IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

Mispronunciation Detection and Diagnosis for Young Arabic Learners Using Transfer Learning

TAHA FANOUSH[1], WASFI G. AL-KHATIB[2], (Member, IEEE), MOHAMMAD AMRO[3], ABDULKAREEM ALZAHRANI[4], and MOUSTAFA ELSHAFEI[5](Senior Member, IEEE)

[1]Computer Science Department, University of Benghazi, Benghazi, Libya (e-mail: taha.fanoush@uob.edu.ly)
[2]Information and Computer Science Department, King Fahd University of Petroleum & Minerals, Dhahran 31261 Saudi Arabia (e-mail: wasfi@kfupm.edu.sa)
[3]Interdisciplinary Research Center for Intelligent Secure Systems, King Fahd University of Petroleum & Minerals, Dhahran 31261 Saudi Arabia (e-mail: mamro@kfupm.edu.sa)
[4]Islamic and Arabic Studies Department, King Fahd University of Petroleum & Minerals, Dhahran 31261 Saudi Arabia (e-mail: akareem@kfupm.edu.sa)
[5]Department of Communications and Information Engineering, Zewail City University of Science and Technology, 6th of October City 12578, Egypt (e-mail: moelshafei@zewailcity.edu.eg)

Corresponding author: Wasfi G. Al-Khatib (e-mail: wasfi@kfupm.edu.sa).

This work was supported by the Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS) at King Fahd University of Petroleum & Minerals under Project No. INSS2211.

ABSTRACT Improving primary school students' reading skills supports their academic growth and communication abilities. Pronunciation accuracy is central to reading, especially in Arabic, where small diacritic changes can alter meaning. This is complicated by Arabic's low-resource nature. This study developed a Mispronunciation Detection and Diagnosis (MDD) system for Arabic learners, allowing teachers and learners to use Computer-Assisted Pronunciation Training (CAPT) for improved instruction and assessment. A pretrained self-supervised learning (SSL) model was fine-tuned to detect phoneme-level pronunciation errors in Modern Standard Arabic using a unique dataset of primary school learner speech from Saudi Arabia. The data were structured, preprocessed, normalized, and aligned to phoneme sequences. The system showed improved phoneme recognition and performance approaching that of a human expert with an F1 score of 71.4%.

INDEX TERMS Arabic Mispronunciation Detection and Diagnosis (MDD), Computer-Assisted Pronunciation Training (CAPT), Phoneme Alignment, Self-Supervised Learning (SSL), Transfer learning, Speech technology for underrepresented languages, Arabic mispronunciation dataset, Artificial Intelligence (AI) in education

I. INTRODUCTION

READING proficiency in early education depends on accurate pronunciation, particularly in languages such as Arabic where minor diacritic variations can substantially alter lexical meaning. Therefore, enhancing pronunciation accuracy is central to developing primary school learners' literacy and communication skills.

Pronunciation training in Arabic presents distinct challenges, owing to the language's complex phonological structure and dense diacritic system. Inadequate articulation impairs reading fluency and also hinders language acquisition and comprehension. Thus, reliable automated pronunciation assessment tools are essential for supporting Modern Standard Arabic (MSA) learners.

Computer-Assisted Pronunciation Training (CAPT) systems are effective language learning tools, delivering automated feedback and adaptive instructions. However, Arabic CAPT systems, particularly those focused on Mispronunciation Detection and Diagnosis (MDD), remain underdeveloped.

This study aimed primarily to investigate whether self-supervised transfer learning models can be fine-tuned for effective mispronunciation detection in Arabic, particularly for continuous speech from young learners in real-world settings. An MDD framework tailored to MSA was introduced and optimized for primary school student speech. Leveraging recent advancements in self-supervised learning (SSL) and speech processing, the proposed system automatically identified and analyzed phoneme-level mispronunciations. This establishes a foundation for consistent, scalable, and accurate feedback to support both pronunciation training and reading assessment in MSA.

II. PROBLEM DESCRIPTION

Over the past decade, Computer-Assisted Language Learning (CALL) systems have expanded rapidly across many languages for children's education. Various Automatic Pronunciation Assessment (APA) systems have been designed for young learners, such as Krajka's "English for Kids" [1] and similar tools targeting early English education. However, research on CALL for Arabic education is still in the early stages [2], [3]. Very few studies have explored the use of deep learning (DL) models for Arabic MDD.

CALL systems are challenged by Arabic's complex phonetic structure, dense diacritic system, and vowel realization. Arabic includes emphatic consonants and short vowels that may not be marked in the written text, thus increasing pronunciation uncertainty [4], [5]. Additionally, the incorrect realization of a short vowel can alter the entire meaning of a sentence, as discussed in Section III-B. Unlike languages with simpler phoneme structures, Arabic pronunciation systems must handle phoneme-level errors accurately, particularly those involving short vowels and gemination. Arabic MDD systems require sophisticated alignment methods, robust phoneme recognition modules, and precise diagnostic feedback mechanisms to detect these linguistic characteristics.

Addressing these challenges requires automated solutions that offer learners effective, consistent feedback in a scalable manner. CALL systems equipped with speech recording and processing technologies can automate oral reading assessments. Within this framework, CAPT systems present a potential solution by analyzing spoken input and providing automated pronunciation feedback.

III. BACKGROUND AND CONTEXT

A. ARABIC LANGUAGE TYPES

Arabic is one of the most widely spoken languages, with over 400 million native and non-native speakers worldwide [6], [7]. The main classifications are Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectical Arabic (DA) [8].

CA is the language of the Holy Quran and early literature, including poetry [9]. MSA, derived from CA, is the official language of 22 countries. It is used in formal contexts, such as education, media, and literature [10]. DA refers to the informal and regional forms of Arabic that are used in daily conversations, on social media, and in broadcasts [11], [12]. These forms differ significantly from CA and MSA, reflecting regional linguistic and cultural developments [8].

B. ARABIC ALPHABET AND VOWELS

The Arabic writing system consists of 28 basic letters. Each letter also has a distinct pronunciation. In the written form, these letters change shape depending on their position within a word [13]. Several commonly used symbols and letter variants, such as Hamza (ء), Alif (ا), and Ya (ي), contribute to a total of 34 symbols. These additional forms represent specific sounds or phonetic distinctions. Positional flexibility contributes to Arabic's fluid, connected script and phonetic richness.

Vowels play a critical role in shaping the meaning of Arabic words. Unlike English, in which vowels are fully represented by letters ("a", "e", "i", "o", and "u"), Arabic distinguishes between short and long vowels. Short vowels are represented by diacritic marks placed above or below consonants to modify them, whereas long vowels are represented by specific letters. Short vowels in Arabic are Fatha (َ), Damma (ُ), and Kasra (ِ). They represent brief vowel sounds and are typically omitted from Arabic writing unless clarity is required. In contrast, long vowels are represented by the letters Alif (ا), Waw (و), and Ya (ي). These are the extended versions of short vowels and have stretched sounds. Long vowels are written using dedicated letters, giving them the same level of importance as consonants. Table 1 presents examples of short and long Arabic vowels.

Table 1: Short and Long Arabic Vowels with English Equivalents

Arabic Vowel	Arabic Example	English Sound	Sound Example
َ (Fatha - short)	سَمَك (samak)	a	as in "travel"
ُ (Damma - short)	كُتُب (kutub)	u	as in "sugar"
ِ (Kasra - short)	بِنت (bint)	i	as in "hint"
ا (Alif - long)	بَاب (baab)	aa	as in "band"
و (Waw - long)	نُور (noor)	oo	as in "school"
ي (Ya - long)	فِيل (feel)	ee	as in "feel"

Gemination, known as شدّة (shadda or tashdeed) in Arabic, refers to adding a silent copy of the preceding consonant. This produces the sound of two consonants in pronunciation. However, in its written form, the word has a single consonant with gemination. Gemination is indicated by placing the shadda diacritic (ّ) above the consonant, often accompanied by a short vowel marker positioned above or below the shadda. Small diacritic and/or long vowel changes can significantly affect how a sentence is read and understood [14], [15], as shown in Table 2.

Table 2: Effects of Diacritic Change on Meaning

Arabic Text	Transliteration (Sound)	English Meaning
مَسَكَ الشَّيْخُ الرَّجُلَ	masaka ash-shaykhu ar-rajula	The sheikh caught the man.
مَسَكَ الشَّيْخُ الرِّجلَ	masaka ash-shaykhu ar-rijla	The sheikh caught the leg.

Since short vowels are typically not written in informal texts, this can lead to misinterpretations, especially

for non-native Arabic speakers. Accurate Arabic pronunciation and comprehension rely heavily on correctly understanding vowel markings within a given sentence. This positions CALL tools as promising approaches for assisting learners in mastering accurate pronunciation.

C. COMPUTER-ASSISTED LANGUAGE LEARNING

CALL systems enhance language learning by leveraging digital platforms and speech-processing technologies to deliver interactive, self-paced, and cost-effective learning with personalized guidance and real-time feedback [16], [17]. These systems also support educators by automating language-proficiency assessments.

In pronunciation-focused applications, CALL systems, referred to as CAPT, aim to improve pronunciation using feedback mechanisms and detailed error analysis. Recent machine learning (ML) advances have enabled the development of automated AI-driven CAPT systems.

Speech assessment, which evaluates broader speech abilities, such as fluency, language skills, and oral-motor function, is often distinguished from pronunciation assessment, which focuses on articulatory accuracy.

APA, also known as automatic pronunciation scoring, uses algorithms to analyze spoken input, detect errors, and provide feedback. It is sometimes referred to as MDD [18]–[25] or Automatic Pronunciation Error Detection (APED) [26]. While APA typically provides holistic proficiency scores or categorical ratings, MDD focuses on identifying and classifying specific mispronunciations.

Both scoring and error detection are important components of CAPT systems, offering complementary feedback to learners. CAPT systems powered by ML and DL techniques are increasingly applied in language learning, pronunciation training, and speech therapy to assist educators, learners, and clinicians with evaluations and progress tracking.

D. MISPRONUNCIATION DETECTION AND DIAGNOSIS

Automatic pronunciation assessment by scoring pronunciation quality or detecting errors is a key MDD research area. These assessments can be applied to different languages, speech styles (oral reading vs. spontaneous speech), and segmentation levels (phonemes, words, or sentences). In particular, phonemes, which are the smallest units of sound, are fundamental to pronunciation assessment [27].

Different languages require different assessment techniques because of the variations in articulation [28]. In Arabic, MDD studies have targeted CA, particularly Quranic recitation [29], [30], and DA [31] to a lesser extent. Most Arabic speech processing efforts, including MDD and Automatic Speech Recognition (ASR), have focused on MSA [8]. Similarly, this study focused on MSA.

MSA pronunciation and orthography differ significantly from those of Quranic and dialectal Arabic languages. MDD in MSA uses isolated phonemes [30], isolated words [32], or both [23], [33], [34] to assess pronunciation; however, sentence-level MDD is more common in English and other languages. Arabic MDD research relies mostly on isolated style data. Earlier studies have often used speaker-dependent corpora, although recent work has shifted toward speaker-independent data.

MDD presents unique challenges in Arabic because of its complex phonetic system, diacritics, and the importance of accurate articulation. Additionally, Arabic has a rich morphology expanding at a rate approximately 2.5 times faster than that of English [35].

E. FEEDBACK

CAPT systems face persistent challenges in delivering accurate and actionable feedback, particularly for automatic diagnosis of pronunciation errors [36]. Although some English-based tools offer basic feedback on sounds (e.g., Sounds App [37]), stress patterns (e.g., eEnglish [38]), and phrases (e.g., Rosetta Stone [39] and Duolingo [40]), few systems offer precise personalized feedback without expert involvement [41].

This limitation is even more evident in the Arabic language. Although MDD research has advanced, it tends to emphasize error detection over providing user-friendly and detailed feedback. Many systems focus on comparing generated phonemes with canonical references and often require expert interpretation. This makes the feedback less practical for learners.

Qvoice [34] is one of the few systems designed for Arabic language learning, offering feedback at both the character and word levels. It provides users with star ratings, color-coded accuracy indicators [ranging from red (inaccurate) to green (accurate)], and tailored feedback for L1 and L2 speakers.

Other studies have explored articulatory feedback in which mispronounced phonemes are analyzed based on articulatory features (AFs), such as the place and manner of articulation, voicing, and emphasis [20]. This approach aims to explain how learners' articulations differ from their targets.

Feedback methods vary widely, ranging from binary correctness labels to highlighting specific mispronounced phonemes. However, many systems do not elaborate on or specify feedback mechanisms. These inconsistencies underscore the need for comprehensive and interpretable feedback designs that offer better assistance for language learners.

In Figure 1, we propose an MDD system that can provide meaningful feedback to learners. This figure illustrates the feedback pipeline used to evaluate student reading by visualizing the end to end process from the learner's reading of the **input** to feedback generation. It also presents the trained model's qualitative assessments

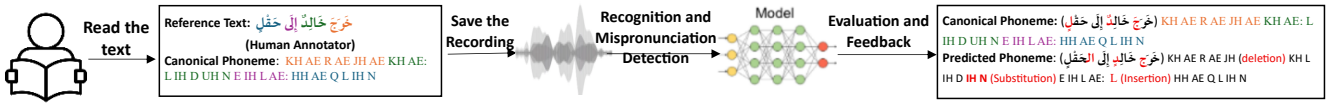

Figure 1: Proposed MDD system with feedback.

after **aligning the learner's predicted phoneme sequence with a canonical phoneme sequence**. The system can identify the following types of errors.

- **Deletion**: خَرَجَ was mistakenly pronounced as خَرَجَ, omitting the Fatha short vowel ـَ on the letter ج.
- **Substitution**: The nunation ـٌ on the final letter د in خَالِدٌ was incorrectly replaced by ـٍ, producing خَالِدٍ.
- **Insertion**: حَقْلٍ was mistakenly pronounced as الحَقْلِ by inserting the definite article ال at the beginning of the word.

IV. ORGANIZATION OF THE PAPER

Section V elaborates on previous studies on Arabic MDD. Section VI describes our proposed framework. The data preparation process is described in Section VII. Section VIII describes the experimental design. Finally, Section IX presents the implementations details and results, followed by our conclusions and recommendations for future work in Section X.

V. LITERATURE REVIEW

Early MDD, also known as APA, relied on traditional techniques such as force alignment, ASR-based segmentation, and handcrafted features [42]. These approaches often required complex pipelines and large labeled datasets. Recent advances favor DL methods, particularly Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs) [33], as well as SSL models, such as Wav2vec 2.0 and HuBERT [43]. These methods extract meaningful speech representations from raw audio with minimal labeled data. MDD approaches can be classified as: (1) conventional & statistical methods, (2) DL-based methods, and (3) comparative & mixed approaches.

A. CONVENTIONAL AND STATISTICAL METHODS

Traditional MDD approaches include ASR-based systems, statistical modeling, and handcrafted acoustic features. Khan et al. [2] developed an ASR method based on a Hidden Markov Model for Arabic pronunciation assessment. This ASR achieved an accuracy of 89.69% using a corpus of native and non-native speakers. Eljazzar and Alagrami [29] proposed a Support Vector Machine(SVM)-based system for evaluating the Holy Quran recitation using threshold scoring. Their dataset comprised 1,257 labeled recordings across eight classes, providing rule-specific feedback on correctness. El Amrani et al. [44] criticized Romanized Arabic phonemes for their complexity and used simplified Arabic phonemes for Quran recitation recognition. Despite achieving a 1.5% Word Error Rate (WER) on the training data, the WER increased to 50% on the unseen data owing to dataset limitations. Calik et al. [30] applied ensemble learning to CA phoneme recognition using Mel-Frequency Cepstral Coefficients (MFCCs) and Mel-spectrograms. Their system, trained on an augmented dataset of 1,450 samples, achieved an accuracy of 95.3% using k-nearest neighbors (KNN) and majority voting. Maqsood et al. [45] used ANNs with acoustic phonetic features to detect mispronunciations of Arabic consonants. By categorizing consonants based on their acoustic similarity, they achieved an average accuracy of 82.27% for a 5,600-sample corpus.

Necibi and Bahi contributed to multiple MSA-focused studies. Their early work [32] assessed children's pronunciation using threshold-based word comparisons. Later, [46] introduced a scoring system using the Global Average Log Likelihood (GLL), which achieved an accuracy of 97.31%. In [47], machine scores were aligned with human ratings by using decision trees and Viterbi decoding. To address subjectivity in scoring, Bahi and Necibi [3] proposed a fuzzy-logic-based assessment system (FuSPA) that categorized pronunciation as "good," "poor," or "acceptable." Despite low inter-rater agreement, the FuSPA aligned with at least one expert's judgment.

Other related studies include Lee and Glass [31]. They used dynamic time warping (DTW) with MFCCs and posterior grams for misalignment scoring in Levantine DA. Applied to 2,064 utterances, their comparison-based method improved the machine-human score correlation and reduced the mean square error (MSE).

These studies used scoring techniques such as phoneme duration, articulation rate, log-likelihood, and goodness of pronunciation (GOP) [32]. Anzola and Moreno [48] evaluated GOP on embedded devices, noting its effectiveness for segmental analysis as well as its limitations in capturing suprasegmental features.

B. COMPARATIVE AND MIXED APPROACHES

Akhtar et al. [33] improved Arabic mispronunciation detection by extracting deep features from AlexNet CNN layers and training KNN, SVM, and Random Forest classifiers. Using an in-house dataset generated by 30 speakers, their CNN-based system achieved an accuracy of 93.20%, outperforming the MFCC-based methods. Ahmed et al. [23] developed an LSTM-based system trained on the MFCC features to detect

mispronunciations in Arabic. Testing this system on 3,120 utterances from 53 speakers yielded an accuracy of 81.52%. Gender recognition was included in the analysis but did not significantly impact performance. Nazir et al. [21] proposed two approaches for Arabic phoneme mispronunciation detection: one using CNN-extracted features and another based on transfer learning from pretrained CNN models. Using a dataset generated by 400 Pakistani learners, both methods outperformed traditional systems. Transfer learning achieved an accuracy of 94.6% and CNN features achieved 91.3%.

C. DEEP-LEARNING-BASED METHODS

Recent DL advances, including transfer learning and SSL, have significantly enhanced MDD accuracy and robustness. These methods support nuanced pronunciation scoring, improve generalization, and perform well in low-resource scenarios.

1) Various APA Applications and Techniques

Lin and Wang [49] introduced a transfer learning approach using deep features from ASR models and an attention-based scoring module. Fine-tuning with human-labeled data improved scoring accuracy, achieving Pearson correlation coefficients (PCCs) of .86 on L2 and .72 on Speechocean762 datasets. Zahran et al. [22] fine-tuned a self-supervised model for phoneme recognition and used a Siamese network to compare the learned embeddings with canonical phonemes. This end-to-end system achieved a PCC of .81 on Speechocean762. Kim et al. [50] explored transformer-based SSL models (e.g., Wav2vec 2.0 and HuBERT) fine-tuned for pronunciation scoring. Using a dataset generated by Korean English as a Second Language (ESL) learners and the Speechocean762 dataset, the bidirectional LSTM-based scoring module achieved PCCs of .82 and .78, respectively.

2) MDD Approaches for Isolated Words & Letters

Asif et al. [4] developed a Deep Neural Network (DNN) model to detect short-vowel mispronunciations in CA. Trained on a dataset of 6,229 vowel recordings across 84 phoneme classes, the system achieved an accuracy of 95.77%. Çalık et al. [51] evaluated transformer-based models (HuBERT, Wav2vec, and UniSpeech) for phoneme detection using a dataset of 29 Arabic phonemes. UniSpeech achieved 94.4% accuracy with a precision and recall of approximately 95%. El Kheir et al. [34] developed QVoice, which is a mobile app for Arabic pronunciation training. It provides end-to-end mispronunciation detection and character-level feedback using an attention-based mechanism. The app processes over 400 words with a sub-1.5-second response time; however, the model specifications were not disclosed.

3) MDD Approaches for Continuous Speech

a: Non-Transfer-Learning Methods

Algabri et al. [20] treated MDD and articulatory feature detection as a multi-label object recognition problem using spectrogram images. Their model was trained on real speech and text-to-speech (TTS) augmented speech from 182 speakers. It achieved a Phoneme Error Rate of 3.83%, an F1 score of 70.53%, and a detection error rate of 2.6%. Zhang et al. [26] proposed an end-to-end ASR system that combined Connectionist Temporal Classification (CTC) and attention mechanisms for APED in Mandarin. Avoiding force alignment, their hybrid model achieved an F1 score of 87.3% using data from the CCTV, PSC-G1-112, and PSC-1176 corpora.

b: Transfer-Learning-Based Methods

Peng et al. [18] used Wav2vec 2.0, with textual modulation gates and contrastive loss, for MDD. Training on the L2-ARCTIC and TIMIT corpora achieved an F1 score of 61.75%. Yang et al. [19] applied momentum pseudo-labeling to fine-tune SSL models for unlabeled L2 speech. Using the L2-ARCTIC and UTD-4Accents corpora, they reduced the Phoneme Error Rate (PER) to 15.3% and achieved an F1 score of 68.4%. Shen et al. [43] fine-tuned Wav2vec 2.0 and WavLM for Mandarin MDD, achieving 91.3% accuracy and an 89.7% F1 score. Their approach performed well in low-resource settings. Xu et al. [52] fine-tuned Wav2vec 2.0, using adaptive pooling layers for pronunciation classification. With L2-ARCTIC and limited labeled data, the model achieved an F1 score of 69.2.

D. ANALYSIS OF EXISTING LITERATURE

Arabic MDD research is therefore still limited, especially for continuous speech, in young learners, and in L2 contexts. Most previous studies have focused on adult English and Mandarin learners. Public large-scale Arabic pronunciation corpora are scarce, which hinders robust model development. To the best of our knowledge, only two studies have applied pretrained SSL or transfer learning models to continuous Arabic for MDD in L2 or young learners. The first was by Nazir et al. [21], who used AlexNet for L2 isolated phoneme detection. The second was by Necibi and Bahi [3], [32], [46], [47], who developed statistical systems for young native Arabic learners. Additionally, QVoice [34] provides character- and word-level MDD through a mobile app but lacks detailed descriptions of model specifications. More recently, El Kheir et al. [55] developed an Arabic dataset consisting of 82.37 hours of recorded speech taken from the Common Voice dataset version 12.0, augmenting it with 52 hours of synthetic data including Quranic recitations using MSA rules. These recitations, recorded in a controlled environment, included mistakes embedded in the data supplied to the readers. Although this is an important advancement in

Arabic benchmark dataset development, mistakes were not naturally generated, which is important for the advancement of MDD systems.

These findings underscore the need for comprehensive utterance-level MDD approaches for Arabic. SSL-based solutions show promise for addressing data scarcity and supporting effective pronunciation training. Integrating a pretrained self-supervised model into a CAPT system tailored for young Arabic learners can improve MDD accuracy. Furthermore, fine-tuning the model with a limited amount of annotated data is expected to enable expert-level feedback and reduce the burden on human instructors.

VI. ARABIC MISPRONUNCIATION DETECTION AND DIAGNOSIS FRAMEWORK

This section presents the proposed MDD framework, focusing on DL and self-supervised speech representation techniques. The framework begins with the model architecture and pretraining phase. It concludes with a performance evaluation.

A. DEEP-LEARNING-BASED FEATURE EXTRACTION

DL methods have been proven to be more reliable than traditional statistical and handcrafted MDD approaches. Unlike GOP, MFCCs, and Filter Banks (fBanks), DL models automatically learn meaningful speech representations from raw audio or spectrograms [4], [19], [26].

Studies such as [20], [21], [33] have demonstrated the effectiveness of CNNs in extracting relevant features and capturing complex speech patterns that traditional techniques may miss. Their ability to learn directly from data makes them well-suited for building robust pronunciation assessment systems, particularly in low-resource contexts, such as Arabic.

B. SELF-SUPERVISED LEARNING PRETRAINED MODELS

SSL models have performed well in low-resource speech applications, making them valuable for MDD [43]. Models such as Wav2vec 2.0 [56], HuBERT [50], [57], WavLM [58], and Whisper [59] offer robust speech representations learned from large-scale unlabeled data. However, HuBERT and WavLM require extensive fine-tuning for Arabic because of their English-centric pretraining [58], [60]. Whisper, while multilingual, focuses on speech-to-text communication and lacks phoneme-level resolution [61].

Wav2vec 2.0 has proven to be more effective for MDD tasks [18], [19], [24], [43], [50], [52], [62]. It captures fine-grained acoustic features and supports phoneme recognition using limited labeled data. Additionally, several Arabic-fine-tuned Wav2vec 2.0 models are available, providing a practical foundation for Arabic mispronunciations detection. Its robustness to noise and native support in platforms like Hugging Face Transformers [63] further support its adoption in real-world CAPT systems. Hugging Face offers an extensive collection of pretrained speech models, including the XLSR-53 variant [64].

1) Wav2vec 2.0

Wav2vec 2.0 comprises a CNN encoder, a transformer network, and a quantization module [56]. The encoder extracts latent features from raw audio data, the transformer contextualizes these features, and the quantization module provides the pretraining targets. During fine-tuning, the quantization module is removed and task-specific layers are added.

This model learns by contrasting true quantized representations with distractors, thereby enabling the development of discriminative phoneme-level embeddings. It performs well when trained on a limited amount of labeled data. Furthermore, it supports accurate phoneme recognition, which is critical for MDD. Wav2vec 2.0 is reported to have a WER $< 5\%$ for LibriSpeech, using only 10 minutes of labeled data [65].

Replacing handcrafted features with raw audio embeddings from Wav2vec 2.0 improves performance, with reported WER reductions of up to 36% in low-resource settings [66]. Its attention mechanism helps detect subtle pronunciation errors by modeling long-range phonetic dependencies [67].

C. PRETRAINED MODEL VARIANTS

Wav2vec 2.0 offers multiple variants designed for different speech-processing scenarios.

- **Base Model:** Lightweight and efficient. Suitable for smaller datasets or low-resource environments.
- **Large Model:** Contains more transformer layers. Ideal for high-resource settings requiring top performance.
- **XLSR-53 (Cross-Lingual):** Trained on 53 languages including Arabic. Useful for multilingual applications.

Many variants are fine-tuned for downstream tasks and are publicly available on platforms such as Hugging Face [63]. Models that are fine-tuned for Arabic ASR tasks provide an advantage for Arabic MDD because they are better at capturing the phonetic features of the language [68]. Among the available Wav2vec 2.0 variants, XLSR-53 [69] offers unique advantages as the only variant explicitly pretrained on multilingual data, including Arabic. The Base and Large variants were primarily trained on English datasets. Consequently, XLSR-53 captures greater phonetic diversity and dialectal variation, making it well-suited for the complexity and phonetic subtleties of Arabic speech.

The Wav2vec2.0 XLSR-53 model architecture, including the CNN encoder and transformer encoder layers, follows the original design proposed by Baevski et al. [56]

and Conneau et al. [69]. It does not feature structural modifications for specific languages. Instead, the CNN encoder employs fixed kernel sizes and strides that are designed to efficiently capture relevant local acoustic features across languages. The transformer network, with its predefined number of attention heads and layers, inherently captures long-range dependencies that are critical for modeling continuous speech. Thus, this study adapted the Wav2vec2.0 XLSR-53 model to Arabic speech solely by fine-tuning the existing model using domain-specific Arabic datasets. This allowed the model to adapt its internal representations implicitly to Arabic phonetic and acoustic characteristics without explicitly modifying the model's original architecture parameters.

This study selected model variants that were already fine-tuned on Arabic speech data and were compatible with existing frameworks. This ensured optimal integration and performance for MDD. We selected two XLSR-53 model variants fine-tuned for Arabic: El-Geish [70] and Grosman [71]. They differ primarily in their fine-tuning specifications. The El-Geish [70] model was initially fine-tuned using the Arabic Speech Corpus [72] dataset and then further fine-tuned using Mozilla's Common Voice Arabic [73] dataset (version 6.1). In contrast, the Grosman [71] model was fine-tuned using the combined training and validation splits of both the Common Voice Arabic [73] dataset (version 6.1) and the Arabic Speech Corpus [72]. These variants demonstrated good performance, reporting the lowest WER scores in Arabic ASR benchmarks compared with the other XLSR-53 variants.

D. PROPOSED APPROACH

The proposed Arabic MDD method comprises two main stages: **phoneme recognition** and **mispronunciations detection**. First, the model processes an input speech file and outputs a sequence of predicted phonemes. These predictions are compared to manually labeled perceived phonemes to calculate the PER, which reflects the model's phoneme recognition accuracy.

Next, mispronunciations are identified by analyzing the discrepancies between the predicted and canonical phonemes and comparing these with the differences identified by human annotators. The F-measure is used to assess the alignment between system-detected and manually identified mispronunciations, offering insights into both precision and recall.

Algorithm 1 outlines the key steps in this approach.

Algorithm 1: Proposed Arabic MDD Procedure

Input: Arabic speech audio A, lesson text T, pretrained MDD model
Output: Mispronunciations (Insertions, Deletions, Substitutions), PER, F1score

Step 1: Data Preparation
- Normalize T to canonical form T'_c;
- Obtain the perceived text T'_p;
- Convert T'_c, T'_p to phoneme sequences P_c, P_p;
- Align P_c and P_p using Needleman-Wunsch method;

Step 2: Model Input Preparation
- Extract features from A;
- Prepare aligned sequences $P_c^{aligned}$ and $P_p^{aligned}$;

Step 3: Phoneme Recognition
- Encode features using CNN and transformer;
- Apply masking;
- Fine-tuning with CTC loss;
- Decode the output using CTC decoding;

Step 4: Mispronunciation Detection
- Align recognized and canonical phoneme sequences;
- Identify mismatches;
- Classify as an insertion, deletion, or substitution;

Step 5: Evaluation
- Compute PER from phoneme alignment;
- Compute F1 score from comparison with human labels;

E. PHONEME RECOGNITION

1) Model Architecture

The phoneme-recognition stage is based on the Wav2vec 2.0 model (see Section VI-B1), which consists of the following components:

- CNN Feature Extractor: Converts raw audio into latent representations.
- Transformer Encoder Layers: Capture temporal context across the speech signal.
- Linear Output Layer: Maps contextualized features to phoneme probability distributions.

Together, these modules enable end-to-end phoneme prediction from speech.

2) Pretraining with Wav2vec 2.0

Wav2vec 2.0 is pretrained using self-supervised contrastive loss on unlabeled speech data. It uses a masking strategy to randomly mask the input segments and predict the corresponding quantized latent speech units, thereby improving the ability of the model to capture context and phonetic structure.

3) Fine-Tuning

Fine-tuning adapts the pretrained model for phoneme recognition using labeled data. This addresses domain mismatches in speech data. Training and evaluation speech data may differ significantly, such as clean-read speech versus learner speech [19]. We followed fine-tuning best practices [19], [24], [52]:

- Add linear layers atop the transformer outputs.
- Remove the pretraining quantization module.
- Freeze CNN layers to retain pre-learned acoustic features.

We used CTC as the loss function, which enables alignment-free learning from variable-length inputs and target phoneme sequences [74]. CTC facilitates modeling of insertions, deletions, and substitutions for frequent occurrences in spontaneous and learner-produced speech. This is achieved by implicitly optimizing the alignment between the audio frames and phoneme labels.

Let $X = (x_1, x_2, \ldots, x_T)$ be the input audio features and $Y = (y_1, y_2, \ldots, y_L)$ be the target phoneme sequence. The CTC loss is given by

$$\text{CTC}_{\text{loss}} = -\log \sum_{S \in \beta(Y)} \prod_{i} P_\theta(s_i|X), \tag{1}$$

where $S = (s_1, s_2, \ldots, s_T)$ is a valid alignment from the input to the output, $\beta(Y)$ is the set of all possible alignments of Y, and $P_\theta(s_i|X)$ is the model's predicted probability of phoneme s_i given X and parameter θ. The log-sum formulation facilitates gradient-based optimization.

Mathematically, as shown in Equation (1), CTC considers all possible alignments between the input acoustic features and annotated phoneme sequences, computes the probabilities of these alignments, and optimizes the network to predict the most likely phoneme sequence. This inherent flexibility of CTC, including its ability to handle blank tokens and repeated phonemes, makes it particularly effective in addressing Arabic-specific pronunciation variations and ensuring robust mispronunciation detection without explicitly aligned training data. Fig. 2 illustrates the complete phoneme recognition workflow, including the pretraining, fine-tuning, and inference stages.

F. MISPRONUNCIATION DETECTION AND DIAGNOSIS

This stage aims to identify discrepancies between the recognized phoneme sequence and canonical pronunciation to detect mispronunciations and provide diagnostic feedback. The process begins by aligning recognized phonemes with the canonical phoneme sequence using dynamic programming algorithms, thereby enabling accurate comparisons despite sequence variations.

Once aligned, the system detects and categorizes the pronunciation errors into three types:

- **Insertion:** An extra phoneme is added to the recognized sequence that is not in the canonical sequence.
- **Deletion:** A phoneme is missing from the recognized sequence that is present in the canonical sequence.
- **Substitution:** A phoneme present in the canonical sequence is replaced with an incorrect one in the recognized sequence.

The system identifies mismatches, classifies them into error types, and localizes the position and nature of each mispronunciation. This fine-grained analysis is essential to deliver targeted feedback to learners.

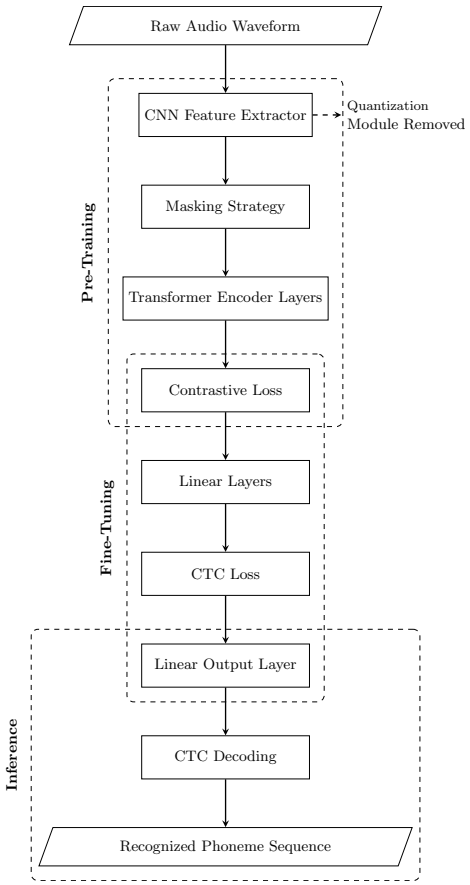

Figure 2: Workflow diagram of phoneme recognition, illustrating pretraining, fine-tuning, and inference.

G. EVALUATION

Following the approaches in related MDD studies [19], [24], [26], [75], we evaluate model performance using two metrics: PER for phoneme recognition accuracy, and the F1 score for mispronunciation-detection effectiveness.

1) Phoneme Error Rate

PER measures how accurately the model predicts phoneme sequences with reference to the ground truth. The recognized phoneme sequence is aligned with the reference sequence using an algorithm, such as Levenshtein distance. Substitutions (S), deletions (D), and insertions (I) are counted relative to the total number of reference phonemes (N).

$$\text{PER} = \frac{S + D + I}{N} \times 100\%. \tag{2}$$

This metric evaluates whether the correct phonemes are predicted in the correct order with minimal distortion.

2) F1 Score

The F1 score evaluates mispronunciation detection as a binary classification task based on the system's ability

to distinguish between correct and incorrect phoneme realizations. We define:

- **True Accept (TA):** Correct pronunciation correctly classified.
- **False Reject (FR):** Correct pronunciation misclassified as incorrect.
- **False Accept (FA):** Incorrect pronunciation misclassified as correct.
- **True Reject (TR):** Mispronunciation correctly detected.

True Rejects (TR) are further divided into:

- **Correct Diagnosis (CD):** Mispronunciation type correctly identified.
- **Diagnosis Error (DE):** Mispronunciation type incorrectly identified.

From these, the Precision (P), Recall (R), and F1 scores are calculated as follows:

$$\mathrm{P} = \frac{\mathrm{TR}}{\mathrm{FR} + \mathrm{TR}}, \tag{3}$$

$$\mathrm{R} = \frac{\mathrm{TR}}{\mathrm{FA} + \mathrm{TR}}, \tag{4}$$

$$\mathrm{F1} = \frac{2 \cdot \mathrm{P} \cdot \mathrm{R}}{\mathrm{P} + \mathrm{R}}. \tag{5}$$

Here, P measures the number of phonemes labeled as mispronunciations that are truly incorrect, whereas R measures the number of actual mispronunciations that are correctly flagged as errors. The F1 score, the harmonic mean of P and R, provides a balanced view of system effectiveness.

This evaluation procedure ensures that both recognition and diagnostic accuracy are rigorously assessed.

VII. DATA PREPARATION

Developing effective MDD systems requires specialized pronunciation datasets that go beyond the standard ASR data. While ASR datasets focus on accurate transcription, MDD datasets must capture fine-grained phoneme articulations, including both correct and incorrect pronunciations.

This section describes the preparation of a dedicated Arabic pronunciation dataset for MDD. This process involved collecting speech recordings with rich metadata, aligning them with canonical text, and generating both canonical and perceived phoneme sequences. The procedures used for data acquisition, text mapping, manual annotation, normalization, and phoneme sequence generation are described below.

A. DATA COLLECTION AND PREPROCESSING

The original dataset was collected using a web-based platform named راصد المقرأة (Rasid al-Miqra'a), which means "Reading Monitor." It was developed to record students' oral reading of displayed Arabic paragraphs. Professor Abdulkareem Alzahrani from the Islamic and Arabic Studies Department at King Fahd University of Petroleum & Minerals (KFUPM) managed the data collection [76]. The key data characteristics were as follows:

- Metadata of the student, including their name, gender, origin, and date of birth.
- Arabic paragraph text content was sourced from the Saudi Arabic primary school textbook لغتي (Lughati), Grades 1-6.
- Over 40 hours of students' reading were collected in 1,040 audio files in uncontrolled environments.

This dataset required extensive cleaning and restructuring. The preprocessing phase included the following steps:

- Remove duplicate recordings and irrelevant ones (e.g., those that do not correspond to a whole lesson).
- Standardize word spellings of metadata fields, such as origin and gender.
- Convert Hijri to Gregorian birth dates.
- Remove misrecorded or irrelevant samples.
- Map each audio file to its corresponding lesson text.

In this ongoing research, 25 speech recordings from different students (primarily in Grade 5) were selected for experimentation. In these recordings, students recited different lessons from the Saudi primary school Arabic textbook لغتي (Lughati). The speakers were from 15 cities across different Saudi provinces. This provided the corpus with a broad sample of Saudi (Gulf) dialectal variation. Because textbook lessons systematically cover every letter of the Arabic alphabet in multiple word positions, the resulting corpus is inherently balanced across phonemes and well suited for evaluating pronunciation. The total duration of these recordings was approximately 60 min. There were 14 boys and 11 girls, ranging in age from 8 to 11 years. The file durations ranged from 48 seconds to 5 minutes 51 seconds, with an average duration of approximately 2 minutes 24 seconds. Each file was paired with its corresponding lesson text for evaluation and annotation.

Guidelines for data annotation were developed and given to a team of Arabic faculty members from Egypt who were well-trained in Arabic text annotation. They listened to the same audio file as a team and produced a single annotation after resolving any issues arising from unclear audio. Their work was coordinated and further reviewed by Prof. Alzahrani. The annotation guidelines are given in Appendix A.

B. TEXT NORMALIZATION

Text normalization is a critical preprocessing step in MDD. It transforms Arabic text into a standardized, pronunciation-aligned form that matches the spoken phonemes. This process was applied to both the

canonical lesson text and manually annotated perceived speech, ensuring consistency and phoneme alignment across the dataset.

1) Normalization Rules

A Python script was developed to normalize the text based on the following linguistic rules:

- **Removal of non-pronounced letters:** Replaces or omits silent glyphs (e.g., ة).
- **Handling of definite article 'the' (ال):** Adjusts pronunciation based on whether the next letter is a 'sun letter' or 'moon letter'.
- **Normalization of irregular words:** Transforms words with pronunciation mismatches (e.g., هذا → هاذا).
- **Punctuation removal:** Eliminates symbols not pronounced in speech.
- **Prefix processing:** Handles attached particles (e.g., و, ب) and their influence on pronunciation.

As illustrated in Fig. 3, the text normalization process involves several systematic steps to ensure accurate phoneme-level representation, which is essential for effective MDD system training.

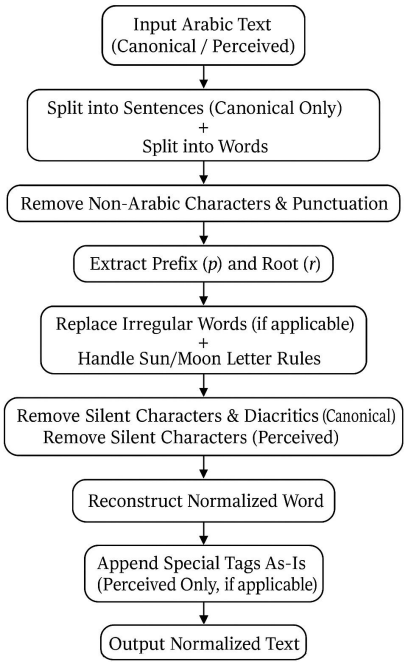

Figure 3: Flowchart illustrating key text normalization steps for canonical and perceived Arabic texts.

2) Normalization Algorithms

Two slightly different pipelines were created for canonical and perceived texts. Algorithms 2 and 3 describe their logic, respectively.

Both algorithms ensure that the text reflects pronunciation by handling script-level phenomena (e.g., silent letters, article assimilation, and diacritics) and speech annotations.

Algorithm 2: Canonical Text Normalization

Input: Canonical Arabic text T_c, sun letters set S, irregular word map W
Output: Normalized text T_c'
Split T_c into sentences;
foreach sentence s in T_c **do**
  Split s into words w_i;
  foreach w_i **do**
    Remove non-Arabic characters;
    Identify prefix p and root r;
    Apply irregular word replacement from W;
    Handle ال (sun/moon letter rules);
    Remove silent characters and diacritics;
    Reconstruct $w_i \leftarrow p + r$;
    Add "sil" for sentence boundaries;
return T_c'

Algorithm 3: Perceived Text Normalization

Input: Annotated text T_p, sun letters set S, irregular word map W
Output: Normalized text T_p'
Remove punctuation and replace '@' with "sil";
foreach word w_i in T_p **do**
  if w_i is a special tag **then**
    Append as-is;
  else
    Clean w_i, extract p, r;
    Replace r if in W;
    Adjust for sun/moon letters;
    Remove silent characters;
    Reconstruct and append;
return T_p'

3) Text Normalization Example

Table 3 shows examples of text before and after normalization.

Normalization removes unpronounced glyphs and maps grammatical features to their realization in speech. It also integrates transcriber annotations, such as:

- **Pauses:** '@' is replaced with the token "sil".
- **Non-speech:** Tags like #ضحك#, #تشويش# are processed appropriately.
- **Special markers:** Symbols indicating emphasis, softening, or hesitation are preserved for accurate representation.

By applying normalization to both canonical and perceived texts, we ensure consistency by generating phoneme sequences that mirror actual pronunciation. This step is critical for improving phoneme alignment and enhancing MDD system performance.

C. PHONEME SEQUENCE PREPARATION

Table 3: Examples of Text Before and After Normalization

Original Text	Normalized Text
جَلَسَتِ الْأُسْرَةُ حَوْلَ الْمَائِدَةِ	جَلَسَتِ لْأُسْرَةُ حَوْلَ لْمَائِدَه
قَالَ فَوَّازٌ: أُمِّي تَسْقِي وَرْدَ الْحَدِيقَةِ	قَالَ فَوَّاز sil أُمِّي تَسْقِي وَرْدَ لْحَدِيقَه

1) Phoneme Sequence Acquisition

Character-to-ASCII transliteration systems such as Buckwalter [77] are useful for text processing. However, they were not designed for phoneme extraction [35]. For MDD purposes, phoneme sequences must accurately reflect pronunciation and account for the influence of surrounding phonemes on articulation over time.

A grapheme-to-phoneme (G2P) converter is required to extract phoneme sequences from diacritized Arabic text. We adopt the rule-based approach described in [78], which was originally developed to generate phonetic dictionaries for Arabic speech recognition. This method applies MSA pronunciation rules using a well-defined phoneme set. Its publicly available implementation and compatibility with large-scale processing make it suitable for our data preparation pipeline.

2) Canonical Phonemes

To generate canonical phoneme sequences for each lesson, normalized lesson text was passed through the G2P system. This outputs a phoneme sequence that accurately reflects the standard pronunciation of the text. Table 4 presents examples of canonical lesson texts and their corresponding phoneme sequences.

3) Perceived Phonemes

A script was used to generate the perceived phoneme sequences by converting each lesson's text into a pronunciation-like form that retained only the pronounced Arabic glyphs. This is based on rules adapted from [79] and refined by Arabic language experts. The resulting text provided a template for annotators to modify such that it reflected the actual speaker pronunciation, including insertions, deletions, and substitutions. This reduced the annotators' manual workload. The annotated text is passed through the G2P system to generate perceived phoneme sequences. Table 5 provides complete examples of the pronunciation-like text, manually perceived text, and final phoneme output.

The dataset could be used for fine-tuning the MDD model once canonical phoneme sequences were created and perceived sequences were generated for each annotated file.

4) Phoneme Sequence Alignment

As described later in Section IX-A, aligned canonical and perceived phoneme sequences are required model inputs. However, these sequences often differ in length because of insertions, deletions, or substitutions in the learner's pronunciation. We resolve this problem using a global sequence alignment algorithm to produce one-to-one phoneme mapping.

Specifically, we apply the Needleman-Wunsch algorithm [80] following the approaches of prior studies [26], [43]. This algorithm aligns two-phoneme sequences by systematically capturing length discrepancies. An aligned output is essential for identifying phoneme-level pronunciation deviations, and these directly inform MDD system training and error detection.

Table 6 presents an example of the canonical and perceived phoneme sequences before and after alignment.

In this example, the G2P algorithm was applied to both texts. The resulting phoneme sequences were aligned with underscores (_) to indicate insertions or deletions. The final alignment ensures that each canonical phoneme is either matched or explicitly accounted for, thus enabling accurate pronunciation analysis.

VIII. EXPERIMENTAL DESIGN

This section outlines the experimental design used to validate our research hypothesis. It focuses on parameter analysis and experimental reproducibility.

A. PARAMETERS

To evaluate our hypothesis, we conducted multiple experiments while analyzing the factors that influence the model performance. These parameters fall into two main categories: data-related and model-related.

Data-Related Parameters

- **Recording Environment:** Variations in noise, microphone quality, and recording conditions may obscure phonetic cues, reducing recognition accuracy.
- **Speaker Variability:** Differences in age, gender, and dialect can influence pronunciation patterns and model generalizability.
- **Annotation Accuracy:** Inconsistent or incorrect phoneme annotations can degrade training quality and model performance.
- **Subset Selection:** Poor train/test splits (e.g., non-stratified sampling) may lead to biased evaluation.
- **Dataset Size:** Insufficient data may cause overfitting and poor generalization.
- **Speaker Representation:** Over-representation of specific demographic groups may result in biased model behavior toward underrepresented groups.

Model-Related Parameters

- **Model Variants:** Using pretrained models not optimized for Arabic may limit performance. Arabic-tuned models are better at capturing relevant phonetic patterns.
- **Learning Rate:** A poorly chosen learning rate can lead to nonconvergence or unstable training.
- **Batch Size and Epochs:** Small batch sizes cause noisy gradients, while larger ones require more resources. Improper epoch count leads to underfitting or overfitting.

Table 4: Examples of Lesson Texts and Canonical Phoneme Sequences

Lesson ID	Lesson Text (Arabic)	Canonical Phoneme Sequence
1113	جَلَسَتِ الْأُسْرَةُ حَوْلَ الْمَائِدَةِ ، فَسَأَلَ الْأَبُ : لِمَاذَا تَأَخَّرَ يَاسِرٌ ؟ قَالَتْ نُورَةُ: هُوَ يَغْسِلُ يَدَيْهِ. قَالَ الْأَبُ: غَسْلُ الْيَدَيْنِ ضَرُورِيٌّ قَبْلَ الْأَكْلِ وَبَعْدَهُ.	JH AE L AE S AE T IH L E UH S R AE T UH HH AE L AE L M AE: E IH D AE H _ F AE S AE E AE L AE L E B _ L IH M AE: DH AE: T AE E AE KH AE R AE Y AE: S IH R _ Q AE: L AE T N UW R AE H _ H UH W AE Y AE GH S IH L UH Y AE D AE H _ Q AE: L AE L E AE B _ GH AE S L UH L Y AE D AE N DD AE R UW R IH Y UH N Q AE B L AE L E AE K L IH W AE B AE AI D AE H
1114	قَالَ فَوَّازٌ: أُمِّي تَسْقِي وَرْدَ الْحَدِيقَةِ ، وَأَبِي يَقْرَأُ الْجَرِيدَةَ ، وَأَخِي يَرْكَبُ الدَّرَّاجَةَ ، وَأُخْتِي تَحْمِلُ دُمْيَتَهَا ،وَأَنَا أَلْعَبُ مَعَ صَدِيقِي خَالِدٍ.	Q AE: L AE F AE W AE: Z _ E UH M IY T AE S Q IY W AE R D AE L HH AE D IY Q AE H _ W AE E AE B IY Y AE Q R AE E UH L JH AE R IY D AE H _ W AE E AE KH IY Y AE R K AE B UH D AE R AE: JH AE H _ W AE E UH KH T IY T AE HH M IH L UH D UH M Y AE T AE H AE: _ W AE E AE N AE: E AE L AI AE B UH M AE AI AE SS AE D IY Q IY KH AE: L IH D

Table 5: Pronunciation-Like Text, Manually Adjusted Perceived Text, and Resulting Phoneme Sequences

Lesson ID	Pronunciation-Like Text	Perceived (Manually Adjusted)	Phoneme Sequence
1113	جَلَسَتِ لْأُسْرَةُ حَوْلَ لْمَائِدَه _ فَسَأَلَ لْأَب _ لِمَاذَا تَأَخَّرَ يَاسِر _ قَالَتْ نُورَه _ هُوَ يَغْسِلُ يَدَيْه _ قَالَ لْأَب _ غَسْلُ لْيَدَيْن ضَرُورِيٌّ قَبْلَ لْأَكْلِ وَبَعْدَه	جَلَسَتِ لْأُسْرَا حَوْلَ لْمَائِدَه _ فَسَأَلَ لْأَب _ لِمَاذَا تَأَخَّرَ يَاسِرْ _ قَالَتْ نُورَا _ هُوَ يَغْسِلُ يَدَيْه _ قَالَ لْأَب _ غَسْلَ لْيَدَيْن ضَرُورِي _ قَبْلَ لْأَكْل وَبَعْدَه	JH AE L AE S AE T IH L E UH S R AE: HH AE L AE L M AE: E IH D AE H _ F AE S AE E AE L AE L E AE B _ L IH M AE: DH AE: T AE E AE KH AE R AE Y AE: S IH R _ Q AE: L AE T N UW R AE: _ H UH W AE Y AE GH S IH L UH Y AE D AE H _ Q AE: L AE L E AE B _ GH AE S L AE L Y AE D AE N DD AE R UW R IY _ Q AE B L AE L E AE K L W AE B AE AI D AE H
1114	قَالَ فَوَّازٌ _ أُمِّي تَسْقِي وَرْدَ لْحَدِيقَه _ وَأَبِي يَقْرَأُ لْجَرِيدَه _ وَأَخِي يَرْكَبُ دَّرَّاجَه _ وَأُخْتِي تَحْمِلُ دُمْيَتَهَا _ وَأَنَا أَلْعَبُ مَعَ صَدِيقِي خَالِد	قَالَ فَوَّازُ _ أُمِّي _ تَسْقِي وَرْدَ لْحَدِيقَه _ وَأَبِي يَقْرَأُ لْجَرِيدَه _ وَأَخِي _ يَرْكَبُ دَّرَّاجَةَ _ وَأُخْتِي _ تَحْمِلُ دُمْيَتُهَا _ وَأَنَا _ أَلْعَبُ مَعَ صَدِيقِي خَالِدٍ	Q AE: L AE F AE W AE: Z UH _ E UH M IY _ T AE S Q IY W AE R D AE L HH AE D IY Q AE H _ W AE E AE B IY Y AE Q R AE E UH L JH AE R IY D AE H _ W AE E AE KH IY _ Y AE R K AE B UH D AE R AE: JH AE T AE _ W AE E UH KH T IY _ T AE HH M IH L D UH N Y AE T UH H AE: _ W AE E AE N AE: _ E AE L AI AE B UH M AE AI AE SS AE D IY Q IY KH AE: L IH D IH N

Table 6: Example of Phoneme Sequence Alignment using the Needleman-Wunsch Algorithm

Canonical	**Text**	مَدْرَسَةُ طُلَّابِ (The students' school)
	Phonemes	M AE D R AE S AE T UH TT UH L AE: B IH
	Aligned	M AE D _ R AE S AE T _ UH TT UH L AE: B IH
Perceived	**Text**	مَدَارَسَة أَلْطُلَّابْ
	Phonemes	M AE D AE: R AE S AE T E L TT UH L AE: B
	Aligned	M AE D AE: R AE S AE T E L TT UH L AE: B _

Table 7: Dynamic Parameters and their Values

Factor	Value 1	Value 2	Value 3
Train Size	Large	Medium	Small
Model Variant	XLSR-53 El-Geish	XLSR-53 Grosman	– –

- **Loss Function and Optimizer:** Incorrect settings can degrade convergence speed or compromise model performance.

Other influential factors include the dropout rate, model depth, hidden layer sizes, and random initialization. Discrepancies in the data pipeline, especially between training and inference, can also affect performance.

We classified the parameters into two types: **static parameters,** whose values are fixed based on the literature or system constraints; and **dynamic parameters,** whose values are tuned during experimentation to optimize results.

Table 7 summarizes the dynamic parameters and their selected values.

Table 8: Dataset Sizes in the Large, Medium, and Small Configurations

Configuration	Train (minutes)	Test (minutes)
Large	42.07 (70%)	17.89 (30%)
Medium	22.15 (50% of Large)	17.89 (same)
Small	11.15 (25% of Large)	17.89 (same)

B. EXPERIMENTAL DESIGN

To systematically evaluate the effects of key variables on MDD system performance, two dynamic parameters were selected: Corpus Size and Model Variant. This selection was informed by their relevance to the research goals, study time constraints, and data availability.

Corpus Size: The optimal quantity of labeled pronunciation data for fine tuning is unknown. To investigate this effect, we tested three training datasets: Small, Medium, and Large. The Large dataset comprised the entire 60 minutes of speech data that was manually annotated (see Section VII-A). This was divided into 70% training (42.07 minutes) and 30% test (17.89 minutes) data. Multiple splits were evaluated, and the best-performing configuration was selected. Medium and Small sets were derived from 50% (22.15 minutes) and 25% (11.15 minutes) of the Large training set, respectively. We used the same test set to ensure consistency. To document the split balance and coverage, we plotted the Arabic phoneme frequency on a log scale for the training and test subsets, as shown in Fig. 4. The resulting training and testing durations for each configuration are listed in Table 8.

Model Variant: Two pretrained models were evaluated: El-Geish and Grosman (see Section VI-C). These variants differ in their architectures and pretraining cor-

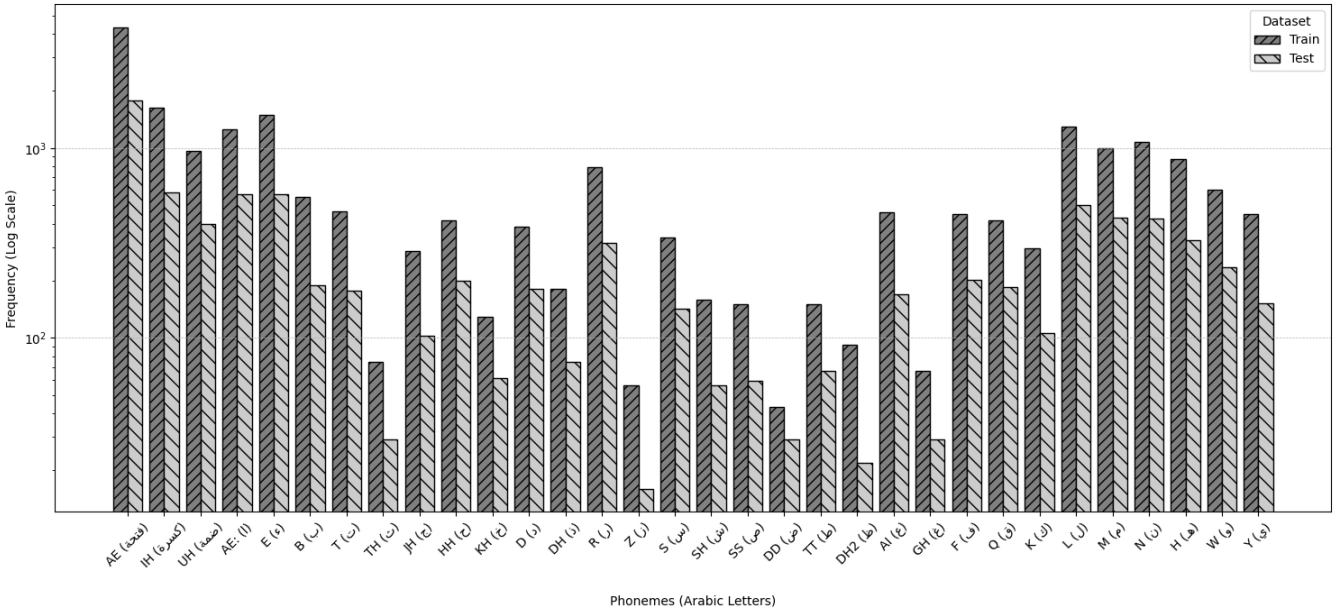

Figure 4: Log-scale frequency distribution of Arabic phonemes in the training and testing subsets after dataset splitting.

Table 9: Experimental Design Showing All Combinations of Dynamic Parameters with Two Replications

Run	Train Size	Model Variant	Replication
1	Small	El-Geish	1
2	Small	El-Geish	2
3	Small	Grosman	1
4	Small	Grosman	2
5	Medium	El-Geish	1
6	Medium	El-Geish	2
7	Medium	Grosman	1
8	Medium	Grosman	2
9	Large	El-Geish	1
10	Large	El-Geish	2
11	Large	Grosman	1
12	Large	Grosman	2

pora. By comparing their fine-tuned performances, we assessed the effect of model selection on MDD accuracy in Arabic.

A full-factorial experimental design was used to explore the effects of both parameters. This included:

- All combinations of the three corpus sizes and two model variants.
- Evaluation of main and interaction effects.
- Two replications per configuration to increase reliability and support statistical analysis.

This resulted in $3 \times 2 = 6$ treatment combinations, and 12 experiments. The full set of experimental runs generated from this factorial design is listed in Table 9. The experiment sequence was randomized using a random number generator to reduce the bias from consistent execution conditions across models or datasets. The experimental logs also document any inconsistencies, model failures, or irregularities to ensure comprehensive evaluation and reproducibility.

IX. IMPLEMENTATION AND RESULTS

This section outlines the implementation and presents the experimental results in our evaluation of the MDD system. The experiments were aimed at assessing the system's performance under varying configurations using the primary metrics defined in Section VI-G.

A. MODEL REQUIREMENTS

Both speech audio and phoneme sequence data are required for training and evaluation. The speech input must be a raw waveform with a 16 kHz sampling rate and 16-bit depth. The textual inputs consist of both canonical and perceived phoneme sequences.

To support the training and fine-tuning, the phoneme sequences were aligned using the process described in Section VII-C4. The model input is structured in JavaScript Object Notation (JSON) format. Each entry corresponds to a pronunciation instance and includes four fields:

- **wav:** Path to the audio file (16 kHz, mono, 16-bit wav).
- **canonical_aligned:** Aligned canonical phoneme sequence.
- **perceived_aligned:** Aligned perceived phoneme sequence.
- **perceived_train_target:** Unaligned perceived sequence used during training.

An illustrative example is shown in Table 10, in which the speaker pronounced مَاجِرَاً instead of the correct مَرْحَباً.

Table 10: Example of a JSON Entry for Model Input

Key	Value
wav	/path/to/audio.wav
canonical_aligned	[M, AE, R, HH, AE, B, AE, _, _, N]
perceived_aligned	[M, AE:, _, HH, IH, B, AE, R, AE, N]
perceived_train_target	M AE: HH IH B AE R AE N

B. IMPLEMENTATION

The implementation was based on the SpeechBrain toolkit [81], which has been widely adopted in recent studies [19], [20], [43]. SpeechBrain is a versatile PyTorch-based toolkit that supports various speech-processing tasks and facilitates feature extraction, model building, and model training.

The experiments used raw audio waveforms sampled at 16 kHz, compatible with the requirements of Wav2vec 2.0. Fine-tuned Wav2vec 2.0 models from Hugging Face were employed (see Section VI-C). Our experimental design follows the configuration proposed by Yang et al. [19], with some adjustments.

Two separate Adam optimizers were used: one for the linear layers (learning rate = .0003) and the other for the Wav2vec 2.0 backbone (learning rate = .00001). Gradient accumulation was employed to reduce the memory overhead by delaying parameter updates until multiple minibatches were processed.

All experiments were executed for 100 epochs using Google Colab's cloud runtime on an NVIDIA A100-SXM4-40GB Graphics Processing Unit (GPU) running Ubuntu 22.04.3 LTS. The system environment included PyTorch 2.5.1, CUDA 12.1, and Python 3.10.12.

C. EXPERIMENTAL RESULTS

For clarity, the experimental results are presented in both tabular and graphical formats. Statistical analyses, including ANOVA, were used to evaluate the significance of the main and interaction effects.

1) Descriptive Results

Table 11 summarizes the average performance across all experimental configurations, including the F1 score, PER, and classification metrics.

2) Visualization of Key Metrics

The key performance metrics were visualized using bar plots to support the interpretation of the experimental findings.

Fig. 5 presents the F1 scores across different corpus sizes and model variants. Corpus size is represented on the x-axis (Small, Medium, Large), and the El-Geish and Grosman models are represented by paired bars. The chart shows that the F1 score improves with increasing data size and that the Grosman model consistently outperforms El-Geish.

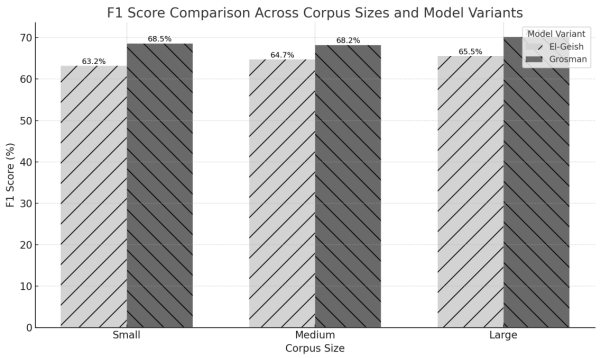

Figure 5: F1 Score comparison across corpus sizes and model variants.

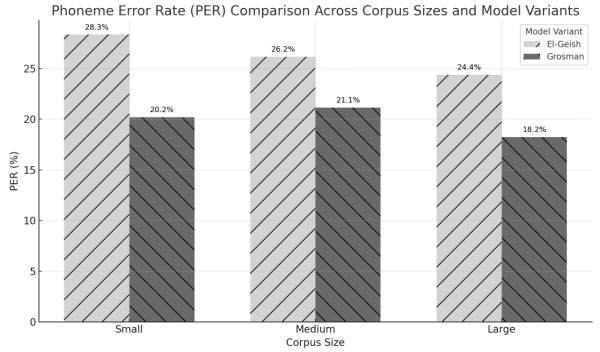

Figure 6: Phoneme Error Rate (PER) comparison across corpus sizes and model variants.

Fig. 6 shows the PER for the same configuration. Similar to the previous figure, this grouped bar chart shows that the PER decreases as the corpus size increases. The Grosman model achieved a consistently lower PER than the El-Geish model.

In Fig. 7, the stacked bar chart displays the distribution of classification outcomes (TA, FR, FA, and TR) for each experimental run. This visualization highlights how the classification behaviors vary across configurations, offering deeper insights into model-decision trends.

3) Statistical Analysis

A two-way ANOVA with two replications was conducted to assess the effects of **Corpus Size**, **Model Variant**, and their interaction on the F1 Score. Each configuration was tested twice, resulting in a total of 12 experiments. The detailed ANOVA results, including the sum of squares, degrees of freedom, effect sizes, and significance levels, are presented in Table 12.

The **Model Variant** factor had a statistically significant effect on the F1 score ($F(1,6) = 45.30$, $p < .001$, partial $\eta^2 = .772$), indicating strong practical significance. In contrast, the effect of **Corpus Size** was not statistically significant ($F(2,6) = 3.04$, $p = .122$, partial $\eta^2 = .104$), nor was the interaction between **Corpus Size**

Table 11: Summary of Experimental Results across Configurations with Corresponding Classification Metrics

Run	Train Size	Model	F1 Score	PER	TA	FR	FA	TR (CD/DE)
1	Small	El-Geish	63.03%	28.68%	5820	2240	815	2605 (2332/273)
2	Small	El-Geish	63.28%	28.01%	5908	2201	818	2602 (2325/277)
3	Small	Grosman	68.62%	20.14%	6537	1634	780	2640 (2417/223)
4	Small	Grosman	68.46%	20.24%	6537	1648	782	2638 (2412/226)
5	Medium	El-Geish	63.30%	28.29%	5934	2008	906	2514 (2247/267)
6	Medium	El-Geish	66.12%	24.02%	6248	1708	887	2533 (2273/260)
7	Medium	Grosman	68.29%	21.01%	6496	1442	899	2521 (2295/226)
8	Medium	Grosman	68.15%	21.27%	6489	1457	899	2521 (2290/231)
9	Large	El-Geish	66.13%	24.02%	6248	1708	887	2533 (2273/260)
10	Large	El-Geish	64.89%	24.73%	6203	1814	906	2514 (2252/262)
11	Large	Grosman	68.86%	19.99%	6559	1375	902	2518 (2280/238)
12	Large	Grosman	**71.42%**	**16.47%**	6665	1358	766	2654 (2438/216)

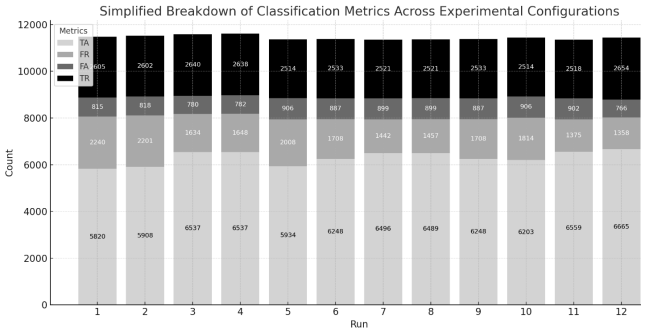

Figure 7: Comparison of classification metrics [True Accept (TA), False Reject (FR), False Accept (FA), and True Reject (TR)] across experimental configurations.

Table 12: Two-Way ANOVA with Replication of Corpus Size and Model Variant on F1 Score, including Effect Sizes (partial η^2)

Source	SS	df	MS	F	p-value	Partial η^2
Corpus Size (A)	8.1886	2	4.0943	3.0420	.1224	.104
Model (B)	60.9752	1	60.9752	45.3030	**< .001**	.772
A × B	1.7800	2	.89	0.6613	.5501	.023
Error	8.0757	6	1.3459	-	-	-
Total	79.0195	11	-	-	-	-

and **Model Variant** ($F(2,6) = .66$, $p = .550$, partial $\eta^2 = .023$). Levene's test indicated that the assumption of homogeneity of variance was satisfied ($p = .49$). Tukey's HSD post-hoc analyses further identified significant differences between specific groups, notably between the El-Geish and Grosman models, within each **Corpus Size** category. This highlights the significant performance advantage of the Grosman model, particularly when **Corpus Size** is Large.

Table 13 connects these findings to the actual performance metrics across configurations.

Small- and Medium Grosman configurations performed similarly. This can be attributed to limited data diversity. Both used overlapping high-frequency phoneme patterns, which likely resulted in similar F1 Scores (68.62% vs. 68.29%) and PER values (20.14% vs. 21.01%).

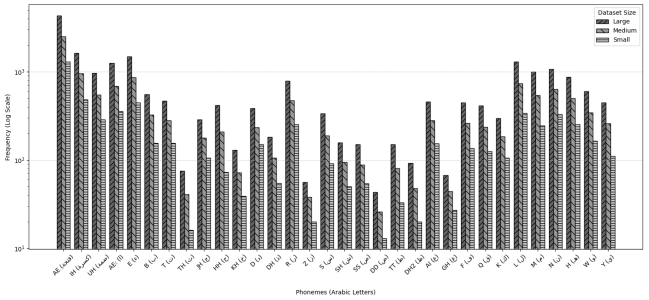

Figure 8: Phoneme frequency distribution across training sets (Small, Medium, and Large).

Fig. 8 shows that these subsets lack rare phonemes, which reduces their generalizability. In contrast, the Large configuration improved generalizability by covering more diverse patterns. It achieved the highest F1 Score (71.42%) and the lowest PER (16.47%).

This comprehensive analysis confirms that model selection significantly affects the F1 Score; however, increasing the corpus size offers diminishing returns unless accompanied by greater phonetic diversity.

4) Phoneme Recognition Performance and Error Analysis

Although an overall F1 score of 71.4% is promising, it does not reveal the system's weaknesses. To clarify these limitations, we have provided a confusion matrix (Fig. 9) and recognition distribution chart (Fig. 10) for the w2v2.0-Grosman-Large configuration, along with qualitative examples (Tables 14–19) highlighting typical substitution, deletion, and insertion errors. The matrix illustrates the distribution of these errors between the predicted and reference phonemes. Diagonal entries indicate correct predictions, whereas off-diagonal entries indicate substitution errors. The last two columns labeled **<INS>** and **<DEL>** denote the number of insertion and deletion errors, respectively.

The confusion matrix provided several critical insights.

Table 13: Performance Metrics across Model Variants and Corpus Sizes

Configuration	Data	TA	FR	FA	CD	ED	PER ↓	F1 ↑
w2v2.0-ElGeish-Small	25%	72.86%	27.14%	23.93%	68.01%	8.10%	28.01%	63.29%
w2v2.0-ElGeish-Medium	50%	74.69%	25.31%	26.49%	65.70%	7.81%	28.29%	63.30%
w2v2.0-ElGeish-Large	100%	78.54%	21.46%	25.94%	66.46%	7.60%	**24.02%**	**66.13%**
w2v2.0-Grosman-Small	25%	80.01%	19.99%	22.81%	70.61%	6.52%	20.14%	68.62%
w2v2.0-Grosman-Medium	50%	81.82%	18.18%	26.26%	67.19%	6.61%	21.01%	68.29%
w2v2.0-Grosman-Large	100%	83.07%	16.93%	22.40%	71.29%	6.31%	**16.47%**	**71.42%**

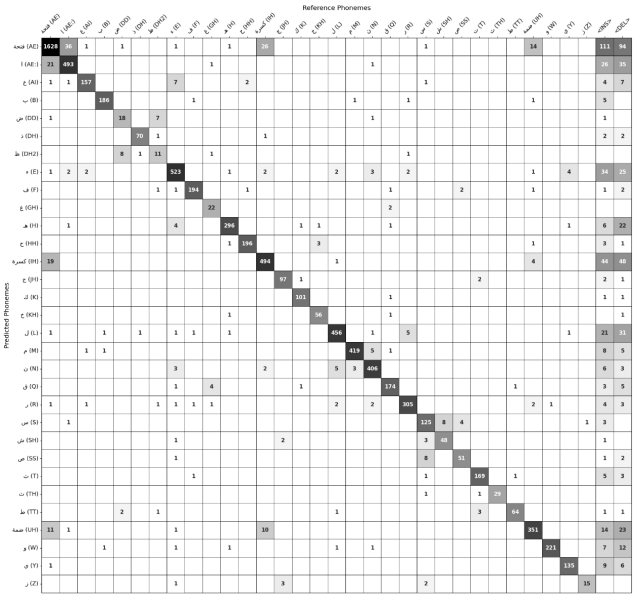

Figure 9: Phoneme confusion matrix showing substitutions, insertions, and deletions in predicted phonemes compared to the reference.

- Phonemes such as ر (R), م (M), ن (N), ف (F), and ل (L) exhibited high recognition accuracy.
- Frequent substitution errors were particularly noticeable among vowels, including ضمة (UH), كسرة (IH), فتحة (AE), and ألف المد (AE:).
- Consonants with phonetic similarities also showed notable confusion, such as ظ (DH2) being confused with ض (DD), and س (S) frequently misrecognized as ص (SS).
- Deletion errors occurred frequently for specific phonemes, notably ه (H) and ل (L).
- High insertion frequencies were observed for the phoneme ء (E).

Fig. 10 presents a stacked bar chart illustrating the proportion of correct vs. incorrect recognition across individual phonemes. This visualization highlights phonemes that are prone to errors, thereby identifying potential targets for model improvement.

Together, these visualizations offer a comprehensive view of the model's phoneme-level performance. The system shows strong recognition of common consonants but struggles with phonetically similar or linguistically complex Arabic sounds. Addressing these challenges can further enhance the system's accuracy and provide more precise learner feedback.

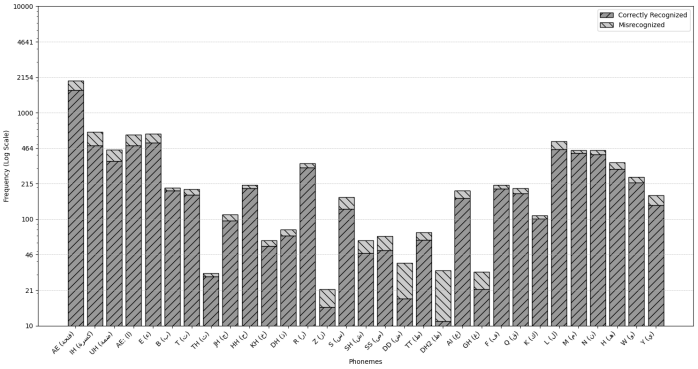

Figure 10: Distribution of phonemes showing proportions of correct vs. incorrect recognition.

Table 14: Substitution: DH2 → DD (ظ → ض)

Correct Arabic	حَاوَلَ أَنْ يُرْضِيَ الجَمِيعَ
Incorrect Arabic	حَاوَلَ أَنْ يُرْضِيَ الجَمِيعَ
Reference phonemes	... UH R [DD] IY ...
Predicted phonemes	... UH R [DH2] IY ...
Transliteration	hawala an yurdhi al-jamee3
Meaning	He tried to please everyone.
File	70-5-1-Z_F_Riyadh_8e86a567-9a53-4d91-9452-b181de052cfb.MOV.wav

a: Qualitative Error Analysis

To complement the confusion matrix and distribution plots, we present examples of the most frequent error types. Each table highlights a representative case, contrasting the correct and incorrect Arabic text. These cases are paired with the corresponding phonemes (reference and predicted), transliterations, semantic meanings, and the files where the errors were observed.

Table 14 shows a substitution where the emphatic consonant ظ (DH2) is misrecognized as ض (DD). Table 15 presents another substitution between the plain consonant س (S) and its emphatic counterpart ص (SS). Tables 16 and 17 illustrate frequent vowel confusions, where short vowels are particularly error-prone. Table 18 provides an example of the deletion of the phoneme ه (H), while Table 19 shows an insertion error involving the glottal stop ء (E).

These analyses revealed the model's most common

Table 15: Substitution: S → SS (س → ص)

Correct Arabic	سَافَرَ عُمَرُ
Incorrect Arabic	صَافَرَ عُمَرُ
Reference phonemes	... [S] AE: F AE R ...
Predicted phonemes	... [SS] AE: F AE R ...
Transliteration	saafara 3umar
Meaning	3umar traveled.
File	69-X-Y-Z_M_Madinah-2_e8968f59-a46c-402d_wf4RonZ.ogg.wav

Table 16: Vowel substitution: IH → AE (كسرة → فتحة)

Correct Arabic	فَزِعَ الحِمَارُ مِنْ صِيَاحِ النَّاسِ
Incorrect Arabic	فَزَعَ الحِمَارُ مِنْ صِيَاحِ النَّاسِ
Reference phonemes	... F AE Z [IH] AE ...
Predicted phonemes	... F AE Z [AE] AE ...
Transliteration	fazi3a al-himaar min siyaah al-naas
Meaning	The donkey panicked from the people's shouting.
File	70-5-1-Z_F_Riyadh_8e86a567-9a53-4d91-9452-b181de052cfb.MOV.wav

Table 17: Vowel substitution: AE → IH (فتحة → كسرة)

Correct Arabic	اعْتَقَدَ أَنَّهُ مَيِّتٌ
Incorrect Arabic	اعْتَقَدَ أَنَّهُ مِيِّتٌ
Reference phonemes	... M [AE] Y IH TT ...
Predicted phonemes	... M [IH] Y IH TT ...
Transliteration	i3taqada annahu mayyit
Meaning	He believed that he is dead.
File	69-X-Y-Z_M_Madinah-2_e8968f59-a46c-402d_wf4RonZ.ogg.wav

Table 18: Deletion: H dropped (هـ)

Correct Arabic	لِمَاذَا أَرْجُلِي طَوِيلَة
Incorrect Arabic	لِمَاذَا أَرْجُلِي طَوِيلَ
Reference phonemes	... L AE [H] sil ...
Predicted phonemes	... L AE [∅] sil ...
Transliteration	limatha arjuli taweelah
Meaning	Why are my legs long?
File	06-5-Y-Z_F_Qatif_7976e7f2-eae1-4f46-95fa-cf1ac600a178.mp4.wav

failures and limitations regarding phoneme recognition accuracy. The examples confirm that emphatic/plain consonant contrasts, short vowel variability, and weak phonemes such as هـ (H) or ء (E) present the most challenging cases. Seemingly small deviations at the phoneme level can alter a word's lexical or semantic interpretations. These qualitative insights complement the overall error analysis and provide practical guidance for future improvements to Arabic MDD systems.

D. COMPARISON TO PREVIOUS STUDIES

This section compares our system's performance with that of previous MDD studies. Direct comparisons are challenging due to the absence of standardized datasets, evaluation protocols, and language targets. Many studies focused on different languages (e.g., English and Mandarin), used isolated words or phonemes, or addressed pronunciation scoring instead of mispronunciation detection.

Even when studies use similar metrics, such as the F1 score or PER, discrepancies in dataset composition, recording conditions, and annotation protocols limit comparability. Nevertheless, we summarize relevant results from other studies that consider continuous-speech MDD and use PER and/or F1 scores as evaluation metrics.

While Table 20 helps to contextualize our system's performance, it should not be interpreted as a claim of our system's direct superiority over others. Most prior studies (e.g., Xu et al. [52], Peng et al. [18], and Yang et al. [19]) focused on English speech of adult L2 learners using standardized corpora such as L2-ARCTIC or TIMIT. In contrast, our system was trained and evaluated using continuous Arabic speech recorded by young learners in uncontrolled environments. These differences, especially in language, age groups, and recording settings, made MDD tasks significantly more difficult in our study. Additionally, each participant recorded audio from their own device in their own environment. This introduced further variability and practical challenges. Our dataset, therefore, contains real-world complexities that are essential for enhancing the accuracy and robustness of MDD systems. Hence, the F1 score of 71.42% reflects robustness under realistic and challenging conditions.

In the Arabic context, most previous studies have targeted isolated words or letters. A notable exception is Algabri et al. [20], who evaluated MDD in continuous Arabic speech using the KSU corpus. This allows for a meaningful, although still limited, comparison with our system. Peng et al. [18], Yang et al. [19], Xu et al. [52] and Shen et al. [43] provided additional benchs for other languages.

Table 20 summarizes the performance metrics from our study and selected others using continuous MDD tasks and comparable evaluation criteria. The w2v2.0-Grosman model achieved an F1 score of 71.42% with only 60 minutes of annotated data, achieving better results than those reported by Yang et al. (56.16%), Xu et al. (61.0%), and Peng et al. (60.44%). It also provided slightly better results than Algabri et al.'s 70.53% F1 score, despite their model benefiting from higher-quality controlled recordings and more than 9 hours of training data.

Although Algabri et al.'s system achieved a lower PER of 3.83%, our system demonstrates better adaptability to real-world educational environments. The KSU dataset benefitted from multiple microphones and

Table 19: Insertion: E (ء)

Correct Arabic	قَرَّرَا الصَّدِيقَانِ
Incorrect Arabic	قَرَّرَاء الصَّدِيقَانِ
Reference phonemes	... AE: [∅] SS ...
Predicted phonemes	... AE: [E] SS ...
Transliteration	qarraraa al-sadeeqaan
Meaning	The two friends decided.
File	69-X-Y-Z__M__Madinah-2__e8968f59-a46c-402d__wf4RonZ.ogg.wav

controlled conditions, whereas our model used natural, noisy, and uncontrolled recordings from primary-school students (see Section VII-A). This makes it more challenging to achieve performance gains. Notably, the Grosman model achieved a TA rate of 83.07% and a FR rate of 16.93%. Among the correctly detected mispronunciations, the system achieved a CD of 71.29%, while the DE rate was 6.31%. These values indicate high recognition accuracy and diagnostic precision. Compared with Algabri et al.'s model, which uses synthetic and clean speech, our system achieved competitive F1 scores.

X. CONCLUSION AND FUTURE WORK

This study contributes to MDD, CAPT, and Arabic speech processing as follows.

- **Arabic Pronunciation Speech Corpus:** Creation and organization of a structured dataset suitable for CAPT, ASR, and phoneme recognition in Arabic.
- **Arabic Text Normalization:** Development of text normalization algorithms to transform Arabic text into a pronunciation-aligned format, while accounting for the language's orthographic complexity.
- **End-to-End Phoneme-Level MDD System:** Proposal of a Wav2vec–2.0–based system for phoneme-level mispronunciation detection in continuous Arabic speech.

Furthermore, our proposed system demonstrates a state-of-the-art F1 score of **71.42%** and a PER of **16.47%** performance under low-resource, dialect-influenced, and noisy spontaneous speech. This performance indicates its suitability for deployment in actual school environments.

Importantly, the model variant significantly affected its performance. However, variations in the corpus size within the tested range did not produce statistically significant changes in the F1 score. This suggests that small annotated datasets with durations as short as 20 minutes may be sufficient to effectively fine-tune SSL models.

Overall, the system addresses the challenges posed by Arabic's complex phonetic structure and low-resource status, enabling automated pronunciation evaluation that can be extended to other under-resourced languages. Although the system demonstrated promising results, several limitations remain.

- **Dataset Diversity :** The dataset primarily includes speech from Saudi primary school students. All speakers were 8–11 years old; therefore, our findings may not transfer to adult pronunciation. Additionally, our findings may not generalize to broader demographics with dialects not encompassed by Saudi dialect clusters.
- **Focus on MSA:** The system does not support DA or CA, limiting its applicability to Quranic or informal speech contexts.
- **Tool Limitations:** The SpeechBrain toolkit [81] does not support non-ASCII characters. This limited our exploration of phoneme sets containing special symbols.
- **Phoneme-Level Focus:** Phoneme-level evaluation may be less intuitive for language learners than word- or sentence-level assessments.
- **Computational Demands:** Wav2vec 2.0 is computationally expensive and may not be suitable for deployment in mobile or edge computing environments.
- **No Paralinguistic Analysis:** The system does not account for prosodic or paralinguistic features such as pitch, stress, or intonation. These elements are important components of natural speech evaluation.
- **Lack of Practical Feedback:** The system output currently requires expert interpretation. There is no built-in feedback mechanism for non-expert users or learners.

These limitations highlight several opportunities to advance future research. First, larger-scale and more diverse datasets should be collected across ages, dialects, and contexts through collaboration with institutions in other Arabic-speaking countries. In particular, future research should extend to Quranic pronunciation, supported by collaboration with Quran teaching associations. Second, to provide more comprehensive learning feedback, phoneme-level evaluation should be extended to other linguistic units such as syllables [82], words, and sentences. Third, it is crucial to develop a practical module to transform outputs into interpretable, learner-friendly feedback. This would enable real-world deployment without expert oversight. Fourth, future work should build lightweight Arabic MDD models that can be efficiently deployed on mobile and edge devices to ensure broader accessibility and real-time usability.

APPENDIX A
DATA ANNOTATION GUIDELINES

1) The transcribed text must match the student's spoken words exactly, even if their pronunciation differs from the written text. This includes:

Table 20: Comparison with Related MDD Studies using Continuous Speech and Reporting F1 Score and/or PER

Model	Dataset	TA	FR	FA	CD	ED	PER ↓	F1 ↑
Ours	In-House, Arabic, Uncontrolled	83.07%	16.93%	22.40%	71.29%	6.31%	16.47%	**71.42%**
Algabri et al. [20]	KSU, Arabic, Controlled	98.12%	1.88%	25.08%	–	14.81%	**3.83%**	70.53%
Yang et al. [19]	L2-A + UTD, English	93.54%	6.46%	45.84%	77.24%	22.76%	14.36%	56.16%
Xu et al. [52]	L2-A, English	–	8.00%	35.70%	–	–	–	61.0%
Peng et al. [18]	TIMIT + L2-A, English	94.30%	5.70%	41.80%	70.72%	29.28%	–	60.44%

- Misreading a word (e.g., كَتَبَ (kataba) instead of كَبَتَ (kabata)).
- Repeating a word (e.g., وَقَدَ قَامَ الأُسْتَاذُ (wa-qad qāma qāma al-ustādhu)).
- Repeating part of a word (e.g., وَلَيْسَ كُلُّ (walay walaysa kullu)).
- Replacing the correct diacritic of a letter with another incorrect diacritic (e.g., مَلَكَ (malaka) instead of مَلِكَ (malika)).

2) The first letter following a sun definite article must have a gemination , if it is correctly pronounced, e.g., الشَّمسُ.
3) Keep the definite article ال without diacritization when it is pronounced properly within the sentence. Make an exception to this guideline when ال appears in a word at the beginning of a sentence. In that case, همزة الوصل (the connecting hamzah) should be written as همزة القطع (cut hamzah) with diacritization, e.g., أَلْوَرْدُ and أَلشَّمسُ.
4) All forms of the pronounced hamzah, viz., أ ؤ ئ ء must be written according to the standard orthographic rules.
5) The cut hamzah is omitted if it was not pronounced due to a reading mistake or the speaker's accent.
6) When a geminated letter is correctly pronounced, the gemination diacritic (ّ) must be written.
7) If pronounced, the nunation diacritic التنوين must be written. If not, it should be removed and the word must be diacritized as it was spoken (e.g., at the end of a sentence).
8) All diacritization marks must be present in the spoken text, except for the سكون (silence) diacritic. This can be omitted.
9) A silence mark, denoted by the '@' character, must be added between two sentences, two words or within a word if a noticeable period of silence exists in that location. If the silence was "ugly", two silence marks are added (e.g., if it changes the meaning such as "Don't pray @@ while intoxicated ...").
10) Add a '+' mark after a soft (مرقق) letter that has been emphasized (مفخم). A '-' mark is added after a letter in the reverse situation.
11) Add the type of noise or unintelligible speech to the transcription, such as laughter or breathing, between two #'s when it occurs (# Laughter # or # unintelligible speech #). Refer to the supplied table containing a list of all such cases.
12) Spoken numbers or digits must be written as text, according to the way they were pronounced.

ACKNOWLEDGMENT

The authors would like to acknowledge the funding support provided by the Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS) at King Fahd University of Petroleum & Minerals. This work was supported and financially funded by the IRC-ISS Project No. INSS2211.

References

[1] Jaroslaw Krajka, "ENGLISH+KIDS," *CALICO J.*, vol. 20, no. 2, pp. 393--404, Jan. 2013, doi: 10.1558/cj.35173.
[2] Ali Fauzi Ahmad Khan, Oudelha Mourad, Amirul Mohamad Khairi Bin Mannan, Hassan Basri Awang Mat Dahan, and Mohammad A. M. Abushariah, "Automatic Arabic pronunciation scoring for computer-aided language learning," in *Proc. ICCSPA*, 2013, pp. 1--6, doi: 10.1109/ICCSPA.2013.6487246.
[3] Halima Bahi and Khaled Necibi, "Fuzzy Logic Applied for Pronunciation Assessment," *Int. J. Comput.-Assist. Lang. Learn. Teach.*, vol. 10, pp. 60--72, Jan. 2020, doi: 10.4018/IJCALLT.2020010105.
[4] Amna Asif, Hamid Mukhtar, Fatimah Alqadheeb, Hafiz Farooq Ahmad, and Abdulaziz Alhumam, "An Approach for Pronunciation Classification of Classical Arabic Phonemes Using Deep Learning," *Appl. Sci.*, vol. 12, no. 1, Art. no. 238, 2022, doi: 10.3390/app12010238.
[5] Ghassan Hasan and Ghassan Al Shatter, "Validity of Oral Tasks Testing in Arabic L2 Teaching," *Linguistica Communicatio*, vol. 18, pp. 27--43, Nov. 2017.
[6] James Lane, "The 10 most spoken languages in the world," *Babbel Magazine*, vol. 6, no. 09, 2019.
[7] George Julian, "What are the most spoken languages in the world," *Retrieved May*, vol. 31, no. 2020, pp. 38, 2020.
[8] Marwa A. Khairy, Tarek M. Mahmoud, Ahmed Omar, and Tarek Abd El-Hafeez, "Comparative performance of ensemble machine learning for Arabic cyberbullying and offensive language detection," *Lang. Resour. Eval.*, pp. 1--18, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260893244
[9] Tareq Moqbel, "Quranic Arabic: From its Hijazi Origins to its Classical Reading Traditions By Marijn van Putten," *J. Islamic Stud.*, vol. 34, May 2023, doi: 10.1093/jis/etad017.
[10] Rahma AlMahrooqi and C. J. Denman, "The Use of Modern Standard Arabic and Arabic Dialects in Oman for Internal Cohesion and External Distinction," in *Modern Arabic Sociolinguistics*, Nova Science Publishers, pp. 281--300, Sep. 2019.
[11] Mohammed ElAmine Chennafi, Hanane Bedlaoui, Abdelghani Dahou, and Mohammed A. A. Al-qaness, "Arabic Aspect-Based Sentiment Classification Using Seq2Seq Dialect Normalization and Transformers," *Knowledge*, vol. 2, no. 3, pp. 388--401, 2022, doi: 10.3390/knowledge2030022.
[12] Yassir Matrane, Faouzia Benabbou, and Nawal Sael, "A systematic literature review of Arabic dialect sentiment analysis," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 35, no. 6, Art. no. 101570, 2023, doi: 10.1016/j.jksuci.2023.101570.
[13] H. A. Al-Muhtaseb, S. A. Mahmoud, R. S. Qahwaji, M. Demiralp, N. Baykara, and N. Mastorakis, "A novel minimal Arabic script for preparing databases and benchmarks for Arabic text recognition research," in *Proc.*

International conferences. Mathematics and Computers in Science and Engineering (WSEAS), June 8 2009.

[14] S.-A. Selouani and Caelen, "Arabic phonetic features recognition using modular connectionist architectures," in *Proc. IEEE IVTTA*, 1998, pp. 155--160, doi: 10.1109/IVTTA.1998.727712.

[15] Jack Halpern, "Word Stress And Vowel Neutralization In Modern Standard Arabic," in *Proc.*, 2009. [Online]. Available from https://api.semanticscholar.org/CorpusID:227123672

[16] Siddique Latif, Aun Zaidi, Heriberto Cuayáhuitl, Fahad Shamshad, Moazzam Shoukat, and Junaid Qadir, "Transformers in Speech Processing: A Survey," *ArXiv*, vol. abs/2303.11607, 2023. [Online]. Available from: https://api.semanticscholar.org/CorpusID:257636830

[17] Ambra Neri, Catia Cucchiarini, Helmer Strik, and Lou Boves, "The Pedagogy-Technology Interface in Computer-Assisted Pronunciation Training," *Computer Assisted Language Learning*, vol. 15, no. 5, pp. 441--467, 2002, doi: 10.1076/call.15.5.441.13473.

[18] Linkai Peng, Yingming Gao, Rian Bao, Ya Li, and Jinsong Zhang, "End-to-End Mispronunciation Detection and Diagnosis Using Transfer Learning," *Appl. Sci.*, vol. 13, no. 11, Art. no. 6793, 2023, doi: 10.3390/app13116793.

[19] Mu Yang, Kevin Hirschi, Stephen Daniel Looney, Okim Kang, and John H.L. Hansen, "Improving Mispronunciation Detection with Wav2vec2-based Momentum Pseudo-Labeling for Accentedness and Intelligibility Assessment," in *Proc. Interspeech*, 2022, pp. 4481--4485, doi: https://doi.org/10.21437/Interspeech.2022-11039.

[20] Mohammed Algabri, Hassan Mathkour, Mansour Alsulaiman, and Mohamed Bencherif, "Mispronunciation Detection and Diagnosis with Articulatory-Level Feedback Generation for Non-Native Arabic Speech," *Mathematics*, vol. 10, Art. no. 2727, Aug. 2022, doi: 10.3390/math10152727.

[21] Faria Nazir, Muhammad Nadeem Majeed, Mustansar Ali Ghazanfar, and Muazzam Maqsood, "Mispronunciation Detection Using Deep Convolutional Neural Network Features and Transfer Learning-Based Model for Arabic Phonemes," *IEEE Access*, vol. 7, pp. 52589--52608, 2019, doi: 10.1109/ACCESS.2019.2912648.

[22] Ahmed I. Zahran, Aly A. Fahmy, Khaled T. Wassif, and Hanaa Bayomi, "Fine-Tuning Self-Supervised Learning Models for End-to-End Pronunciation Scoring," *IEEE Access*, vol. 11, pp. 112650--112663, 2023, doi: 10.1109/ACCESS.2023.3317236.

[23] Abdelfatah Ahmed, Mohamed Bader, Ismail Shahin, Ali Bou Nassif, Naoufel Werghi, and Mohammad Basel, "Arabic Mispronunciation Recognition System Using LSTM Network," *Information*, vol. 14, no. 7, Art. no. 413, 2023, doi: 10.3390/info14070413.

[24] Linkai Peng, Kaiqi Fu, Binghuai Lin, Dengfeng Ke, and Jinsong Zhang, "A Study on Fine-Tuning Wav2vec2.0 Model for the Task of Mispronunciation Detection and Diagnosis," in *Proc. Interspeech*, 2021. [Online]. Available at https://api.semanticscholar.org/CorpusID:239705587

[25] Hyun-Woo Bae, Hyung-Seok Oh, Seung-Bin Kim, and Seong-Whan Lee, "UnitCorrect: Unit Based Mispronunciation Correcting System With a DTW Based Detection," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 17531762, 2025, doi: https://doi.org/10.1109/TASLPRO.2025.3559344

[26] Long Zhang, Ziping Zhao, Chunmei Ma, Linlin Shan, Huazhi Sun, Lifen Jiang, Shiwen Deng, and Chang Gao, "End-to-End Automatic Pronunciation Error Detection Based on Improved Hybrid CTC/Attention Architecture," *Sensors*, vol. 20, no. 7, Art. no. 1809, 2020, doi: 10.3390/s20071809.

[27] W. Freeman Twaddell, "On Defining the Phoneme," *Language*, vol. 11, no. 1, pp. 5--62, 1935. [Online]. Available from: http://www.jstor.org/stable/522070

[28] Youssef Elfahm, Nesrine Abajaddi, Badia Mounir, Laila Elmaazouzi, Ilham Mounir, and Abdelmajid Farchi, "Classification of Arabic fricative consonants according to their places of articulation," *Int. J. Electr. Comput. Eng. (IJECE)*, 2022. [Online]. Available at https://api.semanticscholar.org/CorpusID:243830288

[29] Ali M. Alagrami and Maged M. Eljazzar, "Smartajweed Automatic Recognition of Arabic Quranic Recitation Rules," *arXiv*, 2020. [Online]. Available from: https://arxiv.org/abs/2101.04200

[30] Sükrü Selim Calk, Ayhan Kucukmanisa, and Zeynep Hilal Kilimci, "An ensemble-based framework for mispronunciation detection of Arabic phonemes," *Appl. Acoust.*, vol. 212, Art. no. 109593, 2023, doi: 10.1016/j.apacoust.2023.109593.

[31] Ann Lee and James Glass, "Pronunciation assessment via a comparison-based system," in *Proc. Speech and Language Technology in Education (SLaTE)*, 2013, pp. 122--126.

[32] Khaled Necibi and Halima Bahi, "An Arabic Mispronunciation Detection System by means of Automatic Speech Recognition Technology," in *Proc. ACIT*, Dec. 2012.

[33] Shamila Akhtar, Fawad Hussain, Fawad Raja, Muhammad Ehatisham-ul-Haq, Naveed Khan Baloch, Farruh Ishmanov, and Yousaf Zikria, "Improving Mispronunciation Detection of Arabic Words for Non-Native Learners Using Deep Convolutional Neural Network Features," *Electronics*, vol. 9, Jun. 2020, Art. no. 963, doi: 10.3390/electronics9060963.

[34] Yassine El Kheir, Fouad Khnaisser, Shammur Absar Chowdhury, Hamdy Mubarak, Shazia Afzal, and Ahmed Ali, "QVoice: Arabic Speech Pronunciation Learning Application," *ArXiv*, 2023. [Online]. Available at: https://arxiv.org/abs/2305.07445

[35] Nizar Y. Habash, *Introduction to Arabic Natural Language Processing*, Springer Cham, 2010, doi: 10.1007/978-3-031-02139-8.

[36] John Levis, "Computer Technology in Teaching and Researching Pronunciation," *Annu. Rev. Appl. Linguist.*, vol. 27, pp. 184--202, 2007, doi: 10.1017/S0267190508070098.

[37] Pete Sharma, "Sounds: The Pronunciation App," *ELT J.*, vol. 66, no. 3, pp. 407--409, Jul. 2012, doi: 10.1093/elt/ccs025.

[38] English Computerized Learning Inc., "eEnglish by Pronunciation Power," 2024. [Online]. Available: https://eenglish.com/ Accessed: 2024-11-03.

[39] Auralog, "Tell Me More Language Learning Software," 2013. [Online]. Available: https://www.rosettastone.com/

[40] Duolingo, "Duolingo Language Learning Platform," 2024. [Online]. Available: https://www.duolingo.com/ Accessed: 2024-11-03.

[41] Martha C. Pennington and Pamela Rogerson-Revell, "Using Technology for Pronunciation Teaching, Learning, and Assessment," in *English Pronunciation Teaching and Research*, Palgrave Macmillan UK, 2019, pp. 235--286, doi: 10.1057/978-1-137-47677-7_5.

[42] Kohichi Takai, Panikos Heracleous, Keiji Yasuda, and Akio Yoneyama, "Deep Learning-Based Automatic Pronunciation Assessment for Second Language Learners," in *HCI Int. 2020 - Posters*, Springer, pp. 338--342, 2020.

[43] Yunfei Shen, Qingqing Liu, Zhixing Fan, Jiajun Liu, and Aishan Wumaier, "Self-Supervised Pre-Trained Speech Representation Based End-to-End Mispronunciation Detection and Diagnosis of Mandarin," *IEEE Access*, vol. PP, pp. 1--1, Jan. 2022, doi: 10.1109/ACCESS.2022.3212417.

[44] Mohamed Yassine El Amrani, M. M. Rahman, Mohamed Ridza Wahiddin, and Asadullah Shah, "Building CMU Sphinx Language Model for The Holy Quran using Simplified Arabic Phonemes," *Egyptian Informatics Journal*, vol. 17, no. 3, pp. 305--314, May 2016, doi: 10.1016/j.eij.2016.04.002.

[45] Muazzam Maqsood, Adnan Habib, and Tabassam Nawaz, "An Efficient Mispronunciation Detection System Using Discriminative Acoustic Phonetic Features for Arabic Consonants," *Int. Arab J. Inf. Technol.*, vol. 16, pp. 242--250, Mar. 2019.

[46] Khaled Necibi and Halima Bahi, "A statistical-based decision for Arabic pronunciation assessment," *Int. Journal of Speech Technol.*, vol. 18, Mar. 2014, doi: 10.1007/s10772-014-9248-2.

[47] Khaled Necibi, Hamza Frihia, and Halima Bahi, "On The Use of Decision Trees for Arabic Pronunciation Assessment," in *Proc. Int. Conf. on Intelligent Information Processing, Security, and Advanced Communication*, 2015, Art. no. 74, pp. 1--6, doi: 10.1145/2816839.2816866.

[48] Edward Wilder Caro Anzola and Miguel Mendoza Moreno, "Goodness of Pronunciation Algorithm in the Speech Analysis and Assessment for Detecting Errors in Acoustic Phonetics: An exploratory review," *TechRxiv*, Jun. 2023, doi: 10.36227/techrxiv.23512038.v1.

[49] Binghuai Lin and Liyuan Wang, "Deep Feature Transfer Learning for Automatic Pronunciation Assessment," in *Proc. Interspeech*, Aug. 2021, pp. 4438--4442, doi: http://doi.org/10.21437/Interspeech.2021-931.

[50] Eesung Kim, Jae-Jin Jeon, Hyeji Seo, and Hoon Kim, "Automatic Pronunciation Assessment using Self-Supervised Speech Representation Learning," *arXiv*, 2022. [Online]. Available at: https://arxiv.org/abs/2204.03863

[51] ükrü Selim Çalk, Ayhan Küçükmanisa, and Zeynep Hilal Kilimci, "A novel framework for mispronunciation detection of Arabic phonemes using audio-oriented transformer models," *Appl. Acoust.*, vol. 215, Art. no. 109711, 2024, doi: 10.1016/j.apacoust.2023.109711.

[52] Xiaoshuo Xu, Yueteng Kang, Songjun Cao, Binghuai Lin, and Long Ma, "Explore Wav2vec 2.0 for Mispronunciation Detection," in *Proc. Interspeech*, Aug. 2021, pp. 4428--4432, doi: http://doi.org/10.21437/Interspeech.2021-777.

[53] Mostafa Shahin, Julien Epps, and Beena Ahmed, "Phonological level Wav2vec2-based mispronunciations detection and diagnosis method,"

Speech Communication, Volume 173, 2025, Article 103249. doi: https://doi.org/10.1016/j.specom.2025.103249

[54] Minglin Wu, Jing Xu, Xueyuan Chen, and Helen Meng, "Integrating Potential Pronunciations for Enhanced Mispronunciation Detection and Diagnosis Ability in LLMs," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 15, 2025, doi: https://doi.org/10.1109/ICASSP49660.2025.10887601

[55] Yassine El Kheir, Omnia Ibrahim, Amit Meghanani, Nada Almarwani, Hawau Olamide Toyin, Sadeen Alharbi, Modar Alfadly, Lamya Alkanhal, Ibrahim Selim, Shehab Elbatal, Salima Mdhaffar, Thomas Hain, Yasser Hifny, Mostafa Shahin, and Ahmed Ali, "Towards a Unified Benchmark for Arabic Pronunciation Assessment: Quranic Recitation as Case Study," arXiv preprint arXiv:2506.07722, 2025. Available at: https://arxiv.org/abs/2506.07722

[56] Alexei Baevski, Henry Zhou, Abdel-rahman Mohamed, and Michael Auli, "Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations," *ArXiv*, vol. abs/2006.11477, 2020. [Online]. Available at https://api.semanticscholar.org/CorpusID:219966759

[57] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 34513460, 2021.

[58] Chen, Sanyuan and Wang, Chengyi and Chen, Zhengyang and Wu, Yu and Liu, Shujie and Chen, Zhuo and Li, Jinyu and Kanda, Naoyuki and Yoshioka, Takuya and Xiao, Xiong and Wu, Jian and Zhou, Long and Ren, Shuo and Qian, Yanmin and Qian, Yao and Wu, Jian and Zeng, Michael and Yu, Xiangzhan and Wei, Furu, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE J. Sel. Topic Signal Process.*, vol. 16, no. 6, pp. 1505--1518, Oct. 2022, doi: 10.1109/jstsp.2022.3188113.

[59] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever, "Robust Speech Recognition via Large-Scale Weak Supervision, " in *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Journal of Machine Learning Research*, pages 28492--28518. PMLR, July 23--29, 2023. URL: https://proceedings.mlr.press/v202/radford23a.html.

[60] Omar Mohamed and Salah A. Aly, "Arabic Speech Emotion Recognition Employing Wav2vec2.0 and HuBERT Based on BAVED Dataset," *ArXiv*, 2021. [Online]. Available at: https://arxiv.org/abs/2110.04425

[61] Max Bain, Jaesung Huh, Tengda Han, and Andrew Zisserman, "WhisperX: Time-Accurate Speech Transcription of Long-Form Audio," *ArXiv*, 2023. [Online]. Available at: https://arxiv.org/abs/2303.00747

[62] Qiantong Xu, Alexei Baevski, and Michael Auli, "Simple and Effective Zero-shot Cross-lingual Phoneme Recognition," *ArXiv*, 2021. [Online]. Available at: https://arxiv.org/abs/2109.11680

[63] Hugging Face Inc., "Hugging Face: The AI community building the future," 2023. [Online]. Available from: https://huggingface.co

[64] AI at Meta, "Wav2vec2-XLSR-53," 2021. [Online]. Available at https://huggingface.co/facebook/wav2vec2-large-xlsr-53

[65] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE ICASSP*, 2015, pp. 5206--5210, doi: 10.1109/ICASSP.2015.7178964.

[66] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "Wav2vec: Unsupervised Pre-training for Speech Recognition," *ArXiv*, 2019. [Online]. Available from: https://arxiv.org/abs/1904.05862

[67] Ram Charan Chandra Shekar, Mu Yang, Kevin Hirschi, Stephen Looney, Okim Kang, and John Hansen, "Assessment of Non-Native Speech Intelligibility using Wav2vec2-based Mispronunciation Detection and Multi-level Goodness of Pronunciation Transformer," in *Proc. Interspeech*, August 2023, pp. 984--988, doi: http://doi.org/10.21437/Interspeech.2023-2371.

[68] Marcely Zanon Boito and John Ortega and Hugo Riguidel and Antoine Laurent and Loïc Barrault and Fethi Bougares and Firas Chaabani and Ha Nguyen and Florentin Barbier and Souhir Gahbiche and Yannick Estève, "ON-TRAC Consortium Systems for the IWSLT 2022 Dialect and Low-resource Speech Translation Tasks," in *Proc. IWSLT*, Dublin, Ireland, 2022, pp. 308--318, doi: 10.18653/v1/2022.iwslt-1.28.

[69] Alexis Conneau and Alexei Baevski and Ronan Collobert and Abdelrahman Mohamed and Michael Auli, "Unsupervised Cross-lingual Representation Learning for Speech Recognition," *ArXiv*, 2020. [Online]. Available at: https://arxiv.org/abs/2006.13979

[70] Mohamed El-Geish, "Wav2vec2-Large-XLSR-53-Arabic," 2020. [Online]. Available at https://huggingface.co/elgeish/wav2vec2-large-xlsr-53-arabic

[71] Jonatas Grosman, "Fine-tuned XLSR-53 large model for speech recognition in Arabic," 2021. [Online]. Available at https://huggingface.co/jonatasgrosman/wav2vec2-large-xlsr-53-arabic

[72] Nawar Halabi, "Modern standard Arabic phonetics for speech synthesis," Ph.D. dissertation, Univ. Southampton, 2016.

[73] Rosana Ardila and Megan Branson and Kelly Davis and Michael Henretty and Michael Kohler and Josh Meyer and Reuben Morais and Lindsay Saunders and Francis M. Tyers and Gregor Weber, "Common Voice: A Massively-Multilingual Speech Corpus," in *Proc. LREC*, 2020, pp. 4211--4215.

[74] Jiajun Liu, Aishan Wumaier, Cong Fan, and Shen Guo, "Automatic Fluency Assessment Method for Spontaneous Speech without Reference Text," *Electronics*, vol. 12, no. 8, Art. no. 1775, 2023, doi: 10.3390/electronics12081775.

[75] Kun Li, Xiaojun Qian, and Helen Meng, "Mispronunciation Detection and Diagnosis in L2 English Speech Using Multidistribution Deep Neural Networks," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 1, pp. 193--207, 2017, doi: 10.1109/TASLP.2016.2621675.

[76] Wasfi G. Al-Khatib, Mohammad Ismail Amro, Abdulkareem Alzahrani, Taha Fanoush, Moustafa Elshafei, "Arabic Pronunciation Assessment for Saudi Arabian Students: Corpus Development and System Architecture. in *Proc. International Conference on Artificial Intelligence and its Applications in the Age of Digital Transformation*, Springer, 2024, pp. 28--40.

[77] Nizar Habash, Abdelhadi Soudi, and Timothy Buckwalter, "On Arabic Transliteration," in *Arabic Computational Morphology*, Springer, Dordrecht, 2007, pp. 15--22, doi: 10.1007/978-1-4020-6046-5_2.

[78] Mohamed A. Ali, Moustafa Elshafei, Mansour Al-Ghamdi, and Husni Al-Muhtaseb, "Arabic Phonetic Dictionaries for Speech Recognition," *J. Inf. Technol. Res.*, vol. 2, pp. 67--80, 2009. [Online]. Available at https://api.semanticscholar.org/CorpusID:18162681

[79] Fayçal Imedjdouben and Amrane Houacine, "Automatic Phonetization of Arabic Text," *Stud. Comput. Intell.*, vol. 488, pp. 85--94, 2013, doi: 10.1007/978-3-319-00560-7_13.

[80] Vladimir Likic, "The Needleman-Wunsch algorithm for sequence alignment," Lecture, Univ. of Melbourne, 2008, pp. 1--46.

[81] Mirco Ravanelli and Titouan Parcollet and Peter Plantinga and Aku Rouhe and Samuele Cornell and Loren Lugosch and Cem Subakan and Nauman Dawalatabad and Abdelwahab Heba and Jianyuan Zhong and Ju-Chieh Chou and Sung-Lin Yeh and Szu-Wei Fu and Chien-Feng Liao and Elena Rastorgueva and François Grondin and William Aris and Hwidong Na and Yan Gao and Renato De Mori and Yoshua Bengio, "SpeechBrain: A General-Purpose Speech Toolkit," *ArXiv*, vol. abs/2106.04624, 2021. [Online]. Available at https://api.semanticscholar.org/CorpusID:235377273

[82] Ibrahim Abdalaal, Mohamed Abdelwahed, and Moustafa Elshafei,"Syllable-Based Arabic Speech Recognition Using Wav2vec," *Journal of Computational Linguistics & Arabic Language Processing*, 2024, pp. 120142.King Salman Global Academy for Arabic Language, doi: https://doi.org/10.60161/2521-001-001-006

TAHA FANNOUSH earned the B.Sc. degree in Computer Science from the University of Benghazi, Libya, in 2019, and an M.Sc. degree in Computer Science from King Fahd University of Petroleum and Minerals, Saudi Arabia, in 2024. From 2019 to 2021, he worked as a software developer specializing in mobile applications, after which he served as a Teaching Assistant at King Fahd University of Petroleum and Minerals in Dhahran from 2022 to 2024 while pursuing his master's degree. He is currently a faculty member in the Computer Science Department at the University of Benghazi. His research centers on Arabic speech processing, with particular emphasis on AI-based mispronunciations, diagnostic feedback, and phoneme recognition for language-learning applications.

WASFI AL-KHATIB (Member, IEEE) received the B.S. degree in Computer Science from Kuwait University, Kuwait, in 1990, the M.S. degree in Computer Science from Purdue University, West Lafayette, IN, USA, in 1995, and the Ph.D. degree in Electrical and Computer Engineering from the same institution in 2001. From 2001 to 2002, he was an Assistant Professor at Wright State University, Dayton, OH, USA. He is currently an Assistant Professor at King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia. His research interests include Arabic computing, multimedia computing, content-based retrieval, AI and software engineering. He led funded research projects, supervised numerous graduate students, and contributed to curriculum development and ABET accreditation. He is a member of the ACM and IEEE Computer Societies.

MOHAMMAD AMRO received the M.Sc. degree in Computer Science and the Ph.D. degree in Computer Science and Engineering from King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2009 and 2017, respectively. He has more than 18 years of academic and industrial experience. From 2006 to 2009, he served as a Lab Supervisor at Palestine Polytechnic University, Hebron, Palestine. He joined KFUPM in 2009, progressed from a Research Assistant to Lecturer, and has served as a Technology Consultant and Advisor in the University's IT Department since 2017. He chairs the IT Strategy and Digital Transformation, is a member of the Smart Campus consultancy initiative, and leads KFUPM's collaboration with OpenAI to introduce ChatGPT Edu across the university and join the OpenAI Champion Network. He has taught courses in software testing, Python, C programming, and data structures. In 2023, he became a Guest Affiliate with KFUPM's Interdisciplinary Research Center for Intelligent Secure Systems, focusing on Arabic mispronunciations assessment and the identification of Arabic poetry meters. His research interests include speech recognition, machine translation, and software engineering.

ABDULKARIM AL-ZAHRANI is a Professor in the Department of Islamic and Arabic Studies, College of General Studies, at King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia. His research interests include Arabic linguistics, computational linguistics, and application of digital technologies in language learning. He is the founder of the course "Introduction to the Digitization of the Arabic Language," which bridges classical linguistic theory with AI and modern computational methods. He has led applied scientific research projects on phonetic analysis and the automatic recognition of Arabic poetic meters and holds a patent in this domain. He was a key contributor to the Interactive Arabic Reading Tutor project and, since 2020, has been a Guest Affiliate with the Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS) at KFUPM, focusing on speech-based Arabic language assessment and the digitization of classical Arabic.

MOUSTAFA ELSHAFEI-AHMED (SM'92) received the Ph.D. degree in Electrical Engineering from McGill University (Dean List), Canada, in 1982. Since then, he has accumulated more than 31 years of academic experience and nine years of industrial experience. He was a Visiting Scientist at the Massachusetts Institute of Technology from 2010. He is a Professor of Mechatronics Engineering at the Misr University of Science and Technology. He is a former Professor of Systems Engineering at King Fahd University of Petroleum and Minerals (KFUPM). He is the inventor/co-inventor of 23 U.S. and international patents. He has authored and coauthored five books and published over 170 articles in international journals and professional conferences in the fields of speech processing, digital signal processing, AI, intelligent instrumentation, and industrial control/automation systems. He has been involved in many projects at KFUPM, supported by KFUPM, Saudi Aramco, SABIC, Yokogawa, KACST, Department of Civil Defense, and NSTIP. He is a member of the ISA and SPE.