Automatic Pronunciation Error Detection and Correction of the Holy Quran's Learners Using Deep Learning

Abdullah Abdelfttah [*1] **Mahmoud I. Khalil** [1] **Hazem Abbas** [1]

Abstract

Assessing spoken language is challenging, and quantifying pronunciation metrics for machine learning models is even harder. However, for the Holy Quran, this task is enabled by the rigorous recitation rules (Tajweed) established through the efforts of Muslim scholars, making highly effective assessment possible. Despite this advantage, the scarcity of high-quality annotated data remains a significant barrier. In this work, we bridge these gaps by introducing: (1) A 98% automated pipeline to produce high-quality Quranic datasets – encompassing collection of recitations from expert reciters, segmentation at pause points (waqf) using our fine-tuned wav2vec2-BERT model, transcription of segments, and transcript verification via our novel Tasmeea algorithm; (2) 848 hours of audio (286K annotated utterances); (3) qdat_bench, a benchmark covering phonemes, diacritization, and Tajweed rules (Ghunnah, Qalqalah, Madd) on real recitation errors containing 159 samples; (4) A novel ASR-based approach for pronunciation error detection utilizing our custom Quran Phonetic Script (QPS) to encode Tajweed rules (unlike the IPA standard for Modern Standard Arabic). QPS uses an 11-level script: phoneme level (encoding Arabic letters with short/long vowels) and sifat level (encoding articulation characteristics of every phoneme). We further present comprehensive modeling with our novel multi-level CTC model, which achieved 0.21% and 1.94% average Phoneme Error Rate (PER) on the test set and qdat_bench respectively, with a 75.8% Tajweed F1 score. We release our work as open-source: https://obadx.github.io/quran-muaalem/en/

[1]Faculty of Engineering, Ain Shams University, Cairo, Egypt. Correspondence to: Abdullah Abdelfttah <2101398@eng.asu.edu.eg>, Mahmoud I. Khalil <mahmoud.khalil@eng.asu.edu.eg>, Hazem Abbas <hazem.abbas@eng.asu.edu.eg>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

Assessing pronunciation is not a simple task (Kheir et al., 2023), as it relies not only on correct phoneme pronunciation but also on intonation, prosody, and stress. The Holy Quran presents unique characteristics: it is among the easiest spoken texts to learn despite containing special phonemes absent in other languages.

The Holy Quran is considered the word of Allah by Muslims. It was revealed to Prophet Muhammad (peace be upon him) and holds immense significance for the Muslim community. The Quran possesses unique linguistic characteristics; it is neither poetry nor prose in the traditional sense of Arabic literature but introduces a new art form that takes on a lyrical quality through recitation (Tajweed). Consequently, Quranic recitation involves unique pronunciation characteristics, such as elongation (Madd), nasalization (Ghunnah), and echo (Qalqalah). According to Ibn al-Jazari, these patterns must be pronounced consistently (Ibn al Jazari). As non-Arabic speakers convert to Islam, pronunciation errors can occur. To preserve the exact pronunciation and meaning, ancient Muslim scholars established recitation rules derived from the Quran recitation itself.

The pronunciation of the Holy Quran is governed by rigorously strict rules formally defined by ancient Muslim scholars since the 6th century. Despite being precise and accurate, these rules have not been comprehensively digitized (to our knowledge) for Quranic pronunciation assessment.

This paper focuses on automatic detection and localization of Tajweed Rules and Sifat-related pronunciation errors from audio, enabling formative feedback, automated tutoring, and standardized assessment.

To the best of our knowledge, RDI (Sherif et al., 2007) was the first to apply AI for detecting pronunciation errors alongside Tajweed rules. However, they did not disclose their phoneticization process nor release their data or models publicly. Subsequent research has remained largely incomplete – often focusing on a subset of rules without

Table 1. Vocabulary size per level in the Quran Phonetic Script.

Level	Vocabulary Size
Phonemes	43
Tafkheem / Tarqeeq	3
Shidda / Rakhawa	3
Moraqaq / Muftah	2
Itbaq	2
Safeer	2
Qalqala	2
Tikraar	2
Tafashie	2
Istitala	2
Ghunnah	2

open-source code or datasets. To address these gaps, we introduce:

- A phonetizer that encodes all Tajweed rules and articulation attributes (Sifat) defined by classical scholars (except Ishmam).
- A 98% automated pipeline for generating highly accurate datasets from expert recitations.
- A dataset of 286K annotated utterances (848 hours).
- qdat_bench, a benchmark for phonemes, diacritization, and Tajweed rules (Ghunnah, Qalqalah, Madd) on real recitation errors (159 samples).
- A multi-level CTC model that demonstrates the learnability of QPS (0.21% average PER).

Unlike previous approaches that use IPA-based phoneme sets for Modern Standard Arabic (Omran et al., 2023b), diacritized Uthmani text without Tajweed rules (Khan et al., 2021), or focus on single rules like Qalqalah only (Omran et al., 2023a), our QPS comprehensively encodes all Tajweed rules and Sifat attributes in a unified representation.

Our multi-level CTC architecture offers significant token reduction compared to flat representations. A single-level approach encoding all Tajweed rules would require 99,072 tokens (cross-product of 43 phonemes × 10 Sifat × variations), whereas our hierarchical approach with 11 levels as shown in table 1 achieves the same representational capacity with dramatically fewer output units, enabling more efficient training and better generalization.

The paper is organized as follows: Related Work, Quran Phonetic Script, Data Pipeline, Modeling, Results, Limitations and Future Work, Conclusion, and Appendix.

2. Related Work

2.1. Quran Pronunciation Datasets

Table 2 summarizes the most relevant Quran pronunciation datasets. EveryAyah is the largest openly available dataset with 26 complete *Moshafs* segmented and annotated Ayah-by-Ayah by both experts such as Al Hossary and non-experts such as Fares Abbad. Qdat (Osman et al., 2021) contains 1,509 utterances of single specific Ayahs labeled for three rules: Madd, Ghunna, and Ikhfaa. Although the scale is relatively small, it was widely adopted by the community (Omran et al., 2023a; Shaiakhmetov et al., 2025) due to being open-source. The Tarteel v1 dataset (Khan et al., 2021) consists of 25K utterances with diacritics and no Tajweed rules. The latter is the Tarteel private dataset, a massive 9K-hour collection annotated with diacritics without Tajweed rules. The most recent benchmark is IqraaEval (Kheir et al., 2025), which presents a test set of 2.2 hours from 18 speakers, but uses Modern Standard Arabic (MSA) without Tajweed rules.

None of the above provides explicit, rule-level Tajweed annotations across the full range of Hafs recitation rules. This scarcity has been a fundamental bottleneck.

2.2. Quran Pronunciation Models

The first work addressing automated pronunciation assessment for the Holy Quran is RDI (Sherif et al., 2007), which built a complete system for detecting pronunciation errors (Qalqala, Idgham, Iqlab) but omitted details on phoneticization. Subsequent work (Abdou & Rashwan, 2014; Al-Marri et al., 2018) used DNNs to replace HMMs. Many studies rely on phoneme duration for duration-dependent rules like Madd and Ghunna (Mohammed et al., 2017; Alqadasi et al., 2023) but use limited datasets. Others focus on specific rules like Qalqala (Omran et al., 2023b) or Ghunna and Madd (Shaiakhmetov et al., 2025; Alsahafi & Asad, 2024). The work most aligned with our vision is (Putra et al., 2012), which introduced a multi-level detection system (Makhraj and Tajweed) but relied on HMMs and minimal data.

While prior efforts focused on rule-based detection with constrained corpora, Tarteel (Khan et al., 2021) advanced the underlying ASR infrastructure by developing a robust system for diacritized character recognition, albeit without integrating Tajweed-specific error classification. Notably, they addressed the critical data bottleneck by releasing a crowd-sourced dataset of 25K utterances (68 hours) and later expanding this to 9K hours of private annotated data through application users.

A critical pattern across prior work is limitation in: (a) data scale (verse-restricted corpora), (b) rule coverage (one to three rules), and (c) open-source availability. By con-

Dataset	Task	Size	Tajweed Rules	Sifat
EveryAyah	ASR / Recitation	114 reciters	None	No
Qdat (Osman et al., 2021)	Tajweed Detection	1,509 utterances	3 (Madd, Ghunna, Ikhfaa)	No
Tarteel v1 (Khan et al., 2021)	Diacritization	25K utterances (68 h)	None	No
Tarteel (private)	Diacritization	9K hours	None	No
IqraaEval (Kheir et al., 2025)	Pronunciation Assessment	2.2 hours	None	No
muaalem-annotated-v3 (ours)[1]	ASR + Tajweed Detection	848 h, 286K utterances, 22 reciters	All Hafs Tajweed rules (except Ishmam)	Yes
qdat_bench (ours)[2]	ASR + Tajweed Detection	159 samples (1 verse, Al-Ma'idah 5:109)	Madd, Ghunna, Ikhfaa, Qalqalah	Yes

Table 2. Comparison of Quran pronunciation datasets. The Sifat column indicates whether the dataset explicitly encodes *Sifat* (attributes of articulation of Holy Quran phonemes) at the phoneme level.

trast, this work targets all Hafs Tajweed rules (except Ishmam) using a large expert-recitation corpus and releases both code and data.

2.3. Pretrained Speech Encoders with Self-Supervised Learning (SSL)

Speech pretraining began early (Hinton & Salakhutdinov, 2006) but was constrained by RNNs (Hopfield, 1982). Transformers (Vaswani et al., 2017) enabled large-scale pretraining. BERT (Devlin et al., 2019) introduced masked language modeling, extended to speech with wav2vec (Schneider et al., 2019) and wav2vec2.0 (Baevski et al., 2020). Conformer (Gulati et al., 2020) integrated convolution. Google's Wav2Vec2-BERT (Chung et al., 2021) applied MLM to speech. Facebook extended Wav2Vec2-BERT pretraining (Barrault et al., 2023) to 4.5M hours (including 110K Arabic hours), ideal for low-resource fine-tuning.

Our work bridges the gap by pairing a large SSL encoder with QPS, a phonetic script that makes Tajweed structure explicit at the label level.

3. Quran Phonetic Script

We consider QPS to be the most valuable contribution of our work. By formalizing Holy Quran pronunciation assessment as an ASR problem represented through this script, we provide a comprehensive solution.

Modern Standard Arabic orthography cannot adequately represent Tajweed rules for error detection. Our phonetic script addresses this by capturing all Tajweed pronunciation errors except Ishmam (visual mouth movement without audible output). We based our script on classical Muslim scholarship rather than IPA for three reasons: (1) historical precedence (Muslim scholars from the 6th to 14th centuries defined Quranic errors centuries before modern phonetics), (2) scientific foundation (e.g., Al-Khalil ibn Ahmad systematically described articulations with remarkable accuracy (Al-Hibira, 2023)), and (3) pedagogical relevance (learners' errors align with classical definitions).

Following (Al-Swaid), Quran recitation errors fall into three categories: Articulation Errors (incorrect phoneme exits), Attribute Errors (mistakes in Sifat al-Huruf), and Tajweed Rule Errors (incorrect application of rules). Our script addresses all three through two main output levels: Phonemes Level (letters, vowels, and Tajweed rules) and Sifat Level (10 articulation attributes). Refer to Tables 10 and 11 for the complete vocabularies.

Our script has some important characteristics: Normal Madd appears as consecutive madd symbols (e.g., 4-beat Madd: ااااا); Madd al-Leen is represented with multiple waw/yaa symbols; Stressed Ghunnah (e.g., النون المشددة) is shown as three consecutive noon symbols (ننن); Ikhfa is represented as three consecutive noon_mokhfah (ںں) or meem_mokhfah (ۄۄۄ); إدغام بغنة (Ghunnah assimilation) for the letters yaa and waw is represented by doubling (e.g., مَن يَعْمَلْ → مَييَعمَل); Sakin is indicated by the absence of a following symbol; Imala is represented using fatha_momala and alif_momala; and Rawm is marked by a dama_mokhtalasa marker.

Table 3 illustrates the phonetization process for the word (أَتُحَٰجُّوٓنِّي), showing its conversion into phonetic components and Sifat attributes; refer to the table caption for a detailed row-by-row breakdown.

3.1. Development Methodology

Our phonetization has two steps:

1. **Imlaey to Uthmani Conversion**: We selected Uthmani script as our foundation because it contains specialized Tajweed diacritics (Madd, Tasheel, etc.) and preserves pause rules critical for recitation (e.g., stopping on رحمت). To do so, we created an annotation UI to manually annotate misaligned words in both scripts (e.g., 4), and then developed an algorithm that relies on these annotations to convert Imlaey to Uthmani.

Table 3. Examples of Uthmani to Phonetic Script Conversion with Sifat Attributes. This example shows the phonetization of word (أَنُحَجِّيٓنَّكُمۡ) row by row: The first row shows conversion of (أَ) to (ءَ) with its sifat in the row, following the second row. For the fourth row, showing the madd Lazim rule with 6 beats phoneticized as 6 alifs (ااااااا), same as the sixth row but with damma represented as 6 waw_madd (ووووو). And for the fifth row we notice that we converted (حِّ) to (حِ) disassembling shadda. The last row shows the normal madd of yaa with two beats represented as two yadd_madd (ۦۦ)

Uthmani	Phonetic	H/J	S/R	T/T	Itb	Saf	Qal	Tik	Taf	Ist	Gho
أَ	ءَ	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
نُ	نُ	hams	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
حَـ	حَ	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ـٰٓـ	اااااا	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
حِّ	حِ	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ـُو	وووووو	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
نَّ	نغنِ	jahr	btw	mrq	mnf	no	nql	nkr	ntf	nst	mg
ـِۦ	ۦۦ	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg

Attribute Abbreviations:
H/J: Hams/Jahr S/R: Shidda/Rakhawa T/T: Tafkheem/Taqeeq Itb: Itbaq
Saf: Safeer Qal: Qalqla Tik: Tikraar Taf: Tafashie Ist: Istitala Gho: Ghonna
Value Abbreviations:
shd: shadeed rkh: rikhw btw: between mrq: moraqaq
mof: mofakham mnf: monfateh mtb: motbaq no: no_safeer
nql: not_moqalqal nkr: not_mokarar ntf: not_motafashie
nst: not_mostateel nmg: not_maghnoon mg: maghnoon

Table 4. Example of misalignment between Uthmani and Imlaey Scripts

Imlaey Script	Uthmani Script
يَا ابْنَ أُمَّ	يَبْنَؤُمَّ

2. **Uthmani to Phonetic Script Conversion**
We implemented the process through 26 sequential operations. Each operation contains one or more regular expressions, as shown in the Appendix A.2.

3. **Extracting Sifat**: Next, we extract the 10 attributes (Sifat) defined in Table 11, excluding **Inhiraf** (إنحراف), as it describes the shidda/rakhawa spectrum, and **Leen** (اللين), as it was already handled through our Madd representation.

4. Data Preparation

To prepare the data, we first defined selection criteria. We aimed to collect recitations from the best reciters worldwide to serve as references for judging Quran learners. In our study, we considered only *Hafs* riwayah (رواية حفص) as it's the most popular recitation method globally. Recognizing that manual data annotation requires significant effort and time, we created a 98% automated pipeline as for data collection shown in Figure 4. The steps are:

1. **Digitized Quran:** We chose Tanzil (standard Unicode, both Imlaei and Uthmani versions) over KFGQPC due to stability.

2. **Variant Criteria for Hafs:** We defined variants (e.g., Madd al-Munfasil: 2,4,5, or 6 beats) through Qira'at literature (Al-Dabbaa).

3. **Expert Recitation Collection:** Criteria: perfect recitation, studio recording, complete Mushaf, high audio quality.

4. **Segmentation:** We fine-tuned Wav2Vec2-BERT (Barrault et al., 2023) for frame-level classification. Table 7 shows results on unseen Mushafs.

5. **Transcription:** We used Tarteel ASR (AI, 2023) (Whisper fine-tuned on Quranic recitations (Radford et al., 2023)) with 5.75% WER.

6. **Verification (Tasmeea Algorithm):** A fuzzy matching algorithm based on Levenshtein distance (Levenshtein, 1966) corrects transcription errors. Samples below 85% threshold are sent to human annotators.

We define a *Moshaf* as a complete Quran recitation (chapters 1-114) by a specific reciter. Statistics are summarized in table 5. We manually annotated 5400 samples out of 286,537 utterances, resulting for the automation ratio of 98%.

4.1. Choose a Digitized Version of the Holy Quran

The Quran has multiple digitized versions including Tanzil[3] and King Fahd Complex[4]. We chose Tanzil because:

[3] https://tanzil.net
[4] https://qurancomplex.gov.sa

Table 5. Dataset Statistics per Moshaf

Moshaf ID	Hours	Length
0.0	28.48	9133
0.1	40.31	10764
0.2	49.47	9971
0.3	37.19	12604
1.0	28.41	10939
2.0	51.05	9942
2.1	30.03	10394
3.0	25.19	10444
4.0	29.12	10994
5.0	28.02	11482
6.0	39.39	12435
7.0	28.26	9907
8.0	30.86	10330
9.0	27.95	10642
11.0	24.01	10363
12.0	33.42	9880
13.0	33.99	9377
19.0	30.11	11278
22.0	28.11	10332
24.0	28.51	9868
25.0	16.93	7922
26.0	30.44	11565
26.1	32.71	11850
27.0	28.05	11213
28.0	31.05	10535
29.0	27.79	11061
30.0	29.14	11312
Total	**847.9944402**	**286537**

- It uses standard Unicode characters
- Contains both *Imlaei* and *Uthmani* versions
- Maintains high accuracy

We excluded KFGQPC due to its evolving/unstable nature compared to Tanzil.

4.2. Define Variant Criteria for Hafs

Hafs riwayah contains variants, e.g., *Madd Al-Munfasil* (مد المنفصل) can extend 2, 4, 5, or 6 beats. We rigorously defined these variants through the Qira'at literature (Al-Dabbaa), summarized in the following attributes in the Appendix section A.4.

4.3. Collect Expert Recitations

We collected recitations from 22 world-class reciters with premium audio quality, totaling **893 hours** pre-filtering.

To ensure high-quality reference data for judging Quran learners, reciters were selected based on the following criteria: perfect recitation with no mistakes, non-prayer (studio) recording not during prayer, complete Moshaf (full Quran, chapters 1-114), and high audio quality. All recitations were recorded in high-quality studio conditions with thorough review to ensure high quality.

Table 6. Distribution of Reciters by Country

Country Name	Number of Reciters
Egypt (EG)	15
Saudi Arabia (SA)	3
Syria (SY)	1
Somalia (SO)	1
Kuwait (KW)	1
United Arab Emirates (AE)	1
Total	**22**

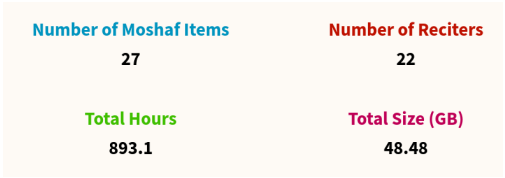

Figure 1. Database Collection Statistics

We developed a web GUI using Streamlit that downloads and extracts metadata for each track, organizes data by Moshaf (each chapter as "001.mp3"), and annotates Moshaf attributes.

4.4. Segment Recitations

Since Tajweed rules are affected by pauses (وقف), accurate segmentation is crucial. We initially tested open-source Voice Activity Detection (VAD) models including SileroVAD (Team, 2024) and PyAnnotate (Plaquet & Bredin, 2023). Poor Quran-specific performance led us to develop a custom segmenter by fine-tuning Wav2Vec2-BERT[5] (Barrault et al., 2023) for frame-level classification.

Preparing Segmenter Data We selected mosahf compatible with SileroVAD v4, using EveryAyah[6] (pre-segmented by ayah) as ground truth. After tuning parameters per Moshaf, we included: threshold, minimum silence duration (to merge segments), minimum speech duration (to discard short segments), and padding (added at segment boundaries).

Data Augmentation Using the Audiomentations (Jordal & Contributors, 2025) library, we replicated SileroVAD's noise setup on 40% of samples, adding `TimeStretch` (0.8x-1.5x) to simulate recitation speeds and sliding win-

[5] https://github.com/obadx/recitations-segmenter
[6] https://everyayah.com/

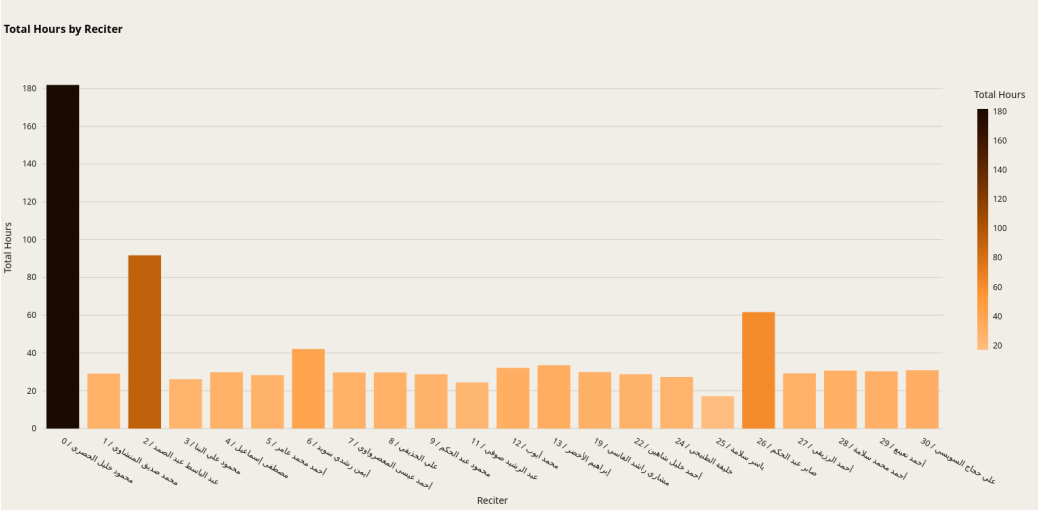

Figure 2. Reciters Statistics

Table 7. Test results of the segmenter on unseen full moshaf. The result is validated by actual usage of the segmenter

Metric	Value
Test Loss	0.0277
Test Accuracy	0.9935
Test F1 Score	0.99476

dow truncation (1-second windows) for long samples instead of exclusion.

Training Segmenter We fine-tuned Wav2Vec2-BERT for frame classification for 1 epoch using the AdamW optimizer with a constant learning rate of 5e-5, default betas and bfloat16 precision on an Nvidia L40 GPU, with a batch size of 50 and a maximum speech sample length of 20 seconds.

Results of our segmenter on unseen mosahf in table 7:

4.5. Transcribe Segmented Parts

We employed Tarteel ASR (AI, 2023) (Whisper fine-tuned on Quranic recitations (Radford et al., 2023)) with a WER of 5.7544%. To handle its 30-second limit, we used sliding window truncation (10-second windows), with verification in the next step.

4.6. Automated Data Pipeline Assessment

In this section, we discuss how we assessed the pipeline performance phase by phase:

Segmentation Verification: For the segmentation phase, we performed manual inspection of 50–75 random samples per Moshaf, in addition to short (< 3 seconds) and long (> 35 seconds) tracks. Any Moshaf with accuracy $< 99\%$ was excluded; specifically, Moshaf 25.0 was excluded due to poor segmentation (90%).

Transcription Verification: We utilized the Tarteel model, which has a 5.7544% Word Error Rate (WER). To correct potential transcription errors, we leveraged the fact that our input consists of 100% correct expert recitations to design the *Tasmeea*-inspired algorithm. This algorithm matches the Tarteel transcription to the Quranic text—analogous to a student reciting to a teacher—using fuzzy matching based on Levenshtein distance (Levenshtein, 1966) with a moving pointer. If the text matching threshold falls below 85%, the samples are referred to human annotators for correction.

The result is a high-quality dataset with no missing verses across 27 full Quran recitations. Refer to the Tasmeea Algorithm in the Appendix 11

5. Modeling The Quran Phonetic Script

Our Quran Phonetic script has two outputs: `phonemes` and `sifat` (which has 10 attributes). We implemented this as a speech encoder $E_\theta(x)$ followed by 11 linear heads W_k, representing 11 parallel transcription levels (`phonemes` and the 10 `sifat` attributes). We chose CTC loss (Graves et al., 2006) without language model integration because we aim to capture what the user actually said, not what they intended to say. We name our architecture **Multi-level CTC**. For each level k, the CTC loss is L_k, and the final weighted loss is:

$$L = \sum_{k=1}^{11} w_k L_k, \quad \text{subject to} \quad \sum_{k=1}^{11} w_k = 1 \qquad (1)$$

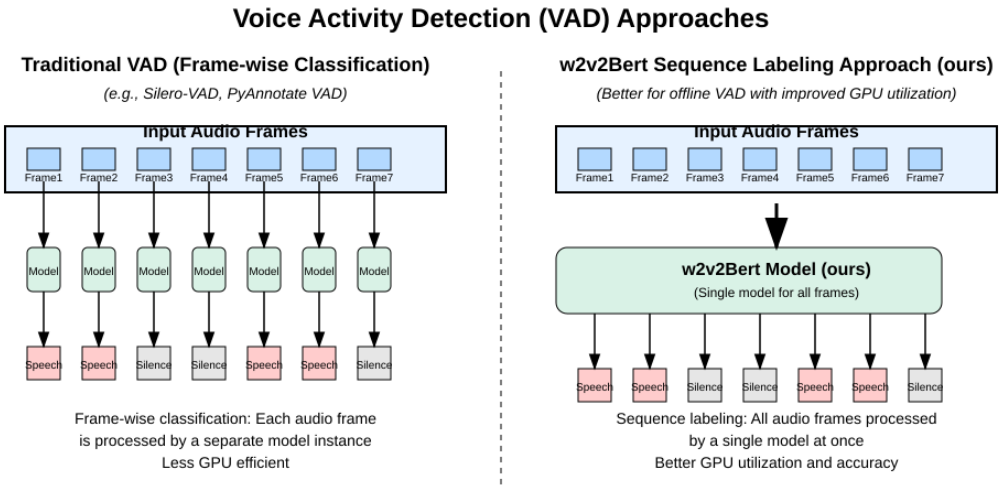

Figure 3. VAD architecture vs. standard streaming models

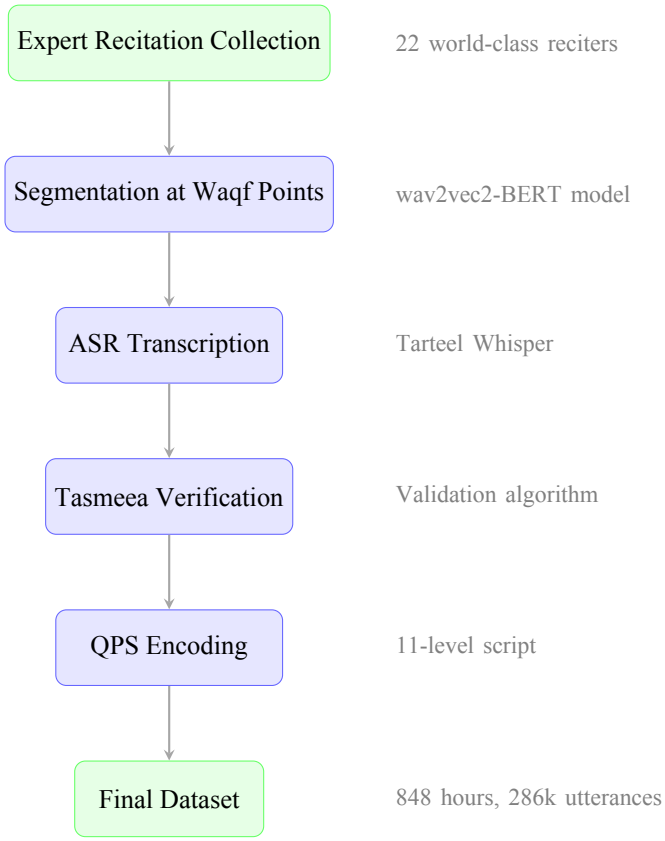

Figure 4. The complete data preparation pipeline showing the six main stages from expert recitation collection to the final dataset generation. Each stage utilizes specialized models and algorithms to ensure high-quality annotations.

The weights are assigned based on the vocabulary size of each level to balance their contribution. Specifically, the `phonemes` level (vocabulary size 44, including blank token) receives a weight of 0.4; the `shidda_or_rakhawa` and `tafkheem_or_taqeeq` levels (each with a vocabulary size of 4 including blank token) receive a weight of 0.0605 each. The remaining eight levels (each with a vocabulary size of 3 including blank token) receive a weight of 0.069875 each.

We fine-tuned Facebook's Wav2Vec2-Bert (Barrault et al., 2023) for a single epoch using an adapter layer and the AdamW optimizer with its default β values. We employed a constant learning rate of `5e-5`, a batch size of 64, and a maximum sample length of 30 seconds, with training performed in `bfloat16` precision. We applied augmentations identical to Silero VAD (Team, 2024) using the `audiomentations` library (Jordal & Contributors, 2025), with additional augmentations: `TimeStretch` and `GainTransition`. We filtered out samples longer than `30 seconds` for efficient GPU utilization, sacrificing only 3k samples out of 250k training samples. The training was conducted on an H200 GPU with 141 GB of GPU memory for 7 hours.

Due to our limited budget, we conducted preliminary experiments to determine the optimal hyperparameters by training on a subset of the training data (4 Mushafs) on Google Colab. We evaluated different training setups, including linear and cosine schedulers, and the impact of SpecAugment (Park et al., 2019). The best results were achieved with a constant learning rate and without SpecAugment, as the latter significantly degraded performance. Based on these findings, we proceeded with the full training on a high-performance GPU.

The convergence behavior of the model is shown in Figure 6, where the evaluation loss decreases consistently throughout the training process, confirming the stability of the multi-level CTC objective.

6. Results

6.1. Expert Recitation Evaluation

We trained on all available Moshaf datasets, reserving three Moshaf (19.0, 29.0, 30.0) for comprehensive testing. These test datasets feature expert male reciters with extensive training in Tajweed rules. The expert nature of these recordings provides an ideal evaluation environment for assessing the model's fundamental phonetic transcription capabilities across different representation levels. Notably, the phonemes level presents the greatest challenge with a 44-character vocabulary (including padding), resulting in the highest Phoneme Error Rate of 0.543% and average PER of 0.21% across all levels as shown in Table 8, while still demonstrating excellent overall performance that validates our multi-level CTC approach.

Table 8. Test Results on Expert Quranic Recitations on three Moshaf datasets (19.0, 29.0, 30.0). The phonemes level has the highest PER (0.543%) due to its 44-character vocabulary including padding.

Metric	Value
loss	0.01162
per_phonemes	0.00543
per_hams_or_jahr	0.00117
per_shidda_or_rakhawa	0.00172
per_tafkheem_or_taqeeq	0.00167
per_itbaq	0.00092
per_safeer	0.00132
per_qalqla	0.00085
per_tikraar	0.0009
per_tafashie	0.0016
per_istitala	0.0008
per_ghonna	0.0013
average_per	**0.0021**

6.2. Learner Error Evaluation (qdat_bench)

To evaluate our model's performance on real recitation errors, we tested on our developed qdat_bench benchmark, which enhances the original (Osman et al., 2021) dataset through extensive expert reannotation and the addition of all Tajweed rules, including phonemes and 10 sifat levels. We performed bootstrap analysis with 10,000 iterations to provide comprehensive view of metric distributions. The results, shown in Table 9, demonstrate the model's effectiveness in detecting pronunciation errors in authentic learner recordings.

qdat_bench shows a higher PER of 0.058 on authentic learner recordings, which is expected given the complexity of detecting real pronunciation errors and handling variability in learner execution. However, 5.8% remains very acceptable despite being trained on expert recitations only. The aggregate Tajweed F1 score of 0.758 represents the mean across three key rules: Noon Moshaddadah (0.869), Ikhfaa (0.453), and Qalqalah (0.953). Similarly, the average Madd RMSE of 0.596 encompasses normal madd (0.464), separate madd (0.687), and aared madd (1.034).

Notably, the Ikhfaa F1 score of 0.453 is considerably lower than other rules, which is expected given the acoustic similarity between Ikhfaa and clear noon pronunciation (see Figure 8). This challenge affects both human perception and automated detection. The model's performance on this rule reflects the inherent difficulty in detecting subtle nasalization differences. Detailed analysis is provided in the appendix A.5.

Notably, despite being trained exclusively on male ex-

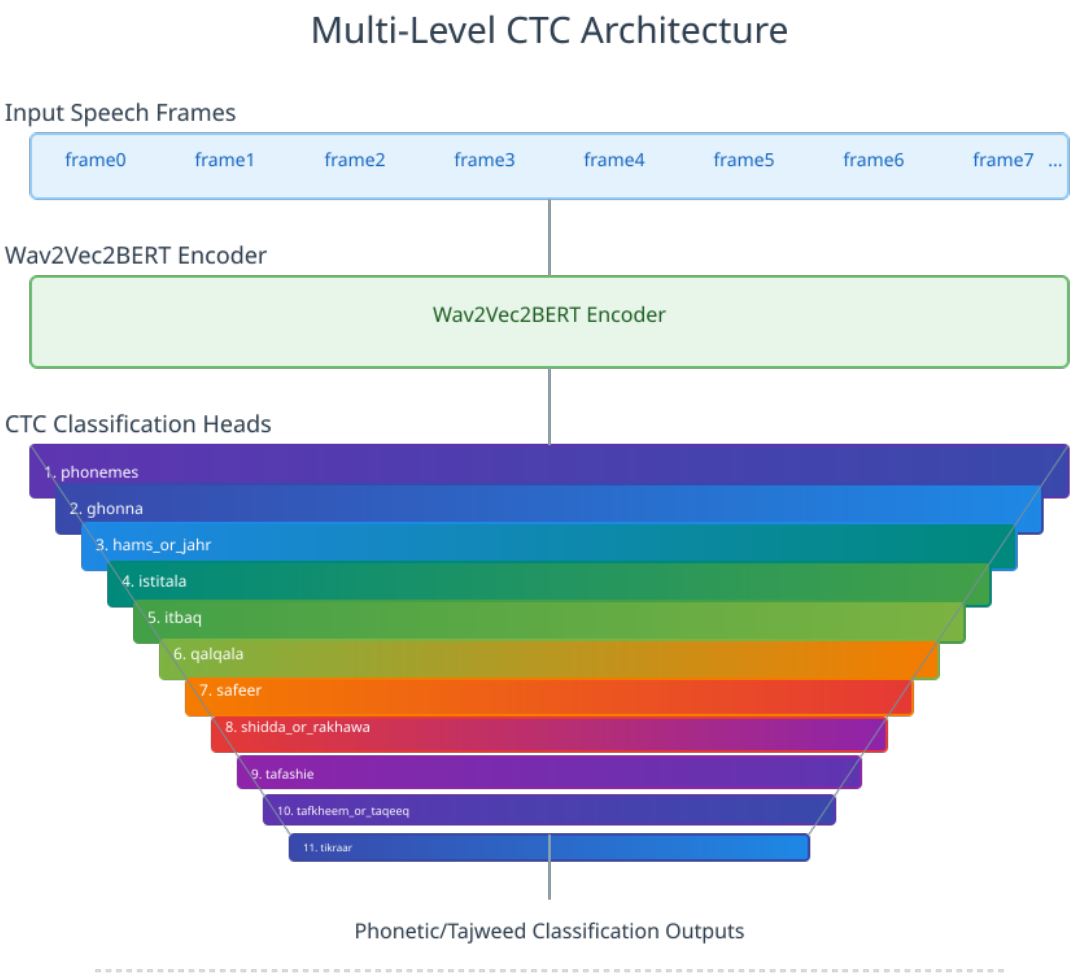

Figure 5. Multi-level CTC loss Architecture composed of 11 Heads for every level and CTC loss for every level with weighted average loss

Table 9. qdat_bench Results on Authentic Learner Recordings. Evaluated on 159 samples (120 female, 39 male) from Surah Al-Ma'idah (5:109). F1 for Tajweed rules: Noon Moshaddadah, Ikhfaa, Qalqlah. RMSE for Madd lengths: normal, separated, and aared madd.

Aggregate Metrics	**Original**	**Bootstrap (n=10,000)**
per_phonemes	0.0578	0.0603 ± 0.0053
avg_per	0.0194	0.0197 ± 0.0032
avg_tajweed_f1	0.7582	0.7575 ± 0.0177
avg_tajweed_acc	0.8470	0.8464 ± 0.0174
avg_madd_rmse	0.5965	0.5960 ± 0.0374
avg_normal_madd_rmse	0.4645	0.4688 ± 0.0501
rmse_madd_aared	1.0340	1.0251 ± 0.0724
rmse_separate_madd	0.6758	0.6868 ± 0.0582
Tajweed Rules F1	**Original**	**Bootstrap (n=10,000)**
Noon Moshaddadah	0.8691	0.8682 ± 0.0324
Ikhfaa (Noon Mokhfah)	0.4526	0.4510 ± 0.0259
Qalqalah	0.9530	0.9532 ± 0.0174
Madd Rules RMSE	**Original**	**Bootstrap (n=10,000)**
Normal Madd (5 rules)	0.4645	0.4688 ± 0.0501
Separate Madd	0.6868	0.6758 ± 0.0582
Aared Madd	1.0340	1.0251 ± 0.0724

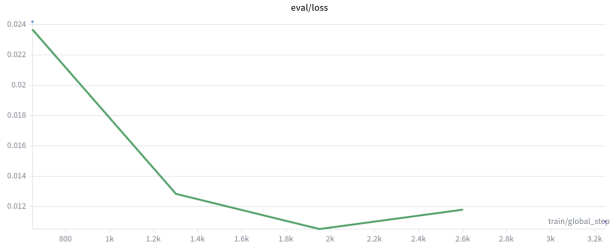

Figure 6. Evolution of the evaluation loss during training.

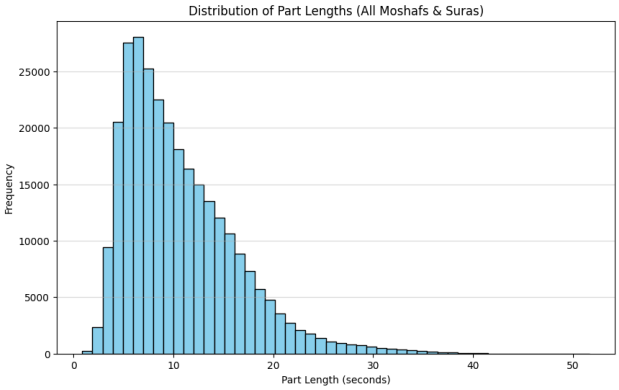

Figure 7. Recitations lengths in seconds for the whole dataset

pert reciters, our model demonstrates promising generalization by achieving 84.7% accuracy on female recitations in qdat_bench, highlighting robustness for real-world deployment. This performance validates the practical applicability of our Quranic phonetic script and multi-level CTC architecture.

Figure 9 visualizes the bootstrap distributions for aggregate metrics, showing stable estimates with narrow confidence intervals.

Despite these promising results, certain model limitations should be acknowledged. Attribute-specific articulation patterns present inherent challenges: certain phonetic attributes apply exclusively to individual letters, such as `Istitala` for (ض) and `Tikrar` for (ر). Future work should focus on annotating real data with these specific errors to improve model understanding (see Section 7 for detailed discussion).

7. Limitations and Future Work

A major limitation appears from attribute-specific articulation patterns: Certain attributes apply exclusively to individual letters, such as `Istitala` for (ض) and `Tikrar` for (ر). Consequently, we expect our model will be unable to capture instances of (ض) without `Istitala` or (ر) without `Tikrar`. This limitation similarly applies to Tajweed rules that occur less frequently in the Holy Quran, such as `Imala`, `Rawm`, and `Tasheel`. The solution is to annotate real data with these errors.

Additionally, the training dataset comprises only expert male reciters, which may limit generalizability to diverse vocal characteristics. The current scope is restricted to Hafs recitation style; future work will extend to other Riwayah (e.g., Warsh). Finally, qdat_bench uses a single annotator and one verse; future versions should include multiple ex-

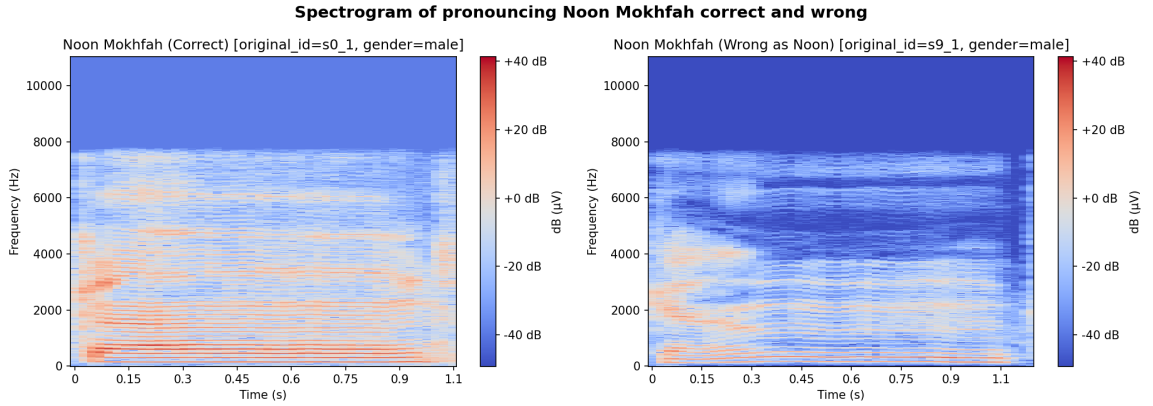

Figure 8. The figure illustrates the challenge of distinguishing Ikhfaa pronunciation from clear noon pronunciation for both humans and AI models. Spectrogram comparison of Noon Mokhfah (Ikhfaa) pronunciation from qdat_bench: (left) correct Ikhfaa pronunciation of أنت by a male reciter (original_id=s0_1), and (right) incorrect elongated noon pronunciation by a male reciter (original_id=s9_0) that the model failed to detect as a pronunciation error.

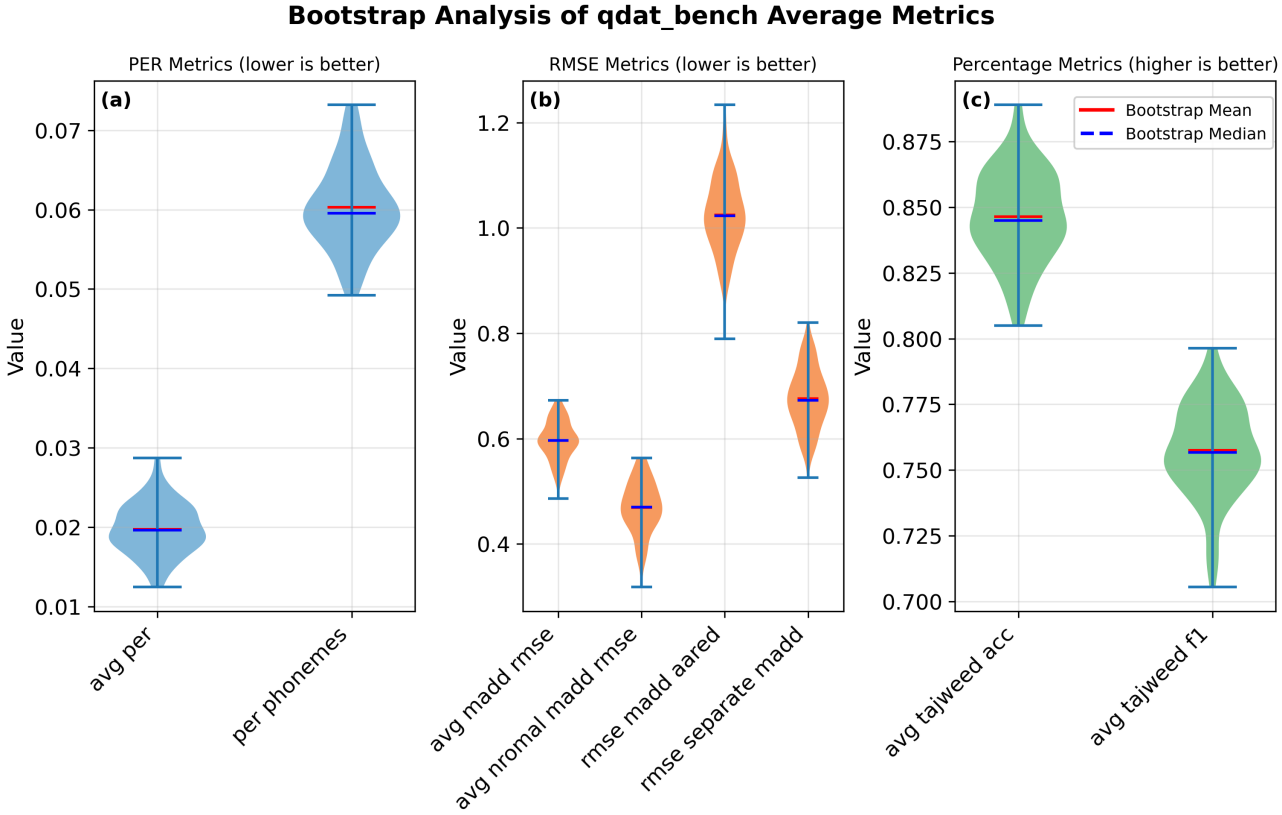

Figure 9. Violin plot for bootstrap analysis (n=10,000) of qdat_bench aggregate metrics. (a) Phoneme Error Rate (PER) metrics (lower is better), (b) RMSE metrics for Madd rules (lower is better), and (c) Tajweed performance metrics (higher is better). Dashed lines indicate original metric values before bootstrapping.

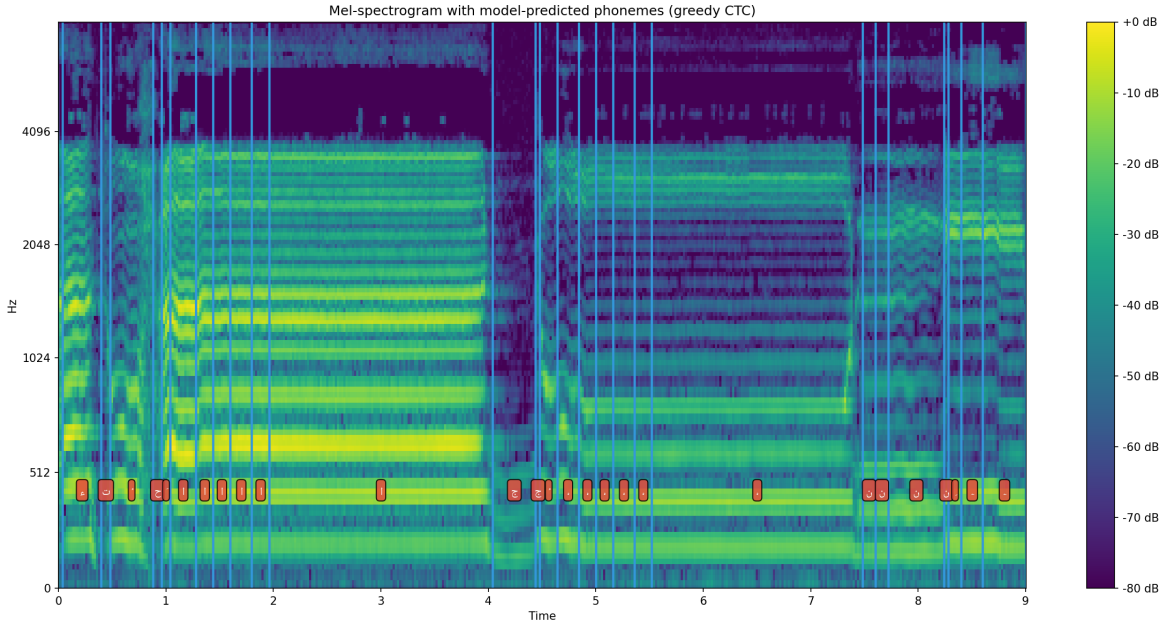

Figure 10. Spectrogram analysis of the aligned model output for the word أَتَحجُونِّي (refer to Table 3 for the complete reference phonetization). Reference: ءءنِنننووووجُجاااااااتُحَاء, Predicted: ءءنِنننووووجُجاااااااتُحَاء. The phoneme boundaries remain near perfect; however, one error is observed: the fatha on the initial hamza (ء → ء) was missed. Additionally, the Madd alif (ااااااا) and Madd waw (ووووو) durations are not balanced, with the final beat of the Madd sequence being notably larger.

perts and broader Quranic coverage.

8. Conclusion

We present a novel approach for assessing pronunciation errors in Holy Quran learners through a multi-level Quran Phonetic Script that captures all pronunciation errors for *Hafs* (except `Ishmam`, as it is a visual diacritic not orally produced). We provide 848 hours of annotated audio data with 286K samples, a 98% automated pipeline for generating similar datasets featuring our Tasmeea verification algorithm, and a novel multi-level CTC model with an 11-level structure (1 phoneme level + 10 sifat levels). Achieving a 0.21% average phoneme error rate on unseen expert test data proves the learnability of the Quran Phonetic Script. Furthermore, our model demonstrates robust generalization on real learner errors through qdat_bench, achieving 75.8% Tajweed F1 score and 84.7% accuracy on female reciters, fundamentally transforming Holy Quran pronunciation assessment methodology.

Acknowledgement

We express our profound gratitude to several individuals and organizations: Azza Abdolmiem, for her unwavering support and for providing a peaceful environment throughout the research period; Sheikh Ahmed Abdelsalam, Sheikh Mustafa Fathy, and Sheikh Mohamed Rabee, for their invaluable guidance in understanding and representing Tajweed rules and learner mistakes. We also thank BA-HPC[7] for providing access to high-performance computing resources and facilitating data processing. Special appreciation goes to Engineer Khaled Bahaa for his assistance with the payment method for GPUs.

References

Abdou, S. M. and Rashwan, M. A computer aided pronunciation learning system for teaching the holy quran recitation rules. In *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, pp. 543–550. IEEE, 2014.

AI, T. Whisper base arabic quran (automatic speech recognition for quranic recitation). https://huggingface.co/tarteel-ai/whisper-base-ar-quran, 2023. Model by Tarteel AI. Company website: https://www.tarteel.ai/.

Al-Dabbaa, A. *Clear Expression in the Disputed Words about Hafs*. Maktabat wa Matba'at Mustafa al-Babi al-Halabi wa Awladuh. Original: صريح النص في الكلمات المختلف فيها عن حَفْصٍ [1380AH (1960CE)].

Al-Hibira, I. Makharij al-huruf al-sihah and their character-

[7] https://hpc.bibalex.org/

istics according to al-khalil al-farahidi: A study in modern linguistics. *Prince Sattar Bin Abdulaziz Journal of Islamic Sciences*, 31:165–190, 2023. doi: 10.37138/emirj.v31i2.1936. Original: مخارج الحروف الصحاح وصفاتها عند الخليل بن أحمد الفراهيدي في ضوء الدرس اللساني الحديث.

Al-Marri, M., Raafat, H., Abdallah, M., Abdou, S., and Rashwan, M. Computer aided qur'an pronunciation using dnn. *Journal of Intelligent & Fuzzy Systems*, 34(5): 3257–3271, 2018.

Al-Swaid, A. R. *The Illustrated Tajweed*. Dar al-Ghubthani for Quranic Studies. Original: التجويد المصور [2021].

Alqadasi, A. M. A., Zeki, A. M., Sunar, M. S., Salam, M. S. B. H., Abdulghafor, R., and Khaled, N. A. Improving automatic forced alignment for phoneme segmentation in quranic recitation. *IEEE Access*, 12:229–244, 2023.

Alsahafi, Y. S. and Asad, M. Empirical study on mispronunciation detection for tajweed rules during quran recitation. In *2024 6th International Conference on Computing and Informatics (ICCI)*, pp. 39–45, 2024. doi: 10.1109/ICCI61671.2024.10485145.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460, 2020.

Barrault, L., Chung, Y.-A., Meglioli, M. C., Dale, D., Dong, N., Duppenthaler, M., Duquenne, P.-A., Ellis, B., Elsahar, H., Haaheim, J., et al. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023.

Chung, Y.-A., Zhang, Y., Han, W., Chiu, C.-C., Qin, J., Pang, R., and Wu, Y. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 244–250. IEEE, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*, pp. 369–376. ACM, 2006. doi: 10.1145/1143844.1143891.

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Hinton, G. E. and Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

Ibn al Jazari, S. a.-D. *Al-Jazariyyah Introduction*. Dar al-Ghubthani for Quranic Studies. Original: المقدمة الجزرية. Edited and annotated by Ayman Rushdi Al-Swaid [2021].

Jordal, I. and Contributors. Audiomentations: A python library for audio data augmentation. https://github.com/iver56/audiomentations, 2025.

Khan, H. I., Abid, A., Moussa, M. M., and Abou-Allaban, A. The tarteel dataset: crowd-sourced and labeled quranic recitation. 2021.

Kheir, Y. E., Ali, A., and Chowdhury, S. A. Automatic pronunciation assessment–a review. *arXiv preprint arXiv:2310.13974*, 2023.

Kheir, Y. E., Ibrahim, O., Meghanani, A., Almarwani, N., Toyin, H. O., Alharbi, S., Alfadly, M., Alkanhal, L., Selim, I., Elbatal, S., et al. Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study. *arXiv preprint arXiv:2506.07722*, 2025.

Levenshtein, V. I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

Mohammed, A., Sunar, M. S. B., and Salam, M. S. H. Recognition of holy quran recitation rules using phoneme duration. In *International Conference of Reliable Information and Communication Technology*, pp. 343–352. Springer, 2017.

Omran, D., Fawzi, S., and Kandil, A. Automatic detection of some tajweed rules. In *2023 20th Learning and Technology Conference (L&T)*, pp. 157–160, 2023a. doi: 10.1109/LT58159.2023.10092350.

Omran, D., Fawzi, S., and Kandil, A. Automatic detection of some tajweed rules. In *2023 20th Learning and Technology Conference (L&T)*, pp. 157–160. IEEE, 2023b.

Osman, H. M., Mustafa, B. S., and Faisal, Y. Qdat: a data set for reciting the quran. *International Journal on Islamic Applications in Computer Science And Technology*, 9(1):1–9, 2021.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

Plaquet, A. and Bredin, H. Powerset multi-class cross entropy loss for neural speaker diarization. In *Proc. INTERSPEECH 2023*, 2023.

Putra, B., Atmaja, B. T., and Prananto, D. Developing speech recognition system for quranic verse recitation learning software. *IJID (International Journal on Informatics for Development)*, 1(2):1–8, 2012.

Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., and Sutskever, I. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pp. 28492–28518. PMLR, 2023.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

Shaiakhmetov, D., Gimaletdinova, G., Momunov, K., and Cankurt, S. Evaluation of the pronunciation of tajweed rules based on dnn as a step towards interactive recitation learning. *arXiv preprint arXiv:2503.23470*, 2025.

Sherif, M., Samir, A., Khalil, A., and Mohsen, R. Enhancing usability of capl system for quran recitation learning. INTERSPEECH, 2007.

Team, S. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad, 2024.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008, 2017.

A. Appendix

A.1. Quran Phoneme Script Vocabulary

A.2. Uthmani to Phonetic Conversion Operations

The 26 sequential phonetization operations:

1. **DisassembleHrofMoqatta** (تفكيك حروف مقطعة): Separates Quranic initials (e.g., الم، الر) into individual letters.
2. **SpecialCases** (حالات خاصة): Handles special words like يبسط that have different pronunciation forms defined in `MoshafAttributes`.
3. **BeginWithHamzatWasl** (البدء بهمزة الوصل): Processes words starting with connecting hamza (ٱ) and converts it to hamza (ء) with appropriate harakah for nouns and verbs.
4. **BeginWithSaken** (البدء بساكن): Manages words beginning with a consonant (sakin) like لْيَقْطَعْ, as Arabic doesn't start utterances with consonants.
5. **ConvertAlifMaksora** (تحويل الألف المقصورة): Converts ى in Uthmani script to either yaa (ي) or alif (ا) based on context.
6. **NormalizeHmazat** (توحيد الهمزات): Standardizes hamza forms (أ إ ؤ ئ) to ء.
7. **IthbatYaaYohie** (إثبات ياء يحيى): Handles words like يُحْيِۦ where two yaa letters occur - resolves conflicts when pausing on words with consecutive consonants (التقاء الساكنين) by adding another yaa at end.
8. **RemoveKasheeda** (إزالة الكشيدة): Deletes elongation marks (ـ) from text.
9. **RemoveHmzatWaslMiddle** (إزالة همزة الوصل الوسطية): Removes connecting hamza (ٱ) in non-initial positions.
10. **RemoveSkoonMostadeer** (حذف الحرف الذي فوقع سكون مستدير): Eliminates letters with circular sukoon diacritics like alif in جَمَعُواْ.
11. **SkoonMostateel** (سكون مستطيل): Removes alif with elongated sukoon mid-word and adds it at the end during pauses (وقف).
12. **MaddAlewad** (مد العوض): Removes alif after tanween fatha mid-word and adds alif while removing tanween at pause positions (وقف).
13. **WawAlsalah** (واو الصلاة): Replaces letter waw (و) with small alif above combined with alif.
14. **EnlargeSmallLetters** (تكبير الحروف الصغيرة): Resizes miniature Arabic letters to standard proportions.
15. **CleanEnd** (تنظيف النهاية): Removes redundant diacritics and spaces at word endings.
16. **NormalizeTaa** (توحيد التاء): Converts ة (taa marbuta) to ت or ه based on context, and converts final ة to haa (ه).
17. **AddAlifIsmAllah** (إضافة ألف اسم الله): Inserts compensatory alif in derivatives of "الله".
18. **PrepareGhonnaIdghamIqlab** (تهيئة الغنة والإدغام والإقلاب): Preprocesses text for nasalization, assimilation, and conversion rules.
19. **IltiqaaAlsaknan** (التقاء الساكنين): Resolves consecutive consonants by inserting vowels.
20. **DeleteShaddaAtBeginning** (حذف الشدة في البداية): Removes shadda (ّ) from word-initial letters.
21. **Ghonna** (غنة): Applies nasalization during pronunciation of sakin noon and tanween.
22. **Tasheel** (تسهيل): Adds a letter representing alif with tasheel easing.
23. **Imala** (إمالة): Converts fatha with imala to `fatha_momala` phoneme and alif with imala to `alif_momala` phoneme.

24. **Madd** (مد): Adds madd symbols for all madd types, inserting `madd_alif` (ا), `madd_waw` (و), and `madd_yaa` (ء).

25. **Qalqla** (قلقة): Adds echoing effect to ق, ط, ب, ج, د letters with sukoon.

26. **RemoveRasHaaAndShadda** (إزالة رأس الحاء علامة السكون): Deletes sukoon diacritic marks.

A.3. Tasmeea Verfication Algorithm

Figure 11. Tasmeea Algorithm

```
Algorithm 1 Tasmeea Algorithm
Input: text_segments, sura_idx, overlap_words=6,
       window_words=30, acceptance_ratio=0.85
Output: List of (match, ratio)
1: aya <- 1
2: penalty <- 0
3: for each segment s_i do
4:     norm_text <- normalize(s_i)
5:     best_ratio <- 0, best_match <- null
6:     for each start p in start_range do
7:         for each window w in [min_win, max_win] do
8:             candidate <- extract(aya, p, w)
9:             dist <- edit_distance(norm_text, candidate)
10:            ratio <- 1 - min(dist, |norm_text|)/|norm_text|
11:            if ratio > best_ratio then update best
12:         end for
13:     end for
14:     if best_ratio <= acceptance_ratio then
15:         output (null, best_ratio); penalty <- max_win; aya <- aya+1
16:     else
17:         output (best_match, best_ratio)
18:         aya <- aya + best_start + best_window; penalty <- 0
19:     end if
20: end for
```

A.4. Moshaf Attribute Definitions

- **rewaya** (الرواية)
  - Values: - `hafs` (حفص)
  - Default Value:
  - More Info: The type of the quran Rewaya.
- **recitation_speed** (سرعة التلاوة)
  - Values:
    - `mujawad` (مجود)
    - `above_murattal` (فويق المرتل)
    - `murattal` (مرتل)
    - `hadr` (حدر)
  - Default Value: `murattal` (مرتل)
  - More Info: The recitation speed sorted from slowest to the fastest سرعة التلاوة مرتبة من الأبطأ إلي الأسرع
- **takbeer** (التكبير)

  - Values:
    * `no_takbeer` (لا تكبير)
    * `beginning_of_sharh` (التكبير من أول الشرح لأول الناس)
    * `end_of_doha` (التكبير من آخر الضحى لآخر الناس)
    * `general_takbeer` (التكبير أول كل سورة إلا التوبة)
  - Default Value: `no_takbeer` (لا تكبير)
  - More Info: The ways to add takbeer (الله أكبر) after Istiaatha (استعاذة) and between end of the surah and beginning of the surah. `no_takbeer`: "لا تكبير" — No Takbeer (No proclamation of greatness, i.e., there is no Takbeer recitation) `beginning_of_sharh`: "التكبير من أول الشرح لأول الناس" — Takbeer from the beginning of Surah Ash-Sharh to the beginning of Surah An-Nas `end_of_dohaf`: "التكبير من آخر الضحى لآخر الناس" — Takbeer from the end of Surah Ad-Duha to the end of Surah An-Nas `general_takbeer`: "التكبير أول كل سورة إلا التوبة" — Takbeer at the beginning of every Surah except Surah At-Tawbah

- **madd_monfasel_len** (مد المنفصل)
  - Values:
    * `2`
    * `3`
    * `4`
    * `5`
  - Default Value:
  - More Info: The length of Mad Al Monfasel "مد المنفصل" for Hafs Rewaya.
- **madd_mottasel_len** (مقدار المد المتصل)
  - Values:
    * `4`
    * `5`
    * `6`
  - Default Value:
  - More Info: The length of Mad Al Motasel "مد المتصل" for Hafs.
- **madd_mottasel_waqf** (مقدار المد المتصل وقفا)
  - Values:
    * `4`
    * `5`
    * `6`
  - Default Value:
  - More Info: The length of Madd Almotasel at pause for Hafs.. Example "السماء".
- **madd_aared_len** (مقدار المد العارض)
  - Values:
    * `2`
    * `4`
    * `6`
  - Default Value:
  - More Info: The length of Mad Al Aared "مد العارض للسكون".
- **madd_alleen_len** (مقدار مد اللين)
  - Values:
    * `2`

    - 4
    - 6
  - Default Value: None
  - More Info: The length of the Madd al-Leen when stopping at the end of a word (for a sakin waw or ya preceded by a letter with a fatha) should be less than or equal to the length of Madd al-'Arid (the temporary stretch due to stopping). **Default Value is equal to** madd_aared_len. مقدار مد اللين عن القوف (للواو الساكنة والياء الساكنة وقبلها حرف مفتوح) ويجب أن يكون مقدار مد اللين أقل من أو يساوي مع العارض

- **ghonna_lam_and_raa** (غنة اللام و الراء)
  - Values:
    - ghonna (غنة)
    - no_ghonna (لا غنة)
  - Default Value: no_ghonna (لا غنة)
  - More Info: The ghonna for merging (Idghaam) noon with Lam and Raa for Hafs.

- **meem_aal_imran** (ميم آل عمران في قوله تعالى: {الم الله} وصلا)
  - Values:
    - waqf (وقف)
    - wasl_2 (فتح الميم ومدها حركتين)
    - wasl_6 (فتح الميم ومدها ستة حركات)
  - Default Value: waqf (وقف)
  - More Info: The ways to recite the word meem Aal Imran (الم الله) at connected recitation. waqf: Pause with a prolonged madd (elongation) of 6 harakat (beats). wasl_2 Pronounce "meem" with fathah (a short "a" sound) and stretch it for 2 harakat. wasl_6 Pronounce "meem" with fathah and stretch it for 6 harakat.

- **madd_yaa_alayn_alharfy** (مقدار المد اللازم الحرفي للعين)
  - Values:
    - 2
    - 4
    - 6
  - Default Value: 6
  - More Info: The length of Lzem Harfy of Yaa in letter Al-Ayen Madd "المد الحرفي اللازم لحرف العين" in surar: Maryam "مريم", AlShura "الشورى".

- **saken_before_hamz** (الساكن قبل الهمز)
  - Values:
    - tahqeek (تحقيق)
    - general_sakt (سكت عام)
    - local_sakt (سكت خاص)
  - Default Value: tahqeek (تحقيق)
  - More Info: The ways of Hafs for saken before hamz. "The letter with sukoon before the hamzah (ء)".And it has three forms: full articulation (tahqeeq), general pause (general_sakt), and specific pause (local_skat).

- **sakt_iwaja** (السكت عند عوجا في الكهف)
  - Values:
    - sakt (سكت)
    - waqf (وقف)
    - idraj (إدراج)

  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word "عوجا" (Iwaja). `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_marqdena** (السكت عند مرقدنا في يس)
  - Values:
    * `sakt` (سكت)
    * `waqf` (وقف)
    * `idraj` (إدراج)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word "مرقدنا" (Marqadena) in Surat Yassen. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_man_raq** (السكت عند من راق في القيامة)
  - Values:
    * `sakt` (سكت)
    * `waqf` (وقف)
    * `idraj` (إدراج)
  - Default Value: `sakt` (سكت)
  - More Info: The ways to recite the word "من راق" (Man Raq) in Surat Al Qiyama. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_bal_ran** (السكت عند بل ران في المطففين)
  - Values:
    * `sakt` (سكت)
    * `waqf` (وقف)
    * `idraj` (إدراج)
  - Default Value: `sakt` (سكت)
  - More Info: The ways to recite the word "بل ران" (Bal Ran) in Surat Al Motaffin. `sakt` means slight pause. `idraj` means not `sakt`. `waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idraj` (no sakt).

- **sakt_maleeyah** (وجه قوله تعالى {ماليه هلك} بالحاقة)
  - Values:
    * `sakt` (سكت)
    * `waqf` (وقف)
    * `idgham` (إدغام)
  - Default Value: `waqf` (وقف)
  - More Info: The ways to recite the word {ماليه هلك} in Surah Al-Ahqaf. `sakt` means slight pause. `idgham` Assimilation of the letter 'Ha' (ه) into the letter 'Ha' (ه) with complete assimilation.`waqf`: means full pause, so we can not determine whether the reciter uses `sakt` or `idgham`.

- **between_anfal_and_tawba** (وجه بين الأنفال والتوبة)
  - Values:
    * `waqf` (وقف)
    * `sakt` (سكت)
    * `wasl` (وصل)
  - Default Value: `waqf` (وقف)

  - More Info: The ways to recite end of Surah Al-Anfal and beginning of Surah At-Tawbah.

- **noon_and_yaseen** (الإدغام والإظهار في النون عند الواو من قوله تعالى: {يس والقرآن}و {ن والقلم})
  - Values:
    * `izhar` (إظهار)
    * `idgham` (إدغام)
  - Default Value: `izhar` (إظهار)
  - More Info: Whether to merge noon of both: {يس} and {ن} with (و) "`idgham`" or not "`izhar`".

- **yaa_ataan** (إثبات الياء وحذفها وقفا في قوله تعالى {آتان} بالنمل)
  - Values:
    * `wasl` (وصل)
    * `hadhf` (حذف)
    * `ithbat` (إثبات)
  - Default Value: `wasl` (وصل)
  - More Info: The affirmation and omission of the letter 'Yaa' in the pause of the verse {آتاني} in Surah An-Naml. `wasl`: means connected recitation without pausing as (آتانيَ). `hadhf`: means deletion of letter (ي) at pause so recited as (آتان). `ithbat`: means confirmation reciting letter (ي) at pause as (آتاني).

- **start_with_ism** (وجه البدأ بكلمة {الاسم} في سورة الحجرات)
  - Values:
    * `wasl` (وصل)
    * `lism` (لسم)
    * `alism` (ألسم)
  - Default Value: `wasl` (وصل)
  - More Info: The ruling on starting with the word {الاسم} in Surah Al-Hujurat. `lism` Recited as (لسم) at the beginning. `alism` Recited as (ألسم). `wasl`: means completing recitation without pausing as normal, So Reciting is as (بئس لسم).

- **yabsut** (السين والصاد في قوله تعالى: {والله يقبض ويبسط} بالبقرة)
  - Values:
    * `seen` (سين)
    * `saad` (صاد)
  - Default Value: `seen` (سين)
  - More Info: The ruling on pronouncing `seen` (س) or `saad` (ص) in the verse {والله يقبض ويبسط} in Surah Al-Baqarah.

- **bastah** (السين والصاد في قوله تعالى: {وزادكم في الخلق بسطة} بالأعراف)
  - Values:
    * `seen` (سين)
    * `saad` (صاد)
  - Default Value: `seen` (سين)
  - More Info: The ruling on pronouncing `seen` (س) or `saad` (ص) in the verse {وزادكم في الخلق بسطة} in Surah Al-A'raf.

- **almusaytirun** (السين والصاد في قوله تعالى {أم هم المصيطرون} بالطور)
  - Values:

    - `seen` (سين)
    - `saad` (صاد)
  - Default Value: `saad` (صاد)
  - More Info: The pronunciation of `seen` (س) or `saad` (ص) in the verse {أم هم المصيطرون} in Surah At-Tur.

- **bimusaytir** (السين والصاد في قوله تعالى: {لست عليهم بمصيطر} بالغاشية)
  - Values:
    - `seen` (سين)
    - `saad` (صاد)
  - Default Value: `saad` (صاد)
  - More Info: The pronunciation of `seen` (س) or `saad` (ص) in the verse {لست عليهم بمصيطر} in Surah Al-Ghashiyah.

- **tasheel_or_madd** (همزة الوصل في قوله تعالى: {آلذكرين} بموضعي الأنعام و{آلآن} موضعي يونس و{آلله} بيونس والنمل)
  - Values:
    - `tasheel` (تسهيل)
    - `madd` (مد)
  - Default Value: `madd` (مد)
  - More Info: Tasheel of Madd "وجع التسهيل أو المد" for 6 words in The Holy Quran: "ءالذكرين", "ءالله", "ءائن".

- **yalhath_dhalik** (الإدغام وعدمه في قوله تعالى: {يلهث ذلك} بالأعراف)
  - Values:
    - `izhar` (إظهار)
    - `idgham` (إدغام)
    - `waqf` (وقف)
  - Default Value: `idgham` (إدغام)
  - More Info: The assimilation (`idgham`) and non-assimilation (`izhar`) in the verse {يلهث ذلك} in Surah Al-A'raf. `waqf`: means the reciter has paused on (يلهث)

- **irkab_maana** (الإدغام والإظهار في قوله تعالى: {اركب معنا} بهود)
  - Values:
    - `izhar` (إظهار)
    - `idgham` (إدغام)
    - `waqf` (وقف)
  - Default Value: `idgham` (إدغام)
  - More Info: The assimilation and clear pronunciation in the verse {اركب معنا} in Surah Hud. This refers to the recitation rules concerning whether the letter "Noon" (ن) is assimilated into the following letter or pronounced clearly when reciting this specific verse. `waqf`: means the reciter has paused on (اركب)

- **noon_tamnna** (الإشمام والروم (الاختلاس) في قوله تعالى {لا تأمنا على يوسف})
  - Values:
    - `ishmam` (إشمام)
    - `rawm` (روم)
  - Default Value: `ishmam` (إشمام)
  - More Info: The nasalization (`ishmam`) or the slight drawing (`rawm`) in the verse {لا تأمنا على يوسف}

- **harakat_daaf** (حركة الضاد (فتح أو ضم) في قوله تعالى {ضعف} بالروم)
  - Values:

    * `fath` (فتح)
    * `dam` (ضم)
  - Default Value: `fath` (فتح)
  - More Info: The vowel movement of the letter 'Dhad' (ض) (whether with `fath` or `dam`) in the word {ضعف} in Surah Ar-Rum.

- **alif_salasila** (إثبات الألف وحذفها وقفا في قوله تعالى: {سلاسلا} بسورة الإنسان)
  - Values:
    * `hadhf` (حذف)
    * `ithbat` (إثبات)
    * `wasl` (وصل)
  - Default Value: `wasl` (وصل)
  - More Info: Affirmation and omission of the 'Alif' when pausing in the verse {سلاسلا} in Surah Al-Insan. This refers to the recitation rule regarding whether the final "Alif" in the word "سلاسلا" is pronounced (affirmed) or omitted when pausing (waqf) at this word during recitation in the specific verse from Surah Al-Insan. `hadhf`: means to remove alif (ا) during pause as (سلاسل) `ithbat`: means to recite alif (ا) during pause as (سلاسلا) `wasl` means completing the recitation as normal without pausing, so recite it as (سلاسلَ وأغلالا)

- **idgham_nakhluqkum** (إدغام القاف في الكاف إدغاما ناقصا أو كاملا {نخلقكم} بالمرسلات)
  - Values:
    * `idgham_kamil` (إدغام كامل)
    * `idgham_naqis` (إدغام ناقص)
  - Default Value: `idgham_kamil` (إدغام كامل)
  - More Info: Assimilation of the letter 'Qaf' into the letter 'Kaf,' whether incomplete (`idgham_naqis`) or complete (`idgham_kamil`), in the verse {نخلقكم} in Surah Al-Mursalat.

- **raa_firq** (التفخيم والترقيق في راء {فرق} في الشعراء وصلا)
  - Values:
    * `waqf` (وقف)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `tafkheem` (تفخيم)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {فرق} in Surah Ash-Shu'ara' when connected (wasl). This refers to the recitation rules concerning whether the letter "Ra" (ر) in the word "فرق" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when reciting the specific verse from Surah Ash-Shu'ara' in connected speech. `waqf`: means pausing so we only have one way (tafkheem of Raa)

- **raa_alqitr** (التفخيم والترقيق في راء {القطر} في سبأ وقفا)
  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `wasl` (وصل)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {القطر} in Surah Saba' when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "القطر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Saba'. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **raa_misr** (التفخيم والترقيق في راء {مصر} في يونس وموضعي يوسف والزخرف وقفا)

  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `wasl` (وصل)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {مصر} in Surah Yunus, and in the locations of Surah Yusuf and Surah Az-Zukhruf when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "مصر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) at the specific pauses in these Surahs. `wasl`: means not pausing so we only have one way (tafkheem of Raa)

- **raa_nudhur** (التفخيم والترقيق في راء {نذر} بالقمر وقفا)
  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `tafkheem` (تفخيم)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {نذر} in Surah Al-Qamar when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "نذر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Al-Qamar. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **raa_yasr** (التفخيم والترقيق في راء {يسر} بالفجر و{أن أسر} بطه والشعراء و{فأسر} بهود والحجر والدخان وقفا)
  - Values:
    * `wasl` (وصل)
    * `tafkheem` (تفخيم)
    * `tarqeeq` (ترقيق)
  - Default Value: `tarqeeq` (ترقيق)
  - More Info: Emphasis and softening of the letter 'Ra' in the word {يسر} in Surah Al-Fajr when pausing (waqf). This refers to the recitation rules regarding whether the letter "Ra" (ر) in the word "يسر" is pronounced with emphasis (`tafkheem`) or softening (`tarqeeq`) when pausing at this word in Surah Al-Fajr. `wasl`: means not pausing so we only have one way (tarqeeq of Raa)

- **meem_mokhfah** (هل الميم مخفاة أو مدغمة)
  - Values:
    * `meem` (ميم)
    * `ikhfaa` (إخفاء)
  - Default Value: `ikhfaa` (إخفاء)
  - More Info: This is not a **standard** Hafs way but a disagreement between **scholars** in our century on how to **pronounce Ikhfa** for meem. Some **scholars** do full merging (إدغام) and the others open the **lips** a little bit (إخفاء). We did not want to add this, but some of the best reciters disagree about this.

A.5. qdat_bench Dataset

qdat_bench is a comprehensive benchmark dataset for evaluating model performance in processing Quranic audio recordings with focus on Tajweed rules. This dataset builds upon the original qdat dataset (Osman et al., 2021) after extensive reannotation and enhancement to include all Tajweed rules and QPS 3 (complete phoneme-level annotations and 10 sifat (characteristic) levels).

The enhanced dataset provides F1 and MSE metrics for Tajweed rules, enabling researchers to compare different representations and approaches to Tajweed rule detection. This comprehensive annotation framework makes qdat_bench particularly valuable for advancing research in automated Quranic pronunciation assessment and error detection.

The dataset addresses several limitations of the original qdat collection. The original dataset suffered from incomplete coverage of all Tajweed rules, with only partial implementation of the comprehensive rule set required for thorough evaluation. Additionally, the original collection contained multiple reciters recording the same verse (each reciter records the same verse 10 times), creating significant redundancy and potential bias in evaluation. In contrast, qdat_bench selects a single recording randomly for every reciter. Finally, the original dataset lacked comprehensive phoneme-level and sifat-level annotations, limiting its utility for fine-grained analysis of pronunciation patterns and characteristics.

qdat_bench contains 159 samples focusing on the verse: قَالُواْ لَا عِلْمَ لَنَآ إِنَّكَ أَنتَ عَلَّٰمُ ٱلْغُيُوبِ from Surah Al-Ma'idah (5:109), providing a concentrated evaluation of key Tajweed rules.

A.5.1. Data Structure

The dataset encompasses several key components designed for comprehensive analysis of Quranic recitation patterns. Each entry contains an audio file recorded in mono channel format, a unique identifiers for each element, reciter gender (male or female) and age, enabling analysis of pronunciation patterns across different population segments. The phonetic_transcript and sifat fields contain the complete transcription in Quran Phonetic Script (QPS) as described in Section 3, Along with Tajweed rules columns to enable different Tajweed representations benchmark on qdat_bench. The dataset was annotated by a single hafiz of the Moshaf with Tajweed knowledge. Below is description of Tajweed rules found on our benchmark:

Madd (Prolongation) Rules: The dataset includes comprehensive annotations for various types of Madd rules, which are fundamental to proper Quranic recitation. The normal Madd rules are captured through several specific metrics: `qalo_alif_len` measures the length of normal Madd alif in the word قالوا on a scale of 0-8, while `qalo_waw_len` similarly measures the normal Madd waw in the same word. Additional normal Madd measurements include `laa_alif_len` for the normal Madd alif in لا and `allam_alif_len` for the normal Madd alif in علام. The Separate Madd is measured through `separate_madd`, which captures the length for the phrase لنا إنك (0-8). Finally, Madd Aared, is evaluated using `madd_aared_len`, measuring prolongation before sukoon (0-8), which typically exhibits the highest variability in implementation.

Ghunnah (Nasalization) Rules: Ghunnah rules are systematically annotated to capture the nasalization characteristics essential for proper Quranic pronunciation. The `noon_moshaddadah_len` metric evaluates the length of noon moshaddadah in the word إنَّك using a binary classification system where 0 indicates partial nasalization and 1 represents complete implementation. Similarly, the `noon_mokhfah_len` measures the Ikhfaa pronunciation in أنت through a three-tiered system: 0 represents a clear noon pronunciation, 1 indicates partial Ikhfaa implementation, and 2 denotes complete Ikhfaa execution.

Qalqalah (Echo) Rules: The Qalqalah rule is captured through the `qalqalah` metric, which identifies the presence or absence of the echo characteristic in the word الغيوب. This binary classification system uses 0 to indicate no Qalqalah implementation and 1 to denote proper Qalqalah execution.

A.5.2. Dataset Statistics

The qdat_bench dataset comprises 159 carefully selected samples. The demographic distribution reflects a diverse participant pool, with 120 female reciters representing 75.5% of the dataset and 39 male reciters accounting for 24.5%. The age diversity across various age groups provides comprehensive coverage of different learning stages and pronunciation patterns as shown in Figure 12. The benchmark has various types of error: 106 reciters have one or more errors while 53 have complete correct recitations as shown in Figure 13. Finally Figure 14 shows the errors per every Tajweed rule as red represents errors and green with correct recitation.

A.5.3. Evaluation Results

The detailed evaluation results on qdat_bench are presented in the following tables, showing performance across different Tajweed rule categories and metrics.

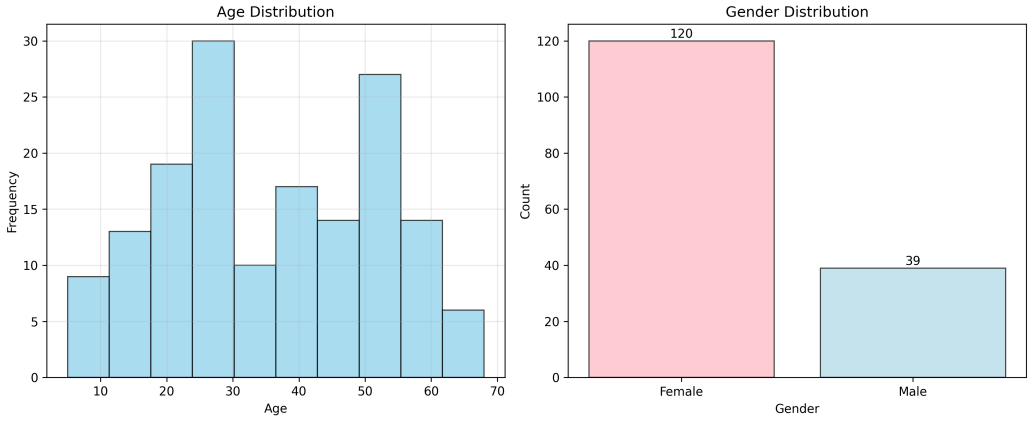

Figure 12. Age and gender distribution of qdat_bench reciters, showing diverse demographic coverage with 75.5% female and 24.5% male participants across different age groups.

A.5.4. Usage

The dataset can be loaded using the Hugging Face datasets library:

```
from datasets import load_dataset
ds = load_dataset('obadx/qdat_bench')
print(ds['train'][0])  # Display first sample
```

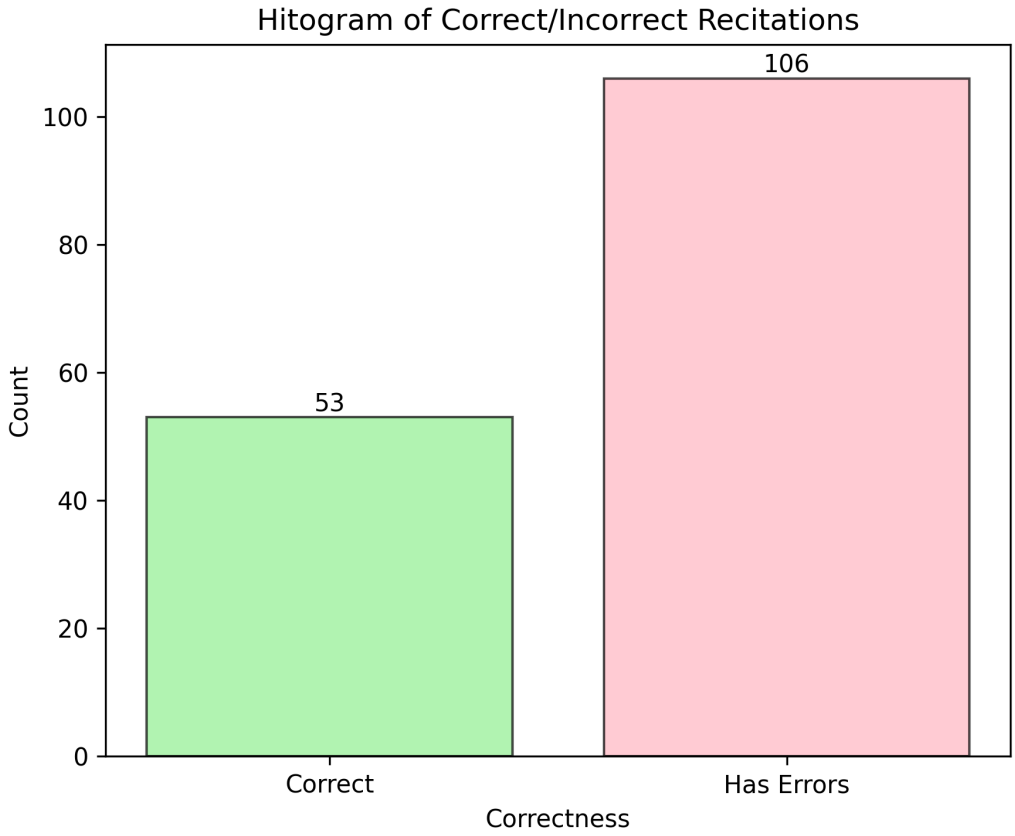

Figure 13. Distribution of recitation correctness across different Tajweed rules in the benchmark: red represents a reciter has one or more errors and green with correct recitation.

Table 10. Phoneme Set (43 Symbols)

Phoneme Name	Symbol
hamza	ء
baa	ب
taa	ت
thaa	ث
jeem	ج
haa_mohmala	ح
khaa	خ
daal	د
thaal	ذ
raa	ر
zay	ز
seen	س
sheen	ش
saad	ص
daad	ض
taa_mofakhama	ط
zaa_mofakhama	ظ
ayn	ع
ghyn	غ
faa	ف
qaf	ق
kaf	ك
lam	ل
meem	م
noon	ن
haa	ه
waw	و
yaa	ي
alif	ا
yaa_madd	ىٓ
waw_madd	وٓ
fatha	ـَ
dama	ـُ
kasra	ـِ
fatha_momala	ـٰ
alif_momala	ـٱ
hamza_mosahala	ـٔ
qlqla	ـۜ
noon_mokhfah	ـۢ
meem_mokhfah	ـۣ
sakt	ـۡ
dama_mokhtalasa	ـۥ

Table 11. Sifat Set (10 Attributes)

Sifat (English)	Sifat (Arabic)	Available Attributes (English)	Available Attributes (Arabic)
hams_or_jahr	الهمس أو الجهر	hams, jahr	همس, جهر
shidda_or_rakhawa	الشدة أو الرخاوة	shadeed, between, rikhw	شديد, بين بين, رخو
tafkheem_or_taqeeq	التفخيم أو الترقيق	mofakham, moraqaq, low_mofakham	مفخم, مرقق, أدنى المفخم
itbaq	الإطباق	monfateh, motbaq	منفتح, مطبق
safeer	الصفير	safeer, no_safeer	صفير, لا صفير
qalqla	القلقلة	moqalqal, not_moqalqal	مقلقل, غير مقلقل
tikraar	التكرار	mokarar, not_mokarar	مكرر, غير مكرر
tafashie	التفشي	motafashie, not_motafashie	متفشي, غير متفشي
istitala	الاستطالة	mostateel, not_mostateel	مستطيل, غير مستطيل
ghonna	الغنة	maghnoon, not_maghnoon	مغنون, غير مغنون

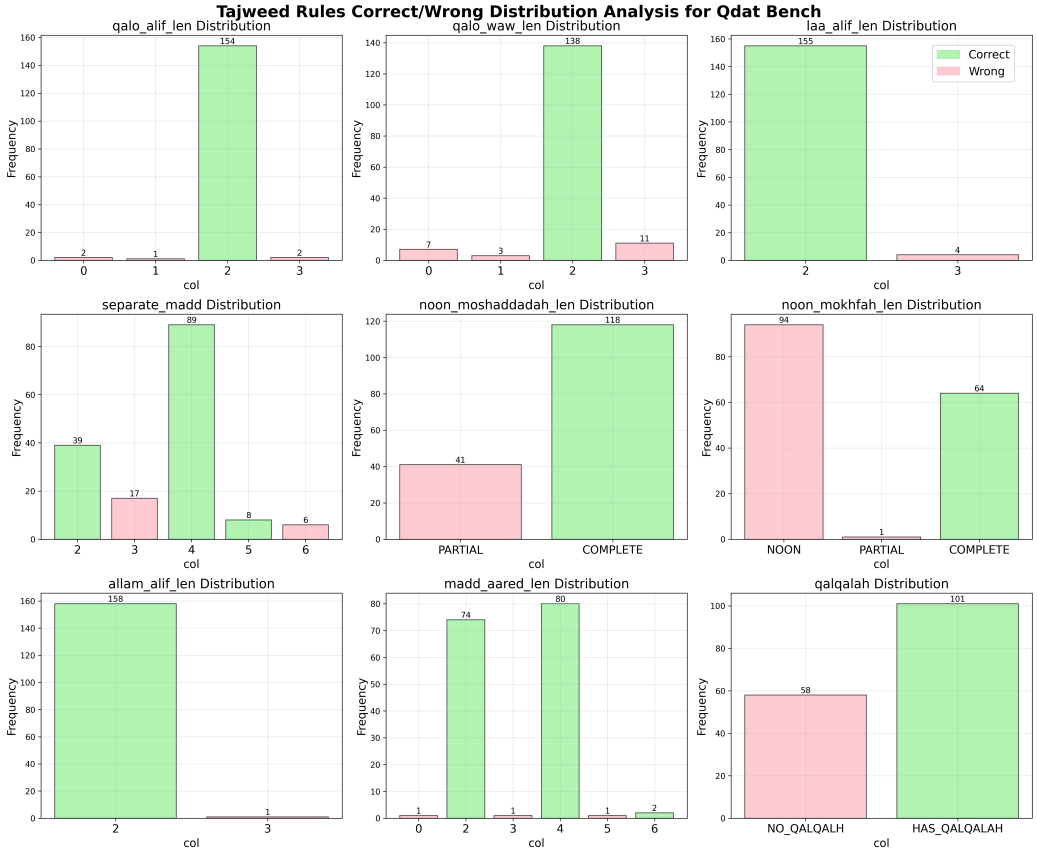

Figure 14. Tajweed rules coverage histogram showing the frequency and diversity of rules evaluated in the benchmark dataset, red bars represents errors and green with correct recitation.

Table 12. Detailed qdat_bench Speech Metrics. Note: Original refers to evaluation on the full test set without bootstrapping.

Metric	Original	Bootstrap (n=10,000)
per_phonemes	0.0578	0.0603 ± 0.0053
per_hams_or_jahr	0.0170	0.0172 ± 0.0033
per_shidda_or_rakhawa	0.0310	0.0315 ± 0.0050
per_tafkheem_or_taqeeq	0.0224	0.0227 ± 0.0044
per_itbaq	0.0117	0.0119 ± 0.0032
per_safeer	0.0103	0.0103 ± 0.0028
per_qalqla	0.0112	0.0112 ± 0.0026
per_tikraar	0.0126	0.0128 ± 0.0027
per_tafashie	0.0156	0.0159 ± 0.0035
per_istitala	0.0093	0.0094 ± 0.0026
per_ghonna	0.0140	0.0139 ± 0.0036
average_per	0.0194	0.0197 ± 0.0032

Table 13. qdat_bench Madd Rules Performance (RMSE). Golden values are for madd are: (2 for normal madd, 2 or 4 for separate madd, and 2, 4, or 6 for aared madd. Note: Original refers to evaluation on the full test set without bootstrapping.

Madd Rule	Original	Bootstrap (n=10,000)
qalo_alif_len (normal)	0.4486	0.4504 ± 0.0676
qalo_waw_len (normal)	0.4556	0.4609 ± 0.0561
laa_alif_len (normal)	0.4044	0.4065 ± 0.0713
separate_madd	0.6868	0.6758 ± 0.0582
allam_alif_len (normal)	0.5494	0.5575 ± 0.0749
madd_aared_len	1.0340	1.0251 ± 0.0724
Average Madd RMSE	0.5965	0.5960 ± 0.0374

Table 14. qdat_bench Noon Moshaddadah Performance. Note: Original refers to evaluation on the full test set without bootstrapping.

Metric	Partial	Complete	Average
Recall	0.6585	1.0000	0.8293
Precision	1.0000	0.8939	0.9470
F1 Score	0.7941	0.9440	0.8691
Accuracy		0.9119	
Bootstrap (n=10,000)			
Recall	0.6604 ± 0.0721	1.0000 ± 0.0000	0.8302 ± 0.0361
Precision	1.0000 ± 0.0000	0.8928 ± 0.0268	0.9464 ± 0.0134
F1 Score	0.7932 ± 0.0531	0.9431 ± 0.0150	0.8682 ± 0.0324
Accuracy		0.9113 ± 0.0222	

Table 15. qdat_bench Noon Mokhfah Performance. Note: Original refers to evaluation on the full test set without bootstrapping.

Metric	Noon	Partial	Complete
Recall	0.4681	0.0000	0.9844
Precision	1.0000	0.0000	0.5676
F1 Score	0.6377	0.0000	0.7200
Average F1		0.4526	
Accuracy		0.6730	
Bootstrap (n=10,000)			
Recall	0.4721 ± 0.0539	0.0000 ± 0.0000	0.9820 ± 0.0165
Precision	1.0000 ± 0.0000	0.0000 ± 0.0000	0.5617 ± 0.0458
F1 Score	0.6395 ± 0.0504	0.0000 ± 0.0000	0.7134 ± 0.0371
Average F1		0.4510 ± 0.0259	
Accuracy		0.6716 ± 0.0376	

Table 16. qdat_bench Qalqalah Performance. Note: Original refers to evaluation on the full test set without bootstrapping.

Metric	**No Qalqalah**	**Has Qalqalah**
Recall	0.9655	0.9505
Precision	0.9180	0.9796
F1 Score	0.9412	0.9648
Macro F1	0.9530	
Accuracy	0.9560	
Bootstrap (n=10,000)		
Recall	0.9611 ± 0.0260	0.9532 ± 0.0217
Precision	0.9244 ± 0.0339	0.9765 ± 0.0156
F1 Score	0.9419 ± 0.0219	0.9645 ± 0.0136
Macro F1	0.9532 ± 0.0174	
Accuracy	0.9562 ± 0.0164	