certain iteration is
            reached.
```

EXPERIMENTAL SETUP

We created a separate text file containing only actual chronological speaker age values. The age distribution ranging from 6 years old child to 80 years old senior speaker is shown in Figure 4 below. The utterances vary in duration; the shortest is 0.28 seconds in the children dataset and the longest is 10.79 seconds in the female dataset. The entire audio is sampled at 8 KHz.

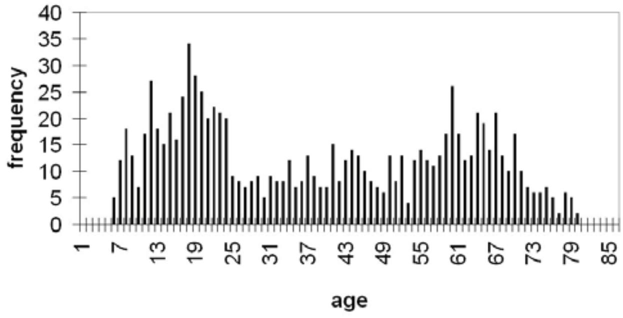


Figure 4. Age structure of the aGender database [6].

We used Matlab on an Intel Core i-3 CPU processor with 8 GB RAM and 2.5 GHz processing speed to carry out the simulation. It approximately takes 4 hours to complete the simulation for a single feature type. Therefore, it has nearly consumed a total of 32 hours to complete the experiment for both male and female datasets. However, we have later confirmed that a core i-7 CPU processor with 16 GB and 3.2 GHz processing speed reduced the processing time 10 times,

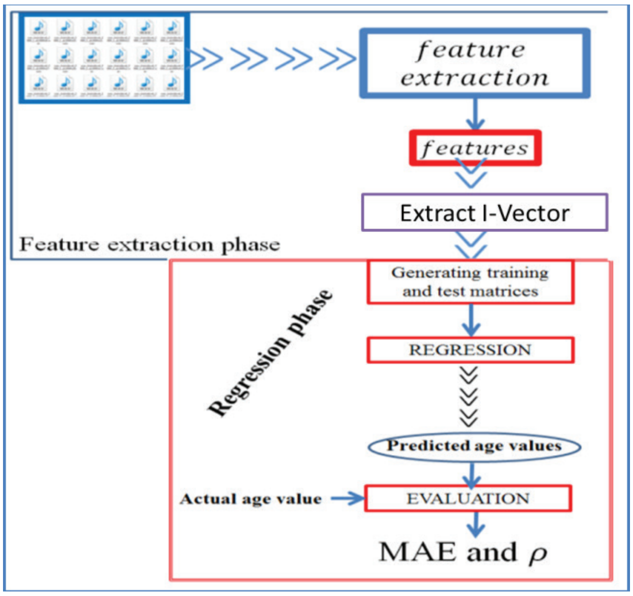


Figure 5. Audio to age prediction regression process.

The experiment begins with accessing training and test utterances from the directory they have been stored. Right after a feature set is generated from a certain audio segment, an i-vector is extracted from it and is appended to a big matrix containing all i-vectors for the entire training audios using Matlab concatenation command. LSSVR follows after the training matrix is created. We used 100 kernels for the RBF in a single layer network and a learning rate of 0.01 in the LSSVR framework. This is basically equivalent to a single layer conventional neural network with 100 neurons whose activation function is equal to the Gaussian radial basis function. An evaluation process is made based on the estimated speaker ages for test sets using the regression model and the actual chronological speaker age. The performance of our model is evaluated using mean absolute error (MAE) and Pearson correlation coefficient (ρ) given by (22) and (23) respectively.

$$\boldsymbol{MAE} = \frac{1}{N}\sum_{i=1}^{N}|(y_i - \tilde{y}_i)| \tag{22}$$

$$\rho = \frac{1}{N-1}\sum_{i=1}^{N}\left(\frac{y_i - \mu_y}{\sigma_y}\right)\left(\frac{\tilde{y}_i - \mu_{\tilde{y}}}{\sigma_{\tilde{y}}}\right) \tag{23}$$

RESULTS AND DISCUSSION

Results

Experimental results for performance evaluation of i-vectors extracted from 4 magnitude-based and one phase-based spectral features applied on LSSVR regression model are presented in this section, which basically exploits RBF for speaker age estimation. Except MFCC, the remaining feature sets have never been tested for speaker age estimation with LSSVR to the best of our knowledge. Performance evaluation of speaker age estimation for the female dataset is displayed on Figures 6 and 7 in terms of mean absolute error (MAE) and ρ respectively.

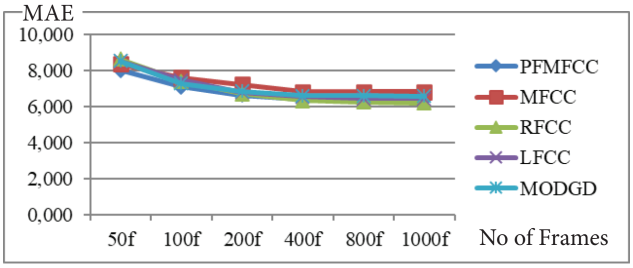


Figure 6. Effect of number of female speech frames on LSSVR performance using the i-vectors extracted from five feature sets expressed in terms of MAE.

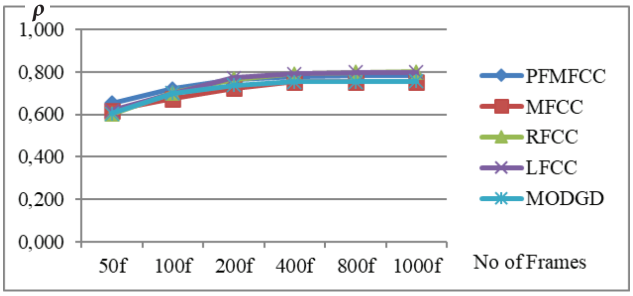


Figure 7. Effect of number of female speech frames on LSSVR performance using the i-vectors extracted from five feature sets expressed in terms of Pearson correlation coefficient ρ.

The experimental results for speaker age estimation performance in terms of MAE and Pearson's correlation coefficient (ρ) for male dataset are shown in Figures 8 and 9 respectively. These results further strengthen the performance improvement observed in the female dataset shown in Figures 6 and 7 above. The results proved that increasing speech duration (number of frames) would likely improve the speaker age estimation performance.

The experimental results displayed in Figures 6 to 9 above indicate significant changes in performance as the number of frames increases from 50 frames to 400 frames but, it slowly saturates and the changes remain sluggish for nearly all the feature sets after the 500th frame. Therefore, we can consider 4-5 seconds of speech as optimal to get an acceptable performance at least compared to speech durations ranging 0.5 to 10 seconds. We can avoid considerable amount of computational overhead with this optimal number of frames. However we suggest further research on speech durations longer than 10 seconds.

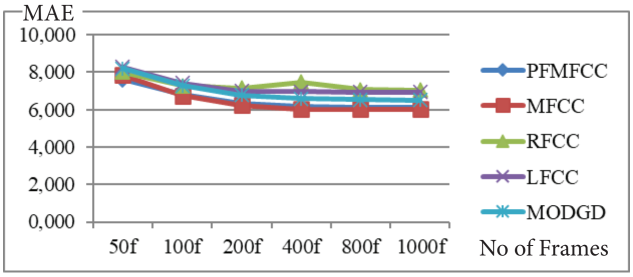


Figure 8. Effect of number of male speech frames on LSSVR performance using the i-vectors extracted from five feature sets expressed in terms of MAE.

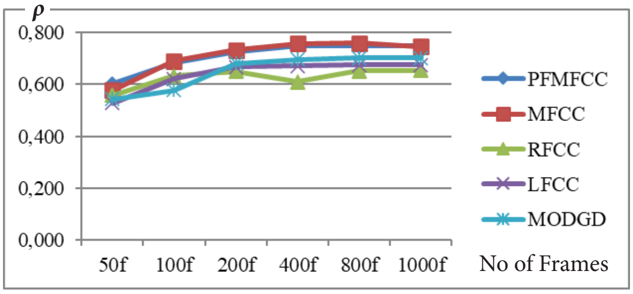


Figure 9. Effect of number of male speech frames on LSSVR performance using the i-vectors extracted from five feature sets expressed in terms of Pearson correlation coefficient ρ.

The performance evaluation of the proposed algorithm for different combination of mismatches in utterance length is shown in Tables 3 and 4 for female and male datasets respectively. The values located along the diagonals represent matched performances while off-diagonal values represent performance of mismatch experimental results. In addition, the bold values show the best performances among mismatch experimental setups.

Performance comparison of LSSVR model on direct acoustic features and i-vectors as a second tier feature extraction for utterance lengths of 3, 5 and 10 seconds as short medium and long speech utterances respectively is shown in Table 5. A similar analysis is carried out employing artificial neural networks (ANN) and long short-term memory (LSTM neural networks [11]. However, the study was carried out on the NIST SRE 2008/2010 database instead of aGender.

Table 3. i-Vector followed by LSSVR performance evaluation for utterance length mismatch in terms of MAE for female dataset. (rows are training and columns are test frames)

		a) PFMFCC-ivector-LSSVR			b) MFCC-ivector-LSSVR		
		Number of test frames			Number of test frames		
		200	500	1000	200	500	1000
Number of training frames	200	6.6330	6.4602	6.4346	7.2220	6.8266	6.7969
	500	6.7843	6.4467	**6.4255**	7.2520	6.8224	**6.7935**
	1000	6.8052	6.4717	6.3630	7.2793	6.8467	6.8160
		c) RFCC-ivector-LSSVR			d) LFCC-ivector-LSSVR		
		Number of test frames			Number of test frames		
		200	500	1000	200	500	1000
	200	6.7310	6.3952	6.3650	6.8040	6.4187	6.3945
	500	6.6540	6.2619	**6.2190**	6.6744	6.3622	**6.3350**
	1000	6.6500	6.2607	6.2190	6.6514	6.3380	6.4570

Table 4. i-Vector followed by LSSVR performance evaluation for utterance length mismatch in terms of MAE for male dataset. (rows are training and columns are test frames)

		a) PFMFCC-ivector-LSSVR			b) MFCC-ivector-LSSVR		
		Number of test frames			Number of test frames		
		200	500	1000	200	500	1000
Number of training frames	200	6.3400	6.1726	6.7498	6.2309	6.1849	6.1544
	500	6.4308	6.1736	6.1982	6.1500	6.0680	6.0424
	1000	6.3770	**6.1555**	6.1285	6.1354	**6.0387**	6.0147
		c) RFCC-ivector-LSSVR			d) LFCC-ivector-LSSVR		
		Number of test frames			Number of test frames		
		200	500	1000	200	500	1000
	200	7.1306	7.0903	7.0717	6.9685	6.9723	6.9873
	500	7.1092	7.0687	**7.0480**	6.9442	6.9328	6.9380
	1000	7.0919	7.0715	7.0459	6.9337	**6.9219**	6.9243

Table 5. Performance of our estimation algorithms on short, medium and long utterances for female and male datasets

Duration	Female MAE/ρ	Feature set used and improvement	Male MAE/ρ	Feature set used and improvement
3s				
Feature + LSSVR	11.704/0.580	RFCC, 44.98%	11.093/ 0.500	MFCC, 44.77%
Feat +i-vector + LSSVR	6.439/0.781	improvement	6.127/0.746	improvement
5s				
Feature + LSSVR	11.628/0.592	RFCC, 46.15%	11.063/0.504	MFCC, 45.15%
Feat +i-vector + LSSVR	6.262/0.796	improvement	6.068/0.7526	improvement
10s				
Feature + LSSVR	11.555/0.594	RFCC, 46.179%	11.012/ 0.506	MFCC, 45.38%
Feat +i-vector + LSSVR	6.219/0.799	improvement	6.015/0.746	improvement

Note: Feat = {MFCC, RFCC, LFCC, PFMFCC, MODGD}

DISCUSSION

Our experimental results clearly show that longer speech segments performed better than the shorter ones. This proves that our assumption on longer speech durations contribute positively is right. All the feature sets showed consistency in proving positive impact of utterance length for speaker age estimation for both male and female datasets. In fact listeners usually need to hear enough before they recognize the speaker or predict speaker age. There was an attempt in a previous article which used LSTM for speaker age estimation to explore performance of 3 speech durations (3 s, 5 s and 10 s) [11]. However, the emphasis was on backend mechanisms rather than the nature of the speech.

Due to the nature of our database which consists of an average length of 2 seconds we preferred to peak only fifty frames at the beginning, middle and end of each utterance and perform effect of their position in speaker age estimation performance. It turns out that the frames at the middle showed insignificant but better performance than the remaining two positions. This could mainly be due to noise at the beginning and end of utterances. And it implies noise reduction can improve the overall performance. Accordingly MAE values of 8.004, 7.845 and 7.981 are recorded using the PFMFCC and i-vector LSSVR method for beginning, middle and end female frames respectively. The same method offered MAE values of 7.598, 7.452 and 7.577 for beginning, middle and end male frames respectively

Utterance length is addressed slightly using least square support vector regression for acoustic feature sets on national institute of standard and technology (NIST 2008 and 2010) speaker recognition evaluation (SRE) database [5]. MAE values of 10.63, 9.77 and 6.47 as well as Pearson correlation coefficient values of 0.76, 0.8 and 0.18 were obtained for male, female and children datasets respectively by applying LSSVR speaker age estimation on acoustic features in a recent study using the same aGender database we used in our experiments. Our (i-vector+ LSSVR) approach has improved these performances by 43.41% and 36.34% for male and female datasets using longer utterances for the same database, respectively.

Training with medium duration utterances and testing with longer ones showed a relatively better performance compared to other combinations in the female dataset. In addition this combination is second best if not the best choice among the list of combinations we used in male dataset. As training dataset is large our experimental ret that using medium utterances could save processing time without affecting the performance significantly. Training with longer utterances on the other hand, takes much time for simulation and could fail to recognize patterns from short duration test samples.

CONCLUSION

In all our experiments, we observed increasing speaker age estimation performance as the length of utterances increases, irrespective of backend regression models or choice of feature extraction techniques. Hence our presumption is confirmed. In addition, the improvement till 400 frames is significant and easily observable whereas it eventually slows down after the 400 frame middle threshold. Hence, the improvement saturates once the increase in number of frames reached the 400 frames point. Accordingly we conclude that the 400-500 frames range is an optimal speech duration based on our simulation results.

The best performance by the center frames compared to other positions indicates that the frames at the beginning and end of speech segments suffer from noise and none speech frames mostly occur at the two ends. This gives more insight in to the significance of speech quality in speaker age estimation.

The mismatches in length of training and test utterances offer poor performances compared to longer utterances for both datasets. However, our regression models trained with medium utterances are capable of getting more information, which is good enough to estimate speaker age from longer test sets than the other mismatch options. Whereas a model trained with longer utterances lacks knowledge of certain patterns from short or medium test sets to make good decisions. This result is consistent in female dataset across all features however; the inconsistency in the male dataset needs further investigation. Our future study focuses on applying these mismatch preliminary investigations to more end to end speaker age estimations such as x-vector architecture using time delay neural networks.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

REFERENCES

[1] Barkana BD, Zhou J. A new pitch-range based feature set for a speaker's age and gender classification. Appl Acoust 2015;98:52−61. [CrossRef]

[2] Schötz S. Perception, Analysis and Synthesis of Speaker Age. thesis/docmono. Lund University; 2006.

[3] Chauhan PM, Desai NP. Mel Frequency Cepstral Coefficients (MFCC) based speaker identification in noisy environment using wiener filter. In: 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE); Mar 2014; pp. 1−5. [CrossRef]

[4] Murthy HA, Gadde V. The modified group delay function and its application to phoneme recognition. In: 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03); Apr 2003; Vol. 1, p. I−68.

[5] Bahari MH, McLaren M, Van hamme H, van Leeuwen DA. Speaker age estimation using i-vectors. Eng Appl Artif Intell 2014;34:99−108. [CrossRef]

[6] Burkhardt F, Eckert M, Johannsen W, Stegmann J. A database of age and gender annotated telephone speech. Proceedings of the Language and Resources Conference (LREC); 2010.

[7] Ajmera J, Burkhardt F. Age and gender classification using modulation cepstrum. In: Odyssey; 2008; pp. 25.

[8] Muller C, Wittig F, Baus J. Exploiting speech for recognizing elderly users to respond to their special needs. In: Eighth European conference on speech communication and technology; Geneva, Switzerland; 1-4 Sep. 2003. [CrossRef]

[9] Braun A, Cerrato L. Estimating speaker age across languages. In: Proceedings of ICPhS; 1999; Vol. 99; pp. 1369−1372.

[10] Ghahremani P, Khorrami P, Lajevardi SM, et al. End-to-end Deep Neural Network Age Estimation. Interspeech 2018;2018:277−281. [CrossRef]

[11] Zazo R, Nidadavolu PS, Chen N, Gonzalez-Rodriguez J, Dehak N. Age estimation in short speech utterances based on LSTM recurrent neural networks. IEEE Access 2018;6:22524−22530. [CrossRef]

[12] Büyük O, Arslan ML. Combination of long-term and short-term features for age identification from voice. Adv Electr Comput Eng 2018;18:101−108. [CrossRef]

[13] Büyük O, Arslan LM. Age identification from voice using feed-forward deep neural networks. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1−4). [CrossRef]

[14] Büyük O, Arslan LM. An Investigation of Multi-Language Age Classification from Voice. In BIOSIGNALS (pp. 85-92), 2019. [CrossRef]

[15] Kitagishi Y, Kamiyama H, Ando A, Tawara N, Mori T, Kobashikawa S. Speaker Age Estimation Using Age-Dependent Insensitive Loss. In 2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC) (pp. 319−324).

[16] Kalluri SB, Vijayasenan D, Ganapathy S. Automatic speaker profiling from short duration speech data. Speech Commun 2020;121:16−28. [CrossRef]

[17] Jacobs JP, Koziel S. Variable-fidelity modeling of antenna input characteristics using domain confinement and two-stage Gaussian process regression surrogates. Int J Numer Model 2020;33:e2758. [CrossRef]

[18] Calik N, Belen MA, Mahouti P, Koziel, S. Accurate modeling of frequency selective surfaces using fully-connected regression model with automated architecture determination and parameter selection based on bayesian optimization. IEEE Access 2021;9:38396−38410. [CrossRef]

[19] Koziel S, Mahouti P, Calik N, Belen MA, Szczepanski S. Improved modeling of microwave structures using performance-driven fully-connected regression surrogate. IEEE Access 2021;9:71470−71481. [CrossRef]

[20] Přibil J, Přibilová A, Matoušek J. GMM-based speaker age and gender classification in Czech and Slovak. J Electrical Eng 2017;68:3–12. [CrossRef]

[21] Reynolds DA, Quatieri TF, Dunn RB. Speaker verification using adapted Gaussian mixture models. Dig Sign Proces 2020;10:19–41. [CrossRef]

[22] Mak MW. Lecture Notes on Factor Analysis and i-Vectors, Technical Report and Lecture Note Series, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Feb. 2016.

[23] Hegde RM, Murthy HA, Gadde VRR. Significance of the modified group delay feature in speech recognition. IEEE Trans Audio Speech Lang Process 2006;15:190–202. [CrossRef]

[24] Vergin R, O'Shaughnessy D. Pre-emphasis and speech recognition. In: Proceedings 1995 Canadian Conference on Electrical and Computer Engineering 1995;2:1062–1065. [CrossRef]

[25] Harris FJ. On the use of windows for harmonic analysis with the discrete Fourier transform. Proc IEEE 1978;66:51–83. [CrossRef]

[26] Osman MM, Büyük O. Parabolic filter mel frequency cepstral coefficient and fusion of features for speaker age classification. Sigma J Eng Nat Sci 2020;38:2177–2191.

[27] Hanilci C. Features and classifiers for replay spoofing attack detection. In: 10th International Conference on Electrical and Electronics Engineering (ELECO). IEEE; 2017 Nov 30 - Dec 2; Bursa, Turkey.

[28] Sadjadi SO. MSR Identity Toolbox. Seattle, WA, USA: Microsoft; 2013.

[29] Moon TK. The expectation-maximization algorithm. IEEE Signal Process Mag 1996;13:47–60. [CrossRef]

[30] Li W, Fu T, Zhu J. An improved i-vector extraction algorithm for speaker verification. J Audio Speech Music Proc 2015;2015:18. [CrossRef]

[31] Alpaydın E. Introduction to machine learning. 2nd ed. Cambridge, Mass.: MIT Press; 2010.

[32] Lodhi H, Saunders C, Shawe-Taylor J, Cristianini N, Watkins C. Text classification using string kernels. J Mach Learn Res 2002;2:419–444.