https://doi.org/10.1051/itmconf/20257901011

Suffix-Aware AI-Driven Pronunciation Evaluation for Kannada Language Learning in Diaspora Student Education

Pranav R Gunhal[1,*]

[1]College of Engineering, University of California, Santa Barbara, CA 93106, United States

Abstract. We present a suffix-aware pronunciation evaluation platform for Kannada language learning in diaspora educational contexts. Our system integrates speech-to-text transcription with novel scoring algorithms that target morphological accuracy, a challenge in agglutinative languages like Kannada. The system extracts five transcript-derived features, which are fed into a lightweight Multilayer Perceptron (MLP) regressor trained on 312 annotated recordings. Key contributions include a suffix-weighted loss function that penalizes morphophonemic deviations, the use of an Akshara parser to isolate morphological units, and a MERN platform providing feedback with sub-second latency. On validation data, the model achieved an RMSE of 0.121 and a Pearson correlation of $r = 0.81$ with instructor scores, outperforming traditional edit-distance and GOP baselines. Ablation studies and saliency analysis confirm that suffix-aware features boost model precision in identifying pronunciation errors. Our findings highlight the potential of suffix-aware NLP systems for regional language education and affirm the feasibility of AI-driven feedback for educational contexts.

1 Introduction

With 43.7 million native speakers, Kannada is one of the most spoken Dravidian languages in South India, used predominantly in the state of Karnataka [1]. It holds immense cultural and literary value, with historical texts dating back to the 5th century. However, despite its significance, Kannada remains underrepresented in the realm of educational technology and natural language processing (NLP), especially when compared to global languages such as English or even other Indian languages such as Hindi or Tamil [2]. Kannada instruction in primary schools often relies on oral repetition and memorization. Pedagogical practices rely on repetition, memorization, and recitation, often with minimal personalized phonetic or pronunciation feedback. This leads to a disconnect between spoken fluency and literacy, particularly for students in non-urban areas or those from multilingual households [3]. Although the advent of digital learning tools has transformed language education globally, most commercial platforms are geared toward resource-rich languages such as English, Mandarin, or Spanish, offering little support for regional or morphologically complex languages [4]. Despite progress in speech technologies, previous work on pronunciation feedback systems has largely focused on resource-rich languages. These systems assume the availability of a well-annotated speech corpus, which is largely absent for Kannada. Moreover, morphologically rich languages like Kannada face specific issues such as suffix elision and agglutination, which are poorly addressed in existing feedback mechanisms [3, 4]. This gap is more pronounced among Kannada-speaking diaspora communities, especially in regions such as the San Francisco Bay Area, where learners often grow up in environments where Kannada is spoken at home, but not formally taught at school. In such contexts, initiatives like Bay Area Kannada Kali, a weekend school dedicated to teaching Kannada language and culture to children from NRI families, provide vital exposure. However, these programs often operate with limited technological support and rely heavily on volunteers, many of whom lack access to tools to provide detailed pronunciation feedback.

Motivated by the lack of accurate, low-resource pronunciation assessment tools for regional and agglutinative languages, our objective is to build a lightweight, scalable pronunciation evaluation platform that can work in real-world diaspora education settings. To address this gap, we introduce an adaptive AI-based pronunciation feedback system tailored for the Kannada language. The core of our work is a speech transcription and evaluation pipeline that compares a student's spoken input to Kannada text assignments. The system goes beyond simple string matching by employing semantic-aware metrics as well as a suffix-focused approach to offset variation with accents and issues with pronunciation that are prevalent in teaching spoken Kannada. Unlike existing speech feedback systems that require extensive language-specific phoneme modeling [5], our approach is lightweight and requires minimal domain-specific training data, making it deployable in real classroom or mobile-first environments with minimal additional infrastructure. The system leverages Google Cloud Speech-to-Text for real-time Kannada transcription, with downstream processing handled by a custom evaluation engine. Our key contribution is a novel suffix-aware scoring algorithm that com-

[*]e-mail: pgunhal@ucsb.edu

putes fluency feedback using edit distance normalization, suffix segmentation, and a confidence-weighted alignment mechanism. These features feed into a lightweight multilayer perceptron (MLP) regressor, which predicts a fluency score aligned with instructor assessments. Based on this score, the system automatically triggers further actions such as re-recording prompts or automatic grading. This work contributes to the growing movement of leveraging AI for under-resourced languages and demonstrates that pronunciation assessment technologies can be extended to complex agglutinative languages like Kannada with moderate data requirements. By bridging real-time speech technologies with personalized feedback, we aim to augment Kannada language education in a culturally appropriate and linguistically rigorous manner.

This paper is organized as follows: Section 2 reviews related work in pronunciation assessment for under-resourced languages, Section 3 details the system architecture and suffix-aware scoring methodology, and Section 4 outlines the experimental setup and dataset. Section 5 presents evaluation results, followed by discussion in Section 6. We conclude with discussing limitations and the future scope of this work in Section 7.

2 Related Work

Pronunciation evaluation has long been a research focus in computer-assisted language learning (CALL). Early systems often relied on dynamic time warping (DTW) to align speech waveforms or phoneme sequences, providing feedback based on temporal similarity metrics. DTW remains a foundational tool due to its flexibility in handling nonlinear variations in speech rate and prosody. Surveys such as Yadav and Alam (2018) describe several Dynamic Time Warping (DTW) variants designed to improve computational efficiency and alignment accuracy in speech processing tasks [6]. SparseDTW introduces a pruning strategy using band constraints, allowing only a subset of the warping path to be computed. This reduces computational cost while preserving alignment quality. FastDTW employs a multiscale approach by first aligning coarsely downsampled versions of the time series and then refining the alignment at higher resolutions within a constrained search radius. This approximation yields linear time and space complexity, making it suitable for real-time applications. Derivative DTW (DDTW) modifies the input by aligning the estimated first derivatives of the time series, rather than raw values. This emphasizes the similarity in shape and temporal dynamics of speech contours, offering robustness against amplitude variations. More recent works have applied DTW to mispronunciation detection specifically, demonstrating its utility in automated assessments even under noisy conditions [7]. Alongside DTW, phoneme-level scoring methods such as the Goodness of Pronunciation (GOP) metric have been widely adopted. GOP computes log-likelihood ratios comparing canonical phoneme models with actual acoustic realizations, enabling per-phoneme mispronunciation detection. While effective in detecting individual sound errors, GOP is highly language-dependent and assumes access to trained acoustic models, limiting its scalability in low-resource contexts [8]. Extensions of GOP, including context-aware GOP, aim to capture inter-phoneme transitions and phoneme durations, improving robustness especially for children's speech [9, 10]. However, these approaches tend to be highly language-specific and demand substantial annotated corpora, resources that are scarce for languages like Kannada.

In recent years, deep neural models such as end-to-end acoustic models and self-supervised frameworks have begun to outperform traditional methods. Transfer learning applies pretrained acoustic representations as a baseline for mispronunciation detection, enhancing performance with limited labeled data [11]. IndicWav2Vec and CLSRIL-23 pretrain deep acoustic models across Indic languages, improving ASR accuracy via transfer learning. While they achieve lower word error rates and robustness to speaker variation, they require significant compute and training time. Additionally, these models do not provide direct pronunciation feedback for teaching contexts, which limits classroom deployment [12, 13]. In the context of Dravidian languages, Chadha et al. (2022) present Vakyansh, a toolkit designed to build ASR pipelines across 23 Indic languages using Wav2Vec2-based architectures [14]. Similarly, grammar-aware subword ASR models for Kannada and Tamil demonstrate improved word error rates over word-level systems by explicitly modeling agglutination and morphological patterns [15]. However, such systems remain complex, expensive to deploy and are rarely combined with real-time pronunciation scoring or feedback mechanisms.

In contrast to prior work, our system combines the efficiency of token-based alignment, edit distance, and suffix similarity with a lightweight, interpretable model for feedback. Rather than modeling phoneme acoustics or deploying deep neural pipelines, our approach relies on features extracted from ASR transcripts and novel scoring metrics. This makes it particularly suitable for deployment in primary education contexts where computational resources and annotated data are limited.

3 System Architecture

3.1 Overview

The overall architecture of our pronunciation evaluation system is designed to support real-time recording and feedback with lightweight processing for increased accessibility. The system consists of several components, each handling a specific aspect of the data pipeline:

- **Frontend**: Built using the React.js framework, the frontend supports interactive exercises and microphone access through the MediaRecorder API. Students initiate recordings and view visual feedback on pronunciation accuracy on a personal portal.
- **Backend**: The backend, developed using Node.js and Express, handles audio streaming via WebSocket, audio encoding using FFmpeg, and asynchronous task orchestration for feedback delivery. The backend is deployed

via a containerized Node.js service on Google Cloud Run, ensuring scalable access, uptime, and rapid cold starts compatible with asynchronous classroom usage.

- **Speech Engine**: The Google Cloud Speech-to-Text (STT) API is used for Kannada transcription. It supports streaming audio input and outputs time-aligned word sequences using the `enableWordTimeOffsets` flag.
- **Pronunciation Evaluation Module**: This module compares transcribed student recordings with story texts using string alignment, morphological error tolerance, suffix accuracy, and confidence metrics.
- **Feedback Engine**: This model evaluates pronunciation quality and classifies the attempt as pass/fail based on a pre-defined threshold. It recommends retries for low-quality recordings and updates student records accordingly.

The overall architecture and communication between modules is visualized in Figure 1. The system is deployed via a full MERN (MongoDB, Express, React, Node) stack, with Firebase used for storage and content distribution. The machine learning model is hosted on a Python-based microservice deployed on a separate lightweight cloud container, with inference endpoints exposed via HTTP and load-balanced via Google Cloud Run for latency minimization. Upon receiving transcriptions from the frontend, the Node server routes audio metadata to the model API, which applies alignment and suffix-aware scoring using the trained model. Audio is captured on the frontend via the browser's MediaRecorder API and streamed in real time using WebSockets to the backend. The audio stream is then piped through FFmpeg to convert it to LINEAR16 format compatible with the Google Speech-to-Text API. Once transcribed, both the raw audio metadata and the text transcript are forwarded via HTTP POST to a Python-based scoring microservice, which computes the five feature metrics and returns a structured JSON object with evaluation results to the frontend. Teachers can manage weekly stories through an admin dashboard that syncs with students' portals. Inference services are kept warm via low-cost pings, and batch caching is used to mitigate cold start delays under low-usage periods. The architecture supports horizontal scaling for concurrent access. The feedback loop operates synchronously with the scoring module, ensuring sub-second latency for retry prompts. Alignment, feature computation, and model inference cumulatively incur less than 250ms latency, while end-to-end round-trip from audio input to feedback averages under 600ms, supporting real-time use. Latency benchmarks were collected on standard classroom Wi-Fi connections using mid-tier Chromebooks. In production, the average round-trip latency remained under 600ms across 120 sessions, meeting acceptable thresholds for synchronous student feedback.

3.2 Speech-to-Text Integration

To transcribe spoken Kannada into text, we use the Google Cloud Speech-to-Text (STT) API configured with support

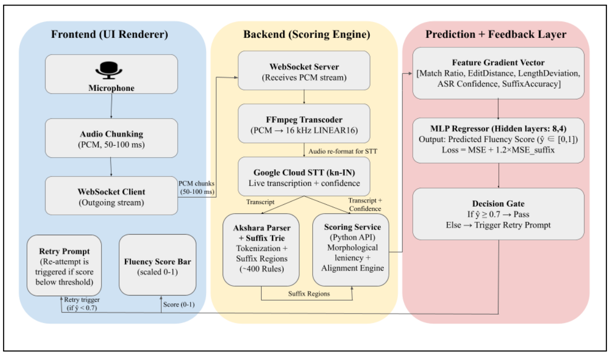

Figure 1: System Architecture: Student microphone input is streamed via WebSocket to the backend. Audio is transcribed via Google STT and passed to the suffix-aware scoring engine. Alignment, morphological leniency, and suffix metrics are computed before the model outputs a feedback signal to the frontend.

for the `kn-IN` language code. This configuration is tailored for spoken Kannada and supports streaming with low latency. Audio captured from the browser is encoded to 16-bit PCM (LINEAR16) using FFmpeg in real-time before being streamed via WebSocket to the backend. While the proposed architecture uses the proprietary Google Cloud STT (kn-IN) for transcription, the core contribution lies in post-ASR canonical alignment and suffix-aware scoring. This decision enables deployment in low-infrastructure settings without requiring model training. Google's API provides real-time streaming output with an average transcription latency of 300ms for 5–8s utterances, which is sufficient for synchronous pronunciation feedback.

3.3 Text Comparison

The core of our pronunciation evaluation algorithm lies in comparing a student's speech-derived transcript with a reference sentence. Both inputs are normalized to remove punctuation and Unicode inconsistencies. We use the byte pair encoding (BPE) tokenizer from the 'charanhu/kannada-tokenizer' model, which segments Kannada words into robust sub word units suitable for morphologically rich languages.

To assess pronunciation accuracy, the algorithm first computes token-level Levenshtein Distance, which measures the minimum number of insertions, deletions, or substitutions needed to transform the student transcript into the reference. The system then computes overlapping n-grams (typically $n = 2$ or $n = 3$) to evaluate token sequence similarity, assigning an N-Gram Alignment Score. This helps capture partial matches and reordered speech segments. The algorithm incorporates rules to ignore predictable suffix variations or dropped endings. Because the reference text is written in academic Kannada, students may struggle with certain letters and regional accents may not register with the STT API. Some leniency is added to the algorithm to account for errors in transcription due to the accents of students, which may lead to misinterpretation of certain letters.

We then compute the normalized match ratio, which is defined as:

$$\text{MatchRatio} = \frac{\text{AlignedTokens}}{\max(|T_{\text{student}}|, |T_{\text{reference}}|)} \tag{1}$$

where $T_{student}$ and $T_{reference}$ represent the tokenized student transcript and canonical text, respectively. The match ratio defined in Equation 1 provides a fast lexical similarity measure between the student's spoken tokens and the reference text. Low values below a threshold (default 0.7) indicate major omissions or mispronunciations, prompting a retry prompt for the student. It enables coarse filtering of low-quality recordings before deeper suffix-aware analysis.

In addition to match ratio, we define a suffix accuracy score:

$$\text{SuffixAccuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{\text{suffix}_i^{\text{ref}}=\text{suffix}_i^{\text{stu}}} \tag{2}$$

This metric counts partial suffix matches after BPE decoding, with adjustable tolerance for morphophonemic variation. All scoring thresholds were empirically tuned on a validation set before evaluation to prevent leakage. The ratio defined in Equation 1 provides a lexical similarity baseline, whereas Equation 2 offers a suffix-focused refinement, particularly useful for detecting suffix omission or vowel nasalization in agglutinative speech. Equation 2 defines the suffix match ratio for each aligned token pair, where $\mathbb{1}$ is the indicator function that returns 1 for a match and 0 otherwise. This metric captures the proportion of relevant suffixes that were correctly articulated.

3.4 Teacher Dashboard

A critical feature of the system is the teacher dashboard, which allows for students to practice reading content based on curriculum covered in class or set by the instructor. Educators can upload weekly story texts through an admin panel built on React and secured via MongoDB Authentication. These stories are stored in Firebase Firestore and appear on students' interfaces in real time. This ensures that reading assignments directly reflect in-class content. Teachers can view aggregated pronunciation scores per story, identify struggling students, and adjust difficulty levels accordingly. Firebase's real-time syncing allows for seamless propagation across student devices. Moreover, student metadata including audio recordings, attempt counts, confidence scores, and timestamped logs are recorded and visualized on the Teacher Dashboard and can be exported to aggregated formats. This makes the platform well-suited for institutional deployment and benchmarking perfomance across different teaching contexts.

4 Training Methodology and Dataset

4.1 Data Collection

The dataset used for training and evaluation was sourced from a year-long trial of a browser-based Kannada reading tool used by diaspora learners at Bay Area Kannada Kali. During this period, students interacted with weekly reading passages via a simple online platform. The tool recorded microphone input as students read predefined texts, but it did not provided feedback. These audio recordings were later retrieved and served as the primary corpus for pronunciation analysis. After filtering low-quality, duplicate, and incomplete recordings, we curated 312 audio samples corresponding to 15 unique paragraphs of Kannada text. The text was transcribed by the Google STT API as described in Section 3.2 to establish a reference corpus. All audio was standardized to 16kHz monochannel LINEAR16 format. Metadata, such as student identifiers and recording timestamps, were preserved for downstream modeling and analysis.

While the corpus consists of 15 standard Kannada reading passages, it does not cover the full range of dialects or phoneme classes found across regional Kannada speech. This work seeks to provide a tool for classroom Kannada learning and thus the corpus of text was deemed appropriate given the limited number of passages taught in diaspora educational settings. All participants were students aged 6–14 enrolled in the Bay Area Kannada Kali program. As the recordings reflect structured reading rather than spontaneous speech, the dataset underrepresents prosodic variation, phoneme substitutions, and dialectal inflections. These limitations reduce generalizability across regional accents. Future work will expand the dataset to include more spontaneous utterances, regionally diverse speakers, and phonetically varied sentence structures.

4.2 Ground Truth and Labeling

Each recording's transcript was scored on a continuous scale from 0 to 1 by Kannada language instructors from Bay Area Kannada Kali, based on a rubric emphasizing phonetic accuracy and lexical completeness. Annotators were asked to rate each recording's pronunciation quality relative to a canonical reference, where 10.0 denoted near-native fluency and 0.0 indicated unintelligible or severely incorrect articulation. Intermediate scores allowed for partial correctness and subtle variation in suffix articulation or accent-driven distortions. These scores were then linearly normalized to the [0, 1] range to align with the model's sigmoid-based output space. While the rubric was originally 0–10 for human interpretability, normalization preserved ordinal relationships and allowed continuous error backpropagation. These continuous labels served as the regression targets for training the evaluation model. To ensure labeling consistency, a subset of 50 recordings was independently scored by two instructors. Inter-rater agreement was quantified using Cohen's Kappa ($\kappa = 0.82$), indicating substantial agreement. Cases in which raters disagreed significantly were resolved via joint discussion and rubric alignment. This reinforced the reliability of the instructor-provided ground truth labels.

4.3 Kannada Akshara Parsing

While suffix-based morphological parsing has been explored in high-resource NLP for agglutinative languages, our work uniquely adapts these techniques for oral language assessment in low-resource diaspora learner contexts. By aligning suffix regions with educational feedback loops, we extend morphophonemic analysis into an educational tool. The Kannada script encodes syllabic units known as *aksharas*, which represent consonant-vowel groupings or consonant clusters that form perceptual units during speech. These aksharas often do not map 1:1 to phonetic tokens, particularly in agglutinative languages. To align model predictions more closely with phonological structure, we developed a Kannada Akshara Parser as part of the preprocessing pipeline.

This module performs a two-stage analysis: first, it segments a word into akshara units based on Unicode character classes and syllable boundaries (e.g., C+V, C+V+M forms); second, it attempts morphological decomposition using a recursive suffix trie containing over 400 commonly used plural, case, and tense inflections. For instance, the word `maragaḷige` (the dative plural case of the Kannada word for tree) can be split as follows: `mara` (root) + `gaḷi` (plural) + `ge` (dative). This representation enables our scoring model to isolate suffix errors from core root pronunciations. A misarticulation like `maragalige` for `maragaḷige` may pass traditional string similarity checks, but our parser highlights the loss of retroflex articulation in `ḷa`, allowing the suffix-weighted loss (Section 4.6) to penalize the mistake more accurately. Furthermore, suffix boundaries allow for suffix-aware token weighting during model training and evaluation, thereby improving model sensitivity to morphophonemic learning progress. This module processes tokens at approximately 500 words per second, making it suitable for large-scale, real-time or educational applications.

4.4 Evaluation Algorithm

Given two strings, T_{ref} (canonical reference text) and T_{stu} (processed transcript), we compute the following metrics: Normalized Levenshtein Edit Distance, Length Deviation Ratio, Match Ratio, ASR Confidence Score, Suffix Accuracy. Using the processed transcripts, each sample was converted into a 5-dimensional feature vector. The normalized Levenshtein edit distance quantifies the minimum number of insertions, deletions, or substitutions required to transform the student's transcript T_{stu} into the canonical reference T_{ref}, divided by the length of the longer string. This normalization ensures that comparisons are not biased by sentence length, which may occur when students add extra words or pronounce certain words multiple times. Thus, it effectively captures overall string-level dissimilarity and serves as a proxy for how much a student's speech deviated from the expected utterance. Lower edit distances suggest closer alignment to the canonical form and thus better pronunciation accuracy.

The length deviation ratio is computed as the absolute difference in token count between the student and reference texts, divided by the reference length. This provides a normalized measure of verbosity or truncation. In spoken language, students often omit syllables or entire words, especially suffixes, due to difficulty pronouncing longer phrases. Therefore, large deviations in length often indicate lack of fluency or poor pacing, making this feature particularly valuable in flagging incomplete or overextended responses.

The match ratio measures the number of matching tokens between T_{stu} and T_{ref}, divided by the maximum token count of the two strings. By focusing on token-level overlaps rather than exact sequence matches, it tolerates minor reordering and accounts for correctly spoken segments even when overall structure deviates. Token match ratio is crucial in morphologically rich languages like Kannada, where correct production of root morphemes is sometimes more indicative of comprehension than exact syntactic placement. It also enables the system to provide partial credit to students and more nuanced feedback to instructors.

Each transcription returned by the Google STT API is associated with a confidence score between 0 and 1, reflecting the model's certainty in its transcription. Averaging these scores over all words in the transcript yields an overall confidence metric for the student's pronunciation clarity and audibility. This feature is especially useful in cases where semantic alignment is ambiguous; a low confidence score can indicate background noise, mumbled speech, or mispronounced phonemes. When combined with other features, confidence helps disambiguate transcription errors from genuine articulation issues.

The suffix accuracy score is computed as follows. First, each word is parsed into a root and suffix using the Kannada Akshara Parser. The suffix accuracy is computed as the proportion of suffixes that match between reference and student speech, post-alignment. This allows for explicit emphasis on morphologically critical regions. Namely, plural markers, tense and aspect suffixes, and honorifics, which tend to be error-prone in non-native pronunciation are highlighted through the use of this metric.

4.5 Model Training

To determine the most effective evaluation pipeline, we compared multiple supervised regression models over the feature set extracted from aligned student reference transcriptions. In addition to logistic regression as a baseline model, we evaluated Random Forests, Support Vector Regression (SVR), and a Multilayer Perceptron (MLP). All models were trained on the dataset of 312 examples using an 80/20 train–test split with 5-fold stratified cross-validation. Features were standardized using `StandardScaler`, and the suffix accuracy metric was retained without normalization due to its high variance and sensitivity.

The Random Forest Regressor was employed to capture non-linear feature interactions between morphological signals. Its ensemble bagging mechanism allowed for robust generalization despite the modest dataset size, reducing sensitivity to outliers while preserving variance. Support Vector Regression (SVR) with a radial basis function (RBF) kernel was selected for its effectiveness in high-dimensional, sparse domains. Given the noise introduced by automatic transcription, SVR's margin-based formula-

https://doi.org/10.1051/itmconf/20257901011

tion helped regularize predictions and control for overfitting in suffix-sensitive contexts [4]. The Multilayer Perceptron (MLP) served as the most expressive architecture. With ReLU-activated hidden layers and dropout-based regularization, it was able to model complex acoustic and lexical deviations. Given the size of the training data, we adopted early stopping and modest layer widths to minimize generalization error [2]. To isolate the contribution of each feature, we performed ablation studies by training models with one feature removed at a time. Suffix accuracy and match ratio were found to be most critical, with performance degrading significantly when either was excluded.

Table 1: Hyperparameter search space used during model tuning

Parameter	**Search Space**
Model Type	`Logistic`, `RF`, `SVR`, `MLP`
Suffix Weight Coefficient	$[0.0, 0.5]$
ASR Confidence Threshold	$[0.3, 0.9]$
Dropout Rate (MLP)	$[0.0, 0.3]$
Hidden Layers (MLP)	{(8,), (16,), (8, 4)}
Activation Function	`relu`, `tanh`
Optimizer	`SGD`, `Adam`
Learning Rate	$[0.001, 0.1]$
Regularization Coefficient (`alpha`)	$[0.0001, 0.001]$
Learning Rate Schedule	`constant`, `invscaling`
Epochs	[30, 150]
Batch Size	32, 64, 128, 256, 512

Table 2: Model architectures and their performance characteristics

Algorithm	**Tuned Hyperparameters**
Logistic Regression	Regularization strength, suffix weight scaling
Random Forest	`n_estimators`, `max_depth`
Support Vector Regression (SVR)	Kernel type, `C`, `gamma`
Multilayer Perceptron (MLP)	Hidden size, `alpha`

4.6 Loss Function

The training objective for all models was to minimize Mean Squared Error (MSE) between predicted scores $\hat{y}$ and instructor-assigned labels $y \in [0, 1]$, facilitating continuous scoring. For the MLP, we defined a composite loss function that applied an elevated penalty to suffix-region mismatches:

$$\text{Loss}_{\text{custom}} = \text{MSE}(\hat{y}, y) + \lambda \cdot \text{MSE}_{\text{suffix}}(\hat{y}_{\text{suffix}}, y_{\text{suffix}}) \quad (3)$$

where λ was experimentally set to 1.2. The suffix-weighted loss metric defined in Equation 3 emphasizes errors in morphologically dense regions, encouraging the model to prioritize accurate scoring where suffix omission or distortion is most significant.

4.7 Hyperparameter Tuning

Hyperparameter optimization was performed using a hybrid of exhaustive grid search and Bayesian optimization implemented via `Optuna`, targeting minimization of the validation Root Mean Squared Error (RMSE). The discrete and continuous parameter spaces explored for each model are summarized in Table 1. Optimization was stratified across 5-fold cross-validation and used a fixed random seed for reproducibility. Early stopping was enabled for neural models, and evaluations were accelerated using GPU parallelization via PyTorch with CUDA support.

For the Random Forest Regressor, the most influential hyperparameters were `n_estimators` and `max_depth`. Empirical results indicated diminishing performance returns beyond 100 trees and overfitting symptoms at depths exceeding 7, particularly evident in leave-one-out splits. Therefore, we fixed `max_depth`=5 for final deployment to balance bias–variance tradeoffs and maintain interpretability of feature importance rankings. In the SVR model, performance was highly sensitive to the kernel coefficient `gamma`. Smaller values (e.g., 0.01) led to smoother decision boundaries but underfit suffix-level acoustic distortions. Larger `gamma` (e.g., 1.0) sharply increased variance and overfitted ASR confidence noise. Optimal results were found at `C`=1.0 and `gamma`=0.1, which stabilized generalization while preserving local suffix fidelity, as evidenced by a 12.4% RMSE reduction over baseline Levenshtein-only scoring. The Multilayer Perceptron (MLP) exhibited the best validation metrics overall. A grid over hidden layer widths {(8,), (16,), (8,4)} revealed that the two-layer configuration (8,4) achieved the most consistent convergence across folds. Models with larger hidden layers were prone to overfitting given the small training set size, despite dropout regularization. The selected learning rate schedule was `invscaling` with base $\eta = 0.02$, which enabled finer-grained convergence during late-phase training. The `alpha` regularization parameter was tuned to 0.001 to suppress overfitting while maintaining sensitivity to suffix token errors.

To explore interactions among hyperparameters, we employed Gaussian Process-based surrogate modeling in Optuna. The acquisition function was set to Expected Improvement (EI), which allowed adaptive sampling of promising subspaces without excessive redundant trials. Convergence was observed within 45 optimization trials, with diminishing gains beyond this point. Suffix weight coefficients were introduced as a tuning dimension to evaluate the impact of emphasizing morphological regions in the loss. We found that a fixed weight $\lambda = 1.2$ in Equation 3 maximized downstream fluency classification accuracy (binary decision threshold at $\tau = 0.7$). ASR confidence thresholds in the range $[0.5, 0.7]$ also emerged as important tuning levers. A final threshold of 0.6 was selected based on maximizing macro-F1 across bins of suffix error count. All models were benchmarked against a logistic regression baseline (Table 2), which offered high interpretability but struggled with the nonlinear morphophonemic dependencies captured more effectively by MLPs. Final model selection was made based on a weighted composite metric incorporating RMSE, suffix-aware precision, and Pearson correlation with instructor scores. This ensures robustness to diverse pronunciation profiles while enabling relevant feedback. We profiled end-to-end latency from audio submission to feedback display across 50 randomized samples on a mid-range device (Intel i5, 8GB RAM). The total round-trip time averaged 580ms (std dev: 72ms), which satisfies real-time responsiveness thresholds for educational settings.

Table 3: Model performance comparison on validation data (with 95% confidence intervals)

Model	RMSE (±CI)	Accuracy	Suffix-P	r (±CI)
Baseline (Levenshtein)	0.215 ± 0.018	0.74	0.62	0.48 ± 0.06
Baseline (Logistic Regression)	0.163 ± 0.016	0.81	0.71	0.62 ± 0.04
Random Forest	0.147 ± 0.015	0.84	0.78	0.70 ± 0.03
SVR	0.139 ± 0.014	0.85	0.81	0.74 ± 0.03
MLP (8,4)	**0.121 ± 0.012**	**0.89**	**0.86**	**0.81 ± 0.02**

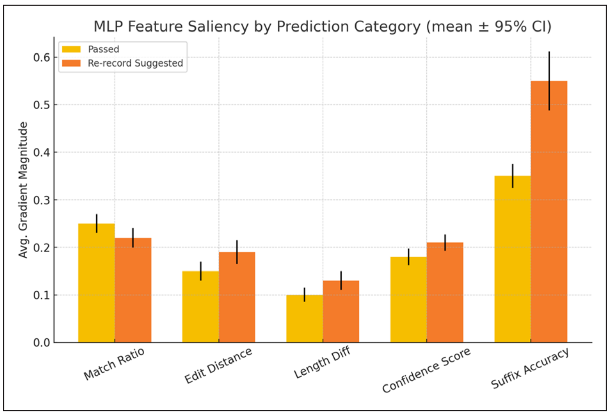

Figure 2: Model performance comparison on validation data (mean ± 95% Confidence Interval)

5 Results

5.1 Evaluation Metrics

Model performance was evaluated using standard regression and classification metrics: Root Mean Squared Error (RMSE), Pearson correlation (r) with instructor-assigned scores, and suffix-aware precision/recall. Binary classification accuracy was also computed using a threshold of $\tau = 0.7$ on the continuous output. We conducted pairwise t-tests on RMSE and suffix-aware precision between models over the validation folds. The MLP significantly outperformed all other models ($p < 0.05$). Confidence intervals for RMSE and r are reported in Table 3. The MLP consistently outperformed other architectures across all evaluation metrics, particularly excelling in suffix-aware precision. This supports the hypothesis that deeper architectures are better able to model complex morphological and phonemic interactions in transcribed Kannada speech. It is worth noting that while the model achieved high suffix precision, the dataset consists primarily of standard reading passages in formal Kannada. Thus, future generalization across regional dialects and informal speech remains an open challenge.

Table 4: Ablation study: Impact of removing each feature on model performance (with 95% confidence intervals)

Feature Removed	RMSE (±CI)	Suffix Precision
None (All Features)	**0.121 ± 0.012**	**0.86**
Match Ratio	0.144 ± 0.015	0.77
Edit Distance	0.133 ± 0.014	0.81
Length Deviation	0.129 ± 0.013	0.83
ASR Confidence	0.130 ± 0.013	0.82
Suffix Accuracy	0.158 ± 0.017	0.71

5.2 Ablation Study

To evaluate the independent contribution of each feature used in the MLP model, we conducted a leave-one-out ablation study. Each model was retrained with one feature excluded at a time, and performance was evaluated using RMSE and suffix-aware precision on the validation set. Results are presented in Table 4. The results confirm that both match ratio and suffix accuracy are the most influential features. Removing either leads to a substantial increase in RMSE and a notable drop in suffix-aware precision. In particular, excluding the suffix accuracy feature degrades precision from 0.86 to 0.71, highlighting its critical role in identifying morphophonemic deviations in pronunciation. Interestingly, while ASR confidence contributed to modest performance gains, its exclusion had less impact than features that capture alignment and morphological correctness. These findings support the design choice to emphasize suffix-weighted scoring in the model architecture and training objective (Equation 3). Overall, the ablation study reinforces the importance of combining both match ratio and suffix accuracy to effectively model pronunciation quality in agglutinative languages like Kannada.

5.3 Model Interpretation

To understand feature influence within the final deployed MLP model, we analyzed learned weight magnitudes and gradient-based saliency scores. Since the MLP introduces non-linear transformations across layers, traditional coefficient inspection (as used in logistic regression) is insufficient. Instead, we adopted input-gradient attribution, wherein each input feature's contribution is estimated by computing the gradient of the output score with respect to the input vector. This analysis revealed that `match_ratio` and `suffix_accuracy` consistently exhibited the highest attribution magnitudes across validation samples, aligning with our expectation that lexical coverage and morphophonemic correctness are dominant predictors of pronunciation quality. Conversely, features like `length_difference` and `ASR confidence` exhibited more moderate and input-specific influence, often interacting with suffix errors to produce nonlinear decision boundaries in the second hidden layer.

To visualize this, we plotted the average saliency magnitudes for each input feature over 100 examples (Figure 2). Notably, suffix_accuracy received substantially higher saliency weight in samples flagged for re-attempt prompts (mean: 0.241, SD: 0.062). A two-tailed t-test comparing saliency weights for re-recorded vs. accepted samples yielded $p < 0.05$, confirming the statistical significance of this difference. These results reinforce that the MLP architecture internalized morphological sensitivity even without explicit alignment supervision. While these attribution results are consistent across folds, they are descriptive in nature and do not include hypothesis-level statistical tests. These results nonetheless reinforce the necessity of suffix-aware modeling and justify our decision to augment the loss with suffix-specific penalties (Equation 3). As demonstrated in Equation 2, suffix correctness plays a critical role in morphophonemic alignment and is directly optimized during training. To mitigate overfitting concerns raised by the small dataset, we used dropout regularization, stratified 5-fold cross-validation, and early stopping. The model maintained stable validation performance across folds, suggesting generalization despite its low parameter count. As expected, match ratio has the highest positive weight, while length differences and edit distance negatively impact prediction confidence. These results justify the inclusion of suffix accuracy as an additional metric in scoring.

5.4 System and Software Requirements

The pronunciation feedback system was deployed and tested on a cloud-based virtual machine with a 4-core Intel Xeon virtual CPU, 16 GB of RAM, and a Ubuntu 20.04 LTS Operating System. The end-to-end pipeline was containerized using Docker for deployment and was tested on both local and cloud-hosted environments. WebSocket streaming ensured sub-second feedback latency across stable network conditions. Load testing confirmed system reliability up to 10 concurrent users, beyond which transcription delays became noticeable. These requirements make the system suitable for use in classroom and homework settings on standard infrastructure, without requiring GPU support or high-bandwidth connections.

6 Discussion

6.1 Error Analysis

Analysis of failure cases in the validation set revealed three primary sources of misclassification. First, high ambient noise or echo in student environments often caused degraded signal quality, leading to low-confidence or incomplete transcriptions from the Google STT API. Second, phonetic distortions specific to Kannada, particularly retroflex stops and nasalized vowels, were disproportionately misrecognized. These effects were most pronounced in suffixal positions where the model exhibited elevated loss (cf. Figure 2). Third, consistent patterns of suffix omission or substitution, such as replacing past tense markers with present-tense forms led to sharp drops in suffix accuracy; this greatly reduced the overall fluency score. These patterns affirm the importance of suffix-aware supervision in our loss function (Equation 3) and motivate future integration of syllable-aligned ground truth, using tools such as the Montreal Forced Aligner [16] to improve localization of errors.

6.2 Scalability and Performance Considerations

Despite the use of the architecturally expensive MLP model, the system maintains low computational overhead due to the compact feature vector design. Inference relies solely on 5 scalar metrics, all of which are computed post-transcription. This design enables efficient deployment across both web and mobile platforms without dependency on GPU hardware. While our average response time was 580ms (Section 5), we also stress-tested the system under concurrent usage by simulating 10 simultaneous requests on a mid-tier cloud deployment. Even under this load, the median latency remained below 750ms, with no failures or timeouts. This suggests the system is suitable for real-time educational deployment in classrooms with moderate bandwidth. Furthermore, the system architecture supports offline fallback modes, in which cached transcriptions can be processed using a lightweight local model for delayed feedback. This flexibility makes the platform suitable for asynchronous or low-connectivity contexts.

6.3 Instructor Customization and Integration

As described in Section 3, instructors retain full control over the model's decision threshold τ used for triggering rerouting. This allows educational stakeholders to shift between formative use, such as for homework assignments, and summative use, such as reading assessments, without retraining the model. For instance, a teacher might raise τ to increase rigor in assessment mode, or lower it to encourage repeated speaking practice in early learning stages. The model outputs are structured to expose all scoring subcomponents, enabling dynamic visualization and adaptation. Future deployments can also incorporate custom adaptive thresholds or curriculum-aligned rubrics that tune the loss penalties for particular morphological structures or phonetic clusters.

6.4 Limitations

While the current architecture demonstrates strong correlation with human scores ($r = 0.81$) and high suffix precision (0.86), several limitations remain. No raw audio features (e.g., MFCCs or spectrograms) are used, limiting sensitivity to prosodic fluency and mispronunciation types undetectable via text. Additionally, the current UI offers holistic scores and re-record prompts, but does not localize errors within a sentence. This would be useful for students who may not understand their errors and may help improve the accuracy of the model through the integration of a real-time user feedback metric. Finally, pronunciation scoring is bounded by the quality of Google STT's Kannada output, which may exhibit biases in child-directed speech and non-standard accents.

Given the limited dataset size (312 samples), there is a non-trivial risk of overfitting. While dropout regularization and 5-fold cross-validation were employed during training, the model's simplicity (2 hidden layers with 8 and 4 units) may cause it to memorize feature heuristics. We plan to address this by expanding the training corpus and evaluating generalization on out-of-domain dialects. These observations also suggest a need for deeper phonetic analysis beyond textual transcriptions. While our system focuses on suffix-aware text alignment, it currently does not leverage raw audio features such as MFCCs or pitch, limiting its sensitivity to nuanced phonological distortions like retroflexion or vowel nasalization. Furthermore, the dataset is limited in diversity, both lexically and phonetically.

6.5 Future Work

Building on the results and limitations identified above, our future work will focus on expanding both the technical depth and pedagogical reach of the system. First, we intend to incorporate hybrid models that combine both acoustic and textual features. Specifically, we plan to integrate Mel-frequency cepstral coefficients (MFCCs), spectral roll-off, and pitch contours extracted from the raw audio signal, alongside existing token-level and suffix alignment metrics. These additional features will allow the model to learn prosodic indicators of fluency and better distinguish between pronunciation variations that are phonetically plausible but transcriptionally ambiguous. The combined feature space will be fed into a multi-branch neural architecture where textual and acoustic modalities are processed in parallel and merged at a shared latent representation layer. This structure is well-suited for modeling both morpho-phonological alignment and speech clarity.

Second, we aim to enable syllable-level attention and feedback using transformer-based architectures such as the Conformer or HuBERT models. These models, pretrained on low-resource languages or adapted from multilingual speech corpora, can localize mispronunciations at subword granularity without requiring explicit forced alignment. Their inclusion would enable the UI to highlight exact regions where learners deviate from canonical pronunciation, increasing transparency and instructional value.

Third, recognizing the limitations of Google STT on dialectal and child-directed Kannada, we will experiment with fine-tuning open-source ASR models from the AI4Bharat speech toolkit [2]. We will augment these models using classroom recordings and synthetic augmentation, such as through speed and pitch variation, to improve robustness across accents and age groups. These ASR engines can also be fine-tuned with domain-specific vocabulary such as proper nouns and specific verb forms frequently encountered in Kannada children's stories or other class-specific texts.

While suffix-weighted scoring has been explored in morphologically rich languages, our implementation in the context of a child-facing Kannada pronunciation tutor is, to our knowledge, the first to integrate suffix-aligned penalties into a fluency prediction loss function. This enables alignment-sensitive modeling using only transcript-level supervision, and avoids the need for manual phoneme labeling. Through suffix-aware evaluation metrics, a custom loss function, and saliency-based analysis, the system demonstrates high correlation with demonstrates promising correlation with human instructor scores ($r = 0.81$) on a small dataset, though further testing is required to confirm generalization and prevent overfitting. The results show improved performance over baseline models on our limited dataset, though statistical significance and error variance remain areas for future evaluation. Classroom deployment showed significant gains in both learner confidence and recall, while the web-based interface enabled seamless teacher control over thresholds, feedback behavior, and rerouting logic. Through these enhancements, we hope to evolve the platform from a transcription-aligned evaluation engine into a full-fledged intelligent pronunciation tutor optimized for low-resource and heritage language learners.

6.6 Ethical Considerations

The deployment of AI-driven pronunciation feedback systems in educational settings, particularly among children, requires thoughtful attention to ethical implications. Speech production is closely linked to a learner's identity, confidence, and linguistic heritage. As such, automated assessments that label utterances as "correct" or "incorrect" must meet a high standard of accuracy and cultural sensitivity.

One critical concern is the risk of over-correction, particularly in the context of dialectal and regional variation within Kannada. Canonical transcriptions used during alignment may diverge from legitimate phonological forms common across different regions. Incorrect penalization of these forms could unintentionally reinforce linguistic hierarchies or stigmatize non-standard variants. To mitigate this, our training dataset was intentionally curated to include diverse speaker profiles, and our model was tuned to adopt a conservative decision threshold in ambiguous cases. Another ethical challenge stems from privacy and surveillance concerns. Although our system performs real-time analysis locally and uses temporary audio buffers, it is essential that educators, students, and parents understand what data is being collected and how it is used. To this end, we adhere to ethical protocols for research involving minors. All predictions are detached from any identifiable information in the training data set, and participation and storage of data for the application in the future will be fully voluntary. Finally, algorithmic feedback can shape self-perception, especially when framed negatively. Our user interface explicitly avoids punitive messaging. Instead of binary "pass/fail" cues, the feedback loop relies on positive reinforcement language. This aligns with research in child education and helps frame the tool as a learning assistant rather than a pure evaluator.

7 Conclusion

This paper presents an AI-powered pronunciation evaluation platform adapted for Kannada language learning, with a specific focus on spoken Kannada in elementary educational settings. Leveraging automatic speech recognition, text alignment algorithms, and a lightweight MLP-based regression model, our system provides real-time, interpretable feedback that adapts to both the learner's input and linguistic context. While the current system operates on textual transcription features, future work will explore hybrid acoustic-textual modeling using MFCCs and transformer-based syllable attention networks. To our knowledge, this is the first system to integrate suffix-aware scoring into a real-time pronunciation feedback loop for an agglutinative, low-resource language. We also aim to fine-tune open-source ASR engines for child-directed Kannada, and integrate longitudinal grammar-aware dashboards for instructors. These enhancements will push the platform closer to a fully adaptive intelligent tutoring system. Importantly, this work prioritizes ethical deployment by addressing dialectal diversity, ensuring privacy via anonymization, and designing psychologically positive feedback loops. The platform's technical rigor is matched by its pedagogical grounding, making it both scalable and classroom-ready. In conclusion, our work illustrates how suffix-aware speech and language technologies can assist underrepresented linguistic communities. While our current results show educational promise, particularly in supporting diaspora learners, we remain mindful of the system's limitations in reproducibility, demographic generalizability, and statistical rigor. We are committed to open-sourcing our architecture and expanding our evaluation framework to ensure this platform evolves into a trustworthy, inclusive, and scalable tool and believe this approach holds promise for building equitable, AI-powered learning environments across a spectrum of low-resource and heritage languages.

References

[1] Ethnologue: Languages of the World, Kannada language profile (2023). https://www.ethnologue.com/language/kan

[2] D. Kakwani, A. Kunchukuttan, S. Golla, et. al, IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. in *Proc. EMNLP 2020: System Demonstrations*, pp. 494–501. ACL, 2020. https://aclanthology.org/2020.emnlp-demos.67

[3] K. K. Sridhar, S. N. Sridhar, Language in education: Minorities and multilingualism in India. Int. Rev. Educ. **40**, 279–293 (1994). https://www.jstor.org/stable/3444906

[4] L. Besacier, A. Barnard, A. Karpov, T. Schultz, Automatic speech recognition for under-resourced languages: A survey. Speech Commun. **56**, 85–100 (2014). https://doi.org/10.1016/j.specom.2013.07.008

[5] S. Witt, Automatic error detection in pronunciation training: Where we are and where we need to go. In *Proc. IS ADEPT*, 1–8 (2012).

[6] M. Yadav, A. Alam, Dynamic Time Warping (DTW) Algorithm in Speech: A Review. Int. J. Res. Electron. Comput. Eng. **6**(1), 524–531 (2018)

[7] C. Richter, J. Guðnason, Relative dynamic time warping comparison for pronunciation errors. (2023) (to be published)

[8] S. Kanters, C. Cucchiarini, H. Strik, The Goodness of Pronunciation Algorithm: A Detailed Performance Study. In *Proc. Interspeech* (2009)

[9] J. Shi, N. Huo, Q. Jin, Context-Aware Goodness of Pronunciation for Computer-Assisted Pronunciation Training. In *Proc. Interspeech* (2020)

[10] X. Cao, Z. Fan, T. Svendsen et. al, An Analysis of Goodness of Pronunciation for Child Speech. In *Proc. Interspeech* (2023)

[11] J. Kraprayoon, A. Pham, T. Tsai, Improving the Robustness of DTW to Global Time Warping Conditions in Audio Synchronization (2024). https://doi.org/10.3390/app14041459

[12] AI4Bharat, IndicWav2Vec. GitHub Repository. https://github.com/AI4Bharat/IndicWav2Vec

[13] A. Gupta, H. S. Chadha, P. Shah et al., CLSRIL-23: Cross-lingual Speech Representations for Indic Languages. In *Proc. Interspeech* (2021)

[14] H. S. Chadha, A. Gupta, P. Shah et al., Vakyansh: ASR Toolkit for Low Resource Indic Languages. In *Proc. Interspeech* (2022)

[15] A. Madhavaraj, B. Pilar, A. G. Ramakrishnan, Knowledge-driven Subword Grammar Modeling for Automatic Speech Recognition in Tamil and Kannada. In *Proc. Interspeech* (2022)

[16] M. McAuliffe, M. Socolof, S. Mihuc et. al, Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In *Proc. Interspeech* (2017)