Improving The Embedded AI Enhanced Learning Environment For College English Teaching

Yuanyang Lei*

The Department of General Education, Henan Medical College, Zhengzhou, 450000, China.

Corresponding author. E-mail: yuany_lei18@outlook.com

Received: Apr. 29, 2026; Accepted: July. 02, 2026

To enhance the English learning environment for college students, this study explores the integration of Artificial Intelligence (AI), focusing on pronunciation improvement through an AI-powered system. Traditional teaching methods face challenges such as large class sizes and limited personalized feedback, which hinder effective pronunciation development. To address these limitations, an AI-driven framework is proposed to provide real-time, targeted feedback for correcting pronunciation errors. The study utilizes English speech data collected from university students with varying proficiency levels, including beginner, intermediate, and advanced learners. The dataset comprises monologues, dialogues, and pronunciation drills, capturing common errors such as misarticulations, vowel and consonant inconsistencies, and stress-related issues. The data are annotated with word-level transcriptions and error labels, along with acoustic features such as pitch, duration, formant frequencies, and Mel-frequency cepstral coefficients (MFCCs). For error detection and classification, a Salp Swarm Integrated Dense Recurrent Neural Network (SS-DRNN) model is employed. The model enables automatic identification and correction of pronunciation errors while delivering adaptive feedback. Experimental results demonstrate high performance, achieving 98.25% accuracy and strong F1-score, precision, and recall. The proposed system enhances learning efficiency, reduces teacher workload, and supports scalable, personalized English instruction in higher education environments.

Keywords: Learning Environment, College English Teaching, Mel-Frequency Cepstral Coefficients (MFCCs), English Speech Data, Feedback.

http://dx.doi.org/10.6180/jase.202611_34.035

1. Introduction

English is the most widely spoken language worldwide, serving as a principal medium of communication across numerous industries [1]. College English Teaching (CET), a required course, has sparked widespread interest. However, a gap exists between existing college ETM and contemporary needs. The application of AI and BD technologies to improve CET reform is essential [2]. CET should match modern society's demand for senior English skills [3]. AI technology has shown impressive outcomes in education [4]. However, AI faces difficulties with embedded devices [5]. AI tools, including speech recognition and NLP, are currently used in English instruction [6]. Recent studies have shown strong performance of deep learning and transformer models in speech recognition and pronunciation assessment tasks [7]. Traditional teaching methods face difficulties such as high-class size and less customized feedback, negatively impacting pronunciation development. To overcome these issues, this research introduces an AI-based online platform that ensures students receive effective, targeted, realtime feedback on pronunciation error correction.

- The research presents an AI-based system that

gives real-time, personalized guidance for improving speech. It helps fix issues like inconsistent vowels and consonants, as well as stress in speech.

- It uses the SS-DRNN model to automatically assess pronunciation, detect errors, and classify mistakes, reducing teachers' workload and enabling large-scale learning.

- The AI method improves learning efficiency by measuring accuracy, F1-score, recognition rate, and precision, providing precise feedback and motivating students of all skill levels to improve their pronunciation.

The paper is organized as follows: Section 2 presents the methodology, Section 3 discusses the results and analysis, and Section 4 concludes with key findings and future directions.

2. Methodology

Data was acquired, preprocessed with word-level transcription, MFCC features extracted, then SS-DRNN detected errors (Fig. 1).

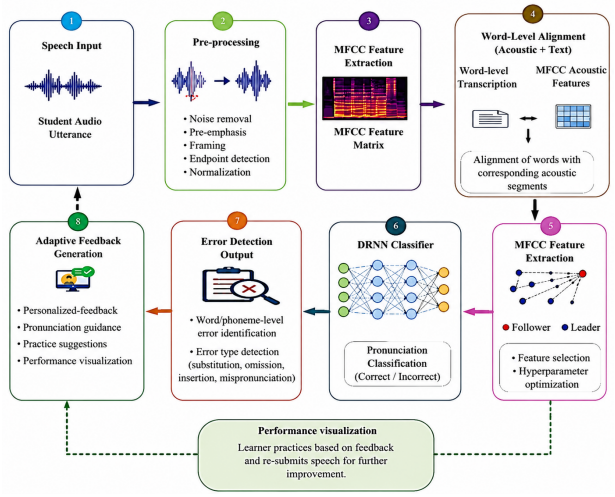

Fig. 1. SS-DRNN model's Overview

2.1. Data Collection

The data was gathered from the open-source Kaggle link: https://www.kaggle.com/datasets/ziya07/english-pronunciation-error-detection-dataset/data.

This dataset contains 200 English speech samples from beginner to advanced college students, with features like MFCCs and word-level transcriptions for AI-based pronunciation error detection.

2.2. Data preprocessing Using word-level transcriptions

Word-level transcriptions align each spoken word with text and acoustic features like pitch and MFCCs, enabling the system to detect, correct, and provide feedback on pronunciation errors. During pronunciation evaluation, word-level transcriptions were synchronized with pitch, duration, and MFCC features using frame-based analysis, enabling accurate error detection through alignment with spoken utterances.

2.3. Feature Extraction using MFCC

MFCC extraction involves preprocessing steps such as noise filtering, silence removal, normalization, framing, and windowing. Each frame is processed using a Hamming window, followed by FFT and a Mel filter bank to generate cepstral coefficients for reproducible speech feature representation. MFCC extracts sound power spectrum mimicking human hearing using Mel-scale filter banks with ear-mimicking center frequencies.

The connection between frequency in Hz (f) and Mel-scale frequency (Mel) is expressed in Equation (1), with a linear relationship below 1 kHz and logarithmic above.

$$\text{Mel} = 2595 \log_{10}\left(1 + \frac{f}{700}\right) \tag{1}$$

Where Mel is Mel-scale frequency, f is Hz. MFCCs extract pronunciation features via Melfilters, log-energy, then DCT reduces dimensionality (Equation 2).

$$C_n = \sum_{l=1}^{M} E_l \cos[n(l-0.5)\pi/M] \tag{2}$$

Where n is the cepstral coefficient index (which ranges from 1 to N), E_l is the log-energy at the l^{th} frequency band, M is the total number of Mel filters, and C_n is the n^{th} cepstral coefficient. MFCC features capture speech frequency and pronunciation variations among learners, enabling the SS-DRNN model to distinguish pronunciation errors across different proficiency levels.

2.4. Error Detection and Classification Using Salp Swarm Integrated Dense Recurrent Neural Network (SS-DRNN) for Pronunciation Issues

SS-DRNN combines DRNN and SS optimization to improve pronunciation error identification using MFCC features for adaptive CET feedback.

2.4.1. Dense Recurrent Neural Network (DRNN)

A DRNN advances standard RNNs by using memory blocks to eliminate vanishing gradients, enabling long-term dependencies for enhanced AI-based CET through tailored feedback and adaptive learning. The DRNN consists of

stacked recurrent layers with 128 and 64 neurons to capture temporal speech features. ReLU is used in hidden layers, softmax in the output layer, and dropout is applied to reduce overfitting and improve generalization.

DRNN predicts language patterns; DSSM extracts features for personalized feedback. Equation (3) generates b_s using tanh with biases and weights. Where, y_s represents the current input sequence at time step (s), out $_{s-1}$ denotes the output from the previous time step, (W) represents the weight matrices, and (b) indicates the bias terms used in the DRNN architecture.

$$b_s = \tanh\left(X_b \cdot y_s + V_b \cdot \text{ out }_{s-1} + c_b\right) \tag{3}$$

The current input gate y_s employs the sigmoid activation to examine how much of the new input should be examined as revealed in Equation (4).

$$j_s = \sigma\left(X_j \cdot y_s + V_j \cdot \text{ out }_{s-1} + c_j\right) \tag{4}$$

Equation (5) depicts the forget gate, which controls how much of the prior memory state is forgotten or retained.

$$f_s = \sigma\left(X_f \cdot y_s + V_f \cdot \text{ out }_{s-1} + c_f\right) \tag{5}$$

The output gate modifies final output (Equation 6), gates update internal state (Equation 7), output scales state with tanh (Equation 8). The input gate j_s controls b_s impact on $state_s$, and forget gate f_s controls state $_{s-1}$ retention or discarding.

$$o = \sigma\left(X_o \cdot y_s + V_o \cdot \text{ out }_{s-1} + c_o\right) \tag{6}$$

$$\text{state }_s = b_s \text{Q} j_s + f_s \text{Q state }_s \tag{7}$$

$$\text{out }_s = \tanh\left(\text{ state }_s\right) \text{Q} o_s \tag{8}$$

DRNN captures long-term relationships, prevents vanishing gradients, and enables adaptive feedback. The DRNN uses two hidden layers (128 and 64 neurons) with ReLU activation, and a softmax output layer, along with dropout to reduce overfitting.

2.4.2. *Salp Swarm (SS) optimization*

SS is a nature-inspired method optimizing neural networks for speech error detection using salp chain leader-follower behavior. Equation (9) updates the leader's location.

$$y_k^1 = \begin{cases} F_k + d_1\left(\left(ub_k - lb_k\right) d_2 + lb_k\right), & \text{if } d_3 \leq 0 \\ F_k - d_1\left(\left(ub_k - lb_k\right) d_2 + lb_k\right), & \text{if } d_3 > 0 \end{cases} \tag{9}$$

Where y_k^1 is the leader's position in k^{th} dimension with the food source F_k, lower/upper limits lb_k and ub_k, balances exploitation/exploration in Equation (10).

$$d_1 = 2e^{-\left(\frac{4t}{t_{\max}}\right)} \tag{10}$$

Where t and $t_{\max}$ show the present iteration and the number of max iterations. After updating the location of the leader, the SS starts to update the location of the followers by utilizing Equation (11).

$$y_k^j = \frac{1}{2}\left(y_k^j + y_k^{j-1}\right) \tag{11}$$

Where y_k^j represents the j follower location within the k^{th} dimension, where j is larger than 1 . Its structure is similar to jellyfish. Fig. 2(a) depicts a salp's form and Fig. 2(b) depicts the salp chain.

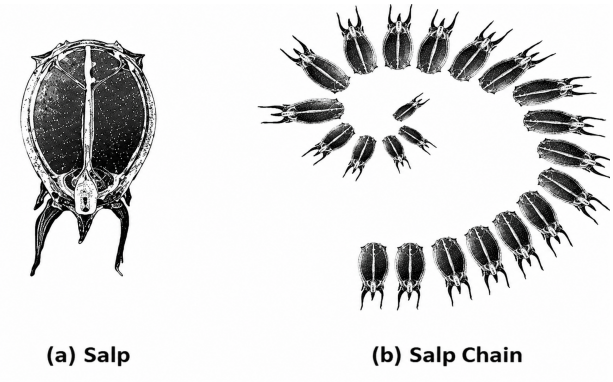

Fig. 2. (a) Form of Salp and (b) Salp Chain

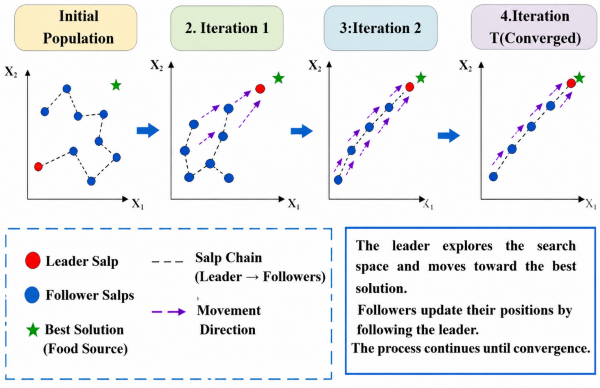

Fig. 3. Salp leader-follower movement in SSO.

Fig. 3 illustrates the Salp Swarm Algorithm, where the leader moves toward the best solution, and followers update their positions along the chain until convergence.

The SS optimizes learning algorithms rapidly, handling dynamic challenges to enhance customized CET learning experiences. Algorithm 1 describes SS-DRNN's final phases. Swarm Optimization offers faster convergence and

better exploration-exploitation than PSO/GWO, improving MFCC feature selection and classification accuracy.

Algorithm 1. Salp Swarm Integrated Dense Recurrent Neural Network (SS-DRNN)

#Initialize the SS–DRNN system
Step 1: Set up DRNN with optimized features from SS.
#Initialize DRNN (LSTM layers, DSSM layers)
For each training epoch:
Input features to DRNN (Extracted MFCC features and optimized parameters)
Apply forward propagation through the LSTM layers.
Apply the input, forget, and output gates (Equations (3)–(Apply the input, forget, and output gates (Equations 3 to 6)
Update internal state and output layers (Equations 7 and 8)
Calculate error (loss function) and update model parameters
Step 2: Classify pronunciation errors and provide personalized feedback to students
#Initialize Population Salp Swarm Optimization (SS)
Step 3: Initialize population X (salp positions)
Repeat until maximum iterations ($t < t_{\max}$):
Step 4: Compute Objective Function (Model Evaluation) for each salp in X:
Evaluate objective function (fitness) for salp (error rate, accuracy, etc.)
Step 5: Update the Best Solution F_{best}
Update the best salp (F _best) based on objective function
Step 6: Update Exploration and Exploitation Parameters (d_1) Calculate d_1 using Equation (10)
Step 7: Update Leader Position using Salp Swarm Position Update Rule
for leader salp in X:
Update leader position using Equation (9)
Step 8: Update Followers' Positions
for follower salp in X:
Update follower position using Equation (11)
Return the final SS-DRNN model

SS-DRNN integrates SS exploration with DRNN learning to improve error identification, pronunciation accuracy, personalized assistance, and adaptability for diverse learners in CET.

3. Results and discussion

This section presents the experimental design evaluating SS-DRNN against existing methods, demonstrating its capability in detecting and correcting faults.

3.1. Experimental Setup

Intel Core i9, 32GB RAM, 1TB SSD, Ubuntu 20.04, and voice libraries enable real-time SS-DRNN pronunciation feedback. Although the dataset contains 200 speech samples, dropout regularization, early stopping, and an 80:20 train-test split were applied to reduce overfitting.

Table 1. Distribution of Speech Samples Across Learner Proficiency Levels

Proficiency Level	Number of Samples	Percentage (%)
Beginner	70	35
Intermediate	65	32.5
Advanced	65	32.5
Total	200	100

Table 1 shows balanced beginner, intermediate, and advanced samples, ensuring fair SS-DRNN evaluation across proficiency levels.

Table 2. Train, Validation, and Test Split of the Pronunciation Dataset

Dataset Portion	Samples	Percentage (%)
Training	140	70
Validation	30	15
Testing	30	15
Total	200	100

As shown in Table 2, the dataset was divided into training (70%), validation (15%), and testing (15%) subsets for evaluation.

GPU acceleration enables real-time feedback. The SS-DRNN model was trained using Adam optimizer (0.001 learning rate), batch size 32, and 100 epochs. Early stopping prevented overfitting.

Table 3. Five-Fold Cross-Validation Results of the Proposed SS-DRNN Framework

Fold	Accuracy (%)
Fold 1	97.84
Fold 2	98.11
Fold 3	98.43
Fold 4	98.02
Fold 5	97.95
Mean ± Std	98.07 ± 0.20

As shown in Table 3, SS-DRNN achieved $98.07\% \pm 0.20$ accuracy across folds, proving robust generalization despite limited data.

3.2. Performance Metrics

All comparative models were implemented and evaluated using the same dataset, preprocessing, training settings,

and metrics for fair comparison with SS-DRNN. The research compared existing methods-Hybrid CTC-attention MDD [8], TDNN [9], and CNN [10]- using accuracy, recall, F1-score, and precision. Recent advances have also explored maximum F1-score training strategies for end-to-end mispronunciation detection [11]. Recognition rate is the ratio of correctly identified samples, indicating system effectiveness in learning assessment and feedback quality.

3.3. Accuracy

Accuracy measures the percentage of correctly categorized instances, evaluating system predictions like grammatical error detection; high accuracy indicates proper learning alignment.

Table 4. Accuracy for Pronunciation Error Detection using SS-DRNN

Methods	Accuracy (%)
Hybrid CTC-attention MDD [8]	52.77
CNN [10]	97.17
SS-DRNN [Proposed]	98.25

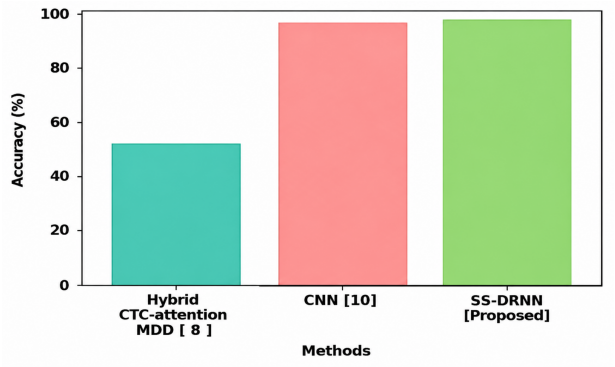

Fig. 4. Graphical Representation of Accuracy for SS-DRNN

As shown in Table 4 and Fig. 4, SS-DRNN achieves 98.25% accuracy, exceeding Hybrid CTC-attention MDD [8] (52.77%) and CNN [10] (97.17%), demonstrating its advantage in pronunciation error detection. All baselines used identical settings. Hybrid CTC-attention MDD's lower accuracy may stem from dataset limitations and feature learning differences.

3.4. Recall

Recall is the percentage of genuine positive forecasts among all real positive cases. It assesses the model's ability to identify all relevant faults, ensuring critical errors are not overlooked.

As shown in Table 5 and Fig. 5, SS-DRNN achieves 94.37% recall, exceeding Hybrid CTCattention MDD [8] (72.20%) and TDNN [9] (90.17%), identifying a greater proportion of pronunciation errors.

Table 5. Recall for Pronunciation Error Detection using SS-DRNN

Methods	Recall (%)
Hybrid CTC-attention MDD [8]	72.20
TDNN [9]	90.17
SS-DRNN [Proposed]	94.37

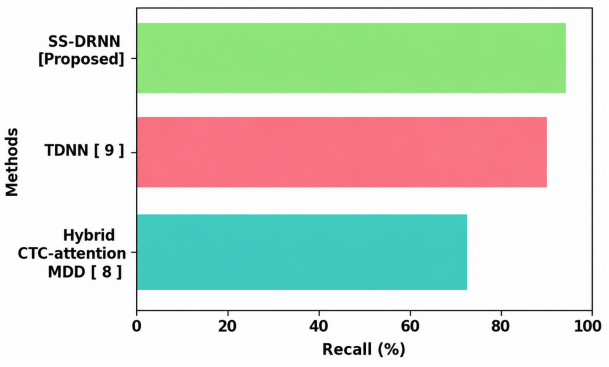

Fig. 5. Graphical Representation of Recall for SS-DRNN

3.5. Precision

Precision measures actual positives among predicted positives, reducing false positives and increasing feedback dependability.

Table 6. Precision for Pronunciation Error Detection using SS-DRNN

Methods	Precision (%)
Hybrid CTC-attention MDD [8]	33.75
TDNN [9]	89.25
SS-DRNN [Proposed]	94.83

As shown in Table 6 and Fig. 6, SS-DRNN achieves the best precision (94.83%), outperforming Hybrid CTC-attention MDD [8] (33.75%) and TDNN [9] (89.25%), reducing false positives.

3.6. F1-Score

The F1 score combines precision and recall to assess model performance, balancing missed errors and wrong corrections for accurate, comprehensive student feedback.

As shown in Table 7 and Fig. 7, SS-DRNN achieves the highest F1-score (95.81%), outperforming Hybrid CTC-attention MDD (46.00%) and TDNN (92.14%).3.7 Confusion Matrix Analysis

Fig. 8 shows confusion matrix with diagonal (22,21,23) correct and only 4 misclassifications across all levels.3.8

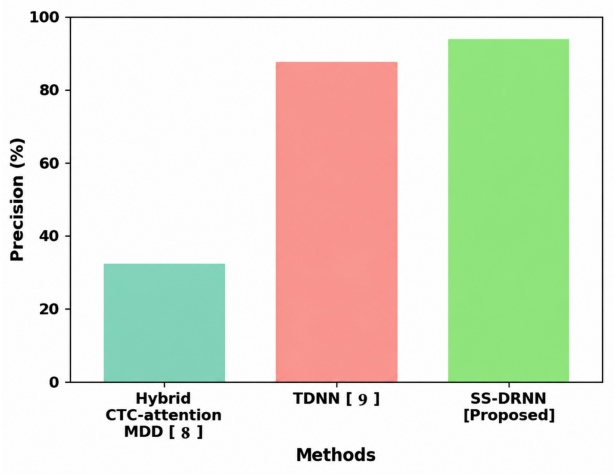

Fig. 6. Graphical Representation of precision for SS-DRNN

Table 7. F1-Score for Pronunciation Error Detection using SS-DRNN

Methods	F1-score (%)
Hybrid CTC-attention MDD [8]	46.00
TDNN [9]	92.14
SS-DRNN [Proposed]	95.81

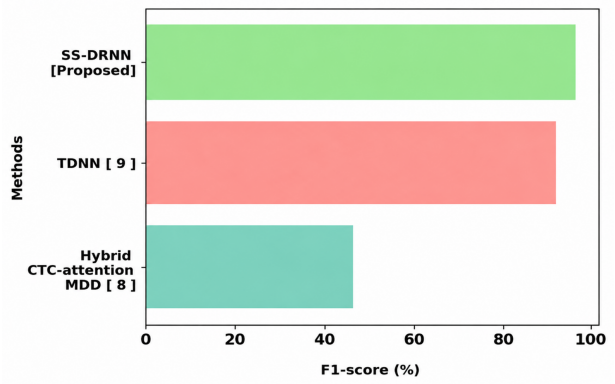

Fig. 7. Graphical Representation of F1-score for SS-DRNN

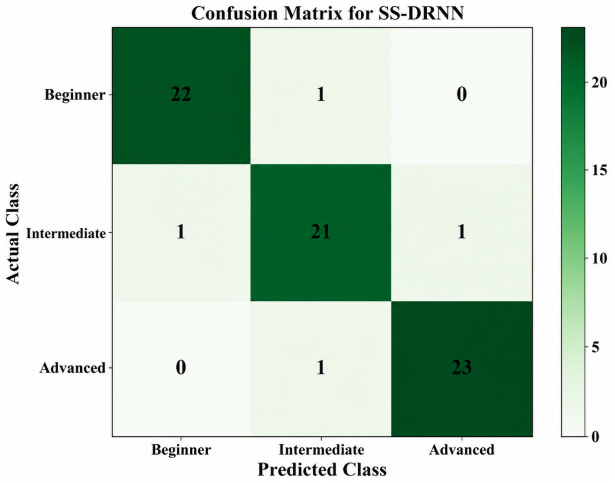

Fig. 8. Confusion Matrix for SS-DRNN Pronunciation Classification

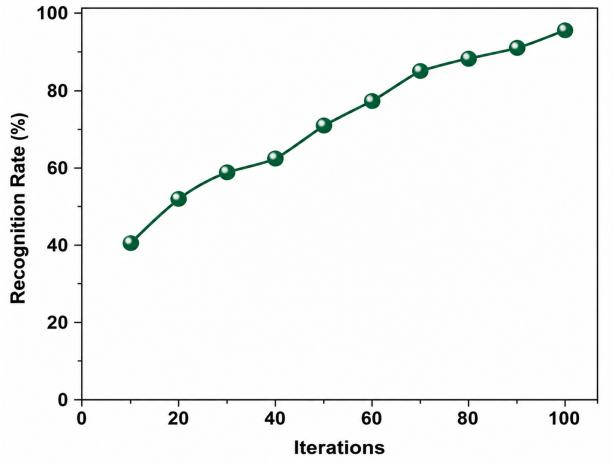

Fig. 9. Graphical Representation of Recognition Rate for SS-DRNN

Model Reproducibility

Table 8 shows five independent runs of SS-DRNN. Low standard deviations (accuracy: 0.09%, F1-score: 0.23%) confirm reproducibility

3.7. Recognition Rate

Recognition rate (97.26% for SS-DRNN in Fig. 9) is the percentage of correctly recognized examples, ensuring effective speech problem identification.

SS-DRNN detects errors efficiently, reduces teacher workload, and personalizes learning. Learner proficiency levels guide adaptive feedback for inclusive instruction and improved engagement. SS-DRNN enables personalized, dynamic English learning environments through enhanced pronunciation assessment. The proposed SS-DRNN framework is computationally efficient and supports real-time assessment. Compared with Transformer [12] and Conformer [13], SS-DRNN offers lower computational complexity and reduced processing overhead. Transformer models address non-English phoneme mispronunciation (e.g., Arabic [14]). Conformer-based architectures like Sampleformer [15] are proposed for automatic speech recognition.

Table 9 presents the SS-DRNN performance across beginner, intermediate, and advanced learners, demonstrating consistent adaptability and effectiveness across different proficiency level

Table 8. Repeated Experimental Results of the Proposed SS-DRNN Framework for Pronunciation Error Detection

Run	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Run 1	98.11	94.52	94.08	95.42
Run 2	98.34	94.75	94.36	95.66
Run 3	98.25	94.83	94.37	95.81
Run 4	98.18	94.41	94.02	95.21
Run 5	98.29	94.67	94.21	95.54
Mean ± Std	98.23 ± 0.09	94.64 ± 0.17	94.21 ± 0.16	95.53 ± 0.23

Table 9. Performance Metrics by Proficiency Level

Proficiency Level	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Beginner	97.84	94.12	93.87	94.35
Intermediate	98.09	94.55	94.2	95.02
Advanced	98.62	95.21	94.96	96.02

4. Conclusion

The incorporation of an AI-driven framework to improve college students' pronunciation learning was effectively demonstrated. By utilizing the SS-DRNN model, which combined SS optimization and DRNN, the system effectively detected and classified pronunciation errors based on MFCC features. The SS-DRNN approach provided real-time, targeted feedback, addressing the challenges of large class sizes and limited personalized instruction. Evaluation metrics such as F1-score (95.81%), recall (94.37%), accuracy (98.25%), precision (94.83%), and recognition rate (97.26%) highlighted the system's effectiveness, ultimately reducing teacher workloads and offering students personalized feedback for improved pronunciation. This AI-powered system contributed to more efficient and scalable CET in college environments. The limitations of this research include dependence on speech data from university-level students, which cannot completely represent different linguistic backgrounds, and the possibility of errors in recognizing complex pronunciation flaws. Limited dataset size may affect generalization; future work will use larger, diverse datasets. Accent variability, phonetic diversity, and native-language interference may impact error detection. SS-DRNN requires multilingual validation to improve robustness across languages, accents, and learner categories. Future research might broaden the dataset to include a larger spectrum of learners, enhance error detection accuracy, and investigate the incorporation of additional AI models to further adapt the learning experience in CET.

Data availability

https://www.kaggle.com/datasets/ziya07/english-pronunciation-error-detection-dataset/data.

Conflicts of interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

Funding statement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author contributions

Yuanyang Lei conceptualized the study, designed the methodology, conducted experiments, analyzed the results, and prepared the manuscript.

Ethical approval

Not applicable

Consent to participate

Not applicable

Consent for publication

Not applicable

Competing interests

The author declares no competing interests.

Code availability

The code used in this study is available from the corresponding author upon reasonable request.

Acknowledgement

The author would like to thank the Department of General Education at Henan Medical College for their support in conducting this research.

References

[1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schluter, and S. Watanabe, (2024) *"End-to-End Speech Recognition: A Survey"* **IEEE/ACM Transactions on Audio, Speech, and Language Processing *32***: 325–351. DOI: 10.1109/TASLP.2023.3328283.

[2] X. Li and X. Huang, (2024) *"Improvement and Optimization Method of College English Teaching Level Based on Convolutional Neural Network Model in an Embedded Systems Context"* **Computer-Aided Design and Applications *21***(S8): 212–227. DOI: 10.14733/cadaps.2024.S8.212-227.

[3] F. Wu, Y. Chen, and D. Han, (2022) *"Development countermeasures of college English education based on deep learning and artificial intelligence"* **Mobile Information Systems *2022***: 1–10. DOI: 10.1155/2022/8389800.

[4] L. Huang, (2022) *"An empirical study of integrating information technology in English teaching in artificial intelligence era"* **Scientific Programming *2022***: 1–12. DOI: 10.1155/2022/6775097.

[5] L. Geng, (2021) *"Evaluation model of college English multimedia teaching effect based on deep convolutional neural networks"* **Mobile Information Systems *2021***: 1–11. DOI: 10.1155/2021/1874584.

[6] M. N. Daoud and M. A. Ben Messaoud. "Phoneme-level mispronunciation detection in Quranic recitation using Shallow Transformer". In: *Proceedings of the Third Arabic Natural Language Processing Conference*. 2025, 457–463. DOI: 10.18653/v1/2025.arabicnlp-sharedtasks.63.

[7] H. Li, C. Tang, X. Yue, and X. Li, (2025) *"Sentence-level consistency of conformer-based pre-training distillation for Chinese speech recognition"* **Frontiers in Communications and Networks *6***(1): 1662788–1662796. DOI: 10.3389/frcmn.2025.1662788.

[8] K. Li, X. Qian, and H. Meng, (2016) *"Mispronunciation detection and diagnosis in L2 English speech using multi-distribution deep neural networks"* **IEEE/ACM Transactions on Audio, Speech, and Language Processing *24***(12): 2528–2539. DOI: 10.1109/TASLP.2016.2621675.

[9] B. C. Yan, H. W. Wang, S. W. F. Jiang, F. A. Chao, and B. Chen. "Maximum F1-score training for end-to-end mispronunciation detection and diagnosis of L2 English speech". In: *IEEE International Conference on Multimedia and Expo (ICME)*. 2022, 1–6. DOI: 10.1109/ICME52920.2022.9858931.

[10] H. Liu, (2021) *"College oral English teaching reform driven by big data and deep neural network technology"* **Wireless Communications and Mobile Computing *2021***: 1–10. DOI: 10.1155/2021/8389469.

[11] L. Peng, Y. Gao, R. Bao, Y. Li, and J. Zhang, (2023) *"End-to-End Mispronunciation Detection and Diagnosis Using Transfer Learning"* **Applied Sciences *13***(11): 6793. DOI: 10.3390/app13116793.

[12] Ş. S. Çalık, A. Küçükmanisa, and Z. H. Kilimci, (2024) *"A novel framework for mispronunciation detection of Arabic phonemes using audio-oriented transformer models"* **Applied Acoustics *215***: 109711. DOI: 10.1016/j.apacoust.2023.109711.

[13] Z. Fan, X. Zhang, M. Huang, and Z. Bu, (2024) *"Sampleformer: An efficient conformer-based neural network for automatic speech recognition"* **Intelligent Data Analysis *28***(6): 1647–1659. DOI: 10.3233/IDA-230612.

[14] H. Kheddar, M. Hemis, and Y. Himeur, (2024) *"Automatic speech recognition using advanced deep learning approaches: A survey"* **Information Fusion *109***: 102422. DOI: 10.1016/j.inffus.2024.102422.

[15] M. A. H. Wadud, M. Alatiyyah, and M. F. Mridha, (2023) *"Non-Autoregressive End-to-End Neural Modeling for Automatic Pronunciation Error Detection"* **Applied Sciences *13***(1): 109. DOI: 10.3390/app13010109.