International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.70, December 2025

A Mispronunciation Detection Model for Certain Arabic Letters and Selected Chapters of Holy Quran Recitation, Designed for Non-native Arabic Speakers, Developed using the Kaldi Toolkit

Nazik O'mar Balula
College of Computer Science
and Information Technology
Sudan University of
Science and Technology

Mohsen Rashwan
Faculty of Engineering
College of Electronics and Communications
University of Cairo,Giza, Egypt

ABSTRACT

The use of Deep Neural Networks (DNN) shows significantly higher accuracy compared to traditional methods like the Hidden Markov Model (HMM) combined with the Gaussian Mixture Model (GMM) for creating acoustic models. This research involved developing and evaluating a baseline GMM-HMM model alongside a hybrid model that merges Time-Delay Neural Networks (TDNN) with LSTM and GMM-HMM for acoustic modelling, utilising the open-source Kaldi ASR toolkit. The main goal is to detect pronunciation errors of Arabic as spoken by Indians, and of the recitation of the Holy Qur'an, focusing specifically on ten Arabic letters (ع ، ظ ، ط ، ض ، ص ، خ ، ح ، د ، غ ، ق) that non-Arabic speakers often mispronounce, confusing them with other letters that have similar articulation points. The speech dataset consisted of around 65 hours of audio, with 58 hours designated for training and 7 hours for validation and testing. The results indicate that the hybrid model, which combines TDNN-LSTM with GMM-HMM, achieved the highest performance of 96.88%, with a Word Error Rate (WER) of 3.12%. This outperforms the GMM-HMM model, which had a performance of 95.2% and a WER of 4.68%. These results confirm the hybrid model's effectiveness in improving the accuracy of identifying pronunciation errors in Indian speech and recitation compared to the GMM-HMM model alone. This represents a significant step forward in the development of more accurate and efficient speech recognition systems.

Gneral Tterms
Artificial intelligent, Speech Recognition,Deep Neural Network

keywords
Deep Neural network, kaldi toolkit, Time Delay neural network with LSTM, The Holy Qur'an Recitation problems, Mispronunciation letters of Indian speakers

1. INTRODUCTION

The Holy Qur'an serves as the holy text for over a billion Muslims globally. It represents the primary religious scripture that encompasses the comprehensive code of conduct for Muslims. This text is the principal source of guidance and regulations for the Muslim community, and it is composed in the Arabic language. It is essential for Muslims to recite The Holy Qur'an accurately, according to the correct pronunciation. The Holy Qur'an must be protected from any form of alteration, distortion, corruption, modification, or errors in both recitation and writing. To learn, read, and recite The Holy Qur'an as demonstrated by our Prophet Mohammed (PBUH), it is imperative to place the words of Allah SWT in their proper context to prevent mistakes during reading [20] . A particular challenge faced by non-Arab individuals is that their pronunciation of Arabic letters often differs from that of native Arabs due to their accents and the confusion of certain letters that share similar articulation points. This issue can result in various errors when reciting verses from The Holy Qur'an, leading to incorrect recitation and pronunciation errors. Therefore, it is crucial to avoid these mistakes, which can be achieved through the assistance of an expert or an Automatic Speech Recognition (ASR) system.

2. THE HOLY QUR'AN RECITATION PROBLEMS

The recitation of The Holy Qur'an often differs significantly from one reciter to another, even when the verses are taken from the same verse. This variation arises from differences in Tajweed rules and the "Qira'at" (such as Hafs, Kaloun, Warsh, etc.) adopted by reciters. Additionally, challenges come out due to the unique characteristics of the Arabic language in The Holy Qur'an, including differences between written and recited forms, consonant-vowel combinations, co-articulation effects of emphatics and pharyngeals, pronunciation rules, Tanween and Ghonna rules, and word-combination rules. Traditional face-to-face teaching of The Holy Qur'an recitation also has its difficulties. A single teacher often handles many students, limiting individual attention. Students may feel shy or afraid to ask questions, and teachers may lack comprehensive information about their students' backgrounds. Access to Tajweed books can be limited, and teaching methods are often uniform, regardless of students' varying levels of understanding. Some students may prefer visual learning, while others benefit from auditory methods. Furthermore, teachers are not always available at all times or places. Non-Arabs often find it difficult to pronounce some Arabic letters correctly because of their accents and confusion between certain letters that share the same articulation points, such as (ظ، ذ) and (ط، ت). As an example, Figures 1, 2, 3 and 4 show the similarity of the wave form and spectrogram form between the recordings of letters the TAA (ت) and TTA (ط) , respectively because they have the same articulation point.

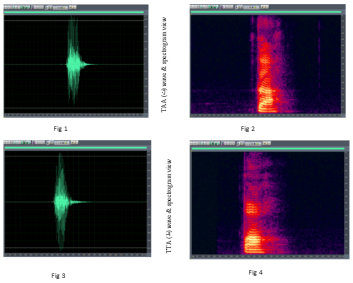

Fig. 1: The wave and spectrogram of the TAA (ت) and TTA (ط).

Therefore, the proposed method is designed to focus on detecting and evaluating improper pronunciation of some Arabic letters for non-native speakers using baseline techniques such as GMM-HMMs and deep learning techniques such as DNNs.

3. RESEARCH OBJECTIVES

The aim of this project is to build an ASR model for Indian speakers to detect and evaluate errors in their recitation (pronunciation errors) to enhance the performance of the HAFSS© application by using deep neural networks (DNNs) based on nnet3 recipes provided by the Kaldi ASR toolkit. The Kaldi ASR toolkit is one of the best toolkits used for speech recognition. The developed model took into its consideration the non-Arabs problems with the non-proper pronunciation of some mispronounced Arabic letters and focused on detecting the mispronunciation of some uttered letters and got a powerful technique to verify The Holy Qur'an recitation, which gave better performance.

4. LITERATURE REVIEW

The most commonly used techniques for verifying the Holy Qur'an recitation are DNNs and HMMs. A key advantage of HMMs is their simplicity and ease of training. This research focuses on detecting mispronounced Arabic letters in the recitation of some Holy Qur'an chapters. Several related studies are highlighted; many of them are relevant as they verify the Holy Qur'an recitation according to incorrect pronunciation and Tajweed rules. Here are some examples. The research conducted by [16] focuses on the recognition of classical Arabic speech, taking into account diacritics by transforming audio signals into discretized text through the use of deep neural network (DNN)-based models. Three distinct models were created for Arabic speech recognition: (i) Time Delay Neural Network with Connectionist Temporal Classification (CTC), (ii) Recurrent Neural Network (RNN) with CTC, and (iii) a transformer model. The study utilized a dataset comprising over 100 hours of Quran recordings, a collection of 72,735 audio recordings of classical Arabic, specifically from the Quran. The findings indicated that the RNN-CTC model achieved state-of-the-art performance, recording the

lowest word error rate of 19.43% and a character error rate of 3.51%. Three character-based speech recognition models were implemented, each employing different DNN architectures in the encoder to identify the most effective model for the dataset. The models based on TDNN, RNN, and transformers were trained on diacritized classical Arabic speech. The RNN-CTC model, which was developed based on the enhanced architecture of the Deep Speech 2 model as proposed in [30], surpassed the recognition capabilities of the original Deep Speech 2, which utilized multiple layers of CNN and RNN. The transformation of raw audio files into various representations, such as Mel-spectrogram tensors, was facilitated by the torchaudio.transforms library in Python. Among the three models evaluated, the RNN-CTC model emerged as the most effective, as mentioned above, achieving the lowest word error rate. The study referenced [2], introduced a system for verifying Quranic recitation. This research put its consideration on recitation of the Quran according to Tajweed rules. Additionally, it attempts to develop an approach that utilizes voice recognition algorithms to automatically detect and determine verses within audio recitations, regardless of the reciters. The study applied the HMM-based Sphinx Framework as its research platform and Sphinx Train to create the acoustic models. Various speech features were extracted, such as elimination of speaker-dependent traits and identification of Arabic phonemes and prosodic characteristics, along with the extraction of spectral features from audio frames using the CMU Sphinx tool. Different classification algorithms, such as the DWT algorithm, were applied to the dataset to obtain classification results utilizing HTK and 16 Gaussian Mixture Models (GMMs). The dataset was categorized based on its patterns, with speakers of Pakistan, including native Urdu, Punjabi, and Pashto speakers, comprising both males and females. The research conducted by [17], utilized the Ar-DAD dataset, which comprises 15,810 Arabic audio clips. Each clip has a sampling rate of 44.1 kHz, a bit depth of 16 bits, and is formatted in stereo WAV, with an average duration of 10 seconds. The dataset features recordings from 30 popular reciters who read 37 chapters from the Holy Quran. Additionally, it includes two plain text files containing the textual content of the same chapters, both with and without vocalization. The dataset was partitioned into 80% for training, 10% for testing, and 10% for validation. The model developed employs a CNN-bidirectional GRU encoder that processes the input feature vector, coupled with a character-based decoder that generates predictions sequentially. For training the encoder, a Connectionist Temporal Classification (CTC) objective function is employed to align the output labels with the input speech. The optimal results achieved include a Word Error Rate (WER) of 8.34% and a Character Error Rate (CER) of 2.42%. An automatic speech recognition (ASR) system for Quranic recitations that caters to female reciters has been developed by [3], using the QRFAM (Quran Recitations by Females and Males) dataset. In this dataset, each female reciter was assigned multiple Surahs from the Quran, with one verse recorded per audio file. The participants were from various Arab nations, including Egypt, Syria, and Algeria. The widely recognized neural speech recognition model, DeepSpeech, developed by Mozilla and available as open-source software, was employed for this project. The QRFAM dataset was prepared by generating CSV files in the format compatible with Mozilla's DeepSpeech, eliminating duplicate audio files, discarding empty and corrupted files, and converting each audio file into a designated format and encoding. The model utilizes a unidirectional recurrent neural network (RNN) architecture with long short-term memory (LSTM) cells. Mel-frequency cepstral coefficients (MFCC) were utilized as feature extractors. A series of four experiments were conducted, with the first experiment yielding the best results. The model achieved a word error rate (WER) of 0.406% and a character error rate (CER) of 0.232%. A Rule-Based Phoneme Duration Algorithm has been developed [6], for the classification of phonemes based on their duration, categorizing them into four types of Medd rules (Medd Wajib, Medd Jayiz, Medd Lazim, Medd Earid LilSukoon). This algorithm utilized the Audacity software for recording and preprocessing the dataset, while Mel-Frequency Cepstral Coefficients (MFCCs) served as a feature extraction method. Additionally, the Hidden Markov Model (HMM) and the HTK toolkit are employed to construct the language model, acoustic model, and classification system. The phoneme dataset comprised 21 Ayats collected from 30 reciters in various environments. The accuracy of the model ranges from 99.87% to 100%, depending on the Medd type. The results achieved by this algorithm are expected to make a significant contribution to models for recognizing Qur'anic recitation. The author [5], proposed a deep learning-based techniques to build a high-performance versatile CAPT system for Mispronunciation Detection and Diagnosis (MDD) and articulatory feedback generation for non-native Arabic learners. The proposed system located the error in pronunciation, recognized the mispronounced phonemes, and detected the corresponding articulatory features (AFs), in both words and even in sentences. This study used the King Saud University speech database (KSU-DB) which is a very rich speech database of Arabic language and designed for speech recognition and speech processing of non-native Arabic speech. To address the problem of the lack of non-native Arabic speech corpora, the non-native Arabic-CAPT corpus (ArabicCAPT) was developed, which comprises 62 speakers from 20 nationalities. For enhancing the generalization of deep models and to reduce overfitting data augmentation techniques have been used to overcome the scarcity of training data. 75% of speakers were used for training and validation and the remaining 25% for testing. Non-native speech corpus has been used to train the MDD-Object model without any initial training. And the alignment of the detected phonemes of, canonical phoneme, and the annotated phoneme is done to calculate the system performance and evaluation. The performance of the various models of the proposed MDD are presented, MDD-E2E system. The best model achieved a 3.83% phoneme error rate (PER) in the phoneme recognition task, a 70.53% F1-score in the MDD task, and a detection error rate (DER) of 2.6% for the AF detection task. The thesis presented in [4], introduced a new phoneme duration model for improving and enhancing the recognition of Medd durations through speech recognition methodologies. A corpus of Qur'an recitations, consists of 21 verses that represent all varieties of Medd, was compiled for the purposes of training and testing Medd duration recognition. A collection of recitations from 100 famous reciters was collected from the internet, while

30 reciters recorded their recitations of each verse four times at varying speeds. The HMM was constructed utilizing this corpus to facilitate the recognition of Qur'anic recitation. Phonemes were classified according to their durations, and a Rule-based Phoneme Duration Algorithm for Medd Classification (RPDMCA) was employed to assign the required duration to each phoneme within a triphone tree. Furthermore, an ANN-based model for Medd duration was proposed to accurately estimate the duration of phonemes. The phoneme classification algorithm demonstrated a high accuracy rate, ranging from 98% to 100%, relying on the type of Medd. The Medd estimation model yielded results that significantly exceeded prior methodologies, as its achieving accuracy of 86% with manual segmentation and 70% with automatic segmentation. development of a dataset that accurately reflects both correct and incorrect pronunciations of short vowel sounds in the Arabic alphabet which includes male and female and contains all the 28 Arabic alphabets with short vowels "Fatha", "Damma", and "Khasra" has been done by [7]. The objective of this dataset is to gather the classical Arabic alphabet along with its short vowel. 31 males and 5 females contributed to the data collection ranging ages from 10 to 60 years. All audio recordings are saved in a wave file format, which is appropriate for the processing of various deep learning models. The recordings are resampled to a frequency of 44.1 kHz. The dataset is partitioned such that 80% is allocated for the training set, while 10% is designated for the testing and the remaing10% for validation sets. The segmentation of the dataset was performed utilizing Python. techniques were employed to increase the size of the dataset. The TensorFlow library in Python was utilized to develop the model, and a CNN sequential model was implemented to evaluate the dataset's efficiency. The result shows the recognition for the alphabet "Alif" is received a successful detection for both the short vowels "Fatha" and "Khasra". However, for the "Damma" short vowel, the model shows a result of 70% the model's performance was checked by evaluating its capability to classify new audio samples. model accuracy was 100% for the short vowels. In [14], a mispronunciation detection system has been established to assist Muslims in learning and reciting the Al-Quran in accordance with Tajweed rules in an easiest way. The dataset encompasses recordings of male, female, and child voices, captured using Android smartphones or the audio recorder feature of WhatsApp in normal environments that may include background noise, and stored in (.wav) format. A total of 29 letters are analysed, with 70% of the data allocated for training purposes and 30% reserved for testing. The audio files are processed using MATLAB. RASTA PLP serves as the feature extraction method to minimize extraneous and additive noise in the speech, thereby enhancing the quality of the signal. For the training and recognition phases, the Hidden Markov Model (HMM) is employed. The 29 letters achieve a perfect accuracy rate of 100% and a 0% Word Error Rate. However, three letters exhibit the lowest accuracy rate and a WER of 91% with a 9% WER, attributed to their similarity to other letters and the complexity of their pronunciation. The overall accuracy percentage stands at 98Reference [35], designed, developed, and evaluated an ASR engine tailored for recitations of The Holy Qur'ān, utilizing a deep learning methodology within the KALDI toolkit. The focus was primarily on the Hafs narration from A'asim, used 32 recitations of Chapter 20 (Sūrat Taha), which were stored in .mp3 format. Each recitation had an approximate duration of 25 minutes, with notable variations in recitation speed among the different reciters. Some audio files were manually segmented using audio editing software, while the rest segmented automatically by using splitting tool, resulting in files saved in .wav format to establish the corpus dataset. This corpus served as the foundation for training and testing the ASR model, employing Mel-frequency cepstral coefficients (MFCC) for feature extraction and a DNN approach for training the acoustic model. Four distinct experiments were carried out to evaluate the ASR system, with the optimal experimental configuration utilizing a TDNN with sub-sampling techniques, achieving a WER ranging from 0.27% to 6.31% and a SER between 0.4% and 17.39%. The study done by S. Hamid [25], who developed an automatic speech recognition system by implementing Computer-Aided Pronunciation Learning (CAPL) System. This system used many algorithms to detect and cover all user mistakes in recitation and gives feedback to the user by the mistakes and the type of that mistakes and also give him/her the correct recitation. A Recitation Rate Normalization (RRN) algorithm was used to overcome the variability in recitation speed which may mislead the phone duration classification module, and HMM-based acoustic model speech recognition engine was implemented to detect the types of recitation mistakes. HMMs used to segment input utterance. Another related work done by Sherif, M. A., Samir, A., Khalil, A.H. and Mohsen, R., CAPL for The Holy Qur'an recitation learning [33], which introduced to enhance the (CAPL) system HAFSS© which was developed for teaching The Holy Qur'an recitation rules and Arabic pronunciations to non-native speakers the verification done by using HMMs, the MLLR techniques used to increment the system performance by adapting the acoustic models.

5. DATASET DESCRIPTION

The speech data plays a considerable role in building, training and testing any speech recognition system, as well as for measuring its performance. Table1 illustrates the dataset which has been used to train, test and evaluate the detection model.

Dataset used	Dataset name	Dataset size	Dataset type	Dataset time of Recordings	Train Dataset &time	Test Dataset & time
From RDI (Research & Development International company)	Hafss-Indian dataset. Containe The Holy Qur'an recitation recited by Indian speakers, 6 chapters (1, 108,109,112,113 and 114) and recording of the 10 mispronounced Arabic letters	94 speakers4025 .wav files With transcription files	Recordings of The Holy Qur'an recitation(.wav files) + Trascription files (.txt))	365 hours of recitation	58 hours79 Indian speakers 860 .wav files 85% from all dataset	7 hours 15 Indian speakers 165 .wav files 15% from all dataset

Table 1. : Dataset description

Developing a robust ASR system requires four essential steps, including data preparation, feature extraction, building the Language Model (LM), and Acoustic Model (AM). The LM is essential for assigning probabilities to word sequences. A widely used and dependable toolkit for creating the LM is the SRILM toolkit. All these steps will be explained and discussed in the following sections.

6. SIRI LANGUAGE MODEL (SRILM)

SRILM, created by the SRI's DecipherTM speech recognition system Technology and Research Laboratory,

serves as an extensive toolkit intended for the development and application of statistical LMs. SRILM is well-known for its robust support of N-gram language modelling, a widely used method for building LMs. It offers various features, including C++ class libraries, designed specifically for creating LMs. The toolkit includes a variety of easy-to-use tools, all of which are detailed and explained in extensive manual pages [36]. N-gram LM is a probabilistic model utilized for forecasting the likelihood of a word by considering the preceding words in a text corpus. This model relies on n-grams, which are consecutive sequences of n words. The SRILM toolkit enables the generation and assessment of different types of language models utilizing N-gram statistics, along with supporting tasks like statistical tagging and handling of N-best lists and word lattice [34]. An n-gram language model interprets the word sequence $\boldsymbol{S}$ as a Markov process characterized by probability $\boldsymbol{P}$, which can be computed as follows:

$$\boldsymbol{P_n(S)} \approx \prod_{i=1}^{k} P\big(w_i \mid w_{i-n+1}, \ldots, w_{i-1+1}\big)$$

In a Markov process, n represents the model's order. For example, when $\boldsymbol{n}$ is 2, it's known as a bigram LM, focusing on word pair co-occurrences. When $\boldsymbol{n}$ is 1, it's called a unigram LM, which uses individual word probabilities. In tasks like speech recognition or machine translation, word arrangement is crucial, often requiring higher-order models like trigrams that examine sequences of three words [24]. N-grams represent the predominant approach employed in language modelling within speech recognition systems. In n-gram representation, all the weight linked to the features of the word $\boldsymbol{w_i}$ following the history $\boldsymbol{h}$ should be allocated to the transition marked with $\boldsymbol{w_i}$ that quit of the state h in the automaton. For instance, if $\boldsymbol{h} = \boldsymbol{w_{i-1}}$, $\boldsymbol{w_{i-2}}$ then the trigram $\boldsymbol{w_{i-2}}\ \boldsymbol{w_{i-2}}\ \boldsymbol{w_{i-1}}\ \boldsymbol{w_i}$ considered a feature, along with the bigram $\boldsymbol{w_{i-1}}\ \boldsymbol{w_i}$ and the unigram $\boldsymbol{w_i}$. Consequently, the weight on the transition$\boldsymbol{w_i}$ departing from state$\boldsymbol{h}$ must be the total of the trigram, bigram, and unigram feature weights [32].

7. THE MODEL'S STRUCTURE

The model structure contains four main parts:

7.1 Data preparation

Data preparation is the first step in the development of any ASR. The data has been prepared to fulfil the requirements of the Kaldi toolkit. As shown in Table 2, the dataset includes 94 recordings of recitation files from Indian speakers. Each speaker has a recitation file of the 6 chapters (Suras) of The Holy Quran and 10 files containing recordings of the 10 Arabic mispronounced letters along with their transcription files. Audio files saved as (.wav) and transcription files saved as (.txt) Additional files were created manually by using Python scripts for each speaker using audio and transcription files. These files were: All the below files are examples of the speaker MS063.

- Utt2Spk: mapping between all utterances of speaker-to-speaker ID

MS036THA MS036
MS036DAA MS036
MS036SUD MS036
MS036DHA MS036
MS036HAA MS036
MS036TTA MS036
MS036AIN MS036
MS036TAA MS036
MS036KHA MS036
MS036GIN MS036
MS036SUR MS036

Table 2. : A partial image of the Utt2Spk file.

- Spk2Utt: mapping between speaker ID to his all utterances.

MS036 MS036-MS036AIN MS036-MS036DAA MS036-MS036DHA MS036-MS036GIN MS036-MS036HAA MS036-MS036KHA MS036-MS036SUD MS036-MS036SURA MS036-MS036TAA MS036-MS036THA MS036-MS036TTA

Table 3. : A partial image of the Spk2utt file.

- Text file: containing transcription of all utterances recorded by each speaker.

MS063-MS063DHA sil ~Z A sil ~Z u sil ~Z i sil ~Z A n sil ~Z I n sil ~Z u n sil f a ~Z ~Z A sil ~@ a ~Z i sil @ a ~Z sil ~Z A l sil ~Z A l a sil y a ~Z sil t a ~Z A sil ~Z A R u sil ~@ a ~Z u sil ~Z u l u sil ~Z A ~@ sil t u ~Z sil ~Z u f u sil ~h a ~Z ~Z A sil ~Z A n1 n a sil ~Z u l i m a sil ~Z A l a m a sil ~@ i ~Z A t u n sil n a ~Z A R a sil ~h i f ~Z A sil ~Z A h a R a sil ~Z u l l a t u n sil g_h A y ~Z A sil ~Z A m a @ u n sil f a ~Z ~Z A n sil ~Z u l a l u n sil ~Z u l m a sil l a ~Z A2 sil y a ~Z u n1 n u sil t a ~Z u n1 n u sil @ a ~@ i ~Z u sil y a ~@ i ~Z u sil ~@ a ~Z m I n sil ~Z A l t a sil @ a ~Z ~Z u l u m a2 t i sil @ a l ~@ i ~Z A2 m i sil ~@ a ~Z i2 m u n sil ~Z u l u m a2 t I n sil @ a ~Z l a m a sil ~Z A2 l i m u2 n a sil f a ~@ i ~Z u2 h u n1 n a sil t a n3 ~Z u R u2 n a sil @ a n1 n a2 ~Z i r i2 n a sil ~Z A2 h i R a sil @ a ~@ ~Z A m u sil m u ~Z l i m a n sil ~Z A ~@ n i k u m sil m a ~h ~Z u2 R A n sil y a ~Z h a R u2 n a sil y a ~Z u n1 n u2 n a sil ~Z A l l a l n a2 sil t a ~Z A2 h a R u2 n a sil ~Z u h u2 r i h i m sil @ u n3 ~Z u R n a2 sil w a t a ~Z u n1 n u2 n a b i l l a2 h i ~Z ~Z u n u2 n a2 sil @ a l l a ~Z i2 @ a n3 q A ~Z A ~Z A h R a k sil f a @ a n3 ~Z a R t u k u m n a2 R A n3 t a l a ~Z ~Z A2 sil @ a f a l a2 y a n3 ~Z u R u2 n a @ i l @ i b i l i sil w a @ a n3 t u m l a2 t u ~Z l a m u2 n sil f i2 l a w ~h I m1 m a ~h f u2 ~Z I n sil

Table 4. : A partial image of the Text file.

- Corpus.txt file: As the same as Text file without Speaker name.

sil ~Z A sil ~Z u sil ~Z i sil ~Z A n sil ~Z I n sil ~Z u n sil f a ~Z ~Z A sil ~@ a ~Z i sil @ a ~Z sil ~Z A l sil ~Z A l a sil y a ~Z sil t a ~Z A sil ~Z A R u sil ~@ a ~Z u sil ~Z u l u sil ~Z A ~@ sil t u ~Z sil ~Z u f u sil ~h a ~Z ~Z A sil ~Z A n1 n a sil ~Z u l i m a sil ~Z A l a m a sil ~@ i ~Z A t u n sil n a ~Z A R a sil ~h i f ~Z A sil ~Z A h a R a sil ~Z u l l a t u n il @ a ~@ u2 ~Z u b i l l a2 h i m i n a s_h s_h a y T A2 n i R R a j i4 m sil b i s m i l l a2 h i R R a ~h m a2 n i R R a ~h i3 sil @ a l ~h a m d u l i l l a2 h i R a b b i l ~@ a2 l a m i4 n sil @ a R R a ~h m a2 n i R R a ~h i4 m sil m a2 l i k i y a w m i d d i3 n sil @ i y y a2 k a n a ~@ b u d u w a @ i y y a2 k a n a s t a ~@ i2 n sil @ i h d i n a2 S S i R A2 T A l m u s t a q i3 m sil S i R A2 T A l l a ~Z i2 n a @ a n a ~@ a m t a ~@ a l a y h i m g_h A y r i l m a g_h ~Z u2 b i ~@ a l a y h i m w a a l ~Z A A A4 l l i2 n sil q u l @ a ~@ u2 ~Z u b i R a b b i n1 n a2 s sil m a l i k i n1 n a2 s sil @ i l a2 h i n1 n a2 s sil m i n3 s_h a r r i l w a s w a2 s i l x A n1 n a2 s sil @ a l l a ~Z i2 y u w a s w i s u f i2 S u d u2 r i n1 n a2 s sil m i n a l j i n1 n a t i w a n1 n a2 s sil q u l @ a ~@ u2 ~Z u b i R a b b i l f a l a q k_l sil m i n3 s_h a r r i m a2 x A l a q k_l sil w a m i n3 s_h a r r i g_h A2 s i q I n @ i ~Z a2 w a q A b k_l sil w a m i n3 s_h a r r i n1 n a f f a2 t_h a2 t i f i l ~@ u q A d k_l sil w a m i n3 s_h a r r i ~h a2 s i d I n @ i ~Z a2 ~h a s a d k_l sil q u l h u w a l l a a2 h u @ a ~h a d k_l sil @ a l l a a2 h u l S A A m a d k_l sil l a m y a l i d k_l w a l a m y u2 l a d k_l sil w a l a m y a k u l l a h u k u f u w a n @ a ~h a d k_l sil w a l ~@ a S R sil @ i n1 n a l @ i n3 s a2 n a l a f i2 x u s R sil @ i l l a l l a ~Z i2 n a @ a2 m a n u2 w a ~@ a m i l u2 S S A2 l i ~h a2 t i w a t a w a2 S A w b i l ~h a q q i w a t a w a2 S A w b i S S A b R sil @ i n1 n a2 @ a ~@ T A y n a2 k a l k a w t_h a R sil f a S A l l i l i R a b b i k a w a n ~h a R sil @ i n1 n a s_h a2 n i @ a k a h u w a l @ a b k_l t a R sil

Table 5. : A partial image of the Corpus file.

- Wav.scp file: Consist of absolute path of audio file.

MS036-MS036AIN /home/hafss/Training&Testing-Data/Training/MS03/MS036AIN.wav MS036-MS036DAA /home/hafss/Training&Testing-Data/Training/MS03MS036DAA.wav MS036-MS036DHA /home/hafss/Training&Testing-Data/Training/MS03/MS036DHA.wav MS036-MS036GIN /home/hafss/Training&Testing-Data/Training/MS036/MS036GIN.wav MS036-MS036HAA /home/hafss/Training&Testing-Data/Training/MS036/MS036HAA.wav MS036-MS036KHA /home/hafss/Training&Testing-Data/Training/MS036/MS036KHA.wav MS036-MS036SUD /home/hafss/Training&Testing-Data/Training/MS036/MS036SUD.wav MS036-MS036SUR /home/hafss/Training&Testing-Data/Training/MS036/MS036SUR.wav MS036-MS036TAA /home/hafss/Training&Testing-Data/Training/MS036/MS036TAA.wav MS036-MS036THA /home/hafss/Training&Testing-Data/Training/MS036/MS036THA.wav MS036-MS036TTA /home/hafss/Training&Testing-Data/Training/MS036/MS036TTA.wav

Table 6. : A partial image of the Wave file.

7.2 Feature extraction

After the process of data preparation, the feature extraction process takes its place. Extracting the optimal parametric representation of acoustic signals is essential for improving recognition performance. The best feature extraction method is MFCC. This method is widely used and has proven effective in the field of signal processing, as mentioned by [15] [3] [6]. The MFCC technique is notable as a prominent method for extracting speech features. MFCC is preferred due to its being based on the variations in the critical frequency bandwidth of the human ear. By applying the first-order derivatives DMFCC and the second-order derivatives DDMFCC from MFCC, more intricate speech features can be obtained [19]. The speech signals of the model were sampled at 16 kHz, and the feature was extracted by applying a 25 ms Hamming window with a 10 ms overlap (25 ms frames shifted by 10 ms each time) in addition to delta and delta-delta coefficients. The MFCCs, which are derived from FFT-based log spectra, were used. The length of the features vector was 40, the length of the parameterized static vector (MFCC0 = 13) plus the delta coefficients (+13) plus the acceleration coefficients (+13) + 1 energy coefficient.

7.3 Building and training of the language model

Language model, commonly known as statistical language modelling (LM), involves estimating a probability distribution that captures the statistical patterns observed in the use of natural language. The LM is essential for determining the probabilities associated with sequences of words. A well-known and reliable toolkit employed

for constructing the LM is the SRILM toolkit. To create and build LM the command: ***ngram-count -order 3 -write-vocab vocab.txt corpus.txt -lm output.arpa -unk*** was used. Where ***ngram-count -order***, ***vocab.text***, ***n-gram count***, ***write-vocab*** and ***unk*** indicate n-gram order, vocabulary used in training file name, input training corpus file name, Output ARPA-format LM file name and OOV marked as <unk>. OOV (out of vocabulary or unknown words (unk)) refers to words that have not been seen previously. This study utilized 2-gram, 3-gram, and 9-gram LMs to assess the likelihood of word sequences, which were subsequently applied to develop the acoustic model (AM). Table 7 is a partial image of the output.arpa file generated by the 3-gram-count command.

```
data\
ngram  1=67
ngram  2=1195
ngram  3=4929
\1-grams:
-5.06918  2  -0.8154395
-5.370211  3  0.336068
-5.67124  4  -0.3907465
-2.739783  </s>
-99  </s>  -2.878459
-1.483128  @  -2.766584
\2-grams:
-0.1538486  2  2  0.2925097
-0.8228216  2  @
-0.5217916  3  sil
-0.5217916  3  y
-0.2207617  4  y
-0.000508244 </s>  sil  -1.757643
\3-grams:
-0.3819621  2  2  2
-0.2870643  A  @  a
-0.5614421  A  @  n
-1.0086  A  @  sil
-1.230449  A  @  u
-1.40654  A  @  u2
```

Table 7. : A partial image of the output.arpa file.

7.4 Building and training of the acoustic model (AM)

AM is a probability distribution used to model and represent the variations in phonological and acoustic-phonetic features observed in speech input to the recognition system [10]. The feature extraction block generates feature vectors that are then used for acoustic modelling of speech utterances. AM is developed using data from a speech database along with linguistic constructs. Acoustic modelling is a vital step in speech recognition, as it connects acoustic data to linguistic units. Most computations in acoustic modelling focus on feature extraction and statistical representation, making it a crucial part of the recognition process. Statistical representations are derived from the extracted features, and the distribution of these features corresponding to specific sounds is modelled in acoustic modelling to link the features with linguistic unit structures [9]. The success of ASR systems depends on smaller components or sub-word units of a word, typically defined by phoneticians or expert linguists. These sub-word units, known as phones or phonemes, can include phones, phonemes, allophones, or other arbitrary sub-words. A key element of standard ASR is the dictionary, which specifies the permissible sequences of phonemes. One critical challenge in ASR systems is accurately defining phoneme boundaries and creating suitable HMMs to represent their duration in speech [29]. Standard ASR systems use HMMs to model acoustic sequences, where each HMM state represents a frame, usually 10 ms, of a spectral representation of the sound wave. These models employ a mixture diagonal covariance GMM. The GMM-HMM AM is considered one of the most effective methods for ASR acoustic modelling [26]. Furthermore, the AM must maximize mutual information between input features and corresponding HMM states. Research in [18] [27] has shown that deep neural networks effectively meet these requirements. A key factor in the success of DNNs is their ability to learn invariant representations that improve class discrimination. Unlike GMMs, DNNs produce discriminative data representations by applying non-linear transformations across multiple hidden layers. The building and training of AM of this study contained two stages: first training with the GMM-HMM and second with the DNN-HMM.

7.5 Building and training of GMM-HMM AM

In this phase, the system was trained using monophones and triphones with 2-gram, 3-gram, and 9-gram LMs. A monophone model is AM without contextual information about the preceding or following phones, serving as a base for triphone models that incorporate such context. Monophone models only capture the acoustic properties of a single phoneme, though phonemes vary greatly depending on their context. In contrast, triphone models represent a phoneme variant within the structure of two adjacent phonemes (left and right) [11]. Initially, AM was trained on monophones, and the monophone AM was used to align the feature vectors of the training data. These monophone alignments were then used to train triphones. The triphone model was re-aligned, followed by training a new delta and delta-delta triphone AM. Next, LDA-MLLT was applied to the new triphone AM to reduce feature dimensionality and de-correlate the reduced features. Finally, Speaker Adapted Training (SAT) was applied on top of the LDA-MLLT features, and fMLLR was used for speaker normalization.

7.6 Building and training of DNN AM

The DNN-HMM model was trained on fMLLR features, which is the final step in GMM-HMM training, using the decision tree and alignments from the SAT-fMLLR GMM model. It utilized TDNN and LSTM neural networks (TDNN-LSTM) and was trained with nnet3 recipes from the Kaldi ASR toolkit.

7.6.1 TDNN.

TDNN is a specific type of DNN designed for handling sequential data. As a feedforward network, it includes delays in the connections between layers, corresponding to the input signals. These delays enable the network to represent data across various time points, allowing TDNN

to respond dynamically to time-series input. Starting with a limited context, the TDNN progressively expands its learning context through additional hidden layers, making it better at capturing contextual relationships. LSTM layers, on the other hand, are a specialized RNN architecture with memory cells and gating mechanisms, enabling efficient processing of sequential data [23].

In TDNNs, temporal context is captured through a hierarchical architecture, with each layer operating at a different temporal resolution. The outputs of the previous hidden layer are concatenated and used as input for the current layer, allowing the current layer to process a wider context. As the network progresses to higher layers, it perceives an even wider context. Transformations within the same layer are tied across time, reducing parameters and ensuring that the transformation is not influenced by shifts in input timing. Subsampling is introduced in TDNNs through splicing configurations, such as -1,1, which splices inputs from one step before and after the current time step, omitting the current frame, and -3,3, which splices inputs three steps before and after, omitting two current frames. This subsampling reduces input dimensions and model size [22].

7.6.2 Related work to TDNN.

The research conducted by [35] focuses on the development of a speaker-independent, large-vocabulary continuous speech recognizer specifically designed for The Holy Qur'an. Their methodology is centered around the recitation of Hafs from A'asim and incorporates advanced ASR techniques from the Kaldi toolkit, along with the integration of various Tajweed rules. The study outlines the meticulous process of creating the Holy Qur'an speech corpus, which served as the foundation for training and evaluating the speech recognizer. The acoustic model training involved four distinct experimental configurations within the KALDI toolkit, each differing in dataset size and the incorporation of Tajweed rules. Notably, one of the experiments utilized TDNN with sub-sampling, resulting in the most favourable outcomes. This particular setup demonstrated a WER ranging from 0.27% to 6.31% and a Sentence Error Rate (SER) ranging from 0.4% to 17.39%. These encouraging findings suggest that the speech recognizer exhibits a high level of accuracy in recognizing The Holy Qur'an based on the recitation of Hafs from A'asim, utilizing the Kaldi toolkit and TDNN architecture. The study [13] presented a TDNN for enhancing speech through comprehensive data learning. A method for full data learning in speech enhancement was suggested to maximize the utilization of training data, including clean-to-clean, noisy-to-clean, and noisy-to-silence data. The experiments were carried out using the TIMIT dataset, demonstrating that the proposed approach outperformed DNN and even showed comparable or superior performance to BLSTM. The data generated had a sampling rate of 8 kHz, with 129-dimensional spectral magnitudes of noisy speech utilized as input features. These features were computed using a Short-Time Fourier Transform (STFT) with a 32 ms length Hamming window and a 16 ms window shift. The ReLU function was used as the activation function for training the deep TDNN. All TDNN-based systems consisted of four hidden layers with 256 nodes in each layer. A total of 30 epochs were required to train the model using the noisy-to-clean data. [31], introduced a TDNN design that can capture long-term temporal dependencies while maintaining training times. The training dataset spans from 3 to 1800 hours, and the network employs sub-sampling techniques to decrease computational load during training. Results from the Switchboard task exhibit a 7.3% enhancement over the standard DNN model. Findings from various LVCSR tasks illustrate the TDNN architecture's capability to learn broader temporal dependencies in both limited and extensive data settings, with an average improvement of 5.5%. A language identification system was developed and trained to differentiate between Arabic, Spanish, French, and Turkish solely based on recorded speech. The MediaSpeech dataset was used as the training data, which consists of approximately 10 hours of speech along with corresponding transcriptions for the aforementioned languages. A series of acoustic models were trained using a preexisting multilingual dataset and the Tedlium TDNN model. The system was equipped with a customized multilingual language model and a pronunciation lexicon that included language names preceding phones were prepared and used. The trained model generated phone alignments for testing data from all four languages, and predictions were made based on a voting scheme that selected the most frequently occurring language prepend in an utterance. The accuracy was evaluated by comparing the predicted languages with the known languages. Very high accuracy in identifying Spanish and Arabic was observed, and lower accuracy was observed in identifying Turkish and French. The Kaldi toolkit was used to build and develop the system. The MFCC method was employed for feature extraction. The model demonstrated an accuracy of over 99% in correctly predicting Arabic utterances as Arabic and Spanish utterances as Spanish, which is considered to be exceptionally high [21]. The LSTM was introduced by Hochreiter and Schmidhuber in 1997. LSTM is a type of artificial recurrent neural network (RNN) architecture specifically designed for learning sequential data, such as numerical or experimental time series. Its main purpose is to address the issue of forgetting in traditional RNNs [12]. The LSTM neural network unit consists of four main components (gates), as Figure 5 shows: an input gate $(\boldsymbol{i_t})$ that is responsible for determining the necessity of the data in the long term; a cell state $(\boldsymbol{c_t})$ that plays a crucial role in transferring important information throughout the sequence processing; a forget gate $(\boldsymbol{f_t})$ that has the ability to reset the internal state of the memory cell once the stored information becomes unnecessary or is no longer needed; and an output gate $(\boldsymbol{o_t})$ that is responsible for generating the hidden state array for the next cell by taking theprevious hidden state and the current input and passing them through a sigmoid function.

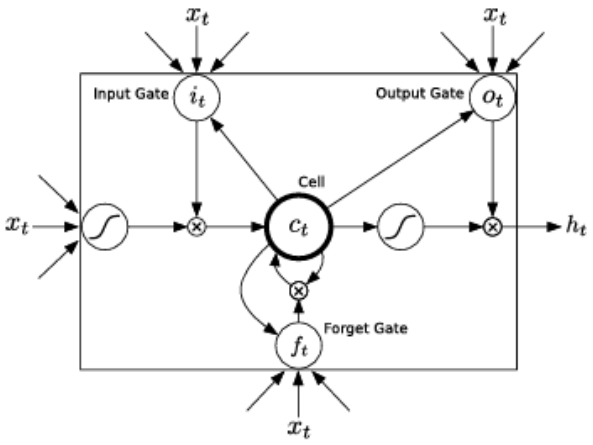

Fig. 2: The LSTM structure and gates.

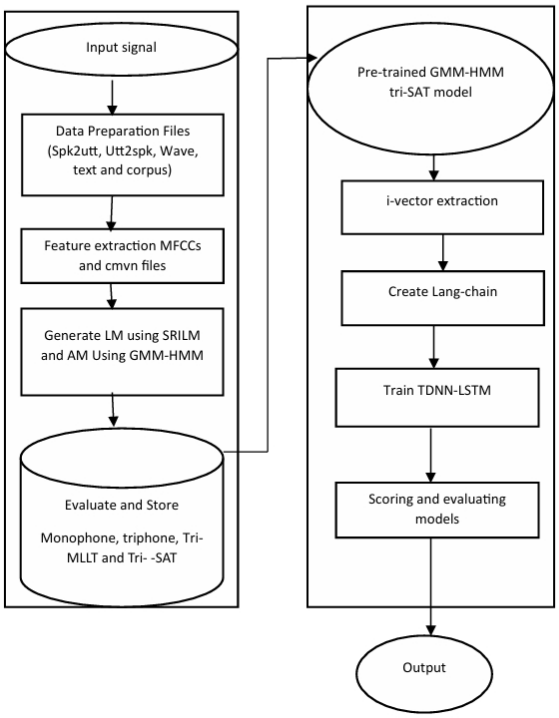

Fig. 3: The structure of the developed model

LSTM effectively overcomes the vanishing gradient problem by ensuring gradients stay sufficiently steep [1]. This leads to faster training and improved accuracy, making the training process both efficient and highly accurate. The DNN-HMM model was built and trained using fMLLR features, with the decision tree and alignments derived from the SAT-fMLLR GMM model. It was trained on a 58-hour dataset using TDNN and LSTM neural networks (TDNN-LSTM) through the nnet3 recipes provided by the Kaldi ASR toolkit. The TDNN-LSTM used 40-dimensional MFCCs combined with 100-dimensional i-vectors as input. Data augmentation was applied through speech perturbations to generate additional training data and improve the performance of DNNs. The network consisted of 8 hidden layers, including 6 TDNN layers and 2 LSTM layers, with each TDNN layer and LSTM cell dimension set to 1024. The ReLU activation function was employed for its advantages, such as good sparsity, fast convergence, simple calculations with lower computational costs, and effectively addressing the gradient vanishing issue associated with sigmoid and tanh. However, a drawback of ReLU is that its gradient remains zero for negative inputs, which can lead to inactive or dead neurons during training. Despite this, ReLU enhances neural network performance, making models easier and faster to train while often yielding better results [8]. The diagram below shows the structure of the model created.

8. EXPERIMENTAL SETUP, RESULTS AND DISCUSSIONS

Experimental results were reported using WER, which calculates the minimum edit distance between the ASR system's output and the correct reference transcriptions. The experiments were performed on Ubuntu 16.04 LTS (64-bit) using a system with a dual-core Intel(R) Core(TM) i5-4200U CPU @ 1.60GHz, 4GB of RAM, and a single GPU with a compute capability of 3.5 and 2GB of memory. The setup employed the CUDA 8.0 toolkit and was compiled with NVIDIA's CUDA GPU. The dataset consists of 1025 audio files from 94 Indian speakers, divided into 860 files for training and 165 files for testing. The model results and their comparisons will be discussed. Performance is measured by WER and accuracy using Kaldi's standard calculations.

$$\boldsymbol{WER} = 100 \times \frac{(I + D + S)}{Total\_number\_of\_words}$$

$$\boldsymbol{Accuracy} = 100 \times \frac{Total\_number\_of\_words - (I + D + S)}{Total\_number\_of\_words}$$

Where ***I, D, S*** indicate the numbers of Insertion, Deletion and Substitution of each utterance.

In the TDNN-LSTM configuration, the techniques of recurrent projection and non-recurrent projection were applied. The first one used to reduce hidden layers connections dimensionality to speed up the training process

by reducing computational complexity, whereas the second one applies to feed-forward paths using full-size hidden layers connections to control model size, and this leads to higher computational cost and computational complexity. Both techniques were set to 512.
There were two experimental setups, and the model was evaluated in each. The first setup used a GMM-HMM classifier technique, involving training and decoding monophones, triphones, Tri-MLLT, and Tri-SAT with three different LMs (2-gram, 3-gram, and 9-gram). This experiment achieved a WER ranging from 30.04% to 4.68%. The evaluation of GMM-HMM was carried out using different language models, such as 2-gram, 3-gram, and 9-gram models.

N-Gram Order	Monophone	Triphone	Tri- MLLT	Tri-SAT
2-Gram	30.04%	20.85%	19.33%	17.43%
3-Gram	20.47%	12.70%	12.15%	10.47%
9-Gram	4.86%	4.70%	4.69%	4.68%

Table 8. : GMM-HMM AMs N-Gram Comparison

Table 8 presents the results, showing that the 9-gram LM outperformed the 2-gram and 3-gram LMs across all AM models, and the Tri-SAT AM model demonstrated superior performance compared to other AM models. Using GMM-HMM-based training on the Tri-SAT model with the 9-gram LM achieved the best result, with a 4.68% WER. The 9-gram Tri-SAT LM outperformed the 2-gram and 3-gram Tri-SAT LMs by 12.75% and 5.79% WER, respectively.

9. ASSESSING THE EFFECTIVENESS OF TDNN-LSTM

The second experimental setup utilized a TDNN-LSTM classifier. Two distinct architectures of this neural network were implemented. The first architecture consisted of 3, 6 and 9 TDNN layers with 1, 2 and 3 LSTM layers respectively., with 2048 hidden units, and ReLU as an activation function, using 2-gram, 3-gram, and 9-gram LMs as shown in table 9.

Architecture	2-gram LM	3-gram LM	9-gram LM
3 TDNN with 1 LSTM and 2048 hidden layers	9.70%	7.11%	3.30%
6 TDNN with 1 LSTM and 2048 hidden layers	8.59%	6.16%	3.15%
9 TDNN with 1 LSTM and 2048 hidden layers	8.16%	6.30%	3.15%

Table 9. : WER of 2048 hidden Architecture with 2-gram, 3-gram and 9-gram LMs

The second architecture consisted of 3, 6 and 9 TDNN layers with 1, 2 and 3 LSTM layers respectively., with 1024 hidden units, and ReLU as an activation function, using 2-gram, 3-gram, and 9-gram LMs as shown in table 10.

Architecture	2-gram LM	3-gram LM	9-gram LM
3 TDNN with 1 LSTM and 1024 hidden layers	9.61%	6.99%	3.26%
6 TDNN with 1 LSTM and 1024 hidden layers	8.70%	6.26%	3.12%
9 TDNN with 1 LSTM and 1024 hidden layers	9.65%	6.51%	3.29%

Table 10. : WER of 1024 hidden Architecture with 2-gram, 3-gram and 9-gram LMs

The experiments above conclude that:

- The hybrid TDNN-LSTM with GMM-HMM model beat the baseline GMM-HMM model alone.
- The architecture with 1024 hidden units, a 6-layer TDNN, and 2 LSTMs achieved an impressive WER of 3.12%. The 9-gram LM performed better than the 2-gram and 3-gram models, proving to be the optimal n-gram order.
- The 1024 hidden layer setup outperformed the 2048 hidden layer setup, being much faster in training. In contrast, the 2048 hidden layer model was more time-intensive.
- The third experiment delivered the best results, achieving the lowest WER of 3.12% and the highest accuracy of 96.88%, showcasing the speech recognizer's effectiveness in detecting and recognizing errors in The Holy Qur'an recitation by non-native speakers
- The difference between the two models (GMM-HH and TDNN), started with 8.7% in a 2-gram and ended with 1.5% with 9-gram.

In addition to WER and accuracy there are several methods have been used to evaluate ASR quality, such as deletion and substitution results. Deletion happens when a phoneme is omitted from the output. For example, the word (عالماً) might be pronounced as (عالِ), where the phoneme (اً) or (تـنـويـن فـتـحـة) is missed from the uttered phoneme. Substitution occurs when the pronunciation of a word changes due to replacing one phoneme with another. For instance, the word (أعوذُ) might be pronounced as (أعوظُ), substituting the letter (ذ) with the letter (ظ) in the uttered phoneme. These errors arise when the model chooses the wrong path, and one possible solution is improving language models [28].
Table 11 presents the deletion and substitution results recorded for each phoneme in the GMM-HMM model. The table evidences that the phoneme DAA (د) has the highest substitution rate at 10.3% (indicating the poorest phoneme accuracy) due to its confusion with the letters TAA (ت) and TTA (ط). On the other hand, the phoneme KHA (خ) achieved the lowest substitution rate at 3.9% (indicating the best phoneme accuracy).

letter	No_of-correct	No_of_Deletion	No_of_Insertion	No_of_substitution	Total_of_utterance	phoneme_substitution%	phoneme_Insertion%	phoneme_Deletion%	Accuracy%	CER%
AAIN ع	7061	314	176	395	7846	5.0	2.2	4.0	88.9	11.1
DDAA د	6419	236	166	783	7604	10.3	2.2	3.1	84.4	15.6
TTHA ط	6770	280	141	490	7681	6.4	1.8	3.6	88.1	11.9
GGIN غ	6659	153	138	327	7277	4.5	1.9	2.1	91.5	8.5
AAIN ع	7051	204	189	379	7823	4.8	2.4	2.6	90.1	9.9
DDAA د	7117	130	178	304	7729	3.9	2.3	1.7	92.1	7.9
TTHA ط	6524	264	144	371	7303	5.1	2.0	3.6	89.3	10.7
GGIN غ	6417	216	137	304	7074	4.3	1.9	3.1	90.7	9.3
TTHA ط	6490	280	123	401	7294	5.5	1.7	3.8	89.0	11.0
GGIN غ	6777	179	161	326	7443	4.4	2.2	2.4	91.1	8.9

Table 11. : GMM-HMM substitution and Deletion Results per phoneme

Also, deletion and substitution results per phoneme for the TDNN-LSTM model have been presented in figure 4. The figure, illustrates that, the phoneme DAA (د) has the highest rates of deletion and substitution at 1.2% and 6.6%,

respectively, representing the worst phoneme accuracy. This is assigned to the confusion of this letter with TAA (ت) and TTA (ط), as mentioned above. On the other hand, the phoneme TAA (ت) shows the lowest rates of deletion and substitution results at 0.5% and 0.9%, respectively, indicating the best phoneme accuracy.

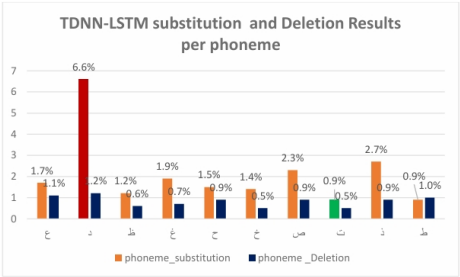

Fig. 4: TDNN-LSTM substitution and Deletion Results per phoneme.

A lack of sufficient training data often leads to a decrease in system accuracy, causing errors like insertions, deletions, and substitutions. Increasing the amount of training data can address these problems and greatly improve the system's performance.
The model detected mispronounced letters in the recitation of SURAs (chapters 1, 108, 109, 112, 113, and 114) using 2-gram, 3-gram, and 9-gram LMs with GMM-HMM as a classifier. Figure 8 and table 12 indicate that the 9-gram LM paired with Tri-SAT AM achieves the best WER and accuracy, at 8.5% and 91.5%, respectively. On the other hand, the 9-gram LM combined with monophone AM yields the worst WER and accuracy, at 22.9% and 77.03%. These findings demonstrate that 9-gram LM combined with Tri-SAT AM outperforms all other AMs.

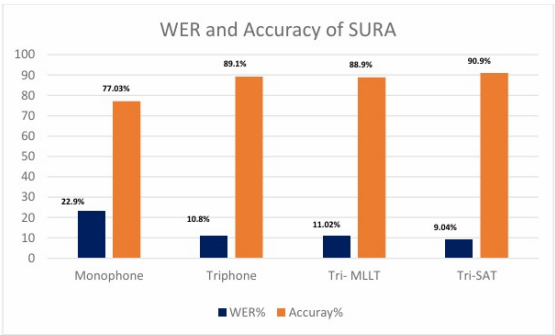

Fig. 5: WER & Accuracy of SURAs using GMM-HMM.

Acoustic-Model	No.correct	No.Substitution	No.Insertion	No.Deletion	Total_ No. S+I+D	Total_ No. Atterence	WER%	Accuracy%
Monophone	8690	1316	190	906	2412	10502	22.97	77.03
Triphone	9363	697	240	197	1134	10497	10.80	89.20
Tri- MLLT	9431	631	288	250	1169	10600	11.03	88.97
Tri-SAT	9674	403	235	255	893	10567	8.45	91.55

Table 12. : GMM-HMM AMs N-Gram Comparison

10. CONCLUSION

This study highlights the use of GMM-HMM and TDNN-LSTM models to detect mispronounced letters of the Arabic language and The Holly Quran recitation by Indian speakers. The model has used HMM and TDNN-LSTM as classifiers. Both GMM-HMM-based and TDNN-LSTM-based AMs were tested. The GMM-HHM-based AM was trained and tested using monophones, triphones, Tri-MLLT, and Tri-SAT with three different LMs (2-gram, 3-gram, and 9-gram). Also, TDNN-LSTM-based AM was trained and tested using two different architectures, 2048 and 1024 hidden layers, each one of the two architectures tested using pretrained GMM-HMM-based Tri-SAT AM with 2-gram, 3-gram, and 9-gram LMs, respectively. The two architectures, 2048 and 1024 hidden layers, used 3 TDNN layers with 1 LSTM layer, 6 TDNN layers with 2 LSTM layers, and 9 TDNN layers with 3 LSTM layers.
This study, conclude that:
GMM-HMM and TDNN-LSTM techniques are able to detect mispronunciation in the Arabic language and The Holly Quran recitation for Indian language speakers with an accuracy about 95.3% and 96.8%, respectively.
The TDNN-LSTM technique outperforms the GMM-HMM technique. And the Tri-SAT AM with 9-gram LM outperforms the monophones, triphones, and Tri-MLLT AMs with 2-gram, 3-gram, and 9-gram LMs.
The TDNN-LSTM classifier with 6 TDNN layers, 1024 hidden layers, and 2 LSTMs using 9-gram LM outperforms the same architecture with 2-gram and 3-gram LMs and at the same time outperforms the 2048 architecture with all other 3 LMs. GM-HMM AM is faster than TDNN-LSTM AM in training, and TDNN-LSTM AM is memory and time consuming.
At last, it can be concluded that even with the difficulty of the Arabic language and the nonavailability of a sufficient amount of non-native Arabic speakers' Quranic dataset, the developed model have achieved a satisfactory result.

11. REFERENCES

[1] Hossein Abbasimehr and Reza Paki. Improving time series forecasting using lstm and attention models. *Journal of Ambient Intelligence and Humanized Computing*, 13(1):673–691, 2022.

[2] Muhammad Rehan Afzal, Aqib Ali, Wali Khan Mashwani, Sania Anam, Muhammad Zubair, and Laraib Qammar. Recitation of the holy quran verses recognition system based on speech recognition techniques. *UMT Artificial Intelligence Review*, 3(2):01–20, 2023.

[3] Suhad Al-Issa, Mahmoud Al-Ayyoub, Osama Al-Khaleel, and Nouh Elmitwally. Towards building a speech recognition system for quranic recitations: A pilot study involving female reciters. *Jordan Journal of Electrical Engineering*, 8(4):307–321, 2022.

[4] AMA Al-Qadasi. Phoneme duration scheme for tajweed medd rules recognition in qur'an recitation,". *PhD, Computer science, Universiti Teknologi Malaysia, Malasia*, 2021.

[5] Mohammed Algabri, Hassan Mathkour, Mansour Alsulaiman, and Mohamed A Bencherif. Mispronunciation detection and diagnosis with articulatory-level feedback generation for non-native arabic speech. *Mathematics*, 10(15):2727, 2022.

[6] Ammar Mohammed Ali Alqadasi, Mohd Shahrizal Sunar, Sherzod Turaev, Rawad Abdulghafor, Md Sah Hj Salam, Abdulaziz Ali Saleh Alashbi, Ali Ahmed Salem, and Mohammed AH Ali. Rule-based embedded hmms phoneme classification to improve qur'anic recitation recognition. *Electronics*, 12(1):176, 2022.

[7] Fatimah Alqadheeb, Amna Asif, and Hafiz Farooq Ahmad. Correct pronunciation detection for classical arabic phonemes using deep learning. In *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, pages 1–6. IEEE, 2021.

[8] Yuhan Bai. Relu-function and derived function review. In *SHS web of conferences*, volume 144, page 02006. EDP Sciences, 2022.

[9] Shobha Bhatt, Anurag Jain, and Amita Dev. Acoustic modeling in speech recognition: a systematic review. *International Journal of Advanced Computer Science and Applications*, 11(4), 2020.

[10] Peter F Brown. *The acoustic-modeling problem in automatic speech recognition*. Carnegie Mellon University, 1987.

[11] Eleanor Chodroff. Kaldi training overview, 2024.

[12] Akshay Madhav Deshmukh. Comparison of hidden markov model and recurrent neural network in automatic speech recognition. *European Journal of Engineering and Technology Research*, 5(8):958–965, 2020.

[13] Cunhang Fan, Bin Liu, Jianhua Tao, Jiangyan Yi, Zhengqi Wen, and Leichao Song. Deep time delay neural network for speech enhancement with full data learning. In *2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pages 1–5. IEEE, 2021.

[14] Javeria Farooq and Muhammad Imran. Mispronunciation detection in articulation points of arabic letters using machine learning. In *2021 international conference on computing, electronic and electrical engineering (ICE cube)*, pages 1–6. IEEE, 2021.

[15] Shikha Gupta, Jafreezal Jaafar, WF Wan Ahmad, and Arpit Bansal. Feature extraction using mfcc. *Signal & Image Processing: An International Journal*, 4(4):101–108, 2013.

[16] Ahmad Al Harere and Khloud Al Jallad. Mispronunciation detection of basic quranic recitation rules using deep learning. *arXiv preprint arXiv:2305.06429*, 2023.

[17] Ahmad Al Harere and Khloud Al Jallad. Quran recitation recognition using end-to-end deep learning. *arXiv preprint arXiv:2305.07034*, 2023.

[18] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[19] Md Afzal Hossan, Sheeraz Memon, and Mark A Gregory. A novel approach for mfcc feature extraction. In *2010 4th international conference on signal processing and communication systems*, pages 1–5. IEEE, 2010.

[20] Noor Jamaliah Ibrahim, Mohd Yamani Idna Idris, and Zulkifli Mohd Yusoff. Computer aided pronunciation learning for al-jabari method: A review. *QURANICA-International Journal of Quranic Research*, 6(2):51–68, 2014.

[21] Benjamin Kepecs and Homayoon Beigi. Automatic spoken language identification using a time-delay neural network. *arXiv preprint arXiv:2205.09564*, 2022.

[22] Boji Liu, Weibin Zhang, Xiangming Xu, and Dongpeng Chen. Time delay recurrent neural network for speech recognition. In *Journal of Physics: Conference Series*, volume 1229, page 012078. IOP Publishing, 2019.

[23] Hui Liu and Longlian Zhao. A speaker verification method based on tdnn–lstmp. *Circuits, Systems, and Signal Processing*, 38(10):4840–4854, 2019.

[24] Xiaoyong Liu and W Bruce Croft. Statistical language modeling for information retrieval. *Annu. Rev. Inf. Sci. Technol.*, 39(1):1–31, 2005.

[25] SEHM Metwalli. Computer aided pronunciation learning system using statistical based automatic speech recognition techniques. *Cairo University: Giza Governorate, Egypt*, 2005.

[26] Abdel-rahman Mohamed. *Deep neural network acoustic models for ASR*. PhD thesis, 2014.

[27] Abdel-rahman Mohamed, Geoffrey Hinton, and Gerald Penn. Understanding how deep belief networks perform acoustic modelling. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4273–4276. IEEE, 2012.

[28] Ammar Mohammed and MS Sunar. Verification of quranic verses in audio files using speech recognition techniques. In *1st International Conference of Recent Trends in Information and Communication Technologies (IRICT 2014)*, pages 370–381, 2014.

[29] Khalid M O Nahar, Mustafa Elshafei, Wasfi G Al-Khatib, Husni Al-Muhtaseb, and Mansour M Alghamdi. Statistical analysis of arabic phonemes used in arabic speech recognition. In *International Conference on Neural Information Processing*, pages 533–542. Springer, 2012.

[30] Michael Nguyen. Building an end-to-end speech recognition model in pytorch. *AssemblyAI. Accessed: Jun*, 8:2022, 2020.

[31] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Interspeech*, volume 2015, pages 3214–3218, 2015.

[32] Brian Roark, Murat Saraclar, and Michael Collins. Discriminative n-gram language modeling. *Computer Speech & Language*, 21(2):373–392, 2007.

[33] MA Sherif, A Samir, AH Khalil, and R Mohsen. Enhancing usability of capl system for quran recitation learning. INTERSPEECH, 2007.

[34] Andreas Stolcke et al. Srilm-an extensible language modeling toolkit. In *Interspeech*, volume 2002, page 2002, 2002.

[35] Imad K Tantawi, Mohammad AM Abushariah, and Bassam H Hammo. A deep learning approach for automatic speech recognition of the holy qur'ān recitations. *International Journal of Speech Technology*, 24(4):1017–1032, 2021.

[36] Jiri Valicek and Petr Mizera. Language models for spontaneous speech recognition. *POSTER 2015*, 14(05), 2015.