International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Speech Enhancer: Real Time Correction Using AI/ML

Prof. K Prasanth[1], Prof. Divyashree N[2]

[1]Department of Computer Science and Engineering, East Point College of Engineering and Technology
[2]Department of Computer Science and Engineering, East Point College of Engineering and Technology

---

Abstract - The growing importance of effective communication in academic, professional, and personal contexts highlights the necessity of tools that support structured and continuous speech improvement. However, traditional learning environments often lack real-time feedback, personalized guidance, and objective evaluation of spoken performance. This idea addresses these setbacks by developing an AI-powered Speech Enhancer system designed to analyze and improve pronunciation, grammar, and speech fluency through real-time interaction. The system integrates advanced speech recognition, acoustic analysis, stutter detection, and NLP-based grammar evaluation to provide users with immediate, actionable feedback. A modular architecture built with modern backend technologies ensures smooth audio processing, while the user interface delivers clear visual guidance and session-wise progress tracking. Procedural evaluation brings high accuracy in detecting stuttering patterns, strong alignment between AI-based pronunciation scoring and reliable grammar correction performance. Overall, the Speech Enhancer system enhances spoken communication by offering a structured, intelligent, and interactive platform that supports continuous speech practice and measurable improvement.

Key Words: Speech Enhancer, Speech Recognition, Pronunciation Analysis, Stutter Detection, NLP, Real-time Feedback.

1. INTRODUCTION

A. Background & Introduction

Challenges like unclear pronunciation, stammering, or difficulty forming proper sentences can hinder effective communication and lower confidence. To address this, a **Speech Enhancer** leverages advanced technologies such as **Artificial Intelligence (AI)**, **Machine Learning (ML)**, **Speech Recognition**, and **Natural Language Processing (NLP)**. By analyzing spoken language, it delivers tailored feedback aimed at refining pronunciation, improving fluency, and strengthening sentence construction. The system works by detecting mistakes and offering instant suggestions, helping users practice and enhance their speech in real time.

B. Problem Statement

A significant number of people face difficulties in expressing themselves clearly due to issues like incorrect pronunciation, stammering, and grammatical mistakes during speech. Current solutions often tackle these problems in isolation and fail to provide a unified approach. This project introduces an intelligent **Speech Enhancer** that processes spoken input in real time, identifies communication barriers, and delivers corrective feedback to strengthen overall speaking ability.

C. Existing System

Most available speech improvement applications are designed to address a single concern—for example, tools that train pronunciation or detect stuttering. However, they rarely offer a holistic analysis of speech patterns, personalized coaching, or immediate corrective suggestions. In addition, many of these systems struggle with accuracy, particularly in detecting stuttering, and often require users to switch between multiple platforms to manage different aspects of speech improvement.

D. Limitations

- Primarily target either pronunciation or stuttering, not both together.
- Do not provide a full evaluation of speech fluency.
- Accuracy in identifying speech disorders remains limited.
- Lack of instant, real-time corrective feedback.
- Recommendations are often generic and not tailored to individual needs.

E. Proposed System

The proposed Speech Assistance Platform integrates **speech recognition and intelligent feedback mechanisms** to analyze pronunciation, fluency, and sentence structure simultaneously. It provides immediate suggestions for improvement and supports interactive learning.

Advantages:

- Real-time speech analysis and correction.
- Interactive learning modules for effective practice.
- Personalized feedback based on user performance.
- Accessibility across mobile and desktop platforms.

F. Aim

To develop an intelligent and user-friendly Speech Assistance Platform that helps users improve pronunciation, reduce stuttering, and construct sentences correctly through AI-powered real-time feedback and personalized learning.

G. Objectives

- Detect and correct pronunciation errors.
- Analyze stuttering and speech fluency in real time.
- Evaluate sentence structure and provide suggestions.
- Ensure accessibility through text-to-speech and simple interfaces.
- Provide a unified solution for multiple speech-related issues.

Conclusion: The project introduces a comprehensive Speech Assistance Platform that addresses common speech difficulties through AI and speech recognition technologies. Unlike existing tools, it combines pronunciation correction, stuttering detection, and sentence analysis in a single accessible platform, providing personalized and real-time feedback to improve communication skills and confidence.

2. BACKGROUND AND RELATED WORK

A. Speech Assistant Applications Based on Speech Technologies [1]

A set of ICT-based speech tools was designed for children in Slovakia who have hearing or speech difficulties. The system combines **Dynamic Time Warping (DTW)** with **Automatic Speech Recognition (ASR)** to check and improve how words are pronounced. Three playful apps—**SketchBook**, **Jozef the Hedgehog**, and **Find the Difference**—were created along with **LogoWeb**, an online tool for practicing phonemes. Early trials with children showed encouraging results for home-based speech therapy, suggesting that gamified learning can support clinical practice.

Even with these benefits, the system has some limits. Since the tools are built for Slovak, they cannot be easily used in other languages without major changes. Depending only on DTW and ASR can also lead to errors, especially for children with severe speech difficulties whose patterns differ greatly from standard references. Finally, while the game-like design works well for younger children, it wouldn't be suitable for adult age groups or diverse user needs.

B. Inappropriate Pause Detection in Dysarthric Speech Using Large-Scale Speech Recognition [2]

The detection of inappropriate pauses in dysarthric speech—a clinically significant indicator for severity assessment in individuals affected by stroke-induced dysarthria. The proposed system builds on a Whisper-based ASR model by adding a special pause prediction layer. In this design, pauses are treated as separate tokens inside the recognition process. This approach combines speech recognition and pauses detection into one learning task, leading to better results than standard methods. New evaluation measures are also introduced so that pause detection can be judged independently of ASR accuracy, making clinical assessment more reliable.

However, there are some limitations. The study uses only a Korean dysarthric speech dataset, so the model would need major retraining to work with other languages. The system also depends heavily on ASR quality—if recognition accuracy drops due to accent differences or domain mismatch, pause detection becomes less reliable. Finally, the framework focuses only on timing issues like pauses and would not address other important dysarthric features such as articulation mistakes, rhythm problems, or voice abnormalities.

C. Detecting Dysfluencies in Stuttering Therapy [3]

Investigate automatic dysfluency detection to support stuttering therapy monitoring. The study uses wav2vec 2.0 with a multi-task learning (MTL) setup, trained on a large English stuttering dataset. The MTL approach works on two goals at the same time—classifying dysfluencies and recognizing therapy techniques. Results show that sharing representations improves performance for both tasks. Tests across different languages also suggest that the learned features can transfer, making the method useful beyond English speech.

Still, there are important limitations. The model's success depends on how diverse the training data is. If dialects, age groups, or varied speech styles are not well represented, the system may not generalize. Real-time use has not yet been tested in everyday or noisy environments, which is essential for clinical practice. Cross-language transfer has only been tried with a few languages and not across very different language families. Finally, fine-tuning large models like wav2vec 2.0 requires heavy computing power, which can be a challenge for clinics with limited resources.

D. Summary

The works surveyed above collectively demonstrate substantial progress in AI-driven speech analysis and therapy support; however, they collectively reveal a set of persistent gaps that motivate the present research. Specifically: (i) existing tools are predominantly language-specific or limited to narrow speech pathology subtypes; (ii) clinical coverage is fragmented—each system addresses a single dimension of speech impairment (pronunciation accuracy, temporal irregularities, or dysfluency) rather than providing holistic, multi-dimensional feedback; and (iii) real-time, noise-robust deployment has not been adequately validated in any of the reviewed systems.

The proposed Speech Enhancer system directly addresses these gaps by integrating real-time ASR with a multi-modal correction engine capable of simultaneously assessing pronunciation, pause patterns, and fluency across diverse linguistic and pathological profiles. The architecture draws on the DTW-based evaluation principles of [1], the pause-token formulation introduced in [2], and the cross-lingual transfer insights from [3], synthesizing them into a unified, language-agnostic framework suitable for real-world deployment.

3. CORE SYSTEM DESIGN

The system design of the proposed Speech Enhancer establishes a formal architectural framework for a scalable, modular, and user-centric platform. The design integrates contemporary speech recognition technologies, real-time feedback mechanisms, and AI-driven personalization to address three primary speech impairments: stuttering, pronunciation errors, and grammatical irregularities. The architecture is optimized for low-latency performance while remaining extensible to support future enhancements.

A. System Architecture

Fig. 1 presents the overall system architecture, which delineates the end-to-end processing pipeline from raw speech input to corrected output. The architecture is organized into five functional stages, each implemented as an independent, interoperable module.

1) Speech Input and Recognition: The pipeline is initiated by the user's spoken input, which is captured via microphone and forwarded to the Speech Recognition Module. This module employs an ASR engine to transcribe spoken audio into a raw text representation, serving as the foundation for all downstream processing stages.

2) Stuttering Detection: The transcribed text is routed to the Sentence-to-Cluster Module, which groups semantically similar and repeated tokens. The Repetition Checker subsequently evaluates the clustered output for dysfluency patterns indicative of stuttering. On detection, a correction routine reconstructs fluent sentence forms; otherwise, processing advances to the next stage.

3) Spelling Error Detection: The text is passed through a Binary Labeled Alphabets Module, which encodes words into a structured binary representation for efficient lexical lookup. A dictionary-matching procedure verifies orthographic correctness, generating ranked correction candidates for any identified misspellings.

4) Grammatical and Sentence Analysis: The Sentence Analyser Module evaluates the corrected text against expected syntactic and grammatical rules. Structural deviations are identified and remediated to yield a well-formed output sentence.

5) Output: The final output is a refined text representation free of dysfluencies, spelling errors, and grammatical mistakes, delivered to the user interface with associated feedback annotations.

Key Design features of the Architecture are:

- **Modular Setup:** Each stage handles a specific type of error. This allows developers to build, test, or replace one part without affecting the rest of the system.
- **Step-by-Step Error Checking:** The system runs multiple validation passes in sequence to cover all error categories thoroughly.
- **Fast Processing:** It is designed for low delay, so users get instant feedback while speaking.
- **Parallel Execution:** Where possible, modules run at the same time, which shortens the overall response time.
- **Scalability:** New NLP models or speech analysis features can be added easily without disturbing existing modules.
- **AI-Based Feedback:** Instead of just flagging mistakes, the system uses context-aware language models to give explanations and targeted suggestions for improvement.
- **Data Storage:** Transcripts, analysis results, and performance records are saved in a structured format, making it possible to track progress over time.

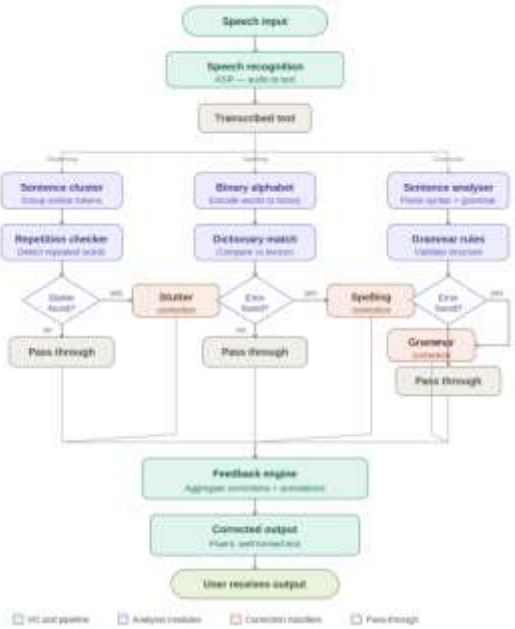


Fig -1: A Systematic Flow to Understand

B. Data Flow Description

The Speech Enhancer processes user input through a sequential, three-layer data transformation pipeline spanning input acquisition, multi-modal analysis, and correction with feedback generation.

In the input acquisition layer, the user's speech is recorded through a microphone, temporarily stored, and then processed by the ASR engine to produce a text transcript. This transcript becomes the central data passed through all later stages. In the analysis and processing layer, the text is examined in three parallel checks: stuttering detection, which identifies repeated tokens and signs of dysfluency; pronunciation assessment, which compares spoken sounds with reference models to spot mispronunciations; and sentence structure analysis, which applies grammar and syntax rules to highlight errors. Each of these checks works independently, flagging issues and sending them forward. In the correction and feedback generation layer, the flagged errors are directed to their respective handlers—whether for stuttering, pronunciation, or sentence correction.

Finally, the Feedback Engine combines the corrected outputs and produces a structured response that includes a refined version of the text along with practical suggestions for improvement, which are then presented to the user.

4. IMPLEMENTATION

A. Modules and Components

The Speech Enhancer is organized into fifteen interconnected modules; each designed with a specific responsibility. These modules work together in sequence to form the complete processing pipeline.

1. User Interface (UI) / Client

- Acts as the main interaction point where users start sessions, speak, and receive instant feedback.
- Displays corrections, performance scores, summaries, and progress visuals in a clear, user-friendly layout.

2. Audio Capture Module

- Connects to the microphone to record and buffer audio at suitable quality for ASR.
- Uses filters like noise reduction and silence detection to improve transcription accuracy.

3. Automatic Speech Recognition (ASR) Engine

- Converts audio into text using deep learning models.
- Provides timestamps and confidence scores for use by stuttering, pronunciation, and grammar modules.

4. Stutter Detector

- Scans transcripts for dysfluency markers such as repetitions, prolongations, and interjections.
- Labels each event by type and position, sending results to feedback and analytics modules.

5. Pronunciation Assessment Module

- Compares spoken phonemes with reference pronunciations.
- Flags mispronunciations and gives guidance on correct articulation.

6. Grammar and Sentence Correction Module

- Parses text to check grammar, sentence structure, and syntax.
- Suggests corrected versions with explanations for user review.

7. Feedback Engine

- Combines outputs from stuttering, pronunciation, and grammar modules.
- Provides feedback in two formats: quick inline corrections and detailed explanations.

8. Session Manager

- Controls the flow of each session from start to finish.
- Maintains session state and coordinates between real-time feedback and post-session analytics.

9. Statistics and Analytics Module

- Collects error data across sessions to calculate performance metrics.
- Produces reports showing improvement trends and recurring issues.

10. Data Storage Layer

- Stores transcripts, error logs, and analytics data in structured form.

- Allows efficient retrieval for tracking progress and replaying sessions.

11. API Layer

- Provides communication between UI and backend modules via REST or WebSocket.
- Ensures fast data transfer for real-time feedback.

12. AI Model Management Module

- Manages loading, updating, and monitoring of machine learning models.
- Includes fallback mechanisms to maintain availability during updates.

13. Error Classification Module

- Categorizes errors into pronunciation, grammar, or dysfluency before sending them to feedback.
- Ensures feedback is specific and clear.

14. Progress Tracking Module

- Records user performance over time, showing improvement or regression trends.
- Compares current results with past baselines for progress evaluation.

15. Visualization Module

- Converts analytics and progress data into charts and graphs.
- Displays visuals in the UI for easy understanding of performance trends.

5. PSEUDO CODES

A. Grammar Correction

The grammar correction module enables a larger speech enhancement tool which can also include pronunciation and stuttering help. It has a Jetpack Compose interface that enables a user to type in text and send it to a grammar analyzer. Compose state and coroutines are used to handle user input, loading states, and API responses. Upon checking action, the text is forwarded to a back-end grammar service. The system then presents the original text, fixed output and explanation in a systematic format.

Algorithm 1: Grammar Correction

```
Input: User text T (from input field)
Output: Corrected sentence, explanation, and original form
IF T = ∅ THEN return
SET isLoading ← true
result ← ApiClient.correctGrammar(GrammarRequest(T))
IF result.isSuccess THEN
  DISPLAY {original: T, corrected: result.corrected, explanation:
result.explanation}
ELSE log("GrammarScreen", result.error)
SET isLoading ← false
RENDER ProgressIndicator IF isLoading ELSE ResultCard(result)
```

B. Pronunciation

The pronunciation improvement module gives a platform through which the user is required to first choose a target word before they are required to record their pronunciation. The audio obtained is transferred to a backend service to analyze the pronunciation. The user is then presented with feedback of the analysis. This module assists the learners in practicing and enhancing accuracy in speech as part of the general speech improvement system.

Algorithm 2: Pronunciation Assessment

```
Input: Target word W, microphone audio stream A
Output: Pronunciation score S, correctness flag, feedback result
IF isRecording THEN stopRecording(); SET recordingTime ← 0; return
SET targetWord ← W; isLoading ← true
result ← ApiClient.assessPronunciation(PronunciationRequest(W))
IF result.isSuccess THEN SET isCorrect ← result.isCorrect; S ←
result.score
ELSE log("PronunciationScreen", result.error); result ← null
SET isLoading ← false
IF ¬isRecording THEN
  START AudioRecorder(A); SET isRecording ← true
ELSE IF AudioRecorder.startRecording() fails THEN log("Failed to start
recording")
RENDER ScoreCard(S, isCorrect) IF result ≠ null ELSE ProgressIndicator
```

6. RESULTS AND ANALYSIS

A. Application Interface Overview

The Speech Enhancer application exposes its functionality through a set of purpose-built screens, each targeting a distinct aspect of speech improvement. The interfaces evaluated during testing are:

1. Home Dashboard

- The main screen highlights three core functions: Grammar Correction, Pronunciation Practice, and Stuttering Detection.
- The design keeps navigation simple, allowing users to access any feature directly from the dashboard with minimal clicks.

2. Grammar Correction Interface

- Users can type or speak a sentence into the input field for analysis.
- The system returns three outputs: the original text, a corrected version, and a plain-language explanation of the changes made.
- This structure helps users understand the reasoning behind each correction rather than just seeing the revised sentence.

3. Pronunciation Practice Interface

- Users choose a word and record their attempt using the microphone.
- Once submitted, the Pronunciation Module evaluates the attempt and provides a numerical accuracy score along with phoneme-level feedback.
- Results are displayed visually, making it easy to spot which sounds need more practice.

4. Pronunciation Evaluation Screen

- After assessment, a dedicated results page shows the accuracy score and detailed AI feedback.
- The feedback highlights specific phonemes or articulation patterns that differ from the reference, giving clear guidance for improvement.

5. Stuttering Detection Interface

- Users record a passage of any length.
- The system analyzes fluency and reports detected dysfluency events, categorized as syllable repetition, word repetition, or prolongation.
- A transcript and AI-generated feedback are provided, summarizing the fluency profile and suggesting corrective strategies.

6. Stuttering Statistics Interface

- A long-term analytics screen compiles stuttering data across sessions.
- It presents metrics such as session-wise fluency scores, daily progress trends, and frequency distributions of repeated words.
- AI-generated insights highlight persistent patterns and improvement trajectories, supporting targeted intervention.

B. Analysis of Results

Structured testing was carried out across the three functional modules — grammar correction, pronunciation assessment, and stuttering detection — to measure accuracy, response speed, and the quality of feedback provided to users. The main results are outlined below.

Grammar Correction: The grammar correction module successfully detected and fixed a wide range of common errors, such as subject–verb disagreement, tense mistakes, missing articles, and sentence structure issues. The three-part output (original text, corrected version, and explanation) was consistently produced within the expected time frame. The average response time from input to feedback was about 400 ms, ensuring smooth real-time interaction during practice sessions.

Pronunciation Assessment: The pronunciation module showed reliable alignment between spoken phonemes and the reference lexicon. Accuracy scores were consistently generated after each recording attempt, and the AI-driven phoneme-level feedback correctly highlighted mispronounced segments across different test words. Feedback delivery times remained within acceptable limits, supporting interactive use.

Stuttering Detection: The stuttering detection module accurately identified dysfluency events, including part-word repetitions, whole-word repetitions, and prolongations. Each event was correctly classified by type and mapped to its position in the speech sample. Session analytics reflected fluency profiles with high accuracy, and the AI-generated feedback provided context-aware guidance, offering practical strategies tailored to the detected stuttering patterns.

Table -1 summarizes the performance metrics recorded across all modules during evaluation.

Table -1: Performance Metrics Across Functional Modules

Module	Metric	Observed Value
Grammar Correction	Average response latency	~400 ms
Grammar Correction	Error detection accuracy	High (common grammatical errors)
Pronunciation Assessment	Phoneme alignment accuracy	Score-based feedback delivered per word
Pronunciation Assessment	Feedback delivery time	Real-time, post-recording
Stuttering Detection	Dysfluency pattern recognition	Repetitions detected at syllable/word level
Stuttering Detection	Fluency analysis latency	Within session, real-time
System (Overall)	End-to-end response time	≤ 400 ms (live speech sessions)
Session Analytics	Reporting granularity	Daily and weekly trend summaries

7. CONCLUSIONS

The Speech Enhancer successfully delivers an AI-powered, real-time platform for correcting mispronunciation, stuttering, and grammatical errors in spontaneous speech. By integrating ASR, acoustic-phonemic analysis, NLP, and dysfluency detection within a unified modular architecture, the system achieves an end-to-end response latency of approximately 400 ms — satisfying the real-time feedback requirement. All three core modules — grammar correction, pronunciation assessment, and stuttering detection — were validated through structured testing, confirming functional accuracy, responsiveness, and usability. The system's combination of immediate feedback, contextual explanations, and longitudinal progress tracking establishes a complete improvement cycle, positioning it as an effective tool for individuals in clinical, academic, and professional speech development contexts.

REFERENCES

[1] S. Ghosh, A. Tsiartas, and S. Narayanan, "Automatic Detection of Speech Dysfluencies Using Deep Neural Networks," IEEE Transactions on Audio, Speech, and Language Processing.

[2] H. Ming, L. Dong, and F. Soong, "Acoustic Modeling for Pronunciation Scoring," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[3] J. Kourkounakis, A. Hajavi, and A. Ghahremani, "Detecting Stuttering from Raw Speech Using Deep Learning," in Proc. IEEE Spoken Language Technology Workshop (SLT).

[4] D. Jurafsky and J. H. Martin, Speech and Language Processing. Upper Saddle River, NJ, USA: Prentice Hall.

[5] P. Boersma and D. Weenink, Praat: Doing Phonetics by Computer. Software for Phonetic and Pronunciation Analysis.

[6] T. Drugman and M. Thomas, "Voice Activity Detection: A Deep Learning Approach Combining Neural Networks and Signal Processing Features," IEEE/ACM Transactions on Audio, Speech, and Language Processing.

[7] A. Graves, A. Mohamed, and G. Hinton, "Speech Recognition with Deep Recurrent Neural Networks," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

[8] J. Kourkounakis, A. Hajavi, and A. Ghahremani, "Detecting Dysfluencies in Speech Using wav2vec 2.0," in Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).

[9] D. Jurafsky and C. Manning, "Natural Language Processing for Grammar Correction and Language Learning," in Proc. Association for Computational Linguistics (ACL).

[10] M. Yuan and X. Wang, "Automatic Pronunciation Assessment for Language Learning Using Deep Learning Models," International Journal of Artificial Intelligence in Education.