

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

Abdullah Abdelfttah

Computer and Systems Engineering  
Faculty of Engineering Ain Shams University  
Cairo, Egypt  
abdullahmlyossef@gmail.com

Mahmoud I. Khalil

Computer and Systems Engineering  
Faculty of Engineering Ain Shams University  
Cairo, Egypt  
mahmoud.khalil@eng.asu.edu.eg

Hazem Abbas

Computer and Systems Engineering  
Faculty of Engineering Ain Shams University  
Cairo, Egypt  
hazem.abbas@eng.asu.edu.eg

**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, 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, Transcript verification via our novel Tasmeea algorithm; (2) 850+ hours of audio (300K annotated utterances); (3) qdat\_bench benchmarks 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): Encodes Arabic letters with short/long vowels. (Sifat level): Encodes articulation characteristics of every phoneme. We further include comprehensive modeling with our novel multi-level CTC Model which achieved 0.21% and 1.94% average Phoneme Error Rate (PER) on the testset and qdat\_bench respectively, with 75.8% Tajweed F1 score and 84.7% accuracy on the benchmark. We release all code, data, and models as open-source: <https://obadx.github.io/quran-muaalem/en/>

**Index Terms**—Mispronunciation Detection Model, Arabic Natural Language Processing, End-to-end Models

## I Introduction

Assessing pronunciation is not a simple task [1], as it does not only rely on pronouncing phonemes correctly but also involves other factors like intonation, prosody, and stress. Does learning these mean one is done? No—other factors include fluency and completeness [1]. However, the Holy Quran presents unique characteristics: it is among the easiest spoken texts to learn despite containing special phonemes absent in other languages.

The pronunciation of the Holy Quran is governed by rigorously strict rules formally defined by ancient Muslim scholars since the 6th century. Despite their beauty and precision,

these rules have not been comprehensively digitized (to our knowledge) for Quranic pronunciation assessment.

Although RDI pioneered computer-aided Quranic instruction [2], they neither disclosed their phoneticization process nor released data/models. Consequently, new research must start from basics: defining phoneticization, data, and models. To bridge this gap, we introduce:

- **A Phonetizer**: Encodes *all* Tajweed rules and articulation attributes (*Sifat*) defined by classical scholars, except *Ishmam* (إشمام)
- **A 98% automated pipeline**: Generates highly accurate datasets from expert recitations
- **A dataset**: ~300K annotated utterances (850+ hours)
- **qdat\_bench**: Benchmarks phonemes, diacritization, and Tajweed rules (Ghunnah, Qalqalah, Madd) on real recitation errors containing 159 samples
- **Integration**: Our multi-level CTC model proves the Quranic phonetic script is learnable (0.21% average phoneme error rate)

The paper is organized as follows:

- **Related Work**: Expands on strengths/weaknesses of prior research
- **Quran Phonetic Script**: Introduces our two-level script: **phonemes** and **Sifat** (10 attributes → 11 total levels)
- **Data Pipeline**: Stages include:
  - 1) Digitized Quran script as foundation
  - 2) *Hafs* methodology criteria
  - 3) Expert recitation collection
  - 4) Segmentation at pause points (وقف)
  - 5) Segment transcription
  - 6) Validation via *Tasmee* (تسميع) algorithm
- **Modeling**: Demonstrates learnability of the phonetic script
- **Results**: Analysis of outcomes
- **Limitations & Future Work**: Next research directions
- **Conclusion**: Summary of contributions
- **Appendix**: Details on *Mushaf* attributes and algorithms

## II Related Work

### II-A Quran Pronunciation Datasets

We discuss the most important datasets here. *everyayah*<sup>1</sup> is the largest openly available dataset with 26 complete *Mushafs* segmented and annotated by Ayah by experts like Al Hossary and non-experts such as Fares Abbad. Qdat [3] contains 1509 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 [4], and [5] due to being open-source. The Tarteel v1 dataset [6] consists of 25K utterances with diacritics and no Tajweed rules. The latter is the Tarteel<sup>2</sup> private dataset, a massive 9K-hour collection annotated with diacritics without Tajweed rules. The most recent benchmark is IqraaEval [7], which presents a test set of 2.2 hours from 18 speakers, but uses Modern Standard Arabic (MSA) without Tajweed rules.

### II-B Quran Pronunciation Models

To our knowledge, the first work addressing automated pronunciation assessment for the Holy Quran is RDI [2], which built a complete system for detecting pronunciation errors. The work does not specify which errors were included or excluded but mentions testing Qalqala, Idgham, and Iqlab rules. It also omits details on Quranic word phoneticization. Subsequent work continued with [8] and [9], using Deep Neural Networks (DNNs) to replace HMMs and improve the system. Many studies rely on modeling phoneme duration for duration-dependent rules like Madd and Ghunna, e.g., [10], [11], but use limited datasets and focus on specific verses rather than the entire Quran. Others concentrate on detecting specific rules like Qalqala [4] or Ghunna and Madd [5], [12]. However, most efforts except RDI work train on small-scale datasets from specific Quranic chapters.

At this point, Tarteel emerges; though lacking Tajweed rules, they built a robust ASR system for diacritized character detection. They developed a crowd-sourced dataset [6] of 25K utterances (68 hours), later extended via application users to 9K hours of private annotated data. The work most aligned with our vision of detecting all error types (including Tajweed and *Sifat*/articulation attributes) is [13]. Although it relies on HMMs and minimal data, it introduces a multi-level detection system: *Makhraj* (phoneme level) and Tajweed rules level.

### II-C Pretrained Speech Encoders with Self-Supervised Learning (SSL)

Speech pretraining began early [14] but was constrained by the sequential nature of Recurrent Neural Networks (RNNs) [15]. The rise of Transformers [16] facilitated greater GPU parallelization, enabling large-scale pretraining. BERT [17] using Masked Language Modeling (MLM) introduce large unsupervised pretraining which has better results on down stream tasks. This soon extended to speech with wav2vec [18] and

wav2vec2.0, which added product quantization [19]. Conformer later replaced vanilla Transformers for speech by integrating convolution [20]. Google’s Wav2Vec2-BERT [21] then applied MLM to speech. Finally, Facebook extended Wav2Vec2-BERT pretraining [22] to 4.5M hours (including 110K Arabic hours), ideal for low-resource language fine-tuning.

## III Quran Phonetic Script

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

Modern Standard Arabic (MSA) orthography cannot adequately represent Tajweed rules for error detection. For example, MSA cannot measure the precise length of Madd rules. Previous research (e.g., [23]) focused on single rules like Qalqalah. Our phonetic script addresses this limitation by capturing all Tajweed pronunciation errors except Ishmam (إشمام), which involves a visual mouth movement without audible output.

We based our script on classical Muslim scholarship rather than the International Phonetic Alphabet (IPA) for these reasons:

- 1) **Historical Precedence:** Muslim scholars from the 6th to 14th centuries rigorously defined Quranic errors centuries before modern phonetics emerged in the West.
- 2) **Scientific Foundation:** Scholars like Al-Khalil ibn Ahmad (6th century AH) systematically described articulations and attributes with remarkable accuracy comparable to modern phonetics [24].
- 3) **Pedagogical Relevance:** Learners’ errors align with classical definitions according to expert Quran teachers.

Following [25], Quran recitation errors fall into three categories:

- **Articulation Errors:** Incorrect pronunciation of phonemes
- **Attribute Errors:** Mistakes in letter characteristics (*Sifat al-Huruf*)
- **Tajweed Rule Errors:** Incorrect application of rules like Ghunnah, Madd, etc.

Our script comprehensively addresses all three aspects through main two output levels:

- **Phonemes Level:** Represents letters, vowels, and Tajweed rules
- **Sifat Level:** Consists of 10 levels representing articulation attributes for each phoneme

Refer to tables: VII VIII for Phonemes and Sifat levels.

Our script has some important characteristics:

- Normal Madd appears as consecutive madd symbols (e.g., 4-beat Madd: ||||)
- Madd al-Leen represented with multiple waw/yaa symbols

<sup>1</sup>everyayah.com

<sup>2</sup>tarteel.ai

- Stressed Ghunnah (e.g., النون المشددة) as three consecutive noon symbols (تنن)
- Ikhfa represented as three consecutive noon\_mokhfah (س) or meem\_mokhfah (ممم)
- Assimilation represented by doubling (e.g., مَنْ يَعْمَلْ → مَيِّيعَمَل)
- Sakin: No following symbol
- Imala: fatha\_momala and alif\_momala
- Rawm: dama\_mokhtalasa marker

Example: In table I shows how our phonetizer works. This example shows the phonetization of word (أَلْحَجُونِي) as a row by row: The first row shows conversion of (أ) to (ء) with its sifat in the row. following the second row. For the fourth row showing the madd lazem rule with 6 beats phonetized as 6 alifs (اااااا) same as the sixth row but with damma represented as 6 waw\_madd (ووووو) And for the fifth row we notice that we converted (ح) to (حْ) as we dissmple shadda. The last row shows the normal madd of yaa with two beats represented as two yadd\_madd (ءء).

### III-A Development Methodology

Our phonetization has two steps:

#### 1) Imlaei to Uthmani Conversion

We selected Uthmani script as our foundation because:

- Contains specialized Tajweed diacritics (Madd, Tasheel, etc.)
- Preserves pause rules critical for recitation (e.g., stopping on رحمت)

In order to do that, we created an annotation UI to manually annotate misaligned words in both scripts. For example II, after that, we developed an algorithm that relies on the annotations to convert Imlaei to Uthmani.

#### 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 VIII-B.

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

## IV 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* riwayat (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 for data collection. The steps are: (1) Choose a digitized Quran script as the project foundation. (2) Define criteria for *Hafs* methodology. (3) Collect expert recitations (4) Segment recitations at pause points (وقف) (5)

Transcribe segments. (6) Validate data through *Tasmee* (تسميع) Algorithm. (7) Develop Quran Phonetic Script.

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

### IV-A Choose a Digitized Version of the Holy Quran

The Quran has multiple digitized versions including Tanzil<sup>3</sup> and King Fahd Complex<sup>4</sup>. We chose Tanzil because:

- 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.

### IV-B Define Variant Criteria for Hafs

*Hafs* riwayat 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 [26], summarized in the following attributes in the Appendix section VIII-D.

### IV-C Collect Expert Recitations

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

Number of Moshaf Items	Number of Reciters
27	22
Total Hours	Total Size (GB)
893.1	48.48

Fig. 1. Database Collection Statistics

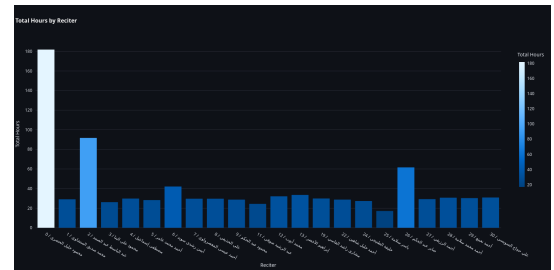


Fig. 2. Reciters Statistics

We developed a web GUI using Streamlit<sup>5</sup> that:

- Downloads and extracts metadata for each track
- Organizes data by Moshaf (each chapter as "001.mp3")
- Annotates Moshaf attributes

<sup>3</sup><https://tanzil.net>

<sup>4</sup><https://qurancomplex.gov.sa>

<sup>5</sup><https://streamlit.io/>

TABLE I

Examples of Uthmani to Phonetic Script Conversion with Sifat Attributes. This example shows the phonetization of word (أَمْحُجُو) as a row by row: The first row shows conversion of (أ) to (ʔ) with its sifat in the row. following the second row. For the forth row showing the madd lazem rule with 6 beats phonetized as 6 alifs (اااااا) same as the sixth row but with damma represented as 6 waw\_madd (وووووو) And for the fifth row we notice that we converted (ح) to (حْ) as we dissmple shadda. The last row shows the normal madd of yaa with two beats represented as two yadd\_madd (ءء)

Uthmani	Phonetic	H/J	S/R	T/T	Ith	Saf	Qal	Tik	Taf	Ist	Gho
أ	ʔ	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
م	ṡ	hams	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ح	ḥ	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
اااااا	اااااا	hams	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
وووووو	وووووو	jahr	shd	mrq	mnf	no	nql	nkr	ntf	nst	nmg
وووووو	وووووو	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg
ءء	ءء	jahr	btw	mrq	mnf	no	nql	nkr	ntf	nst	mg
ءء	ءء	jahr	rkh	mrq	mnf	no	nql	nkr	ntf	nst	nmg

## Attribute Abbreviations:

H/J: Hams/Jahr S/R: Shidda/Rakhawa T/T: Tafkheem/Taqeeq Ith: Itbaq  
Saf: Safeer Qal: Qalqla Tik: Tikraar Taf: Tafashie Ist: Istitala Gho: Ghonna

## Value Abbreviations:

shd: shadeed rkh: rikhw btw: between mrq: mraqaq  
mof: mofakham mnf: monfateh mtb: motbaq no: no safeer  
nql: not\_moqlalqal nkr: not\_mokarar ntf: not\_motafashie  
nst: not\_mostateel nmg: not\_maghnoon mg: maghnoon

TABLE II

Example of misalignment between Uthmani and Imlaey Scripts

Imlaey Script	Uthmani Script
يَا أَيُّهَا الْمَدِينَةُ	يَا أَيُّهَا الْمَدِينَةُ

## IV-D 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 [27] and PyAnnotate [28]. Poor Quran-specific performance led us to develop a custom segmenter by fine-tuning Wav2Vec2-BERT [22] for frame-level classification.

*IV-D1 Preparing Segmenter Data:* We selected mosahf compatible with SileroVAD v4, using EveryAyah<sup>6</sup> (pre-segmented by ayah) as ground truth. After tuning parameters per Moshaf:

- Threshold
- Minimum silence duration (merges segments)
- Minimum speech duration (discards short segments)
- Padding (added at segment boundaries)

*IV-D1a Data Augmentation:* Using the Audiomentations [29] library, we replicated SileroVAD's noise setup on 40% of samples, adding:

- TimeStretch (0.8x-1.5x) to simulate recitation speeds
- Sliding window truncation (1-second windows) for long samples instead of exclusion

*IV-D2 Training Segmenter:* We fine-tuned Wav2Vec2-BERT for frame classification (1 epoch):

Results of our segmenter on unseen mosahf in table IV-D2:

TABLE III

Dataset Statistics per Moshaf

Moshaf ID	Hours	Length
0.0	28.47721296	9133
0.1	40.31257093	10764
0.2	49.46541671	9971
0.3	37.18758118	12604
1.0	28.40784367	10939
2.0	51.04665234	9942
2.1	30.02847051	10394
3.0	25.19377593	10444
4.0	29.12333379	10994
5.0	28.01777693	11482
6.0	39.38568468	12435
7.0	28.25627201	9907
8.0	30.85935158	10330
9.0	27.95178738	10642
11.0	24.00685052	10363
12.0	33.42429862	9880
13.0	33.99108879	9377
19.0	30.11410843	11278
22.0	28.10947704	10332
24.0	28.51243509	9868
25.0	16.92910042	7922
26.0	30.44461112	11565
26.1	32.71190443	11850
27.0	28.05097968	11213
28.0	31.05318768	10535
29.0	27.78900316	11061
30.0	29.14366461	11312
<b>Total</b>	<b>847.9944402</b>	<b>286537</b>

TABLE IV

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

<sup>6</sup><https://everyayah.com/>

#### IV-E Transcribe Segmented Parts

We employed Tarteel ASR [30] (Whisper fine-tuned on Quranic recitations [31]). To handle its 30-second limit, we used sliding window truncation (10-second windows), with verification in the next step.

#### IV-F Verification of Segmentation and Transcription

**Segmentation Verification:** Manual inspection of 50-75 random samples per Moshaf. Moshaf 25.0 was excluded due to poor segmentation.

**Transcription Verification:** *Tasmeea*-inspired algorithm: (1) Match segments to Quranic text. (2) Identify missing surah parts. (3) Manual correction.

Refer to the *Tasmeea* Algorithm in the Appendix 1

After matching, we catalogued missing Quranic portions per surah. Then correct transcription errors identified through the above process.

### V Modeling The Quran Phonetic Script

Our Quran Phonetic script has two outputs: *phonemes* and *sifat* (which has 10 attributes). We modeled this as follows: Imagine you are given an input speech utterance and want to output transcripts in Arabic, English, French, and German simultaneously. We implemented this as a speech encoder with a linear layer for each language. Replacing languages with our 11 levels (phonemes and the 10 *sifat*), we obtain 11 parallel transcription levels. We chose CTC loss [32] 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**.

We compute the loss by averaging all CTC losses for the 11 levels, assigning a weight of 0.4 to the phonemes level as it has the largest vocabulary size (43) compared to other levels.

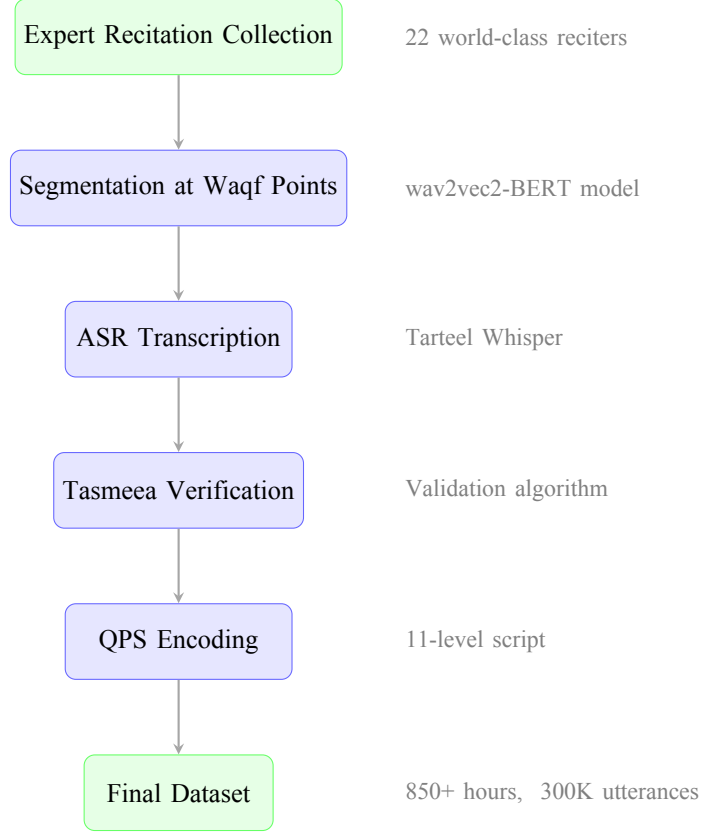


Fig. 4. The complete data preparation pipeline showing the six main stages from expert recitation collection to the final dataset generation. Each stage utilizes specialized models and algorithms to ensure high-quality annotations.

### Voice Activity Detection (VAD) Approaches

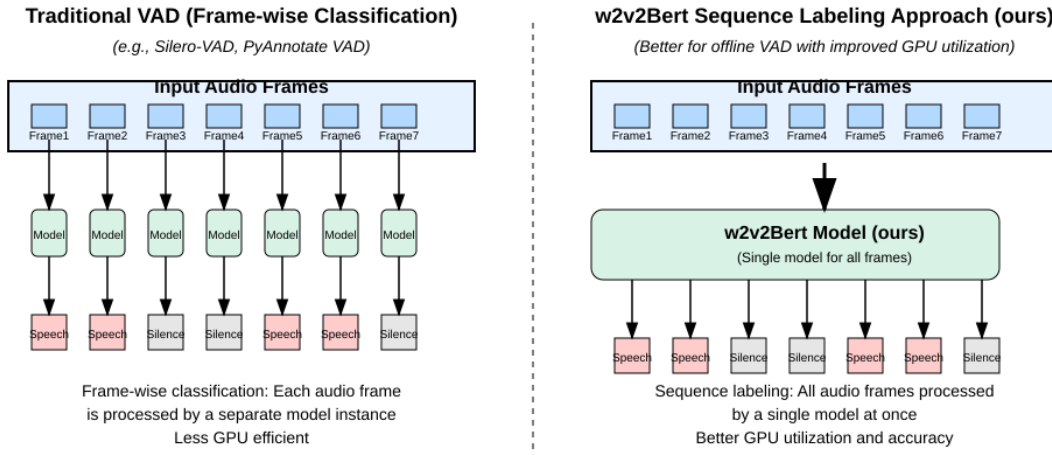


Fig. 3. VAD architecture vs. standard streaming models



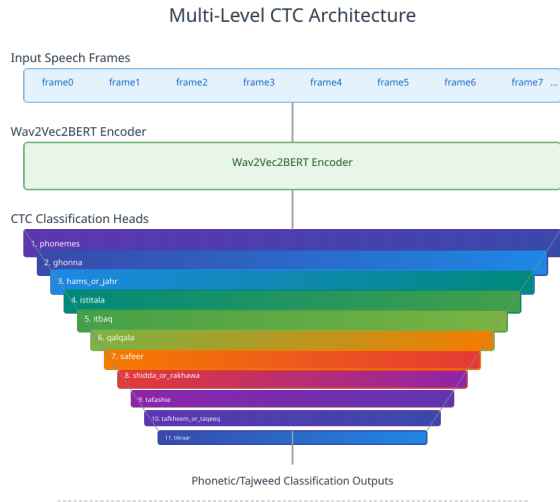


Fig. 5. Multi-level CTC loss Architecture composed of 11 Heads for every level and CTC loss for every level with weighted average loss

We fine-tuned Facebook’s Wav2Vec2-Bert [22] for a single epoch with a constant learning rate of  $5e-5$  with 64 batch size. We applied augmentations identical to Silero VAD [27] using the audiomentations library [29], with additional augmentations: TimeStretch and GainTransition. We filtered out samples longer than 30 seconds not due to model limitations, but for efficient GPU utilization - sacrificing only 3k samples out of 250k training samples.

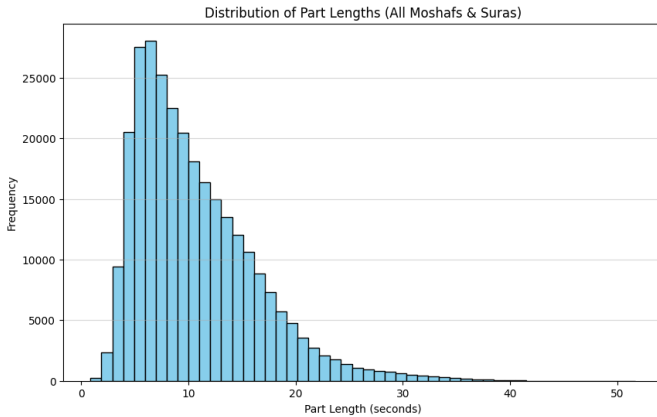


Fig. 6. Recitations lengths in seconds for the whole dataset

The training was done using an H200 GPU with 141 GB of GPU memory for 7 hours.

## VI Results

We trained on all available Mushaf datasets, reserving three Mushaf (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 V, while still demonstrating excellent overall performance that validates our multi-level CTC approach.

TABLE V

Test Results on Expert Quranic Recitations. Evaluation conducted on three Mushaf datasets (19.0, 29.0, 30.0) featuring expert male reciters with extensive Tajweed training, recorded under controlled acoustic conditions. The phonemes level demonstrates the highest error rate (0.543%) because it uses the largest vocabulary of 44 characters including padding, making it the most challenging classification task among all representation levels.

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_safer	0.00132
per_qalqla	0.00085
per_tikraar	0.0009
per_tafashie	0.0016
per_istitala	0.0008
per_ghonna	0.0013
average_per	<b>0.0021</b>

To evaluate our model’s performance on real recitation errors, we tested on our developed qdat\_bench<sup>7</sup> benchmark, which builds upon [3] after comprehensive reannotation and addition of all Tajweed rules including phonemes and 10 sifat levels. This enhanced benchmark provides F1 and MSE metrics for (F1 for: Noon Moshaddadah, Ikhfaa and Qalqlah. MSE for Madd lengths including: normal madd, separated madd and aared madd), enabling researchers to compare different representations of Tajweed rules. The results, shown in Table VI, demonstrate that despite being trained exclusively on expert male reciters, our model achieves remarkable performance on female recitations (120 samples) and overall error detection.

qdat\_bench shows a higher PER of 0.058 (5.8%) on authentic learner recordings, which is expected given the increased complexity of detecting real pronunciation errors and handling variability in learner execution of Tajweed rules. But (5.5%) remains very acceptable despite trained on expert recitations only. Despite this performance gap, the model maintains excellent Tajweed F1 scores (75.8% average) on learner data, demonstrating robust generalization capabilities for practical educational applications. The aggregate Tajweed F1 score of 0.758 represents the mean performance across three key rules: Noon Moshaddadah (0.869), Ikhfaa (0.453), and Qalqalah (0.953). Similarly, the average Madd RMSE of 0.596 encompasses performance across normal madd rules (0.464), separate madd (0.687), and aared madd (1.034).

<sup>7</sup>Available at: [https://huggingface.co/datasets/obadx/qdat\\_bench](https://huggingface.co/datasets/obadx/qdat_bench)

TABLE VI

qdat\_bench Results - Comprehensive Evaluation on Authentic Learner Mistakes. This benchmark builds upon the original qdat dataset [3] through extensive expert reannotation that systematically labeled each audio segment across multiple dimensions: complete phoneme-level transcription, 10 sifat characteristics, and comprehensive Tajweed rule classifications. The enhanced dataset contains 159 samples (120 female, 39 male reciters) focusing on Quranic verse from Surah Al-Ma'idah (5:109), providing a concentrated evaluation of key pronunciation challenges.

Aggregate Metrics	Value
per_phonemes	0.058
avg_per	0.019
avg_tajweed_f1	0.758
avg_tajweed_acc	0.847
avg_madd_rmse	0.596
Tajweed Rules F1	Value
Noon Moshaddadah	0.869
Ikhfaa (Noon Mokhfah)	0.453
Qalqalah	0.953
Madd Rules RMSE	Value
Normal Madd (5 rules)	0.464
Separate Madd	0.687
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. This challenge affects both human perception and automated detection, as few reciters can reliably distinguish Ikhfaa when it is recited with characteristics similar to noon moshaddah. The model's performance on this rule reflects the inherent difficulty in detecting subtle nasalization differences. Detailed analysis of these patterns and methodology considerations are provided in the appendix VIII-E.

Notably, despite being trained exclusively on male expert reciters, our model demonstrates strong generalization capability by achieving 75.8% Tajweed F1 score and 84.7% accuracy on female recitations in qdat\_bench, highlighting the robustness of our approach for real-world deployment where diverse learner populations are expected. This performance on authentic learner mistakes validates the practical applicability of our Quranic phonetic script and multi-level CTC architecture. The comprehensive nature of qdat\_bench provides a robust foundation for evaluating Quranic pronunciation models.

## VII Limitations and Future Work

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

## VIII 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 850+ hours of annotated audio data with ~300K 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.

## Acknowledgment

We express our profound gratitude to several individuals and organizations: 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<sup>8</sup> for providing access to high-performance computing resources and facilitating data processing. Special appreciation goes to Engineer Khaled Bahaa for his assistance with paying method for GPUs.

## References

- [1] Y. E. Kheir, A. Ali, and S. A. Chowdhury, "Automatic pronunciation assessment—a review," *arXiv preprint arXiv:2310.13974*, 2023.
- [2] M. Sherif, A. Samir, A. Khalil, and R. Mohsen, "Enhancing usability of capl system for quran recitation learning," *INTERSPEECH*, 2007.
- [3] H. M. Osman, B. S. Mustafa, and Y. Faisal, "Qdat: a data set for reciting the quran," *International Journal on Islamic Applications in Computer Science And Technology*, vol. 9, no. 1, pp. 1–9, 2021.
- [4] D. Omran, S. Fawzi, and A. Kandil, "Automatic detection of some tajweed rules," in *2023 20th Learning and Technology Conference (LT)*, 2023, pp. 157–160.
- [5] D. Shaikhmetov, G. Gimaletdinova, K. Momunov, and S. Cankurt, "Evaluation of the pronunciation of tajweed rules based on dnn as a step towards interactive recitation learning," *arXiv preprint arXiv:2503.23470*, 2025.
- [6] H. I. Khan, A. Abid, M. M. Moussa, and A. Abou-Allaban, "The tarteel dataset: crowd-sourced and labeled quranic recitation," 2021.
- [7] Y. E. Kheir, O. Ibrahim, A. Meghanani, N. Almarwani, H. O. Toyin, S. Alharbi, M. Alfadly, L. Alkanhal, I. Selim, S. Elbatal *et al.*, "Towards a unified benchmark for arabic pronunciation assessment: Quranic recitation as case study," *arXiv preprint arXiv:2506.07722*, 2025.
- [8] S. M. Abdou and M. Rashwan, "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)*. IEEE, 2014, pp. 543–550.
- [9] M. Al-Marri, H. Raafat, M. Abdallah, S. Abdou, and M. Rashwan, "Computer aided qur'an pronunciation using dnn," *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 5, pp. 3257–3271, 2018.
- [10] A. Mohammed, M. S. B. Sunar, and M. S. H. Salam, "Recognition of holy quran recitation rules using phoneme duration," in *International Conference of Reliable Information and Communication Technology*. Springer, 2017, pp. 343–352.
- [11] A. M. A. Alqadasi, A. M. Zeki, M. S. Sunar, M. S. B. H. Salam, R. Abdulghafor, and N. A. Khaled, "Improving automatic forced alignment for phoneme segmentation in quranic recitation," *IEEE Access*, vol. 12, pp. 229–244, 2023.
- [12] Y. S. Alsahafi and M. Asad, "Empirical study on mispronunciation detection for tajweed rules during quran recitation," in *2024 6th International Conference on Computing and Informatics (ICCI)*, 2024, pp. 39–45.

<sup>8</sup><https://hpc.bibalex.org/>

- [13] B. Putra, B. T. Atmaja, and D. Prananto, "Developing speech recognition system for quranic verse recitation learning software," *IJID (International Journal on Informatics for Development)*, vol. 1, no. 2, pp. 1–8, 2012.
- [14] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.
- [15] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.
- [16] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.
- [17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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.
- [18] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.
- [19] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [20] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.
- [21] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "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)*. IEEE, 2021, pp. 244–250.
- [22] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, M. Duppenhaler, P.-A. Duquenne, B. Ellis, H. Elsahar, J. Haaheim *et al.*, "Seamless: Multilingual expressive and streaming speech translation," *arXiv preprint arXiv:2312.05187*, 2023.
- [23] D. Omran, S. Fawzi, and A. Kandil, "Automatic detection of some tajweed rules," in *2023 20th Learning and Technology Conference (L&T)*. IEEE, 2023, pp. 157–160.
- مخرج الحروف الصالح وصفاتها عند الخليل بن أحمد الفراهيدي قراءة في ضوء "هيلجيمز", مجلة جامعة الأمير عبد القادر للعلوم الإسلامية "الدرس اللساني الحديث", vol. 31, pp. 165–190, 02 2023.
- 2021, دار الغوثاني للدراسات القرآنية. التجويد المصور, أموي 2553 سويد
- مكتبة ومطبعة مصطفى البابي. صريح النص في الكلمات المختلفة فيها عن حفص. علي 26 فبراير
- المؤلف: علي الضياع (توفي 0831 هـ), 1380 هـ, الحلي وأولاده
- [27] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier," <https://github.com/snakers4/silero-vad>, 2024.
- [28] A. Plaquet and H. Bredin, "Powerset multi-class cross entropy loss for neural speaker diarization," in *Proc. INTERSPEECH 2023*, 2023.
- [29] I. Jordal and Contributors, "Audiomentations: A python library for audio data augmentation," <https://github.com/iver56/audiomentations>, 2025.
- [30] T. AI, "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/>.
- [31] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.
- [32] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning (ICML 2006)*. ACM, 2006, pp. 369–376.

TABLE VII  
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	ـ
zaa_mofakhama	ـ
ayn	ع
ghyn	غ
faa	ف
qaf	ق
kaf	ك
lam	ل
meem	م
noon	ن
haa	ه
waw	و
yaa	ي
alif	أ
yaa_madd	ء
waw_madd	و
fatha	و
dama	
kasra	ـ
fatha_momala	
alif_momala	هـ
hamza_mosahala	أ
qlqla	ج
noon_mokhfah	ن
meem_mokhfah	م
sakt	س
dama_mokhtalasa	,

## Appendix

### VIII-A Quran Phoneme Script Vocabulary

### VIII-B Uthmani to Phonetic Conversion Operations

The 26 sequential phonetization operations:

- 1) **DisassembleHrofMoqatta** (تفكيك حروف مقطعة): Separates Quranic initials (e.g., الم, ال) into individual letters.
- 2) **SpecialCases** (حالات خاصة): Handles special words like يبسط that have different pronunciation forms defined in MoshafAttributes.



TABLE VIII  
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, رخو
tafkheem_or_taqeeq	التفخيم أو الترقيق	mofakham, moraqaq	مفخم, مرقق
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

- 3) **BeginWithHamzatWasl** (البدء بهمزة الوصل): Processes words starting with connecting hamza (أ) 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 يَلْقَظُ, 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 يَحْيَى 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 (أ) in non-initial positions.
- 10) **RemoveSkoonMostadeer** (حذف الحرف الذي فوق سكون): Eliminates letters with circular sukoon diacritics like alif in جَمْعُوْا. (مستدير).
- 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 ت (taa marbuta) to ت or ه based on context, and converts final ت 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) **ItiqaaAlsaknan** (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 (ي).
- 25) **Qalqla** (قلقلة): Adds echoing effect to ق, ط, ب, ج, د letters with sukoon.
- 26) **RemoveRasHaaAndShadda** (إزالة رأس الحاء علامة السكون): Deletes sukoon diacritic marks.

**Algorithm 1** Tasmeea Algorithm

---

**Require:**  $text\_segments = [s_1, s_2, \dots, s_n]$ ,  $sura\_idx$ ,  
 $overlap\_words = 6$ ,  $window\_words = 30$ ,  
 $acceptance\_ratio = 0.5$ , flags for special phrases

**Ensure:** List of tuples ( $match, ratio$ ) per segment

- 1:  $aya \leftarrow 1$  {Start at first verse}
- 2:  $penalty \leftarrow 0$
- 3: **for** each segment  $s_i$  in  $text\_segments$  **do**
- 4:  $norm\_text \leftarrow normalize(s_i)$  {Remove spaces/diacritics}
- 5:  $min\_win \leftarrow window\_words - 10$ ,  $max\_win \leftarrow window\_words + 10$
- 6:  $start\_range \leftarrow [-(overlap + penalty), (overlap + max(window\_words, max\_win) + penalty)]$
- 7: **if** first segment **and**  $include\_istiaatha$  **then**
- 8: Check istiaatha special case
- 9: **else if** last segment **and**  $include\_sadaka$  **then**
- 10: Check sadaka special case
- 11: **end if**
- 12:  $best\_ratio \leftarrow 0$ ,  $best\_match \leftarrow null$
- 13: **for** each start position  $p$  in  $start\_range$  **do**
- 14: **for** each window size  $w \in [min\_win, max\_win]$  **do**
- 15:  $c \leftarrow$  extract candidate at ( $aya, p, w$ )
- 16:  $dist \leftarrow edit\_distance(norm\_text, c)$
- 17:  $ratio \leftarrow \frac{1}{min(dist, |norm\_text|) / |norm\_text|}$
- 18: **if**  $ratio > best\_ratio$  **or** ( $ratio = best\_ratio$  **and**  $|p| < |best\_start|$ ) **then**
- 19: update  $best\_ratio$ ,  $best\_match$ ,  $best\_start$ ,  $best\_window$
- 20: **end if**
- 21: **end for**
- 22: **end for**
- 23: **if**  $best\_ratio < acceptance\_ratio$  **then**
- 24: output ( $null, best\_ratio$ )
- 25:  $penalty \leftarrow max\_win$
- 26:  $aya \leftarrow aya + 1$  {Default advance}
- 27: **else**
- 28: output ( $best\_match, best\_ratio$ )
- 29:  $aya \leftarrow aya + best\_start + best\_window$
- 30:  $penalty \leftarrow 0$
- 31: **end if**
- 32: **end for**
- 33: **Complexity:**  $O(N \cdot W \cdot L^2)$  { $N$ =segments,  $W$ =window size,  $L$ =segment length}

---

## VIII-D 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 (فوق 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 إلى 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 ”السما”.
- **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 (للاوا الساكنة والياء الساكنة وقبلها حرف مفتوح) ويجب أن يكون مقدار مد اللين أقل من أو يساوي مع العارض
- **ghonna\_lam\_and\_raa** (غنة اللام والراء)
  - 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 ومدّها حركتين)
    - \* wasl\_6 (فتح الميم ومدّها 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:
    - \* tahqeeq (تحقيق)
    - \* general\_sakt (سكت عام)
    - \* local\_sakt (سكت خاص)
  - Default Value: tahqeeq (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 {ماله هلك} بالحاقة)
  - Values:
    - \* sakt (سكت)
    - \* waqf (وقف)
    - \* idgham (إدغام)
  - Default Value: waqf (وقف)
  - More Info: The ways to recite the word {ماله 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 القرآن} 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} بالنقل)
  - 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 (آتاني). 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 البدء 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: {الذكرين} بموضعي)
  - 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: "ءائن", "ءالله", "ءالذكرين".
- **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 {يَلْهَثْ ذَلِكَ} in Surah Al-A'raf. waqf: means the reciter has paused on (يَلْهَثْ)
- **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\_tamna** (الإشمام والروم (الاختلاس) في قوله تعالى {لا تأمنا} على يوسف)
  - Values:
    - \* ishnam (إشمام)
    - \* rawm (روم)
  - Default Value: ishnam (إشمام)
  - More Info: The nasalization (ishnam) or the slight drawing (rawm) in the verse {لا تأمنا على يوسف}
- **harakat\_daaf** (حركة الضاد (فتح أو ضم) في قوله تعالى {ضعف})
  - 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: {سلاسل})
  - Values:
    - \* hadhf (حذف)
    - \* ithbat (إثبات)
    - \* wasl (وصل)
  - Default Value: wasl (وصل)
  - More Info: Affirmation and omission of the 'Alif' when pausing in the verse {سلاسل} in Surah Al-Insan. This refers to the recitation rule regarding whether the final "Alif" in the word "سلاسل" 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 (سلاسل) wasl means completing the recitation as normal without pausing, so recite it as (سلاسل وأغلا لا)
- **idgham\_nakhlukum** (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)
  - 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 مخففة 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.

### VIII-E 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 [3] after extensive reannotation and enhancement to include all Tajweed rules and QPS III (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: قَالَوْا لَا عِلْمَ لَنَا بِإِنَّكَ أَنْتَ عِلْمُ الْغُيُوبِ from Surah Al-Ma'idah (5:109), providing a concentrated evaluation of key Tajweed rules.

*VIII-E1 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 III, Along with Tajweed rules columns to enable different Tajweed representations benchmark on qdat\_bench. Below is description of Tajweed rules found on our benchmark:

*VIII-E1a 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 إِنَّكَ (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.

*VIII-E1b 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 إِنَّكَ 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 أَنْتَ through a three-tiered system: 0 represents a clear noon pronunciation, 1 indicates partial Ikhfaa implementation, and 2 denotes complete Ikhfaa

execution.

**VIII-E1c 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.

**VIII-E2 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 7. The benchmark has various types of error: 106 reciters have one or more errors while 53 have complete correct recitations as shown in Figure 8. Finally Figure 9 shows the errors per every Tajweed rule as red represents errors and green with correct recitation.

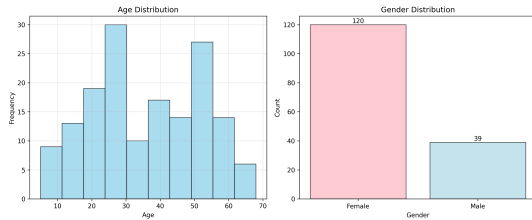


Fig. 7. Age and gender distribution of qdat\_bench reciters, showing diverse demographic coverage with 75.5% female and 24.5% male participants across different age groups.

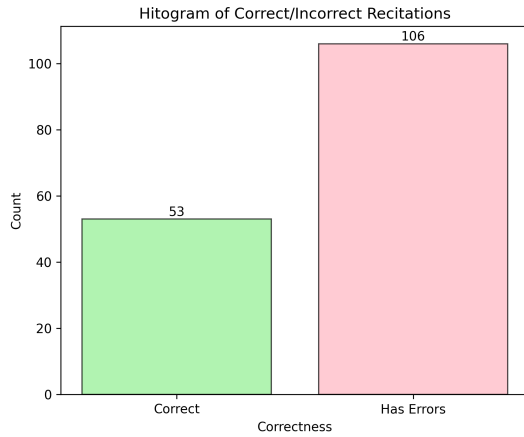


Fig. 8. Distribution of recitation correctness across different Tajweed rules in the benchmark: red represents a reciter has one or more errors and green with correct recitation.

**VIII-E3 Evaluation Results:** The detailed evaluation results on qdat\_bench are presented in the following tables, showing performance across different Tajweed rule categories and metrics.

**VIII-E4 Performance Analysis and Discussion:** The benchmark has many limitations: the size is very limited with

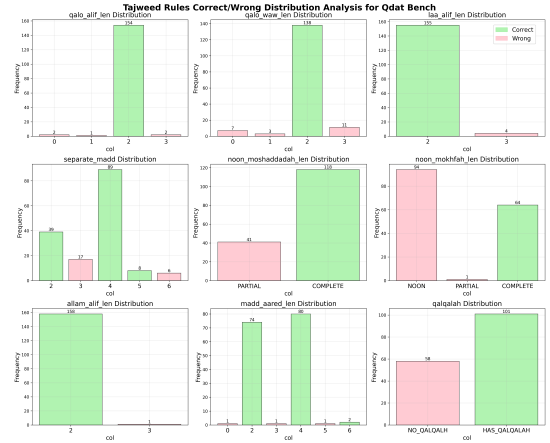


Fig. 9. Tajweed rules coverage histogram showing the frequency and diversity of rules evaluated in the benchmark dataset, red bars represents errors and green with correct recitation.

TABLE IX  
Detailed QDat Bench Speech Metrics

Metric	Value
per_phonemes	0.058
per_hams_or_jahr	0.017
per_shidda_or_rakhawa	0.031
per_tafkheem_or_taqeeq	0.022
per_itbaq	0.012
per_safeer	0.010
per_qalqla	0.011
per_tikraar	0.013
per_tafashie	0.016
per_istitala	0.009
per_ghonna	0.014
average_per	0.019

TABLE X  
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.

Madd Rule	RMSE
qalo_alif_len (normal)	0.449
qalo_waw_len (normal)	0.456
laa_alif_len (normal)	0.404
separate_madd (separate)	0.687
allam_alif_len (normal)	0.549
madd_aared_len	1.034
<b>Average Madd RMSE</b>	<b>0.596</b>

TABLE XI  
QDat Bench Noon Moshaddadah Performance

Metric	Partial	Complete	Average
Recall	0.659	1.000	0.829
Precision	1.000	0.894	0.947
F1 Score	0.794	0.944	0.869
Accuracy		0.912	

TABLE XII  
QDat Bench Noon Mokhfah Performance

<b>Metric</b>	<b>Noon</b>	<b>Partial</b>	<b>Complete</b>
Recall	0.468	0.000	0.984
Precision	1.000	0.000	0.568
F1 Score	0.638	0.000	0.720
<b>Average F1</b>		0.453	
<b>Accuracy</b>		0.673	

TABLE XIII  
QDat Bench Qalqalah Performance

<b>Metric</b>	<b>No Qalqalah</b>	<b>Has Qalqalah</b>
Recall	0.966	0.950
Precision	0.918	0.980
F1 Score	0.941	0.965
<b>Macro F1</b>		0.953
<b>Accuracy</b>		0.956

imbalanced gender. Along with not all Tajweed rules covered like (Tasheel). Neither all phonemes like letter thaal: ﺛ. But it is a move towards benchmarking Tajweed rules.

*VIII-E5 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
```