Mehran University Research Journal of Engineering and Technology
Vol. 40, No. 2, 279 - 297, April 2021
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2102.03

An Arabic Mispronunciation Detection System Based on the Frequency of Mistakes for Asian Speakers

Faria Nazir[1a], Muhammad Nadeem Majeed[1b], Mustansar Ali Ghazanfar[2], Muazzam Maqsood[3]

RECEIVED ON 23.04.2019, ACCEPTED ON 15.09.2020

ABSTRACT

Over the last few decades, the field of artificial intelligence and machine learning has evolved. Due to the advancement in these fields, much work has been done to assist language learning with the help of computers called Computer-Assisted Language Learning (CALL). Mispronunciation detection is one of the significant tasks of the CALL system. An efficient mispronunciation detection model has a positive impact on the life of second language learners by providing phoneme level feedback. In this paper, we introduce the phone grouping technique for mispronunciation detection that is based on mistakes probability. We consider mispronunciation detection as a classification problem, traditionally for this purpose, a separate classifier is trained for each phoneme mistake that requires a lot of memory and time. Instead of training a separate classifier, we group the phoneme based on their mistakes probability that helps in reducing the number of the classifiers to be trained and also saves memory and time. We use the Support Vector Machine (SVM) classifier and test the results on the Arabic dataset (28 Phonemes). The performance of our proposed method is evaluated by using accuracy. The results of the model are evaluated using the confusion matrix and gives an accuracy of 88%. Our approach outperforms the existing systems developed for Arabic phonemes in terms of accuracy and is also time/memory efficient.

Keywords: Mispronunciation Detection, Support Vector Machine, CALL

1. INTRODUCTION

Due to advancements in technology world has become a global village. People can easily communicate with one another living in different parts of the world, so there is an increasing demand for new language learning [1]. Arabic is the fifth prevalent language in terms of native speakers [2]. Speech technology has improved dramatically over the last decade, so by using speech technology and machine learning techniques, many intelligent CALL systems are developed which are more useful and intelligent than ever. These systems detect pronunciation mistakes of a learner and provide feedback [3].

There are many tasks performed by these CALL systems which include automatic speech recognition, pronunciation scoring, and mispronunciation detection. Among these different tasks of CALL, mispronunciation detection is the most important task. Many people consider both pronunciation scoring and mispronunciation detection the same, but actually, both these tasks are different from each other [4]. In

[1] Department of Software Engineering, University of Engineering and Technology Taxila, Pakistan.
Email: [a] faria.nazir@uettaxila.edu.pk (Corresponding Author), [b] nadeem.majeed@uettaxila.edu.pk
[2] Department of Computer Science, The School of Architecture, Computing and Engineering, University of East London, London, United Kingdom. Email: Mghazanfar@uel.ac.uk
[3] Department of Computer Science, COMSATS Institute of Information and Technology, Islamabad, Attock Campus, Pakistan. Email: muazzam.maqsood@cuiatk.edu.pk

pronunciation scoring, an overall pronunciation score is calculated on the global level. These global scores are not very useful when used in pronunciation training because in pronunciation training people are more concerned with the nature of errors made in pronunciation rather than the overall scoring. Pronunciation scoring determines the speaker's proficiency in the language and used to test different pronunciation applications. Mispronunciation detection requires calculating the pronunciation scores on the local level which is usually phoneme level. So both pronunciation scoring and mispronunciation detection have different goals and different results.

On the other hand, mispronunciation detection can point out pronunciation mistakes and provide feedback at the phoneme level. There are many reasons for mispronunciation such as the speaker's native speaking style or speaker's unfamiliarity with words and so on. Pronunciation errors are classified into phonemic errors and prosodic errors [5, 6]. Phonemic errors are related to phones, phonemes may be substituted with another similar phoneme, some phones may be added or deleted, and all these changes make a difference in sound. Prosodic errors on the other hand are difficult to categorize because they include errors based on stress, rhythm, and annotation [4].

The second language learner makes pronunciation mistakes frequently. Particularly when the non-native language contains a few phonemes that are not found in foreign native language, second language learners replace these phonemes with ones existing in their native language. Automatic detection of such errors is a fundamental and essential procedure in CALL frameworks [7].

In this paper, we propose a classifier-based approach for mispronunciation detection of Arabic phonemes. We consider mispronunciation detection as a classification problem. Traditionally to detect mispronunciation we train a separate classifier for each phone mistake, a separate classifier is trained, that takes a lot of memory and time for training. To cater to this problem, we categorize the data into groups so we train only one classifier for the whole group instead of training a separate classifier for each pronunciation mistake. This grouping technique enhances the performance of classifiers as well as it is more efficient in terms of space and time.

The remaining paper is organized as follows: In section 2, we present a detailed overview of related approaches for mispronunciation detection. In section 3 we describe our proposed methodology and details of each step are also provided. In section 4, we deliberate the experiments and results and also provide a comparison of our approach with state of the art approaches.

2. LITERATURE REVIEW

Mispronunciation detection systems can be categorized into three main groups: posterior probability-based methods, classifier-based methods, and Deep-learning-based methods.

2.1 Posterior probability-based Methods:

The initial work in this field started in the 1990s and different scoring algorithms were proposed for error detection. Kim *et al.* [8] presented three Hidden Markov Model (HMM) based scores: 1) HMM-based log-likelihood scores 2) HMM-based posterior probability scores, which later on turned into an accepted standard, 3) segment duration based scores. Similarly, the Goodness of Pronunciation (GOP) score utilizes a log-probability-based score. In posterior probability-based methods, different methodologies have been used. Witt *et al.* [9] introduced the GOP strategy to check the quality of pronunciation and the combined standard GOP strategy with a few refinements that provide improvements in scoring performance. GOP score can be computed in equation (1) as

$$\mathrm{GOP}(s,q) = \frac{|\log(p(q|s)|}{d} = \frac{\frac{|p(St|q)_{p(q)}|}{\sum_{i=1}^{Q} p(St|qi)_{p(qi)}}}{d} \qquad (1)$$

where;
s = the sequence of observation
q = the labels
d = time interval of the audio examination in the form of frames.

Zhang *et al.* [10] proposed the Scaled Log-Posterior Probability (SLPP) and weighted phone SLPP method to improve the degree of pronunciation quality. Hindi *et al.* [11] calculated the GOP score to identify pronunciation mistakes in five Arabic phonemes that were frequently mispronounced by non-native Arabic speakers. In the same manner, Kawai *et al.* [12] utilized log-probability scores in constrained arrangement mode. Extended versions of probability-based scores were effectively utilized by Mak *et al.* [13]. Posterior probability-based methods can detect the pronunciation quality but these scoring algorithms are not capable to detect the type and exact location of error so for this purpose classifier-based techniques are used.

2.2 Classifier-based Methods

In classifier-based approaches, Truong *et al.* [14] used Linear Discriminant Analysis or a decision tree for mispronunciation detection of three sounds (A, Y, and X) that are frequently mistaken by L2-students (foreign/second language students) of Denmark. Ito *et al.* [15] proposed a decision-based clustering technique to enhance the accuracy of error detection. They developed the clusters of pronunciation rules and defined a threshold for each cluster. Amdal *et al.* [16] differentiated among short and long vowels of speech by using acoustic-phonetic features and consolidated them in a Linear Discriminant Analysis (LDA) classifier. Georgoulas *et al.* [17] used SVM to detect the speech articulation of sound and for the classification of speech sounds. Strik *et al.* [3] compared four different approaches (GOP, decision tree, LDA-APF (Acoustic phonetic feature) and LDA-MFCC (Mel Frequency Cepstral Coefficient)) for automatic pronunciation error detection. The comparative analysis showed that LDA-APF and LDA-MFCC both strategies yielded preferred outcomes over GOP scores and the decision tree. Wei *et al.* [4], presented the SVM framework, with pronunciation space models to enhance execution. Tongmu Zhao *et al.* [18] developed a system for error detection on eight confusing phonemes of Chinese using SVM classifier with structural features. Yoon *et al.* [19] presented the confidence scoring method and landmark-based SVMs method to detect mispronunciation. The combination of both methods did not provide significant improvement when data was not trained appropriately. Yang *et al*. [20] used six different classifiers (decision trees, random forest, gradient boosting, SVM with a linear kernel, SVM with radial basis function and Binomial logistic regression) for classification and among those support vector classifiers and logistic regression performed best. Maqsood *et al.* [21] developed acoustic-phonetic feature-based Computer Assisted Pronunciation Training (CAPT) system for most confusing Arabic phoneme pairs (/ ط / vs ت / /) and (/ ح / vs / خ / or / هـ /). They applied four classifiers (Random forest, Naïve Bayes, Ada-boost, and K-NN) on a dataset of 200 speakers and compared the performance of the classifiers and the result showed that Random Forest classifier performed better as compared to other classifiers. Maqsood *et al.* [22] developed a system for mispronunciation detection of five phonemes of Arabic using the SVM classifier.

2.3 Deep-Learning based Methods:

In Deep-learning based approaches, Lee *et al.* [23], used Deep Belief Network (DBN) posteriorgrams to detect the word level mispronunciation. DBNs have been effectively utilized for phone recognition with input coefficients that are MFCCs or filterbank [24, 25]. Li *et al.* used Deep belief networks for lexical stress detection, and demonstrated that the DBN performed better than the Gaussian Mixture Model. Hu *et al.* [11] proposed the Deep Neural Network (DNN) based approach to acoustic modeling of a tonal language. Joshi *et al.* [26] proposed a method for vowel mispronunciation detection using DNN with cross-lingual training. Gao *et al.* [27], aimed at the robust detection of Pronunciation Erroneous Tendency (PET) and proposed the DNN-HMM framework for error detection and used three types of acoustic features namely MFCC, Perceptual Linear Predictive (PLP) and filter band. Hu *et al.* [28] enhanced the performance of mispronunciation detection by training the acoustic model using a Deep neural network instead of conventional GMM-HMM based training. They used the Neural Network (NN) based Logistic Regression classifier, where a neural network with shared hidden layers was used to extract speech features, and two class logistic regression classifiers were trained as phone specific output layer nodes.

Hu *et al.* [29], extended the GOP algorithm from traditional GMM-HMM to DNN-HMM to detect phone-level mispronunciation and tone diagnosis of the L2 learner. Li *et al.* [30] focused on mispronunciation detection on the segmental and sub-segmental levels. They used speech attributes (voicing and aspiration) and Deep neural network classifiers to address mispronunciation detection and diagnostic feedback. At the sub-segmental level, they used speech attribute scores to measure the pronunciation quality, and then they integrated scores using NN classifiers to produce segmental level pronunciation scores. Li *et al.* [31] proposed Acoustic Phonological Model that used multi-distribution DNN for mispronunciation detection and diagnostics.

2.4 Features used for mispronunciation detection

Apart from methods used to detect pronunciation mistakes, an important aspect of a mispronunciation detection technique is to extract discriminative features that efficiently represent pronunciation variations. Different types of features have been used by researchers including confidence measures and log-likelihood scores based features, acoustic-phonetic features, statistical features, structural features, and combination of many features. However, the most discriminative pronunciation features are still to be identified. Literature has highlighted that the performance can be achieved through the use of better classifiers [17, 19, 25] but that causes an increase in the computational cost. Therefore, there is a need for a system that can effectively and efficiently detect and classify mistakes in phonemes. We have used acoustic-phonetic features in our research to improve the efficiency of the system. Table 1 represents the details of features used by the researchers for pronunciation training.

3. PROPOSED METHODOLOGY

The flowchart of our proposed methodology is shown in Fig. 1. The first step is to extract the features from the audio signals labeled with phoneme class C= {c_1, c_2, c_2….c_n}, where n represents a total number of phoneme classes. In the next step, we use pre-processing to clean the data and remove sparsity, and then we apply dimensionality reduction for selecting the most discriminative features. After dimensionality

Table 1: Features used for mispronunciation detection systems

Author	Year	Language	Features	Evaluation metrics		
				Accuracy	EER	F-Score
Witt *et al.* [9]	2000	English	GOP and its refinements	_	_	_
Truong *et al.* [14]	2004	Dutch	ROR (Rate Of Rising), amplitude, highest ROR value, and duration	87%-95%	_	_
Ito *et al.*[15]	2005	English	MFCC, ΔMFCC	90%	_	_
Georgoulas *et al.* [17]	2006	Greek	The standard deviation of the wavelet coefficient, entropy of the normalized entropies	77.78%	_	_
Bolanos *et al.* [32]	2008	English	phone-level features MFCC along with single and double derivatives, energy	-	22%	-
Zhang *et al.* [10]	2008	Mandarin	Enhanced GOP score	-	16.3%	-
Amdal *et al.* [16]	2009	Norwegian	Acoustic-phonetic features	92.3%		
Strik *et al.* [3]	2009	Dutch	Acoustic features, MFCCs	-	-	-
Li *et al.* [33]	2009	English	extended MFCC features with standardized formant trajectory information		17.5	
Wei *et al.* [4]	2009	Mandarin	Log-likelihood ratios	-	-	-
Su-Youn Yoon [19]	2010	English	spectral features			0.67
Al Hindi *et al.* [11]	2014	Arabic	GOP	87% - 100%		
Franco *et al.* [34]	2014	Spanish	Acoustic features		8%	
Yang *et al.* [35]	2016	Mandarin, English	MFCC, Formants	_	-	-
Maqsood *et al.* [21]	2017	Arabic	Pitch, energy, MFCC feature, Single and double derivative of MFCC, zero-crossing rate and spectral features	-	-	-

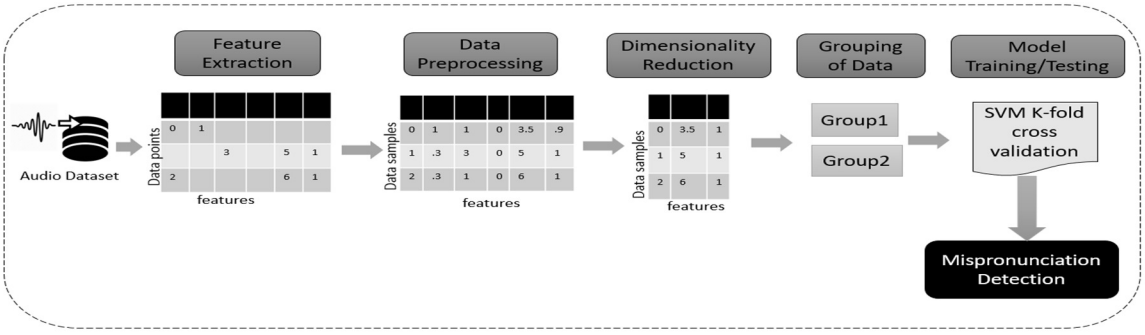

Fig 1: Flowchart of the proposed methodology

reduction, we divide the Arabic dataset into two groups: frequently mistaken phonemes and less mistaken phonemes. Finally, we train the model using SVM, Naïve Bayes, and KNN on those discriminative features to get optimal results. We use the k-fold cross-validation in which test data from each fold is passed to the trained classifier to find the phoneme label.

Algorithm 1 represents the sequence of steps. First of all, we extract the acoustic-phonetic features from an audio file present in dataset D (Line 2). If dataset D contains missing values then we apply pre-processing steps like data cleaning and impute missing values (Line3 and 4). After that, we apply a feature selection algorithm to choose the most discriminative features for mispronunciation detection (Line 6,7). After feature selection step we divide the dataset into two groups, frequently mistaken phonemes P_{fm} and less mistaken phonemes P_{lm} (Line 9). We train two separate classifiers for each group using k-fold cross-validation (Line 11, 12) to detect mispronunciation. We present the detail of each step in the subsection.

Algorithm 1: Phonemes Mispronunciation Detection Algorithm

Input: Arabic Phonemes dataset D
Output : Phoneme Class C= 1,2,3……28
STEPS: 1. **For each** Audio signal in the dataset, **D** do 2. Extract Acoustic-Phonetic features(APF) 3. ***If*** D has missing values 4. Apply Pre-processing steps 5. ***end*** 6. Let SF represent the selected features from dataset D 7. SF = (f_1,f_2,f_3,…….f_{135}) 8. ***end*** 9. Let dataset D be dividing into Frequently mistaken phonemes $\mathbf{P_{fm}}$ and Less mistaken phonemes $\boldsymbol{P_{lm}}$ 10. Use **k=10** fold cross-validation for classifier learning 11. LearnClassifier for P_{fm}= (SF, DataLabels_P_{fm}) 12. LearnClassifier for P_{lm}= (SF, DataLabels_P_{lm}) 13. ***Return*** Phonemes Class to detect mispronunciation

3.1 Feature Extraction:

To extract features from an audio file, first, we divide the speech signal into small frames of 20ms and 10ms, overlap and apply signal processing techniques to extract acoustic-phonetic features like pitch, MFCC, energy, and formats from these frames. These features are called low-level descriptors. We also extract global statistical features like mean, min, standard deviation, and slope by combining different frames and these features represent the global trend of a signal. Table 2 shows the acoustic-phonetic features used in this research work.

Table 2: Acoustic-phonetic features

Features	Description
Pitch	Pitch in Hertz
Roll-Off	Steepness
Entropy	Entropy Features
Cepstrum	14 MFCC and their single and double derivatives
Zero Cross	Number of Zero cross
Low Energy	The low energy of each frame
Root Mean Square	Root mean square energy
Spectrum	Spectral features
Global statistical features	mean, Period_frequency, slope, Period_amplitude, Period_Entropy, standard deviation

We use these feature for our research work as these are the main features used in the literature work as listed in Table 1 and we also combine the statistical features with other features to obtain good results. Each local level descriptor combines with each global statistical feature to form multiple features. For example pitch, a local level descriptor combines with all global descriptors mean, Period_frequency, slope, amplitude, standard deviation, period and entropy to form six features (Pitch_mean, Pitch_std, Pitch_slope, Pitch_preiod Frequency, Pitch_period Amplitude,

Pitch_Entropy) and similarly all other features are combined to form 284 features. We provide details of each feature as follows.

3.1.1 Pitch

We define the pitch of the sound as a frequency of vibrations. When compelled air from the lungs passes through the choral folds sound is produced. The fundamental frequency or pitch of the sound is that frequency where vocal tracts vibrate. Automatic speech recognition applications widely use the pitch of the sound. It has also been proved useful for mispronunciation detection.

3.1.2 Roll-Off

The roll-off is defined as the frequency below which 95% of the power of the signal is determined. It is also a measure of spectral shape and produces higher values for high frequencies. Therefore, it can be assumed that a strong correlation exists between these features. The roll-off is computed in equation (2) as

$$\sum_{k<f} X_k = 0.95 \sum_k X_k \quad (2)$$

Here X_k represents the discrete Fourier transform of $x(t)$. The left-hand side of the equation denotes the summation of the power underneath the frequency value f and the right side of condition shows the 95% of the aggregate vitality of the signal.

3.1.3 Entropy

The entropy feature has been used in speech recognition applications to detect the voiced and unvoiced region of a speech signal. Spectral entropy is a measure of signal complexity. It captures the formats or the peakiness of a distribution. Formants and their locations assume an imperative part in speech track. We compute the entropy of speech signals in equation (3) as

$$E(s) = -\sum_{i=1}^{N} p_i \times \log_2 p_i \quad (3)$$

where

$$p_i = \frac{X_i}{\sum_{i=1}^{n} X_i} \quad (4)$$

Xi represents speech signal Power Spectral Density (PSD), and pi represent normalized PSD, Xi can be calculated in equation (5) as

$$X_i = \frac{1}{N} |S_i|^2 \quad (5)$$

where S_i shows the spectrum of the speech signal.

3.1.4 Cepstrum

Mel scale is a scale of pitches that are equal in distance from each other employed by MFCC's. The normal frequency f in hertz can be changed to the Mel scale range by equation (6) ss follow

$$M(f) = 1127.01048 \log\left(1 + \frac{f}{700}\right) \quad (6)$$

The Cepstrum is a measure used to gain information from a person's speech signal. We apply logarithm on the signal spectrum and then take inverse Fourier transform to obtain Cepstrum. Mathematically it is expressed in equation (7).

Backward Fourier Change (IFT) of the logarithm of the evaluated range of a flag is Cepstrum.

$$C(n) = DFT^{-1}[\log|DFT\{X(n)\}|] \quad (7)$$

where DFT represents the Discrete Fourier transform and DFT^{-1} is the Inverse DFT. The Cepstrum contains the information rate of change in spectrum bands. Spectrum is first transformed by the Mel scale to give MFCC's which are used for speech recognition. We retain the high coefficients if we are interested in excitation signals and on the off chance that we are occupied with the vocal tract, we keep the low coefficients. Cepstral coefficients are a compressed representation of the spectral envelope. It can be shown that cepstral coefficients are not correlated. This information is useful that is why speech recognition applications widely use cepstral coefficients.

3.1.5 Zero-Crossing Rate:

Zero-Crossing Rate (ZCR) is the extent of how often a signal passes the zero axes or in other words, it

counts the number of times in a given frame a signal amplitude changes sign from positive to negative and vice versa. ZCR is a time-domain feature and is a very robust and discriminative feature to differentiate sound signals. We compute ZCR for signal S with length T in equation (8) as follows:

$$\mathrm{ZCR} = \frac{1}{\mathrm{T}}\sum_{t=1}^{\mathrm{T}}|s(t) - s(t-1)| \quad (8)$$

where
$s(t) = 1 \quad t \geq 0$
$s(t) = -1 \quad t < 0$

Zero cross value for the periodic sound is low, and its value for noisy sound is high. The zero-crossing rate is a time-domain feature that is determined by the signal frequency. Furthermore, to notice zero crossings of the input speech signal, the sampling rate should be very high. Another important aspect is to normalize the input signal before calculating the zero-crossing. The zero-crossing rate is an important parameter for mispronunciation detection techniques. Zero cross value for the periodic sound is low, and its value is high for noisy sounds.

3.1.6 Energy features

In a speech signal, the power of the signal at a given time is called energy. Energy can also be defined as the pressure exerted by the lungs and passed through the vocal track. The signal amplitude differs with time due to variation in pronunciation. The spoken section amplitude changes altogether when contrasted with an unspoken section of the speech signal. Correct pronounced phonemes have different amplitude variation as compared to the mispronounced phonemes. These amplitude variations are represented by short-time energy, so energy is considered as a potential feature to discriminate speech signals. The energy of the discrete-time signal $S(t)$ is computed in equation (9) as

$$\mathrm{E_s} = \sum_{t=-\infty}^{\infty}|\mathrm{Sg}(t)|^2 \quad (9)$$

where $\mathrm{Sg}(t)$ represents the time signal power and is computed in equation (10) as

$$\mathrm{Pw} = \lim_{T\to\infty}\frac{1}{2T+1}\sum_{t=-\infty}^{\infty}|\mathrm{Sg}(t)|^2 \quad (10)$$

Low short-term energy can be characterized as the number of speech frames whose short-time vitality esteem is not as much as the 0.5 times of the normal short-time vitality in one-moment. We compute energy in equation (11) as:*

$$\mathrm{E_{st}} = \frac{1}{2\mathrm{T}}\sum_{t=0}^{\mathrm{T}-1}\mathrm{sgn}(0.5\mathrm{STE_{avg}} - \mathrm{STE}(t) + 1) \quad (11)$$

$$\mathrm{STE_{avg}} = \sum_{t=0}^{\mathrm{T}-1}\mathrm{STE}(t)$$

where
T = total number of frames,
(t) = short time vitality of the t^{th} frame
$\mathrm{STE_{avg}}$= average short time vitality in a one-second

3.1.7 Root Mean Square

Root Mean Square (RMS) value represents the average power of a signal and it is related to the amplitude of a signal. We compute RMS by squaring the signal amplitude, averaged over time-period and then the square root of the result is calculated in equation (12):

$$\mathrm{RMS} = \sqrt{\frac{1}{\mathrm{n}}\sum_{i=1}^{\mathrm{n}}\mathrm{x_i}} \quad (12)$$

RMS is proportional to the effective power of the signal and an important feature to discriminate correctly pronounced and mispronounced audio signals.

3.1.8 Spectral Features

The spectral features can be expressed as qualities of the speech signal in the frequency domain other than the fundamental frequency f_0. Formants are the most generally utilized spectral features of a speech signal. The speech spectrum is gone through a bank of band-pass channels whose middle frequencies depend on human recognition scales and are exponential. To show these frequencies, analysts have proposed two unique techniques the Bark Mel and Scale. Bark Scale is characterized as in equation (13)

$$\text{Bark}(fr) = 13\arctan(0.00076fr) + 3.5\arctan\left(\frac{fr}{3500}\right)^2 \quad (13)$$

Spectral features are then taken out from these signals [36, 37].

3.2 Data Pre-Processing

Data is often incomplete and inconsistent, so it is essential to preprocess the data before applying any machine learning algorithm, so effective analysis is performed to achieve optimal results. Our dataset is sparse (contains missing values) so we apply a numerical cleaner filter [38] that detects and marks missing values in the dataset and then applies to replace missing value filter. This filter imputes missing values in data by a mean value of data distribution. Cleaned and completed data is then fed to feature selection process that enhances the performance of the training model.

3.3 Feature Selection:

Feature selection aims at picking those features that are discriminative to distinguish among classes. The dataset contains 284 features, but all features are not significant, a subset of discriminative features plays an important role in decision making for classification. In our proposed methodology we use Relief-F attribute evaluation technique for feature selection. Relief-F is an addition to Relief feature selection procedure that deals with only binary classification data, but Relief-F is optimized to deal with multiclass problems [39].

Relief-F attribute selection methods arbitrarily selects any instance Xi and afterward looks for n closest neighbors from the same class called closest hits H, and n closest neighbors each from an alternate class called closest misses M and refreshes the weights of all attributes. The weights are calculated by using equation (14):

$$\text{Weight}(A_i) = \text{Weight}(A_i) - \sum_{k=1}^{n}\text{difference}\frac{(A,X_i,H_k)}{(m,n)} + \sum_{C\neq\text{class}(X_i)}\left[\frac{P(C)}{1-P(\text{class}(X_i))}\sum_{k=1}^{n}\text{difference}(A,X_i,M)\right]/(m,n) \quad (14)$$

This process is repeated m times and the parameter n defined by the user, controls the number of nearest hits and misses. Relief-F attribute evaluation filter provides a ranked list of attributes using the ranker search method, and a threshold is also required. The ranker method is used in combination with the feature evaluation method and ranks features by their separate evaluations. We set n=10 and threshold value to 0.0181 by ranking the data on multiple thresholds and this threshold value provides the best features for further processing. We discard all the values below that threshold from the ranked list of attributes and retain values above that threshold. We retain 135 features on a defined threshold.

Algorithm 2: Feature Selection using Relief-F

Input:	Arabic Phonemes dataset D with Attribute Ai and class values Ci
Output:	Weights of the attributes
STEPS:	
1.	Initialize weights of all attribute to Zero **weight[Attribute]=0**
2.	**for l =1 to k**
3.	Arbitrarily select an attribute **Xa** from training instances
4.	select n closest neighbors (Hits) **Hn**
5.	Find n closest misses Mn from class Cl where Cl does not belong to Cl(Xa)
6.	**for** j= **1 to a**
7.	$\text{Weight}(\text{Attribute}_l) = \text{Weight}(\text{Attribute}_l) - \sum_{n=1}^{m}\text{difference}^{(\text{Attribute},X_a\,H_n)}/_{(k,m)}$
8.	$+\sum_{Cl!=Cl(X_a)}\left[\frac{Pr(Cl)}{1-Pr(Cl(X_a))}\sum_{n=1}^{m}\text{difference}(\text{Attribut}$
9.	end

Algorithm 2 describes the steps to calculate the weighted attribute list. Initially, all the attributes are initialized with zero weight (Line1). Relief-F algorithm arbitrarily chooses an instance Xa (line3), and finds closest hits Hn, n of its closest neighbors from a similar class (line 4), and n closest misses Mn (Cl), nearest neighbors from various classes (lines 5 and 6). It updates the weight estimation. Weights [Attributes] for all instances rely upon their estimations of Xa, hits Hn, and misses Mn (Cl) (lines 6, 7, and 8). The formula for Relief (lines 7 and 8), utilizes a considerable number of hits and all of the misses. The commitment for each class of the misses is weighted with the earlier likelihood of that class Pr

(Cl). As the class of hits is absent in the total, we need to separate every likelihood weight with factor 1− Pr (Cl (Xa)). The procedure repeats for 'k' times.

3.4 Grouping of Phonemes:

The Arabic language consists of 28 phonemes. Non-native Arabic speaker makes pronunciation mistakes due to the number of reasons. A Pakistani national while learning the Arabic language confuses some phonemes (replace one phone with other similar phones called confusing pairs). Fig 2 shows confusing pairs of Arabic mistaken by the Pakistani national. Sulaiman *et al.* [40] discovered Arabic phonemes mispronounced by Pakistani nationals and also found which phoneme sound is replaced or substituted by the other phonemes to provide confusing phoneme pairs.

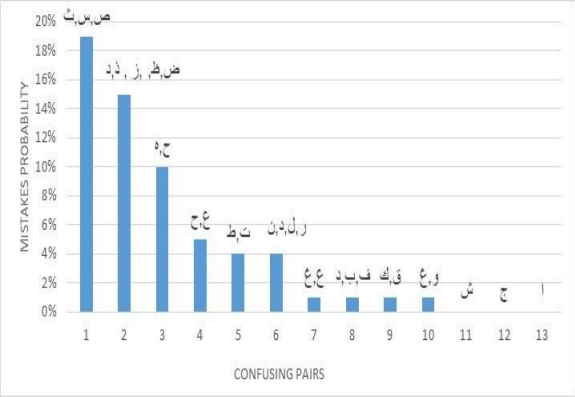

Fig 2: Arabic Confusing phoneme pair for Pakistani

When we take mispronunciation detection as a classification task, we have to train a separate classifier for each confusing pair that needs a lot of memory and training time. To make efficient use of memory and time, we group the phonemes into two main groups. These groups are based on the pronunciation errors made by Pakistani nationals [40]. The phonemes with a high probability of mistakes are placed in Group1 and phonemes that have a low probability of mistakes placed in Group2. We set a threshold that all the phoneme pairs having mistaken probability greater and equal to 10% are placed in Group1 (frequently mistaken phonemes) and phoneme pairs having mistaken probability below that threshold are placed in Group2 (Less mistaken phonemes). Table 3 shows mispronounced phonemes along with their mistakes probability.

3.5 Classifiers

There are many classifiers used for mispronunciation detection. In this research work, we use convolutional neural network features from different layers to detect mispronunciation and for classification of deep features, we use SVM [43], Naïve Bayes, and KNN [44].

3.5.1 SVM Classifier:

The support vector algorithm outputs an optimal hyperplane which categorizes the data by labeled classes. The SVM is best for binary classification, but it is optimized to deal with multiclass problems. In two-dimensional space, the SVM classifier makes a direct hyperplane that isolates the two classes. If the data is not linear so we have to tune the SVM using some parameters like kernel trick. Kernel function transforms the nonlinear data to linear in high dimensional space. We apply SVM on a multiclass dataset that contains correctly pronounced phonemes and mispronounced phonemes.

The earliest utilized approach for SVM multi-classification is one versus all strategy. In this strategy, k SVM models are developed where k is a number of classes. Another significant approach is one versus one. It was presented in [15]. This technique develops (K-1)/2) classifiers where everyone is prepared for information from two classes. One versus one approach is more productive when contrasted with one versus all approach in terms of various classifiers prepared and we utilize one against one strategy. Our dataset consists of Arabic phonemes and some phonemes resemble other phonemes, so it is really hard to classify the phonemes, and data is not linearly separable so for mispronunciation detection of phonemes, we utilize the kernel functions to change the information to higher dimensional space for better classification performance. We have utilized linear, polynomial, and Radial Basis Function (RBF) Kernel and they can be expressed numerically in equation (15) as

Table 3: Pronunciation Error by Pakistani Nationals									
Correct Sound	**IPA Symbols**	**Error sound**	**IPA Symbols**					**Mistakes probability**	**Groups**
ص	/ṣ/	ث ,س	/sˤ /		/θ /			19%	Group1
ض	/ɗˤ/	د, ذ, ز, ظ	/ɗ /	/z/	/d̪/	/ð /		15%	
ح	/h/	ه	/h /					10%	
Other phonemes								<= 5%	Group2

$K(\nu, \upsilon_i) = \upsilon^T \upsilon_i + c$ Linear

$K(\nu, \upsilon_i) = (\gamma \upsilon^T \upsilon_i + c)^d$, $\gamma > 0$ Polynomial (15)

$K(\nu, \upsilon_i) = \exp(-\gamma \|\nu - \upsilon_i\|^2)$, γ>0, Radial Basis

where (ν) represent input vector and (ν_i) shows support vector; c is a constant term and d represents the degree of the polynomial and these parameters are adjustable.

3.5.2 Naïve Bayes

Naïve Bayes classifier is a basic classifier with a strong naïve assumption between features and based on Bayes theorem. Naïve Bayes classifier assumes that features are independent of each other. For the classification purpose, we assume that there are a fixed number of phoneme classes, C∈{c_1, c_2, c_3,…, c_k}, where k is the aggregate number of classes that represent a unique phoneme, each with a fixed set of features. Each sample is characterized by n-dimensional vector Ph = {$ph_1, ph_2, ph_3 \ldots\ldots ph_n$}, where n is the quantity of features {A_1, A_2, A_3,…,A_n}. Given a phone sample Ph, the classifier will anticipate that phone Ph has a place with an accurately articulated class or misspoke class relies upon the highest posterior probability of a class, molded on Ph. The phone sample Ph belongs to class Cl if and only if

$\Pr(Cl_a/Ph) > \Pr(Cl_b/Ph)$, for 1≤b≤a, $b \neq a$ (16)

We search the class that maximizes $\mathrm{pr}\left({}^{Cl_a}/_{Ph}\right)$. The class Cl_a for which $\mathrm{pr}\left({}^{Cl_a}/_{Ph}\right)$ is highest is assigned to the phone sample Ph. Utilizing Baye's hypothesis, the likelihood of a phoneme Ph having a place with specific class Cla can be processed in equation (17) as:

$$\mathrm{pr}\left({}^{Cl_a}/_{Ph}\right) = \mathrm{pr}\left({}^{Ph}/_{Cl_a}\right)\frac{\mathrm{pr}(Cl_a)}{\mathrm{pr}(Ph)} \quad (17)$$

$\mathrm{pr}\left({}^{Cl_a}/_{Ph}\right)$ is the posterior probability of a class, $\mathrm{pr}\left({}^{Ph}/_{Cl_a}\right)$ represents likelihood and pr(Ph) shows evidence. As pr(Ph) is the same for all classes, the only $\mathrm{pr}\left({}^{Ph}/_{Cl_a}\right)\mathrm{pr}(Cl_a)$ needs to be maximized. If the classes from the earlier probabilities, P(Cla), are not known, at that point it is generally accepted that the classes are equally likely. The probability is processed in equation (18) as

$$\mathrm{pr}(Ph/Cl_a) = \mathrm{pr}((ph_1, ph_2, \cdots, ph_n)/Cl_a) = \prod_{k=1}^{n} \mathrm{pr}(ph_k/Cl_a) \quad (18)$$

The probabilities $\mathrm{pr}({}^{ph_1}/_{Cl_a})$, $\mathrm{pr}\left({}^{ph_2}/_{Cl_a}\right)$, $\mathrm{pr}\left({}^{ph_3}/_{Cl_a}\right)$ …… $\mathrm{pr}({}^{ph_n}/_{Cl_a})$ can be evaluated from the training set. Here ph_k denotes the estimation of attribute A_k for phone sample P_h.

3.5.3 K-Nearest Neighbor

KNN is a simple and important instance-based machine learning classification algorithm [41]. KNN is utilized for classification and regression. For classification, an instance is classified by majority votes of its K-Nearest neighbors. The nearest neighbor is selected by the linear search method, but other searching methods are also used. These search methods by default use the Euclidean distance as the selection parameter. K is a positive whole number and on the off chance we set k=1 then the instance is classified based on one closest neighbor that means the instance is allotted the same class as a neighbor. We apply KNN on our dataset to detect correctly pronounced phonemes and mispronounced phonemes. Firstly, we trained the classifier with labeled training data T. In KNN; a training data T is utilized to decide the label of anonymous sample A. KNN classifier finds its K closest neighbors of sample A based on Euclidian distance. If we have two samples a and b

then Euclidian distance can be calculated in equation (19) as

$$|a - b| = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \quad (19)$$

where n represents a number of features describing a and b. The label is assigned to sample A according to majority voting rule which states that that label is assigned to sample A that frequently occurs among nearest neighbors. This classification scheme improves performance by defining nonlinear decision boundaries. We choose the value of K after trying different values of K and find optimal results at K=10.

4. EXPERIMENTAL RESULTS

4.1 Dataset:

There are numerous CALL frameworks available for various languages like English, Mandarin, Dutch, French, and Arabic. We used an Arabic dataset that was recorded from 400 speakers of Pakistani, learning Arabic as their second language. The dataset was recorded in an open office environment with the help of a microphone in stereo using a 44100 Hz sampling frequency. We used Audacity software to record the dataset and for manual segmentation. Data recorded in the office environment contains noise, so we used a fifth-order high pass Butterworth filter to remove low-frequency noise. The reading material includes isolated Arabic consonants. Arabic language consists of 28 consonants and Table 5 shows the details of the phonemes used in this research work. The recording process was held in five different sessions, and each speaker recorded 28 phonemes three times. The repetition per speaker was used to find the best-recorded consonant. The detail of the dataset used for this experiment is given in Table 4. The dataset was created by considering an equal number of male and female speakers as their ages ranged from 10-50 and having different mother tongues like Punjabi, Pushto, and Urdu. Some speakers were highly proficient, and some were at the beginning stage of learning the Arabic language.

The labeling of the dataset was carried out by five Arabic language experts. Each language expert labelled the data separately as correct and incorrect pronunciation classes. If three or more language experts assigned the same label to a certain phoneme then that class (data label) was assigned to that phoneme.

Table 4: Detail of Dataset used in this experiment

	No. of speakers	No. of phonemes
Native	160	4480
Non-Native	240	6489
Total	400	10969

4.2 Evaluation Metrics

To evaluate CALL frameworks, distinctive evaluation matrices like accuracy, recall, precision, and Mean Absolute Error were utilized. In our research work, we use accuracy, Recall, Precision, and Receiver Operating Characteristic (ROC) curve as an evaluation parameter that is based on the confusion matrix. Accuracy can be computed in equation (20):

$$\text{Accuracy} = \frac{N_c}{N_t} \times 100 \quad (20)$$

where N_c represents the number of mispronunciations detected correctly and t represents total number of mispronunciations detected. Recall and Precision can be defined in equation (21) and (22) as

$$\text{Recall} = \frac{T_p}{T_p + F_N} \quad (21)$$

$$\text{Precision} = \frac{T_N}{T_N + F_p} \quad (22)$$

where T_p and T_N represent the number of mispronunciation detected correctly, whereas, F_p and F_N represent number of mispronunciations detected incorrectly. We also use a ROC curve (a graphical representation of sensitivity versus specificity) to evaluate the performance of our model. The area under the curve shows the performance of the model, the greater the area under the curve, the more accurate the model.

4.3 Results and Discussions

This section presents the results for both groups of phonemes, frequently mistaken phonemes (P_{fm}), and less mistaken phonemes (P_{lm}). P_{fm} contains ten

Table 5: Details of Phonemes used for this experiment														
Phonemes	ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	ا
Names	Saad	Sheen	Seen	Zayn	raa'<	That	Daal	khaa'<	haa'<	Jeem	thaa'<	taa'<	baa'<	Alif
IPA Symbols	/ṣ/	/šˤ /	/sˤ /	/z /	/r/	/ð /	/ḍ /	/x /	/h /	/g/	/θ /	/t /	/b /	/ʔ /
Phonemes	ي	و	ه	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط	ض
Names	yaa'<	Waaw	haa'<	Noon	Meem	Laam	Kaaf	Qaaf	faa'<	Rayn	"ayn	zaa'<	taa'<	Daad
IPA Symbols	/y /	/w/	/h /	/n /	/mˤ /	/l /	/k /	/q /	/f /	/ɣ /	/ʕ /	/đ /	/ţ /	/ďˤ /

phoneme classes and P_{lm} contains eighteen phonemes classes**.** Each phoneme represents a unique class. Table 6 represents the list of phonemes included in frequently mistaken phonemes P_{fm} and less mistaken phonemes P_{lm}.

Three different classifiers, Naïve Bayes, K-Nearest Neighbor and SVM were tested for P_{fm} and P_{lm}. We used k-fold cross-validation (k=10) to divide the dataset for training and testing. We used almost equal number of samples for each phone, and we used all the classifiers with default settings.

The performance of all the three classifiers has been evaluated for frequently mistaken phonemes P_{fm}.

Average accuracies for frequently mistaken phonemes P_{fm} are found to be 78%, 79%, 89.8.8% respectively.

The performance of the same three classifiers Nearest Neighbor, and SVM has been evaluated for less mistaken phonemes P_{lm}. Average accuracies for for less mistaken phonemes P_{lm} are found to be 61.6%, 72.1%, 86.7% respectively. The results for each group are presented in Table 7.

The results show that the classifier-based approach efficiently handles mispronunciation detection in both groups. It is also concluded from the results that the SVM classifier outperforms the Naïve Bayes and KNN classifiers. Naïve Bayes classifier shows worst results due to its simplicity and cannot cope up with a complex problem like mispronunciation detection while SVM classifier performs best due to its robustness and generalized ability as shown in Fig 3.

To check the effectiveness of the feature reduction technique we executed our algorithm twice, once using the defined feature reduction technique on P_{fm} and P_{lm} and once without using the feature reduction technique on P_{fm} and P_{lm}. Table 8 represents the effectiveness of the feature reduction technique. Accuracies achieved by Naïve Bayes, KNN and SVM on P_{fm} group without feature reduction technique are 78%, 80%, and 88% respectively and after feature reduction, accuracies achieved by these classifiers are 78%, 81%, and 90% respectively. Accuracies achieved by Naïve Bayes, KNN and SVM on P_{lm} group without feature reduction technique are 57%, 73%, and 85.4% respectively and after feature reduction accuracies achieved by these classifiers are 61.6%, 76.1%, and 86.9% respectively. The comparative analysis of the results shows that the feature reduction technique enhances the accuracy of the algorithm by around 2%.

Table 6: Groups of less and frequently mistaken phonemes																			
Frequently mistaken phonemes P_{fm}									ه	ح	ظ	ض	ز	ذ	د	ص	س	ث
									/h/	/h/	/đ/	/ďˤ/	/z/	/ð/	/ḍ/	/ṣ/	/sˤ/	/θ/
Less mistaken phonemes P_{lm}	ي	و	ن	م	ل	ك	ق	ف	غ	ع	ط	ش	ر	خ	ج	ت	ب	ا
	/y/	/w/	/n/	/mˤ/	/l/	/k/	/q/	/f/	/ɣ/	/ʕ /	/ţ/	/š/	/ɾ/	/x/	/g/	/t/	/b/	/ʔ/

Table 7: The percentage accuracies for P_{fm} and P_{lm}						
Classifiers	Frequently mistaken phonemes $\mathbf{P_{fm}}$	Less mistaken phonemes $\mathbf{P_{lm}}$	Evaluation Metric			
			Avg Accuracy	Precision	Recall	F1 Measure
Naïve Bayes	78.04%	61.69%	69.5%	0.70	0.698	0.69
KNN	81%	76%	78%	0.813	0.79	0.791
SVM	89.8%	86.9%	88%	0.87	0.86	0.864

Fig 4 represents the selection procedure of features; the y-axis represents the weights of attributes and features with a weight value greater than 0.0181 are selected for this research work. Relief-F filter provides a list of weighted attributes and all the attributes whose weights are greater than cut-off or threshold value are retained for mispronunciation detection process, all other features with weights less than the threshold are discarded.

As the comparison of different classifier inferred that SVM outperforms the Naïve Bayes and KNN, so we test the accuracy of our algorithm by applying different kernels (linear, polynomial and Gaussian) on both groups P_{fm} and P_{lm}. Table 9 represents the percentage of accuracies of the SVM classifier using different kernels. Results show that accuracies achieved by SVM using linear kernel are of 74.2% for P_{fm}, and 76.7% for P_{lm}.

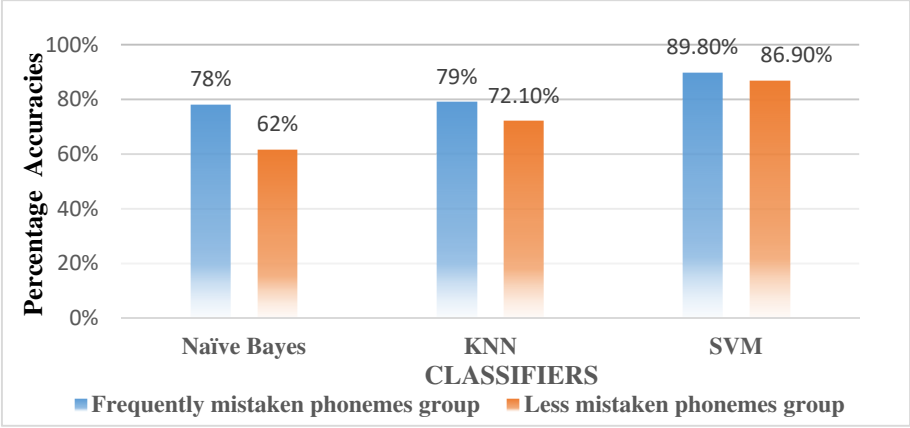

Fig 3: Accuracy of each classifier for P_{fm} and P_{lm}

Table 8: Effectiveness of Feature Selection Method

Feature Selection	Classifiers	Group1	Group2	Avg Accuracy
No	Naïve Bayes	78%	57%	67.5%
	KNN	80%	73	76%
	SVM	88%	85.4%	86.7%
Yes	Naïve Bayes	78.04%	61.69%	69.5%
	KNN	81%	76%	78%
	SVM	89.8%	86.9%	88%

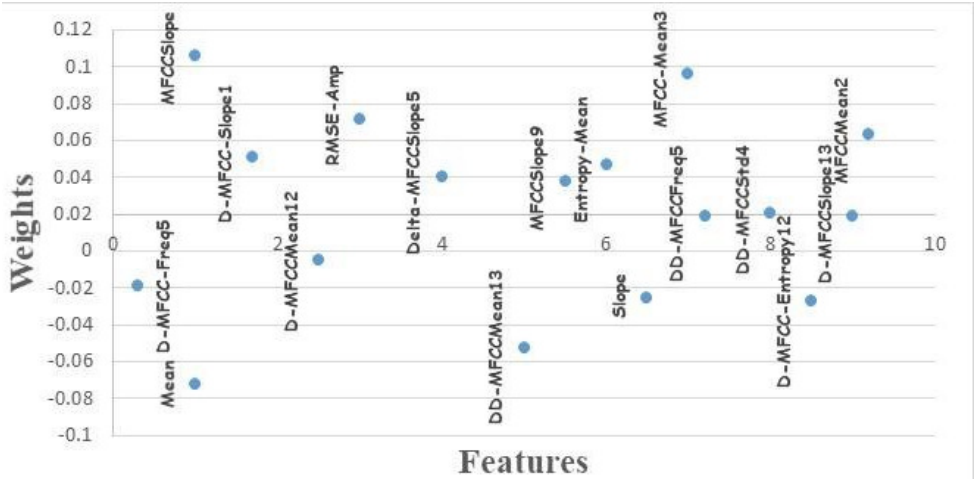

Fig 4: Selected attributes above threshold = 0.018

Accuracies achieved by using polynomial kernel are 89.8% for P_{fm}, and 86.9% for P_{lm} and Gaussian kernel gives an accuracy of 90.2% for P_{fm}, and 84.9% for P_{lm}.

Table 9: Performance evaluation for SVM Classifier

Kernel	SVM		Average Accuracy
	Group1	Group2	
Linear	74.2%	76.7%	75%
Polynomial	89.8%	86.9%	88%
Gaussian	90.2%	84.9%	87%

The average accuracy obtained by SVM using linear, polynomial, and Gaussian kernels is 75%, 88%, and 87% respectively. Comparative analysis shows that SVM with polynomial kernel performs best as compare to linear and Gaussian kernels as shown in Fig 5.

Fig. 6 and Fig. 7 shows the performance of our method, with a polynomial kernel of degree 3, on both groups P_{fm} and P_{lm} in terms of the confusion matrix. It shows that our approach achieved reasonable performance on most of the phoneme classes. The

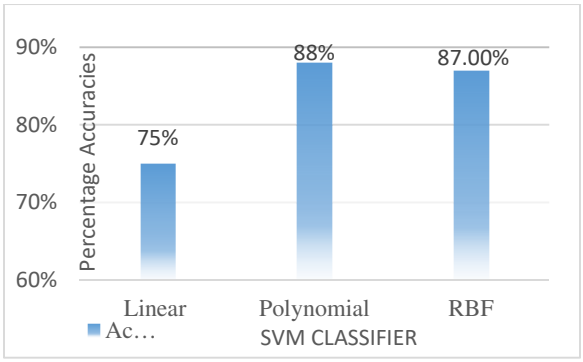

Fig 5: Comparison of Linear, polynomial and RBF kernels

confusion matrix of the P_{fm} group shows that misclassification occurs only on confusing phones. In P_{fm} group, we take three frequently mistaken phonemes pairs ظ ض ز ذ د, ص س ث, ه ح so misclassification occurs only within one confusing pair while the decrease in performance of P_{lm} group is due to the presence of all remaining phones with low mistakes probability but still have confusing pairs, which can mislead classifiers.

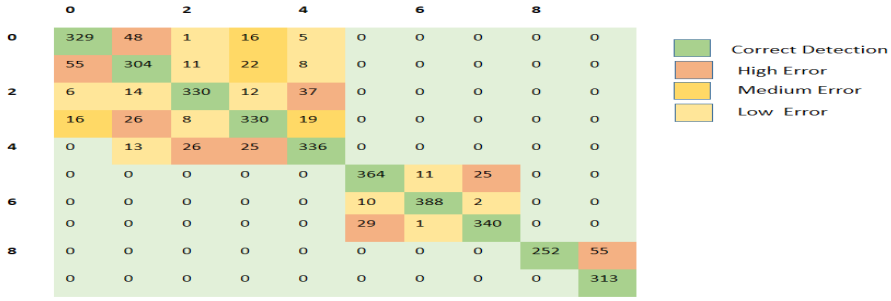

Fig 6: Confusion Matrix of Group1

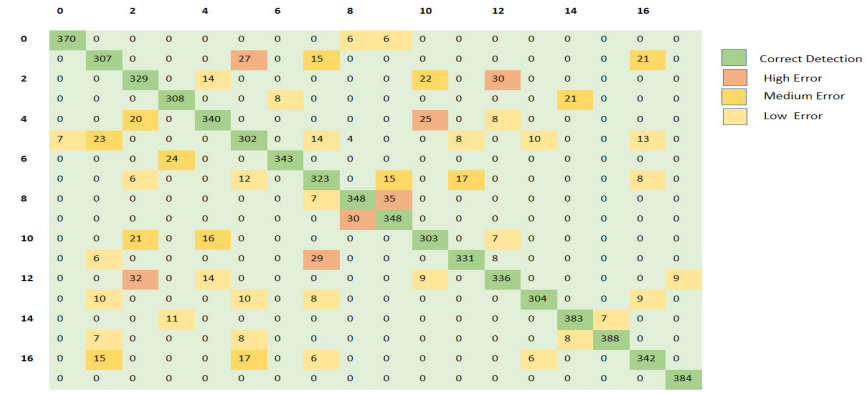

Fig 7: Confusion Matrix of Group2

The results of our method for mispronunciation detection are also presented using the ROC curve on both groups P_{fm} and P_{lm} as shown in Fig 8. The curve plots the true positive rate against false positive rate. In this research work, a multiclass classification problem is addressed, so the ROC curve is drawn for P_{fm} group by taking an aggregate for the ten classes and P_{lm} group by taking an aggregate for the eighteen classes. Each point on the curve represents sensitivity and specificity pair value for a specific decision threshold. A perfect ROC curve for classification passes through the top left corner. The ROC curve for P_{fm} and P_{lm} group is closer to unity which shows that our approach demonstrates a reasonable classification performance.

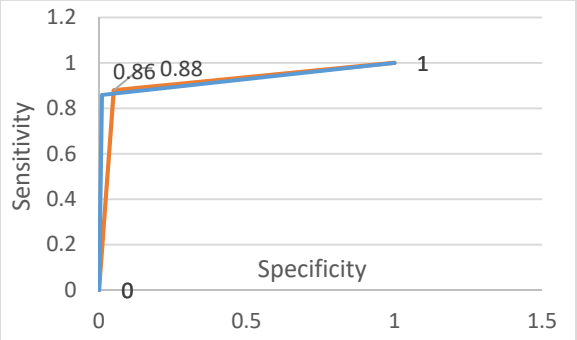

Fig 8: ROC curve

4.5 Discussion

The results of our model show an accuracy rate of 88%, which is higher than the accuracies of other similar models [2, 17, 42] and less than Al Hindi *et al.* [11] work. Our method is more efficient as compared to Georgoulas *et al.* [17], Kun Li *et al.* [42], Abdou *et al.* [2], Kun Li *et al.* [31] in terms of accuracy due to phonemes grouping technique. We also compare our work with Muazzam *et al.* [45] work that uses the same dataset as we used in our proposed method. Our proposed method performed better as compared to their work. We group frequently mistaken confusing pairs in one group; one confusing pair is different from other confusing pairs. In our group of frequently mistaken phonemes, we take three confusing pairs with a high probability of mistakes. The first confusing pair consists of three phonemes so these three phonemes have matching sounds and confused with each other while the second confusing pair consists of five phonemes and confused with each other. The second confusing pair is not confused with the first confusing pair, because they have different sounds, so that is the reason that the proposed classifier achieves better accuracy as compared to the previous approaches. The accuracy rate of Al Hindi *et al*. work is 92.5% that is higher because they focused on only five Arabic phonemes and considered mispronunciation detection as binary classification while our proposed model is based on the multi-label classification of 28 Arabic consonants.

4.4 State of art comparison

We complete experimentation and results in the discussion section by comparing our approach with state of the art as shown in Table 10. Our proposed method outperforms the mentioned state of the art methods in terms of accuracy. It should be noted that the performance of our method is enhanced due to the grouping of phonemes.

Table 10: Comparison with state of the art methods

Techniques	Language		Accuracy
	Arabic	Other languages	
Proposed Method	✓		88%
Muazzam *et al.* [45]	✓		82.7%
Al Hindi *et al.* [11]	✓		92.5%
Georgoulas *et al.* [17]		Greek	77.8%
Kun Li *et al.* [42]		English	80%
Abdou *et al.* [2]	✓		52%
Kun Li *et al.* [31]		Mandarin	83.3%

5. CONCLUSION

In this paper, we proposed a novel approach to deal with pronunciation mistakes of Arabic made by Pakistani nationals. This proposed work demonstrated the development of an efficient mispronunciation detection framework for language learning systems. We considered mispronunciation detection as a

classification problem. When we deal with mispronunciation detection as a classification problem the main drawback is that we have to train a separate classifier for each phoneme's mistake resulting in the increased use of memory and time. To handle this issue we grouped the dataset of 28 phonemes into two groups based on mistakes probability of phonemes. Group1 contained frequently mistaken phonemes and the second group contained less mistaken phonemes. We trained the SVM for both groups instead of training separate classifiers for each phoneme mistake. This grouping technique saves memory and helps in minimizing the number of classifiers to be trained for each phoneme mistake.

Moreover, most states of the art methods focused on one or two confusing pairs while the proposed model deals with all Arabic consonants. This grouping technique is not only efficient in terms of space and time but also enhances the performance of the classifier and achieves an accuracy of 88%. Our approach also outperforms the state of the art methods by around 6% in terms of accuracy. The system is implemented to detect the pronunciation mistakes of a second language learner and provides feedback to make language learning more efficient.

ACKNOWLEDGMENT

Authors would like to thank Dr. Tabassam Nawaz, Chairman, and Associate Professor of the Software Engineering Department, University of Engineering and Technology, Taxila, Pakistan for providing the cooperative environment and infrastructure for this research work to perform experiments.

REFERENCES

1. Abdou R. M., Al-Barhamtoshy H., Jambi K., Al-Jedaibi W., "Enhancing the Confidence Measure for an Arabic Pronunciation Verification System", *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training*, pp. 6-8, Stockholm, Sweden, June 2012.
2. Abdou S. M., Hamid S. E., Rashwan M., Samir A., Abdel-Hamid O., Shahin M., et al., "Computer aided pronunciation learning system using speech recognition techniques", *Proceedings of the Ninth International Conference on Spoken Language Processing*, pp 849-852, Pittsburgh, PA, USA, 17-21 September 2006.
3. Strik H., Truong K., De Wet F., Cucchiarini C., "Comparing different approaches for automatic pronunciation error detection", *Speech Communication*, Vol. 51, pp. 845-852, 2009.
4. Wei S., Hu G., Hu Y., Wang R.-H., "A new method for mispronunciation detection using support vector machine based on pronunciation space models", *Speech Communication*, Vol. 51, pp. 896-905, 2009.
5. Weigelt L. F., Sadoff S. J., Miller J. D., "Plosive/fricative distinction: The voiceless case", *The Journal of the Acoustical Society of America*, Vol. 87, pp. 2729-2737, 1990.
6. Witt S. M., "Automatic error detection in pronunciation training: Where we are and where we need to go", *Proceedings of the International Symposium on Automatic Detection of Errors in Pronunciation Training*, Vol. 1, Stockholm, Sweden, 2012.
7. Van Doremalen J., Cucchiarini C., Strik H., "Automatic detection of vowel pronunciation errors using multiple information sources", *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 580-585, Merano, Italy, 13-17 December 2009.
8. Kim Y., Franco H., Neumeyer L., "Automatic pronunciation scoring of specific phone segments for language instruction", *Proceedings of the Fifth European Conference on Speech Communication and Technology (EUROSPEECH)*, Rhodes, Greece, 2 2-25 September 1997.
9. Witt S. M., Young S. J., "Phone-level pronunciation scoring and assessment for interactive language learning", *Speech Communication*, Vol. 30, pp. 95-108, 2000.
10. Zhang F., Huang C., Soong F. K., Chu M., Wang R., "Automatic mispronunciation detection for Mandarin", *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5077-5080, Las Vegas, NV, 2008.
11. Al Hindi A., Alsulaiman M., Muhammad G., Al-Kahtani S., "Automatic pronunciation error detection of nonnative Arabic Speech", *Proceedings of the 11th International IEEE/ACS*

Conference on Computer Systems and Applications (AICCSA), pp. 190-197, Doha, 2014.

12. Kawai G., Hirose K., "A CALL system using speech recognition to teach the pronunciation of Japanese tokushuhaku", *Proceedings of the ETRW on Speech Technology in Language Learning (STill)* , pp. 73-76, Marholmen, Sweden, 24-27 May 1998.
13. Mak B., Siu M., Ng M., Tam Y.-C., Chan Y.-C., Chan K.-W., Leung K.-W., Ho S., Cong F.-H., Wong J., Lo J., "PLASER: pronunciation learning via automatic speech recognition", *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, Vol. 2, pp. 23-29, 2003.
14. Truong K., Neri A., Cucchiarini C., Strik H., "Automatic pronunciation error detection: an acoustic-phonetic approach", *Proceedings of the InSTIL/ICALL2004 - NLP and Speech Technologies in Advanced Language Learning Systems,* pp. 135-138, Venice, 17-19 June 2004.
15. Ito A., Lim Y.-L., Suzuki M., Makino S., "Pronunciation error detection method based on error rule clustering using a decision tree", *Proceedings of the Ninth European Conference on Speech Communication and Technology*, pp. 173-176, Lisbon, Portugal, 4-8 September 2005.
16. Amdal I., Johnsen M. H., Versvik E., "Automatic evaluation of quantity contrast in non-native Norwegian speech", *Proceedings of the International ISCA Workshop on Speech and Language Technology in Education*, pp. 21-24, Wroxall Abbey State, Warwickshire, England, 3-5 September 2009.
17. Georgoulas G., Georgopoulos V. C., Stylios C. D., "Speech sound classification and detection of articulation disorders with support vector machines and wavelets", *Proceedings of the 28th Annual International IEEE Conference of the 28th Annual International Conference on Engineering in Medicine and Biology Society,* New York, NY, USA, pp. 2199-2202, August 2006.
18. Zhao T., Hoshino A., Suzuki M., Minematsu N., Hirose K., "Automatic Chinese pronunciation error detection using SVM trained with structural features", *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pp. 473-478, Miami, Florida, USA, 2-5 December 2012.
19. Yoon S.-Y., Hasegawa-Johnson M., Sproat R., "Landmark-based automated pronunciation error detection", *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 614-617, Makuhari, Chiba, Japan, 26-30 September 2010.
20. Yang X., Loukina A., Evanini K., "Machine learning approaches to improving pronunciation error detection on an imbalanced corpus", *Proceedings of the IEEE Spoken Language Technology Workshop (SLT)*, pp. 300-305, South Lake Tahoe, Nevada, USA, 7-10 December 2014.
21. Maqsood M., Habib H., Anwar S., Ghazanfar M., Nawaz T., "A Comparative Study of Classifier Based Mispronunciation Detection System for Confusing", *The Nucleus*, Vol, 54, 2017.
22. Maqsood M., Habib H. A., Nawaz T., Haider K. Z., "A Complete Mispronunciation Detection System for Arabic Phonemes using SVM", *International Journal of Computer Science and Network Security*, Vol. 16, No. 3, p.30, Mar 2016.
23. Lee A., Zhang Y., Glass J., "Mispronunciation detection via dynamic time warping on deep belief network-based posteriorgrams," *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 8227-8231, Vancouver, BC, Canada, 26-31 May 2013.
24. Mohamed A.R., Dahl G. E., Hinton G., "Acoustic modeling using deep belief networks," IEEE *Transactions on Audio, Speech, and Language Processing*, Vol. 20, pp. 14-22, 2012.
25. Dahl G., Mohamed A.R., Hinton G. E., "Phone recognition with the mean-covariance restricted Boltzmann machine", *Advances in Neural Information Processing Systems Proceedings*, pp. 469-477, 2010.
26. Joshi S., Deo N., Rao P., "Vowel mispronunciation detection using DNN acoustic models with cross-lingual training", *Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association*, pp. 697-701, Germany, September 2015.
27. Gao Y., Xie Y., Cao W., Zhang J., "A study on robust detection of pronunciation erroneous tendency based on deep neural network",

Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association, pp. 693-696, Germany, September 2015.

28. Hu W., Qian Y., Soong F. K., Wang Y., "Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers", *Speech Communication*, Vol. 67, pp. 154-166, 2015.
29. Hu W., Qian Y., Soong F. K., "An improved DNN-based approach to mispronunciation detection and diagnosis of L2 learners' speech", *Proceedings of the Workshop on Speech and Language Technology in Education (SLaTE),* pp. 71-76, Leipzig, Germany, 4-5 September 2015.
30. Li W., Siniscalchi S. M., Chen N. F., Lee C.-H., "Improving non-native mispronunciation detection and enriching diagnostic feedback with DNN-based speech attribute modeling", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6135-6139, Shanghai, China, 20-25 March 2016.
31. Li K., Qian X., Meng H., "Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks", I*EEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol 25, pp. 193-207, 2017.
32. Bolanos D., Ward W., Wise B., Vuuren S. V., "Pronunciation error detection techniques for children's speech", *Proceedings of the Ninth Annual Conference of the International Speech Communication Association*, pp. 1725-1728, Australia, September 2008.
33. Li H., Liang J., Wang S., Xu B., "An efficient mispronunciation detection method using GLDS-SVM and formant enhanced features", *IEE International Conference on Acoustics, Speech and Signal Processing,* pp. 4845-4848, Taipei, Taiwan, 19-24 April 2009.
34. Franco H., Ferrer L., Bratt H., "Adaptive and discriminative modeling for improved mispronunciation detection", *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 7709-7713, Florence, Italy, 4-9 May 2014.
35. Yang X., Kong X., Hasegawa-Johnson M., Xie Y., "Landmark-based pronunciation error identification on Chinese learning", S*peech Prosody*, pp. 247-251, Jan 2016.
36. Kim S., Georgiou P. G., Lee S., Narayanan S., "Real-time emotion detection system using speech: Multi-modal fusion of different timescale features", *Proceedings of the 9th IEEE Workshop on Multimedia Signal Processing,* pp. 48-51, Crete, Greece, 1-3 Oct 2007.
37. Shaukat A., Chen K., "Towards automatic emotional state categorization from speech signals", *Proceedings of the Ninth Annual Conference of the International Speech Communication Association*, pp. 2771-2774, Australia, Brisbane, Australia, 22-26 September 2008.
38. Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., Witten I. H., "The WEKA data mining software: an update", *ACM SIGKDD Explorations Newsletter*, Vol. 11, pp. 10-18, 2009
39. Durgabai R., "Feature selection using ReliefF algorithm", *International Journal of Advanced Research in Computer and Communication Engineering,* Vol. 3, No. 10, pp. 8215-8218. 2014.
40. Alsulaiman M., Ali Z., Muhammad G., Al Hindi A., Alfakih T., Obeidat H., Al-Kahtani S., "Pronunciation errors of non-Arab learners of the Arabic language", *Proceedings of the International Conference on Computer, Communications, and Control Technology (I4CT)*, pp. 277-282, Langkawi, 2-4 September 2014.
41. Bhatti A. M., Majid M., Anwar S. M., Khan B., "Human emotion recognition and analysis in response to audio music using brain signals", *Computers in Human Behavior*, Vol. 65, pp. 267-275, 2016.
42. Li K., Qian X., Kang S., Meng H., "Lexical stress detection for L2 English speech using deep belief networks", *Proceedings of the 14th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 1811-1815, Lyon, France, 25-29 August 2013.
43. Tariq N., Ijaz I., Malik M. K., Malik Z., Bukhari F., "Identification of Urdu Ghazal Poets using SVM", *Mehran University Research Journal of*

Engineering and Technology, Vol. 38, No. 4, pp. 935-944, 2019.

44. Akram B. A., Akbar A. H.. "Wi-Fi Fingerprinting Based Room Level Indoor Localization Framework Using Ensemble Classifiers", *Mehran University Research Journal of Engineering and Technology*, Vol. 38, No. 1, pp. 151-174, 2019.
45. Maqsood M., Habib H.A., Nawaz T., An efficieny 5. Maqsood M., Habib H. A., Nawaz T., "An efficient mispronunciation detection system using discriminative acoustic phonetic features for Arabic consonants", *The International Arab Journal of Information Technology*, Vol. 16, pp. 242-250, 2019.