information

Article

Arabic Mispronunciation Recognition System Using LSTM Network

Abdelfatah Ahmed [1,*], Mohamed Bader [2], Ismail Shahin [2], Ali Bou Nassif [3], Naoufel Werghi [1] and Mohammad Basel [3]

[1] Department of Electrical and Computer Engineering, Khalifa University of Science Technology and Research, Abu Dhabi 127788, United Arab Emirates
[2] Department of Electrical Engineering, University of Sharjah, Sharjah 27272, United Arab Emirates
[3] Department of Computer Engineering, University of Sharjah, Sharjah 27272, United Arab Emirates
* Correspondence: 100059689@ku.ac.ae

Abstract: The Arabic language has always been an immense source of attraction to various people from different ethnicities by virtue of the significant linguistic legacy that it possesses. Consequently, a multitude of people from all over the world are yearning to learn it. However, people from different mother tongues and cultural backgrounds might experience some hardships regarding articulation due to the absence of some particular letters only available in the Arabic language, which could hinder the learning process. As a result, a speaker-independent and text-dependent efficient system that aims to detect articulation disorders was implemented. In the proposed system, we emphasize the prominence of "speech signal processing" in diagnosing Arabic mispronunciation using the Mel-frequency cepstral coefficients (MFCCs) as the optimum extracted features. In addition, long short-term memory (LSTM) was also utilized for the classification process. Furthermore, the analytical framework was incorporated with a gender recognition model to perform two-level classification. Our results show that the LSTM network significantly enhances mispronunciation detection along with gender recognition. The LSTM models attained an average accuracy of 81.52% in the proposed system, reflecting a high performance compared to previous mispronunciation detection systems.

Keywords: artificial intelligence; deep learning; long short-term memory; Mel-frequency cepstral coefficients; pronunciation error; recurrent neural network

Citation: Ahmed, A.; Bader, M.; Shahin, I.; Nassif, A.B.; Werghi, N.; Basel, M. Arabic Mispronunciation Recognition System Using LSTM Network. *Information* **2023**, *14*, 413. https://doi.org/10.3390/info14070413

Academic Editor: Katsuhide Fujita

Received: 27 April 2023
Revised: 13 July 2023
Accepted: 14 July 2023
Published: 16 July 2023

1. Introduction

The widespread use of CALL (computer-assisted language learning) systems attests to their success in helping people improve their language and speech skills. CALL is predominantly concerned with addressing pronunciation errors in non-native speakers' speech. Accurate mispronunciation detection, voice recognition, and accurate pronunciation evaluation are all activities that may be accomplished with CALL. Similarly, there are a plethora of studies on speech processing that have been implemented in numerous languages with the aim of facilitating language learning. Breakthroughs in AI and other areas of computer science have permitted extensive study of CALL. Due to the inability of their mouth muscles to articulate the intricacies of a particular language, speakers of different languages are prone to committing pronunciation problems while speaking a particular language. For this reason, academics often explore mispronunciation in English, Dutch, and French, while Arabic literary studies are scarce. However, Arabic studies have increased in recent years. Arabic, the most widely spoken language with approximately 290 million native speakers and 132 million non-native speakers, and one of the six official languages of the United Nations (UN), has two major dialects, Classical Arabic (CA) and Modern Standard Arabic (MSA). Classical Arabic is the language of the Quran, whereas Modern Standard Arabic is a modified form of the Quran used in daily conversation. In order to retain the right meaning of the phrases, the rules for pronouncing the Quranic language are quite well-defined. This study emphasizes the recognition of incorrect pronunciations of Arabic letters [1]. In addition, there is a distinction between letters and sounds; sounds are pronounced and formed with our articulators, while letters are written by the individual. As a result, we gathered the most mispronounced Arabic phonemes from various speech pathologists; these phonemes will be represented as letters. Table 1 illustrates the most mispronounced Arabic letters in the field of pronunciation. Therefore, the effect of employing long short-term memory as a classifier blended with Mel-frequency cepstral coefficients as the feature extractor is observed. The LSTM network is well suited for speech recognition due to its ability to model the complex temporal relationships in speech signals, adapt to variations in the input data, and handle sequences of variable lengths.

Table 1. Most common disordered Arabic letters.

No.	Arabic Letter	Phonetic Symbol
1	س	/s/
2	ر	/r/
3	ق	/q/
4	ج	/ʒ/
5	ك	/k/
6	خ	/x/
7	غ	/ɣ/
8	ض	/ḏ/
9	ح	/ḥ/
10	ص	/Ṣ/
11	ط	/ŧ/
12	ظ	/ə/
13	ذ	/ð/

Despite the advancements in existing research, to our knowledge, no studies address the classification of pronunciation errors in conjunction with the gender of the speaker. This lack of gender-based analysis in Arabic mispronunciation recognition creates a critical gap in our understanding and the effectiveness of language learning systems. Motivated by this, our study proposes a two-level detection framework that can identify both mispronunciation and the speaker's gender. This focus is particularly crucial considering the potential influence of gender on pronunciation and the resultant implications for personalized language learning approaches. By leveraging the capabilities of deep learning, specifically long short-term memory (LSTM) networks, we aim to improve the accuracy and robustness of mispronunciation recognition systems, thus contributing to the evolution of computer-assisted language learning (CALL) systems.

The substantial contributions of the exhibited framework are presented as follows:

- A two-stage diagnostic system for recognizing the mispronunciation of Arabic letters using MFCC features and the LSTM model was implemented.
- To the best of our knowledge, this is the first attempt to recognize both mispronunciation and the gender of the speaker through two-level classification.
- The first benchmark for mispronunciation prediction for both native and non-native Arabic speakers is provided in the paper.
- Grid search was utilized for the proposed framework to identify the optimum model hyperparameters.

- Empirical analysis was conducted to investigate the impact of speech features on the mispronunciation recognition system.

2. Literature Review

Numerous mispronunciation detection and diagnosis (MD&D) research methods attempt to utilize both auditory and linguistic input elements. However, the absence of a substantial quantity of annotated training data at the phoneme level constrains the improvement of performance. Recent advancements in speech recognition, such as the LAS-Transformer, which is an enhanced Transformer based on the Local Attention Mechanism, suggest potential applications in improving mispronunciation detection by utilizing more advanced attention mechanisms in processing auditory inputs [2]. To construct a more robust MD&D system, authors combined the embedding properties of acoustic, phonetic, and linguistic data, abbreviated as APL. In [3], the suggested method has a detection accuracy of 9.93% higher than the baseline, a diagnosis error rate of 10.13% lower, and an F-measure of 6.17% higher than the baseline. These results were acquired through experimental work performed on the L2-ARCTIC database. The authors introduced a phoneme-level MD&D system that utilizes acoustic embedding (acoustic characteristics), phonetic embedding, and linguistic embedding (canonical phoneme sequence) as inputs in order to predict the spoken phoneme sequence. In [4], an acoustic-graphemic phonemic model (AGPM) utilizing multi-distribution deep neural networks (MD-DNNs) is proposed, whose input features consist of acoustic data, graphemes, and canonical transcriptions (encoded as binary vectors). The AGPM is capable of intuitively modeling both grapheme-to-likely-pronunciation and phoneme-to-likely-pronunciation conversions, which are incorporated into acoustic modeling. Using the AGPM, in this paper, a unified MDD framework that functions similarly to freephone recognition is constructed. Experiments indicate that the proposed technique yields an 11.1% phone error rate (PER). The false rejection rate (FRR), the false acceptance rate (FAR), and the diagnostic error rate (DER) for MDD are 4.6%, 30.5%, and 13.5%, respectively. While this model showed promising results, it is notable that it resulted in relatively high false rejection and acceptance rates, indicating room for improvement in model accuracy.

Computer-aided pronunciation training systems necessitate reliable automated pronunciation error detection techniques to recognize human faults. Yet, the overall number of mispronounced speech data utilized to train these algorithms and their manual annotation reliability greatly affect their performance [5]. To resolve this issue, the authors in [5] employed anomaly detection methods to identify mispronunciation. Their anomaly detection model was the One-Class SVM, using phoneme-specific models. A bank of binary DNN speech attribute detectors retrieved manners and locations of articulation for each model. Multi-task learning and dropout were implemented to reduce DNN speech attribute detector overfitting. The model was trained using the WSJ0 and TIMIT standard datasets, which contain solely native English speech data, and then assessed it using three datasets: a native English speaker corpus with fake mistakes, a foreign-accented speech corpus, and a children's disordered speech corpus. Lastly, the proposed approach was compared to the usual goodness-of-pronunciation (GOP) algorithm to prove its efficacy. The technique lowered false-acceptance and false-rejection rates by 26% and 39% compared to the GOP technique. Furthermore, in [6], the authors presented a speech recognition system capable of detecting mispronunciations. The dataset contains 89 students, 46 of whom are female. Ten times 28 Arabic phonemes are voiced. MFCCs are retrieved from 890 utterances for modeling with five different machine-learning models. K nearest neighbor (KNN), support vector machine (SVM), naive Bayes, multi-layer perceptron (MLP), and random forest (RF) are some of them. The experimental findings show that the random forest approach achieves an accuracy rate of 85.02%, which is higher than that of other machine learning models. Shareef et al. [7] emphasize the extensive comparison of feature extraction algorithms for the purpose of identifying impaired Arabic speech. The feature extraction approach is based on several wavelet transformation variants. LSTM and CNN-LSTM models are built to identify the impairment in Arabic speech. The combination of MFCCs and LSTM achieves the highest classification accuracy (93%), followed by CNN-LSTM (91%).

Table 2 summarizes the various approaches used in these studies for mispronunciation detection and diagnosis. However, these studies have largely overlooked the influence of gender on pronunciation, an aspect that may hold key insights for personalized language learning. To address these limitations, our paper proposes a two-level detection system that not only recognizes the mispronunciation of Arabic letters but also identifies the gender of the speaker. By implementing this dual-level approach, our model aims to provide a more comprehensive understanding of Arabic pronunciation errors, thus contributing to more effective personalized language learning strategies.

Table 2. Comparative analysis of different mispronunciation detection and diagnosis methods.

Work	Classification Algorithm	Data Utilized	Performance Metrics	Results
Ye et al. [3]	Acoustic, Phonetic, and Linguistic Data Embedding	L2-ARCTIC database	Detection Accuracy, Diagnosis Error Rate, F-Measure	Accuracy: 9.93% DER: 10.13% F-measure: 6.17%
Li et al. [4]	Acoustic-Graphemic Phonemic Model (AGPM) Using Multi-Distribution Deep Neural Networks (MD-DNNs)	Not specified	Phone Error Rate (PER), False Rejection Rate (FRR), False Acceptance Rate (FAR), Diagnostic Error Rate (DER)	PER: 11.1%, FRR: 4.6%, FAR: 30.5%, DER: 13.5%
Shahin and Ahmed [5]	One-Class SVM, DNN Speech Attribute Detectors	WSJ0 and TIMIT standard datasets	False-Acceptance Rate, False-Rejection Rate	Lowered FAR and FRR by 26% and 39% compared to the GOP technique
Arafa et al. [6]	Random Forest (RF)	89 students' Arabic phoneme utterances	Accuracy	85.02%
Shareef and Al-Irhayim [7]	LSTM and CNN-LSTM	Not specified	Classification Accuracy	LSTM: 93%, CNN-LSTM: 91%

The rest of the paper is arranged as follows: Section 3 provides the methodology, which illustrates the utilized framework. Section 4 presents the experimental setup. Section 5 discusses the experimental results. Section 6 concludes our work.

3. Methodology

3.1. Speech Corpus

Throughout this manuscript, we collected our database since there is no standard database available that suits our requirements. Our corpus was obtained via the implementation of two distinctive and prearranged sessions, specifically denoted as the "training" and "testing" sessions. The first session comprises 30 native Arabic speakers (15 male and 15 female). This part of the dataset provides us with the correct pronunciation. Each speaker was asked to utter different Arabic letters and words several times in a neutral talking condition. The total collected number of utterances used for training was 7800 ((30 speakers $\times$ 13 letters $\times$ 5 repetitions/letter) + (30 speakers $\times$ 39 words $\times$ 5 repetitions/word)). The letters and words are displayed in Table 3. On the other hand, the testing session comprises 11 native Arabic speakers and 42 non-native Arabic speakers (23 male and 29 female). This part of the dataset provides us with correct and incorrect pronunciation. Individual participants were prompted to enunciate different Arabic letters and words three times only. The total collected number of utterances that were used for testing was 3120. Our database was recorded by a speech acquisition board using a 16-bit linear coding A/D converter and sampled at a sampling rate of 44.1 kHz. For noise reflection reduction, we utilized microphone isolation shield sound absorber foam. Furthermore, Figure 1 illustrates the country distribution of our speakers. As we can see, our dataset comprises speakers from 17 different countries. Also, Figure 2 shows that the speaker's age spans 9 to 65 years. Based on the method of acquiring the dataset, our system is speaker-independent and text-dependent. Furthermore, according to a speech-language therapist, the effective method of evaluating mispronunciation is by uttering the letter alone and the letter in three different locations in the word (beginning, middle, and end). It should be noted that this dataset is utilized to detect only two classes, correct or incorrect pronunciation. Along with that, it can also be used for gender recognition tasks. Consequently, we proposed a gender and mispronunciation recognition system.

Table 3. Arabic mispronunciation database.

Letters	Word Initial	Word Medial	Word Final
/ʒ/ج	/ʒbl/ جبل	/nʒm/ نجم	/zwʒ/ زوج
/ḥ/ح	/ḥbl/ حبل	/lḥm/ لحم	/mlḥ/ ملح
/x/خ	/xjmh/ خيمة	/nxl/ نخل	/jdwx/ يدوخ
/ð/ذ	/ðʔb/ ذئب	/ʔðhb/ اذهب	/mnð/ منذ
/r/ر	/rml/ رمل	/brmjl/ برميل	/ʔmr/ أمر
/s/س	/sjf/ سيف	/nsf/ نسف	/jlbs/ يلبس
/Ṣ/ص	/Ṣjf/ صيف	/bṢl/ بصل	/lṢ/ لص
/d/ض	/db/ ضب	/mdʔ/ مضى	/nbd/ نبض
/ŧ/ط	/ŧjb/ طيب	/mnŧʔd/ منطاد	/hbwŧ/ هبوط
/∂/ظ	/∂l/ ظل	/m∂lh/ مظلة	/mlfw∂/ ملفوظ
/ɣ/غ	/ɣʔbh/ غابة	/bbɣʔʔ/ ببغاء	/blɣ/ بلغ
/q/ق	/qlm/ قلم	/lqmh/ لقمة	/jtfq/ يتفق
/k/ك	/khf/ كهف	/mkʔn/ مكان	/djk/ ديك

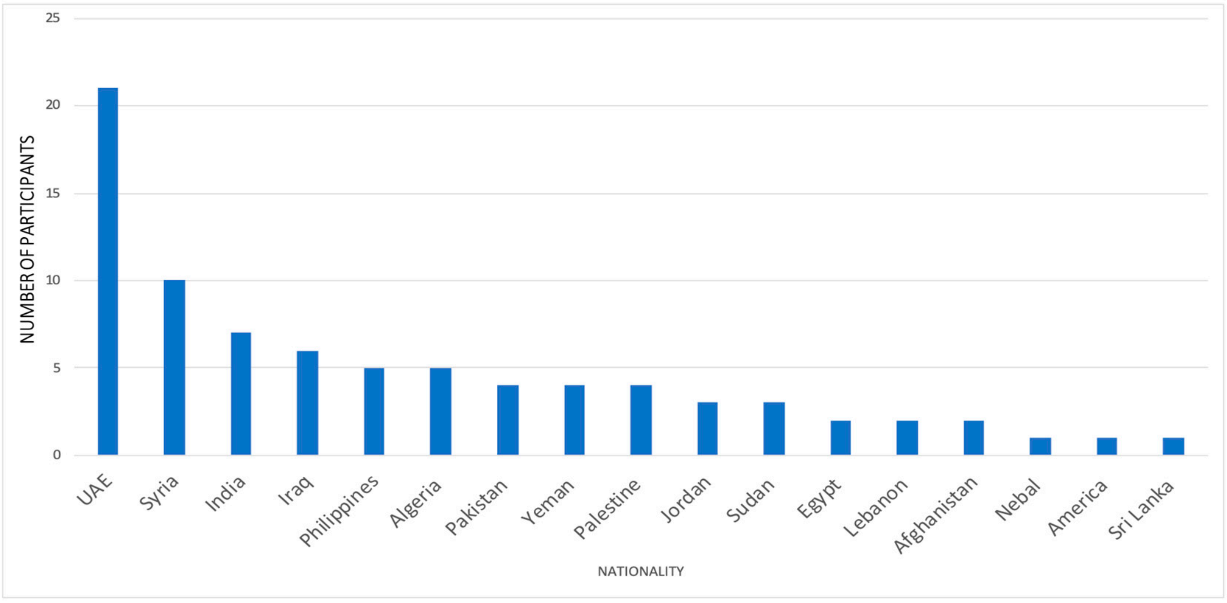


Figure 1. Country distribution.

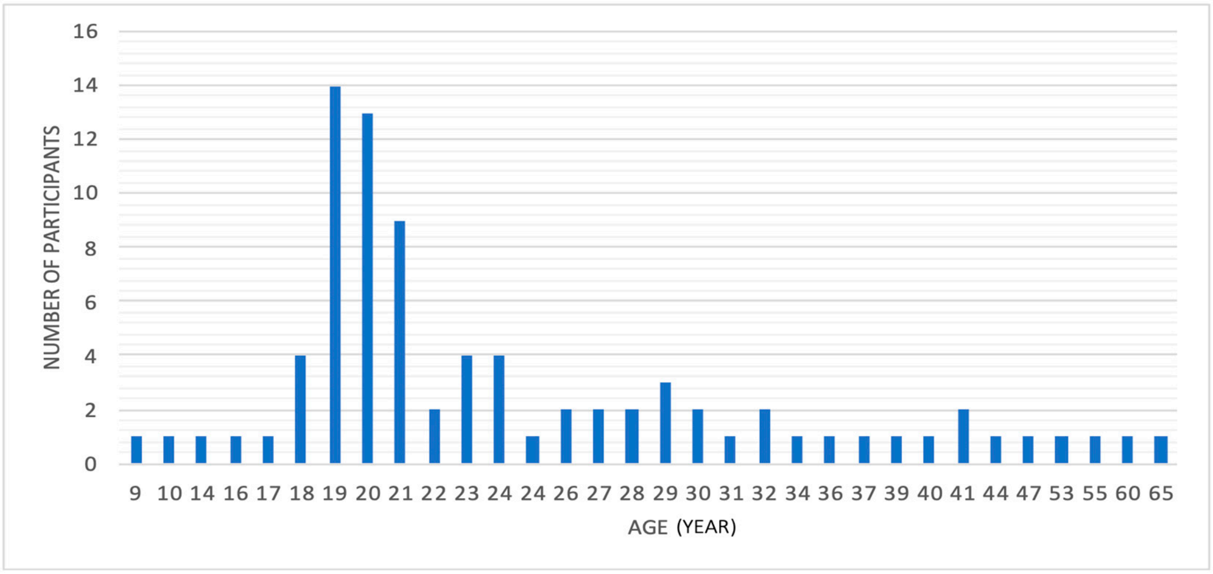
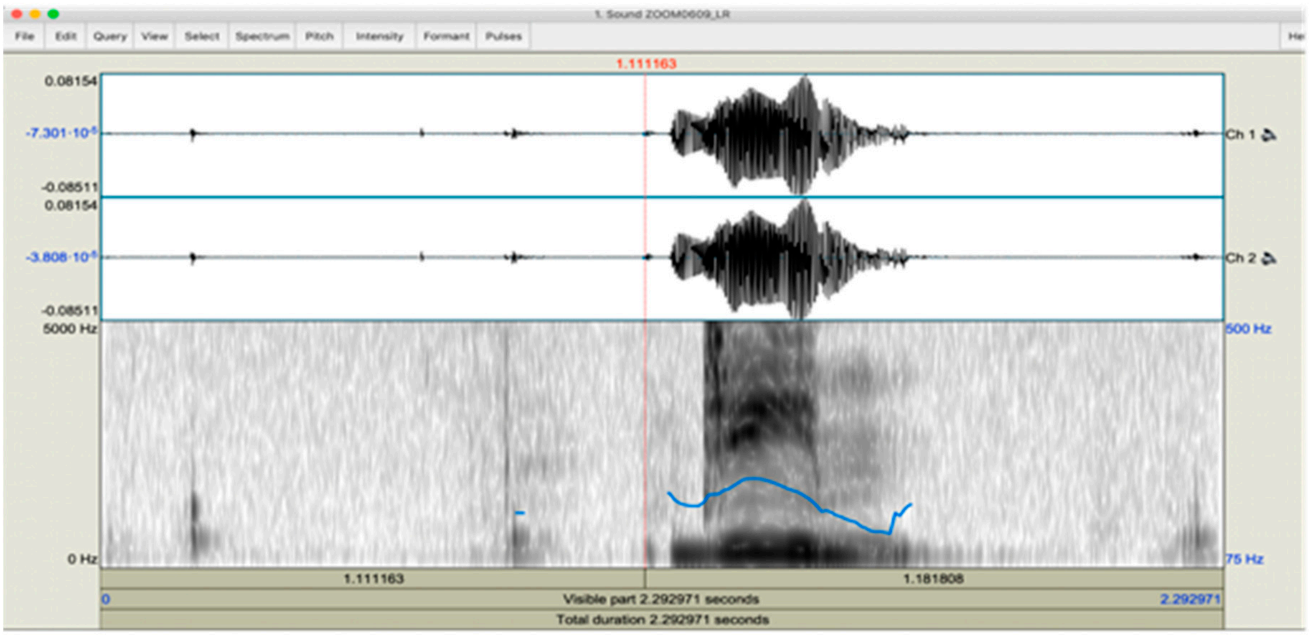


Figure 2. Age distribution.

3.2. Speech Preprocessing

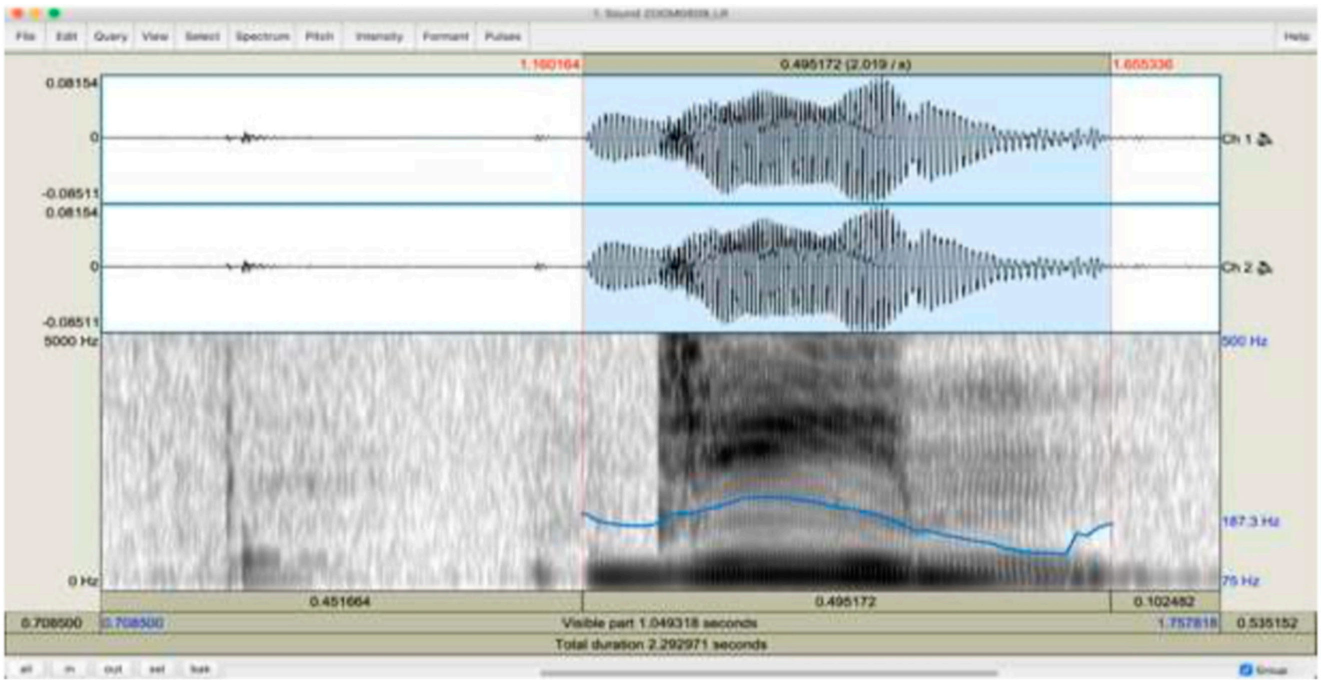
To optimize the efficiency of our analysis, it was deemed necessary to engage in preprocessing procedures for each captured utterance. Considering this, we utilized the PRAAT software tool to extract the section of recorded audio that contains speech from the surrounding silence, as depicted in the graphical illustration provided in Figure 3 [8]. Additionally, it may be postulated that the original speech signal may be symbolized as $x(n)$, while the cleaned speech signal with the presence of noise can be mathematically represented by $d(n)$. Thus, the subsequent expression [9] represents the speech signal in its unprocessed state:

$$x(n) = s(n) + d(n) \tag{1}$$

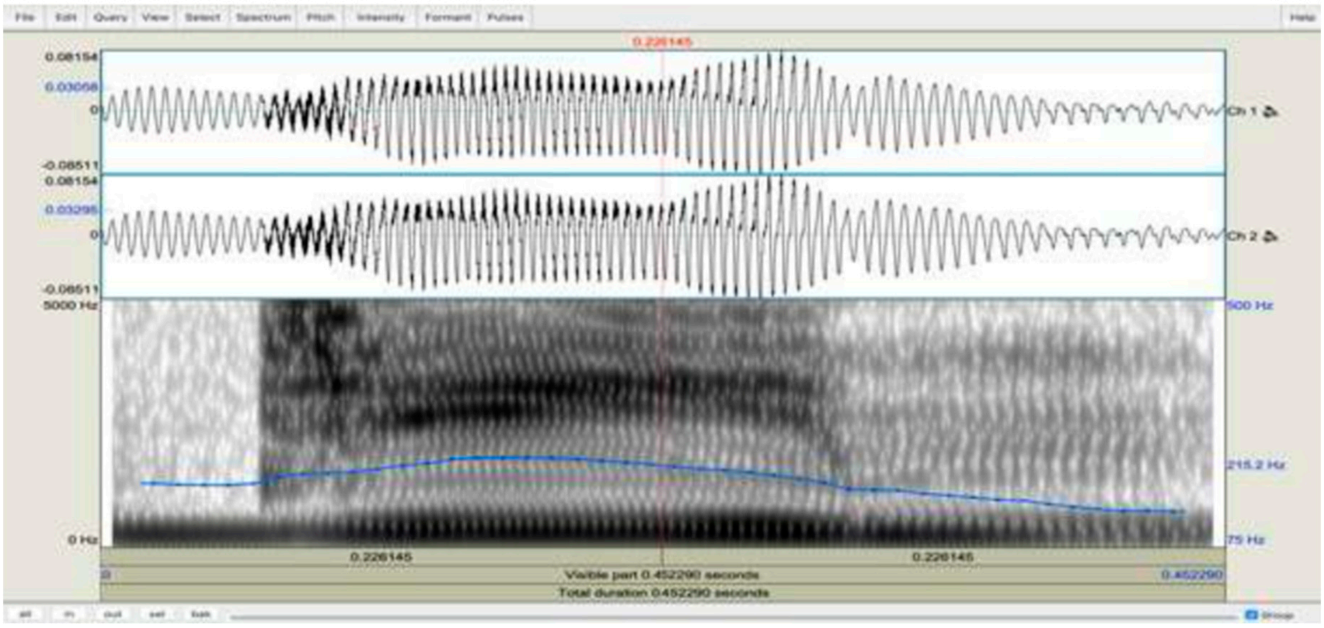


(**a**)

Figure 3. *Cont.*



(**b**)



(**c**)

Figure 3. (**a**) The original speech signal without removing silence; (**b**) selecting the part without silence; (**c**) the speech signal without silence.

3.3. Proposed Framework

In this work, we proposed a two-level classification to obtain both speaker's gender and the pronunciation state. Figure 4 shows the methodology of our proposed framework. In the proposed design, the collected dataset undergoes the preprocessing phase, where it gets filtered from all the noises, and all the silence portions are eliminated. Hence, this process leads to a remarkable increment in the system's performance. Furthermore, whenever the preprocessing is completed, the feature extraction process is initiated, and its determination is essential for enhancing the system's performance. These extracted features are fed to the LSTM network through which training and testing are conducted on the proposed system. Lastly, a fully connected network classifies the gender of the speaker along with the presence of mispronunciation. The complete description of each block is specified in the following subsections.

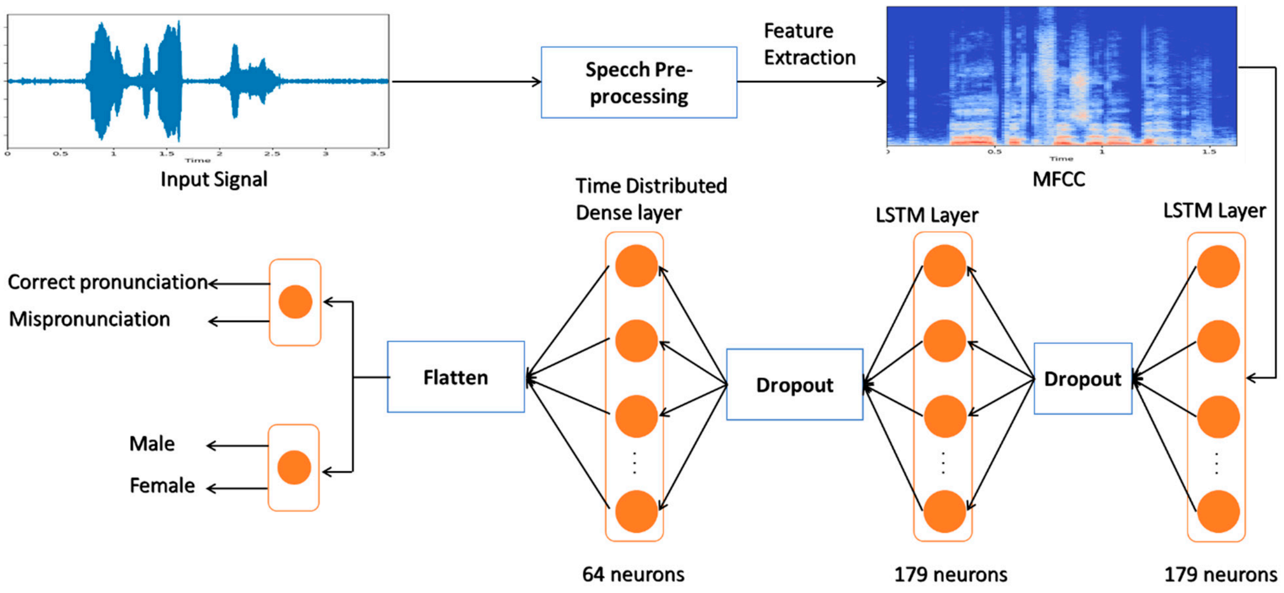


Figure 4. Block diagram of the proposed system.

3.3.1. Recurrent Neural Network (RNN)

Classification is a fundamental mapping of input vectors to a finite set of N classes. In the context of this research endeavor, we utilized cutting-edge deep neural networks. These neural networks facilitate the generation of output classes by activating N output neurons, where the neuron that corresponds to the class of the input vector is characterized by an activation value of 1, and all other outputs remain quiescent with an activation value of 0. As described in [10], this innovative technique is frequently employed in speech recognition systems to correlate speech frames with phoneme classes. In addition to deep neural networks, we also explored recurrent neural networks (RNNs), which are a subset of neural networks used predominantly for predicting future data sequences based on past data samples. RNNs are proficient in modeling sequence data such as speech and text. However, the extensive use of RNNs has been limited due to the difficulties posed by their training processes, which stem from their inability to capture long-term dependencies accurately [11,12]. The computation of the output of the recurrent neural network (RNN) is performed by recursively iterating the following mathematical equations from time t = 1 to t = T [13]:

$$h_t = \mathcal{H}(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{2}$$

$$y_t = W_{hy}h_t + b_y \tag{3}$$

Within the conceptual framework of the recurrent neural network (RNN), x signifies the input variable, y represents the output sequence, and h denotes the hidden vector sequence. Additionally, the architectural integrity of the RNN is founded upon the incorporation of the weight matrices W and the bias vector b, in conjunction with the underlying hidden layer function $\mathcal{H}$ [13]. The rationale for employing recurrent neural networks (RNNs) instead of conventional neural networks in speech recognition systems is grounded in the premise that conventional neural networks presuppose the independence of inputs and outputs. However, the non-linear temporal dynamics of speech necessitate a more sophisticated modeling approach. Furthermore, the prediction of every term in a given sentence requires the inclusion of information pertaining to the previously utilized terms [14].

3.3.2. Long Short-Term Memory (LSTM)

The long short-term memory (LSTM) architecture exhibits the ability to retain data information over prolonged time intervals while simultaneously facilitating the retrieval of previously stored information. Furthermore, with regard to the specific LSTM variant utilized within our system, the canonical representation of an individual LSTM cell unit can be mathematically articulated through the following set of equations [13]:

$$f_t = \sigma\Big(W_f[h_{t-1},\ x_t] + b_f\Big) \tag{4}$$

$$i_t = \sigma(W_i[h_{t-1},\ x_t] + b_i) \tag{5}$$

$$\widetilde{C}_t = tanh(W_C[h_{t-1},\ x_t] + b_C) \tag{6}$$

$$C_t = f_t \times C_{t-1} + i_t \times \widetilde{C}_t \tag{7}$$

$$o_t = \sigma(W_o[h_{t-1},\ x_t] + b_o) \tag{8}$$

$$h_t = o_t \times \tanh(C_t) \tag{9}$$

The formulation of a single long short-term memory (LSTM) cell unit involves the integration of several components, specifically the forget gate (f), input gate (i), output gate (o), new memory cell content ($\widetilde{C}$), memory cell content (C), and the sigmoid function (σ). The sigmoid function is instrumental in the generation of three gates within the memory cell, while the hyperbolic tangent function (tanh) serves to augment the memory cell output, as expounded upon in reference [13]. Each LSTM block comprises a memory cell and three distinct gate units, namely the input gate, output gate, and forget gate, which regulate the memory block's functionality. The forget gate serves to reset the cell variable, thus expunging stored input C_t from memory, whereas the input and output gates facilitate the input of data from the feature vector x_t and the output of data to h_t, respectively. As a consequence, the network is capable of retaining input data over prolonged time intervals and leveraging an acquired amount of long-range temporal context, thereby enhancing its self-learning capabilities [15].

3.3.3. Feature Extraction

Speech features refer to the numerous parameters encapsulated within sound waves and serve as fundamental inputs for speech-processing algorithms. The accurate determination and normalization of these parameters are pivotal steps that heavily influence the efficacy of the system. Normalization ensures that all our features maintain a consistent scale, minimizing potential bias and aiding the model's learning process. Within the context of the present study, the following categories of features were extracted:

- **Zero-crossing rate (ZCR):** These features embody the number of times the sound signal transitions from positive to negative and vice versa, thereby providing insight into the temporal characteristics of the signal. The zero-crossing rate (ZCR) is a crucial indicator of the dominant frequency component of the signal, which serves as a crucial determinant in the appraisal of the signal's acoustic properties [9,11,16].
- **Mel-frequency cepstral coefficients (MFCCs):** The Mel-frequency cepstral coefficients (MFCCs) constitute an inherent feature that is widely utilized in the fields of speaker and emotion recognition. This is attributable to the high-level representation of human auditory perception that MFCCs afford, thus rendering them a critical component in the assessment of acoustic properties [17–20]. The computation of MFCCs necessitates the utilization of a psychoacoustical-motivated filter bank, which

is subsequently followed by logarithmic compression and discrete cosine transform (DCT) techniques. The MFCCs are calculated based on the following formula [21,22]:

$$C(n) = \sum_{m=1}^{M} [log\ Y(m)]\ cos\left[\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right] \tag{10}$$

In the given formula, $Y(m)$ represents the output obtained from the M-channel filter bank, where the index "m" indicates the specific filter. Additionally, the cepstral coefficient index is indicated by "n", and it plays a crucial role in the calculation of MFCCs.

- **ΔMFCC:** MFCC's first-order time derivative. This expression is utilized for calculating differential coefficients [11,23]:

$$D_t = \frac{\sum_{n=1}^{N} n(c_{t+n} - c_{t-n})}{2\sum_{n=1}^{N} n^2} \tag{11}$$

  where the delta coefficient of the frame t is denoted by Dt, and ct+n to ct-n are the static coefficients.
- **Δ2 MFCC:** Another important feature in speech processing algorithms is the second-order derivative of Mel-frequency cepstral coefficients (MFCCs), which is calculated through a differential computation.
- **Linear prediction cepstral coefficients (LPCCs):** The coefficients derived from the impulse response of the linear prediction coefficients (LPCs) are modeled to mimic the vocal tract of the human speech production system, providing a resilient and noise-tolerant speech feature compared to LPCs. LPC analysis evaluates the speech signal by approximating the formants, removing their effects from the speech signal, and estimating the residual concentration and frequency that remain [24,25].
- **Gammatone-frequency cepstral coefficients (GFCCs):** Derived from gammatone filter banks, they offer a more precise representation of speech features and are less affected by noise and distortion compared to MFCCs. Their integration into speech processing algorithms has resulted in better performance and accuracy in tasks such as speaker recognition, speech emotion recognition, and music information retrieval [26].
- **Log-frequency cepstral coefficients (LFCCs):** A comparable technique to Mel-frequency cepstral coefficients (MFCCs) is employed; however, this approach leverages a spatially distributed filter bank situated along a linear frequency spectrum [25].

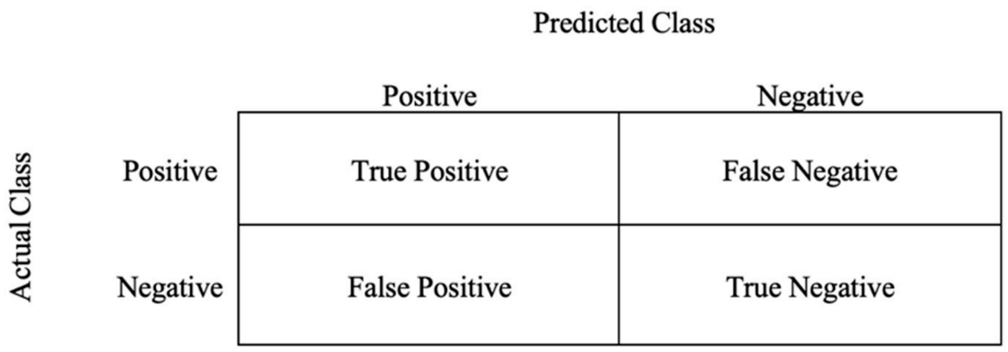
3.3.4. Grid Search

Prior to training, a machine learning model's hyperparameters are determined, necessitating their adjustment to conform the model to a specific dataset. Optimal hyperparameter configurations for one dataset may not be ideal for another, complicating the hyperparameter optimization process. Grid search represents an advancement in conventional hyperparameter optimization, dedicating the search to a specified segment of the training algorithm's hyperparameter domain. In a grid search, the range of potential parameters is manually designated. Subsequently, the software conducts an exhaustive search for these parameters. Fundamentally, a brute-force approach is employed to explore all hyperparameter permutations, followed by model evaluation using cross-validation techniques [27].

4. Experimental Setup

4.1. Evaluation Metrics

To evaluate the LSTM models, we used the performance metrics of precision, recall, F1 score, area under curve (AUC), and accuracy. A confusion matrix is also used to illustrate LSTM performance based on the four classes: True Positive, False Positive, True Negative, and False Negative; these four classification classes are used to calculate each performance metric. Figure 5 shows the confusion matrix classification.



		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive	False Negative
	Negative	False Positive	True Negative

Figure 5. Basic confusion matrix.

In our system, we can define the four different classes of confusion matrix as follows:

- True Positive is an incorrect positive pronunciation prediction.
- False Positive is a correct positive pronunciation prediction.
- True Negative is an incorrect negative pronunciation prediction.
- False Negative is a correct negative pronunciation prediction.

Furthermore, accuracy is defined as the measurement of how correctly a model predicts the pronunciation corresponding to the actual pronunciation. The subsequent formula delineates the computation of accuracy within a given context [13,28]:

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Negative + True\ Negative + False\ Positive} \tag{12}$$

Precision is a statistical measure that assesses the ratio of accurately predicted positive outcomes to the total number of predicted positive outcomes. The formula provided can be utilized to compute the precision metric [13,28]:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{13}$$

Recall is a metric that indicates the proportion of accurately predicted positive outcomes that are correctly identified by the model. To calculate the recall metric, the following mathematical expression is employed [13,28]:

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{14}$$

The F1 score is a performance metric that characterizes the harmonic mean between recall and precision. This metric is particularly useful for imbalanced datasets that feature disproportionate class frequencies. The calculation of the F1 score can be expressed using the following equation [13,28]:

$$F1\ Score = 2 \times \left(\frac{Recall * Precision}{Recall + Precision} \right) \tag{15}$$

4.2. Platform

We built our project using the Python language that was written in a development environment called PyCharm. Computation was performed on an Intel Core i7-4750HQ CPU with 16 GB RAM and NVIDIA GeForce GTX 950M. The training process was completed 53 times. For each letter, the training was performed for the letters alone, the letter at the beginning of the word, the letter in the middle of the word, and the letter at the end of the word. Therefore, training took approximately 18 h, including the optimization steps.

5. Experimental Results

5.1. Ablation Study

5.1.1. Optimum Model Hyperparameters

The optimization of model hyperparameters was performed using both manual and grid search techniques. We divided the dataset into training (70%), testing (20%), and validation (10%) sets. Using the grid search optimization method, we determined the ideal parameters for the network: learning rate, epochs, dropout, and batch size. These optimal values are shown in Table 4. It is important to note that the learning rate, epochs, and batch size are common hyperparameters to tune in deep learning models. A lower learning rate ensures that the model learns at a steady pace, whereas a high number of epochs means that the model goes through the training set multiple times to better understand the patterns in the data. A smaller batch size means the model gets to update its weights more frequently, resulting in a more robust learning process. It should be noted that the optimization of parameters is performed within the training session.

Table 4. Classification network parameters.

Parameters	Learning Rate	Epoch	LSTM Layers	LSTM Units	Fully Connected Layers	Fully Connected Units	Dropout
Value	10^{-3}	190	2	179,179	2	64, 1	0.1

Moreover, the manual optimization technique is illustrated below:

Number of LSTM Cells in Each Layer

We used two hidden layers in our LSTM network, which is the most common number used. Moreover, every hidden layer has several LSTM cells or neurons. To identify the best number of cells for each layer, we trained the network using various values and found that setting the cells in every layer to 179 minimized the MSE, as shown in Figure 6. The experiment was conducted with a dropout of 0.2, 50 training epochs, and a batch size of 32 [29,30].

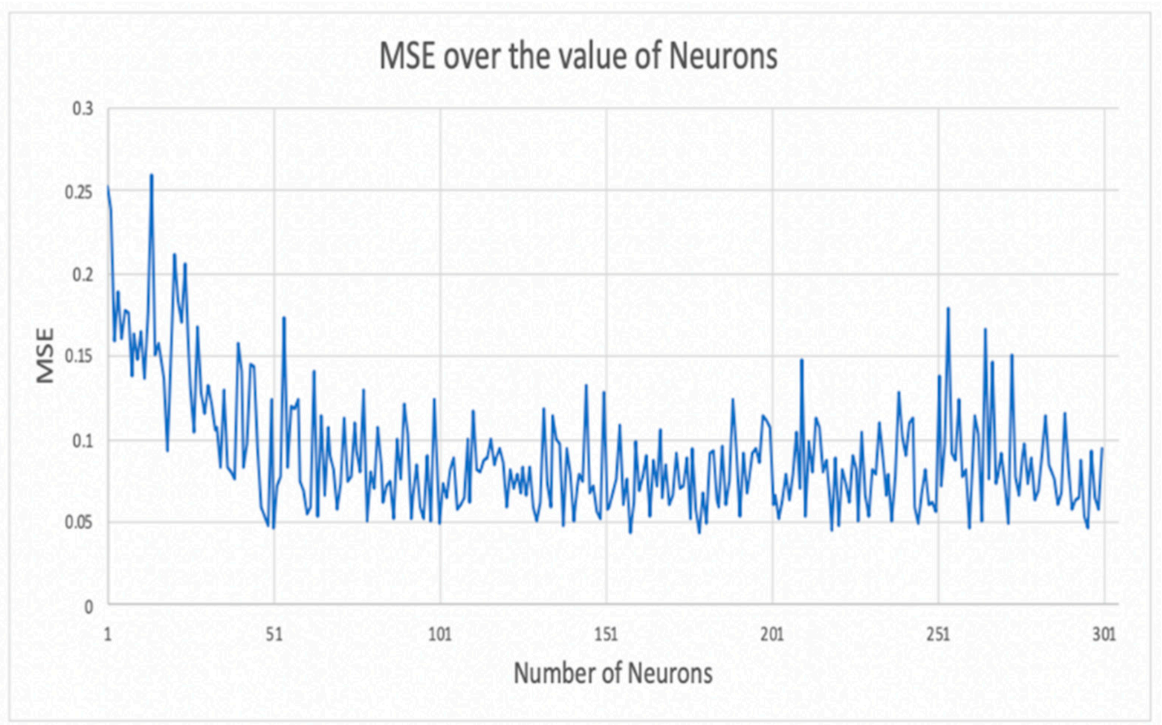


Figure 6. Selecting the optimum value of neurons by the mean square error value.

Dropout

Overfitting is a vital dilemma that affects the system's accuracy and needs to be solved. Therefore, dropout is a layer that prevents overfitting; it randomly chooses any cell in a layer according to the probability chosen and sets its output to 0. Furthermore, to find the optimal dropout rate, a test was performed and implemented in all of the hidden layers. We built and trained our LSTM model, then set the dropout to different values. Based on Figure 7, the results indicate that the most effective dropout value for our scenario is 0.1, as it is associated with the lowest MSE. Therefore, the dropout was set to 30%. While performing this test, the LSTM cells in the two hidden layers were set to 179, epoch to 50, and batch size to 32 [29,30].

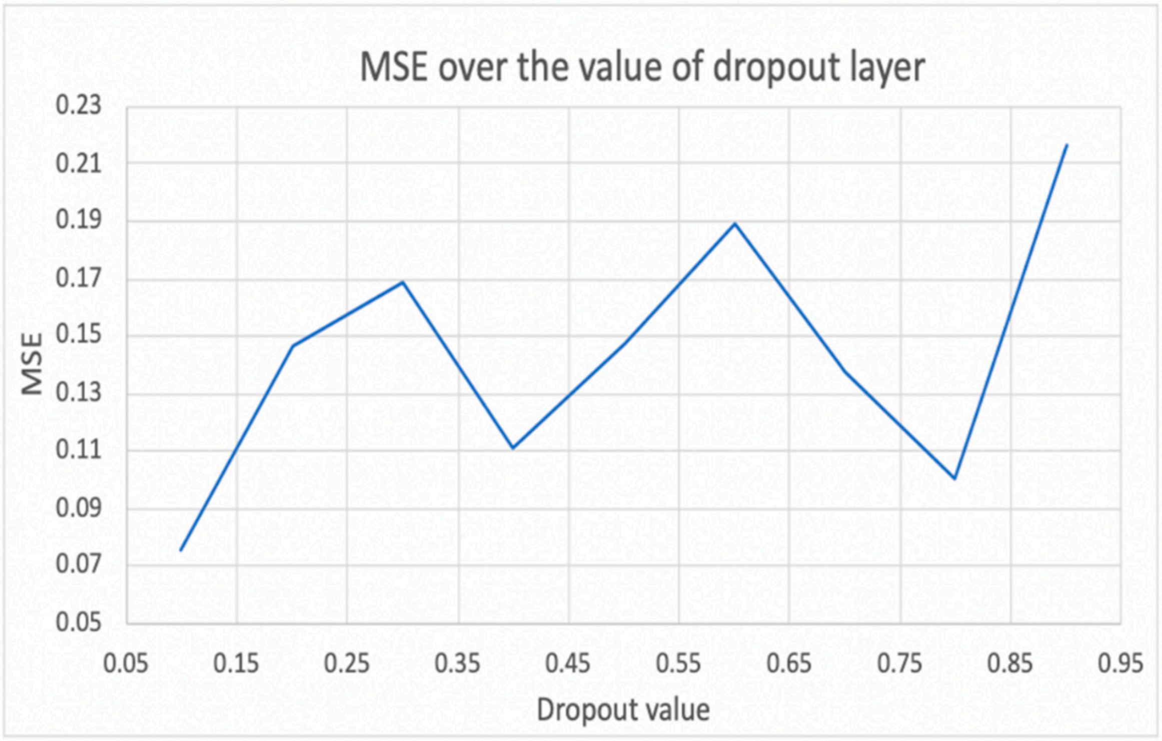


Figure 7. Selecting the optimum value of dropout by the mean square error value.

Batch Size

Batch size refers to the number of training data utilized in one iteration or epoch. We split the training data into a batch size before starting the training process. Splitting the training data will make the training more manageable and more efficient. Also, the higher the batch size used, the more memory space is needed. Figure 8 shows the optimal batch size value; in our case, it is 16 because that value has the lowest MSE. While performing this test, the LSTM cells in the two hidden layers were set to 179, dropout to 0.1, and epochs to 32 [29,30].

Number of Epochs

An epoch refers to a single iteration of the entire training dataset being passed through the neural network. During this process, the training data are partitioned into batches, where a batch size of 16 was chosen in this study. This means that the LSTM network is first trained on the first 16 samples (0–16), then on the next 16 samples (16–31), and so on, until all samples have been propagated through the network. As depicted in Figure 9, the optimal number of epochs in our study was determined to be 190 based on the smallest MSE. During this experiment, the LSTM cells in the two hidden layers were set to 179, dropout was set to 0.1, and batch size was set to 16 [29,30].

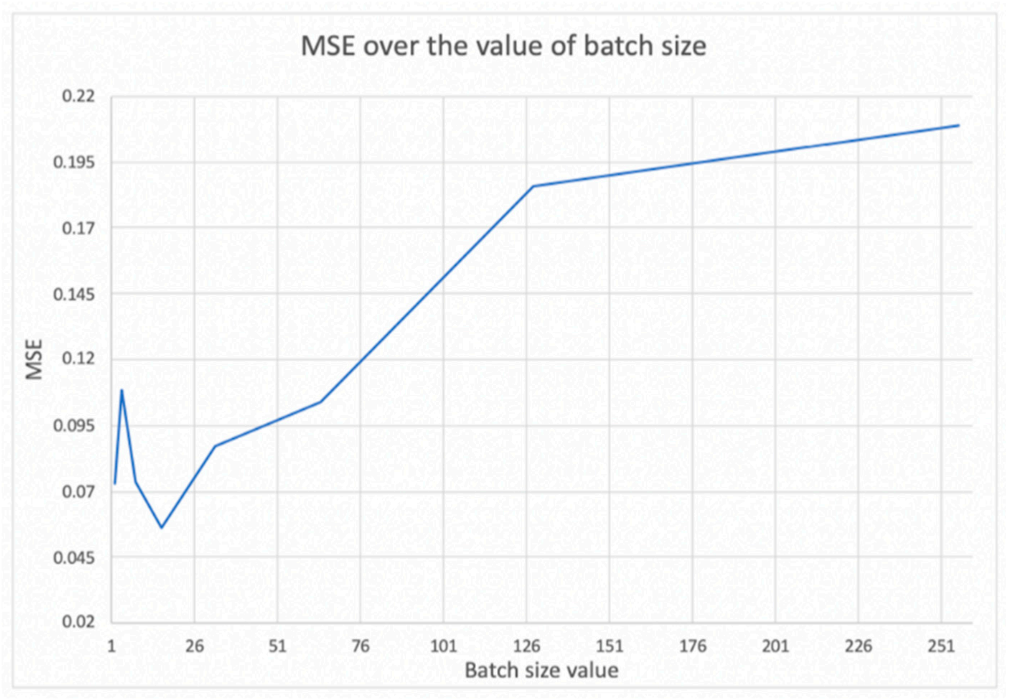


Figure 8. Selecting the optimum value of batch size by the mean square error value.

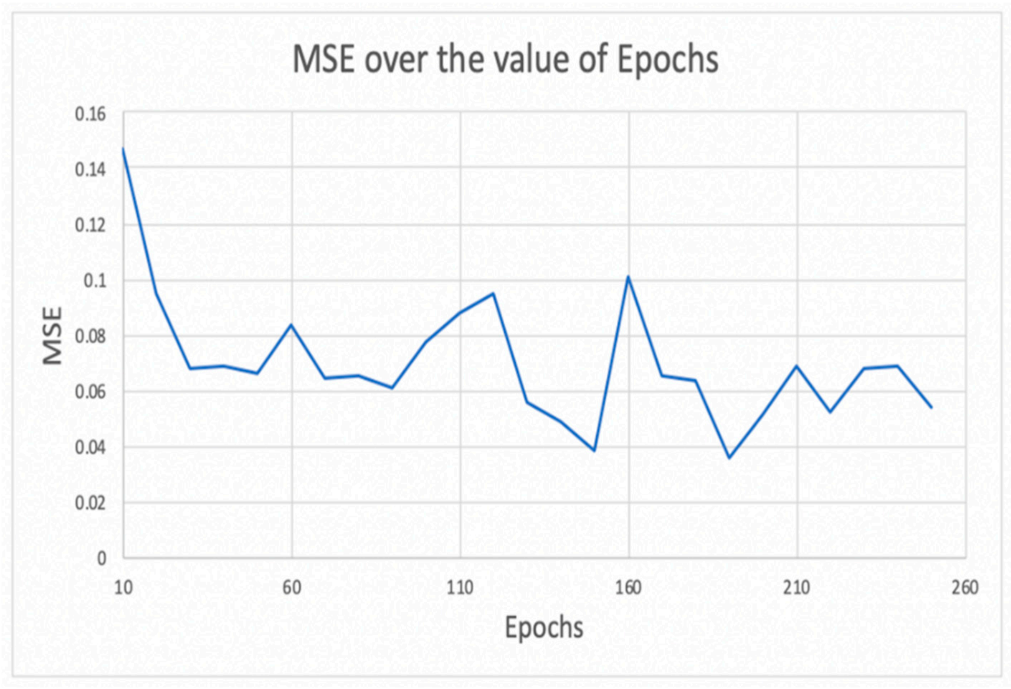


Figure 9. Selecting the optimum value of epochs by the mean square error value.

Consequently, the values obtained manually are similar to the grid search parameters. Therefore, the usage of the grid search is effective and requires less training time.

5.1.2. Optimum Speech Features

Feature extraction is vital to the system's overall effectiveness. Hence, seven distinct auditory features were acquired from the captured dataset sounds and forwarded to our framework classification model. In Figure 10, we illustrate the features' importance based on system performance. To obtain these results, we used only the letter ر/r/. The majority of letters also showed the same results alongside the letter ر/r/. Furthermore, it can be observed that the MFCC features have the highest accuracy. Thus, we utilized the MFCCs to be our optimum feature extraction method. However, we can also implement the system using other features or their combination since they achieved good accuracy.

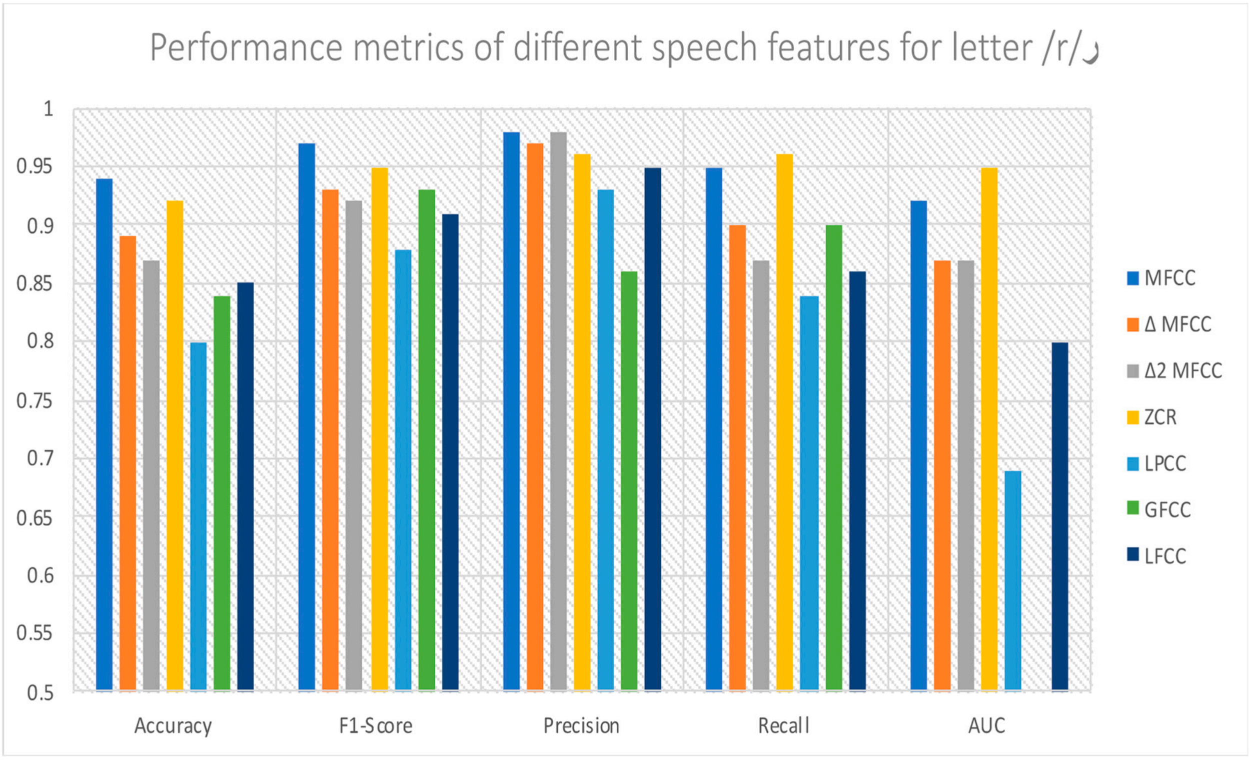
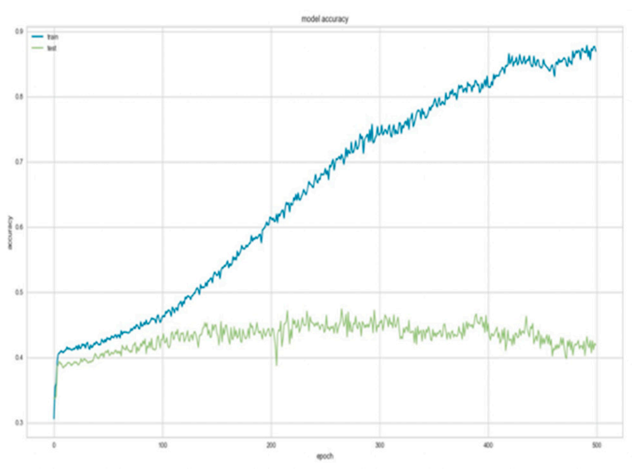


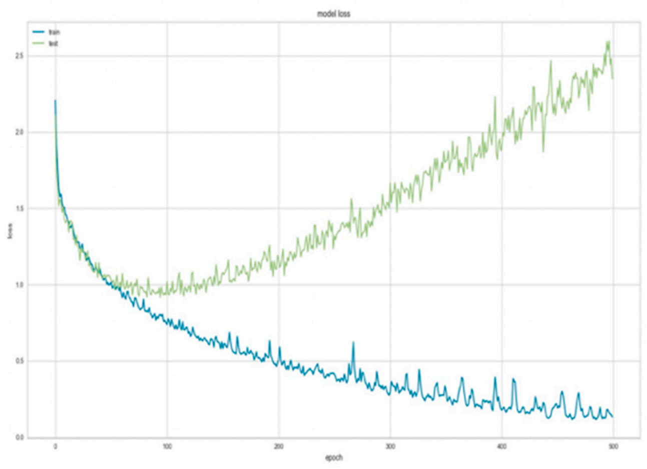
Figure 10. Performance metrics of different speech features.

5.2. Overall Performance of the System

In order to accurately gauge the performance of our LSTM model, we meticulously executed both the training and testing phases, each integral to the overall system. During the training phase, the primary task was to convert the input training sets into corresponding vector representations. Once these were prepared, they were introduced into the LSTM network. This initiated a process where the model tried to learn the underlying structure of the input data, using backpropagation to gradually adjust the network's weights and improve its predictions. Here, it is essential to explain that backpropagation is a critical part of training deep learning models. It involves computing the gradient of the loss function with respect to the weights of the network and using this information to adjust the weights to minimize the loss. In the testing phase, we utilized input test signals to probe the network's learning. The output values generated by the network are determined by the weights the model has learned during the training phase. This phase is crucial as it allows us to evaluate the LSTM model's capability to capture relevant features and relationships in the data and predict unseen data points. During the training and testing process, we encountered a common issue in machine learning models, overfitting, where the model learns the training data too well and performs poorly on unseen data. However, we successfully mitigated this problem by optimizing the LSTM parameters, as discussed in Section 5.1.1. Figure 11 provides a visual representation of an instance of overfitting that we encountered. It is a situation where the model, while performing well on the training data, fails to generalize well to unseen data due to excessive complexity. Furthermore, Figure 12 illustrates the model's accuracy and loss metrics during the training and validation process after optimization. A balanced and improved performance in these metrics signifies an effective learning model capable of making reliable predictions.

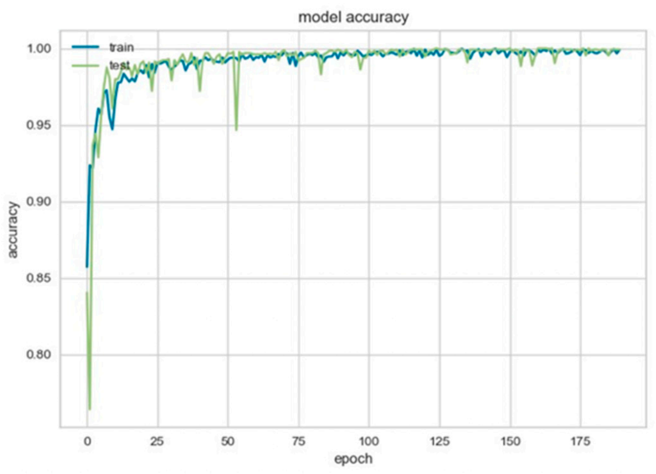


(**a**)

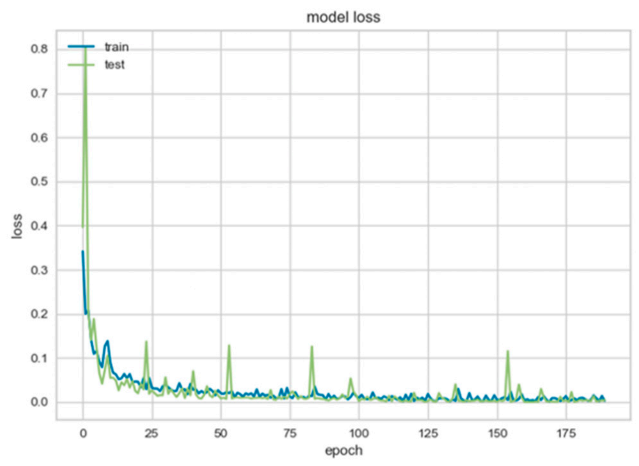


(**b**)

Figure 11. The overfitting plot of (**a**) model accuracy on training and the validation dataset; (**b**) model loss on the validation dataset.



(**a**)



(**b**)

Figure 12. The plot of (**a**) model accuracy on training and the validation dataset; (**b**) model loss on training and the validation dataset.

Our testing resulted in the performance metrics of the LSTM models in identifying mispronunciations, reported in Tables 5–8. Each table refers to a different set of conditions: respectively, uttering letters alone, at the beginning of the word, in the middle of the word, and at the end of the word. Analyzing the results, we find that the highest accuracy is consistently achieved by the letter ج/ʒ/. When pronounced alone, or at the beginning or middle of a word, the accuracy rates were 82.25%, 85.39%, and 91.84%, respectively, as seen in Tables 5–7. The letter ك/k/ also demonstrated high accuracy, specifically 82.34%, when articulated at the end of a word (Table 8). Conversely, the lowest accuracy across all conditions was demonstrated by the letter ر/r/. When this letter was pronounced alone, or at the beginning, middle, or end of a word, the system's accuracy was 75.67%, 75.49%, 86.53%, and 66.05%, respectively. These numbers, presented across Table 5 through Table 8, reflect the challenge our system faced in accurately identifying the pronunciation of this particular letter in different contexts. However, these results do not necessarily mean that our system is universally more effective with some letters than others. The variance could be due to factors such as the inherent difficulties of certain sounds, differences in the number of training samples for each letter, or biases in our training data. Future work could investigate these possibilities in more detail.

Following the evaluation of mispronunciation detection, we extended our system to integrate gender recognition capabilities. Given the challenges our system faced with the letter ر/r/, as highlighted in the previous section, we chose this letter to test the combined system's robustness. The system was trained and tested to simultaneously recognize both gender and mispronunciation. Table 9 shows the results of these tests. When creating a separate model for genders, the system attained an average accuracy of 81.52%. Without a separate model for genders, the accuracy was slightly higher, at 83.77%. Interestingly, the system's accuracy did not improve upon adding the gender recognition task, but it maintained a reasonable accuracy level. This suggests that our LSTM model was able to handle the added complexity of gender recognition without significantly compromising its performance on mispronunciation detection. While the results are encouraging, they also highlight areas for improvement. Future work can aim to refine the model, exploring different architectures or training strategies, to boost accuracy in both tasks. The goal would be a more robust system capable of detecting mispronunciations and recognizing speaker gender with even higher accuracy.

Table 5. Performance metrics for letters alone.

Sample	Precision	Recall	F1 Score	Accuracy
Letter ر/r/	0.8255	0.7759	0.7964	0.7567
Letter ظ/ð/	0.8164	0.7880	0.8019	0.7717
Letter ض/d/	0.8161	0.7873	0.8014	0.7709
Letter ذ/ð/	0.8110	0.7834	0.7970	0.7633
Letter غ/ɣ/	0.8162	0.7808	0.7981	0.7652
Letter ك/k/	0.8165	0.7874	0.8017	0.7714
Letter ق/q/	**0.8297**	0.8090	0.8192	0.8114
Letter س/s/	0.8127	0.7877	0.8000	0.7864
Letter ص/Ṣ/	0.8188	0.8125	0.8156	0.8130
Letter خ/x/	0.8188	0.8089	0.8138	0.8098
Letter ج/ʒ/	0.8233	**0.8226**	**0.8235**	**0.8225**
Letter ط/ŧ/	0.8235	0.8097	0.8165	0.8109
Letter ح/ḥ/	0.8240	0.8222	0.8231	0.8222

Table 6. Performance metrics for letters at the beginning of words.

Sample	Precision	Recall	F1 Score	Accuracy
Letter ر/r/	0.8369	0.7582	0.7756	0.7549
Letter ظ/ð/	0.8458	0.8050	0.8249	0.8027
Letter ض/d/	0.8459	0.8023	0.8235	0.8004
Letter ذ/ð/	0.8100	0.7679	0.7884	0.7876
Letter غ/ɣ/	0.8362	0.8245	0.8303	0.8258
Letter ك/k/	0.8255	0.8239	0.8247	0.8240
Letter ق/q/	0.8243	0.8172	0.8207	0.8176
Letter س/s/	0.8232	0.8189	0.8210	0.8181
Letter ص/Ṣ/	0.8568	0.8067	0.8310	0.8351
Letter خ/x/	0.8561	0.8193	0.8373	0.8460
Letter ج/ʒ/	**0.8568**	**0.8574**	**0.8571**	**0.8539**
Letter ط/ŧ/	0.8442	0.8404	0.8423	0.8410
Letter ح/ḥ/	0.8429	0.8431	0.8430	0.8422

Table 7. Performance metrics for letters in the middle of words.

Sample	Precision	Recall	F1 Score	Accuracy
Letter ر/r/	0.9061	0.8699	0.8876	0.8653
Letter ظ/∂/	0.9131	0.9157	0.9144	0.9102
Letter ض/d/	0.9137	0.9115	0.9126	0.9071
Letter ذ/ð/	0.9159	0.9140	0.9149	0.9112
Letter غ/ɣ/	0.9188	0.9141	0.9164	0.9137
Letter ك/k/	0.9200	0.9151	0.9175	0.9157
Letter ق/q/	0.9191	0.8978	0.9083	0.8996
Letter س/s/	0.9186	0.8841	0.9010	0.8873
Letter ص/Ṣ/	0.9196	0.9179	0.9187	0.9179
Letter خ/x/	0.9178	0.9137	0.9157	0.9125
Letter ج/ʒ/	**0.9200**	**0.9182**	**0.9191**	**0.9184**
Letter ط/ŧ/	0.9199	0.9056	0.9127	0.9075
Letter ح/ḥ/	0.9188	0.9178	0.9183	0.9171

Table 8. Performance metrics for letters at the end of words.

Sample	Precision	Recall	F1 Score	Accuracy
Letter ر/r/	0.7113	0.6926	0.7018	0.7105
Letter ظ/∂/	0.7756	0.7704	0.7730	0.7733
Letter ض/d/	0.7756	0.7708	0.7732	0.7637
Letter ذ/ð/	0.7744	0.7607	0.7725	0.7727
Letter غ/ɣ/	0.7743	0.7669	0.7606	0.7778
Letter ك/k/	0.7772	0.7266	0.7469	0.7765
Letter ق/q/	0.7471	0.7386	0.7428	0.7394
Letter س/s/	0.8251	0.8110	0.8180	0.8104
Letter ص/Ṣ/	0.8280	**0.8246**	**0.8263**	**0.8234**
Letter خ/x/	**0.8288**	0.8189	0.8238	0.8148
Letter ج/ʒ/	0.7607	0.7578	0.7592	0.7582
Letter ط/ŧ/	0.8029	0.7975	0.7902	0.8180
Letter ح/ḥ/	0.7747	0.7723	0.7735	0.7726

Table 9. System accuracy based on different techniques.

Techniques	Average Accuracy (%)
Creating a model for genders	81.52
Without creating a model for genders	83.77

6. Concluding Remarks

Our research introduces an effective speaker-independent and text-dependent system aimed at detecting articulation disorders in Arabic language learners. This system leverages the power of speech signal processing, specifically using Mel-frequency cepstral coefficients (MFCCs) as the optimal features, and long short-term memory (LSTM) for classification. It delivers robust performance with an average accuracy rate of 81.52%, representing a significant improvement over previous mispronunciation detection systems. One of the innovative aspects of this system is the incorporation of a gender recognition model, enabling two-level classification. While this incorporation led to some interesting observations, it did not significantly improve the overall accuracy of the system. In fact, the results suggested that Arabic mispronunciation patterns might not be distinctly gender-specific. This observation has enriched our understanding and opened up a new line of inquiry for future work. Although the gender recognition aspect did not greatly impact mispronunciation detection in the current study, we see potential in further exploring its influence. It represents an uncharted territory in the field of Arabic mispronunciation detection and could reveal new insights and dimensions. We plan to investigate more sophisticated gender recognition mechanisms or other sociolinguistic variables that might influence pronunciation patterns in the future. Future work also could focus on improving the system's efficiency for people with different disorder cases and incorporating additional teaching tools, such as a 3D animation of the vocal tract, on providing the appropriate teaching tools for non-Arabic speakers and children to learn the proper Arabic language. Overall, the proposed system has the potential to provide significant support for Arabic language learners by identifying and correcting mispronunciation errors, which can improve language and speech skills. This study represents a valuable contribution to the field of CALL, demonstrating the potential for using AI and machine learning in language learning and teaching.

Author Contributions: A.A.: Conceptualization, Methodology, Software, Writing—Original Draft. M.B. (Mohamed Bader): Conceptualization, Methodology, Writing—Original Draft. I.S.: Conceptualization, Methodology, Investigation, Writing—Original Draft, Writing—Review & Editing, Project administration, Supervision. A.B.N.: Conceptualization, Methodology, Formal Analysis, Investigation, Writing—Review & Editing, Supervision. N.W.: Writing—Review & Editing. M.B. (Mohammad Basel): Methodology, Writing—Original Draft. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funding through the competitive research project entitled "Analysis and Investigation of Emirati-Accented Corpus in Emotional and Stressful Talking Environments for Speaker and Emotion Recognition based on Capsule Networks", Project ID: No. 23020403251.

Data Availability Statement: The authors generated the dataset used in this study. Due to ethical restrictions, it is not publicly available but can be accessed for research purposes upon request. Any distribution of the data will adhere strictly to confidentiality and ethical guidelines.

Acknowledgments: The authors would like to thank the University of Sharjah in the United Arab Emirates for funding this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Calik, S.S.; Kucukmanisa, A.; Kilimci, Z.H. An ensemble-based framework for mispronunciation detection of Arabic phonemes. *arXiv* **2023**, arXiv:2301.01378.
2. Fu, P.; Liu, D.; Yang, H. LAS-Transformer: An Enhanced Transformer Based on the Local Attention Mechanism for Speech Recognition. *Information* **2022**, *13*, 250. [CrossRef]
3. Ye, W.; Mao, S.; Soong, F.; Wu, W.; Xia, Y.; Tien, J.; Wu, Z. An Approach to Mispronunciation Detection and Diagnosis with Acoustic, Phonetic and Linguistic (Apl) Embeddings. In Proceedings of the ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, 23–27 May 2022; pp. 6827–6831. [CrossRef]
4. Li, K.; Qian, X.; Meng, H. Mispronunciation Detection and Diagnosis in L2 English Speech Using Multidistribution Deep Neural Networks. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2017**, *25*, 193–207. [CrossRef]
5. Shahin, M.; Ahmed, B. Anomaly detection based pronunciation verification approach using speech attribute features. *Speech Commun.* **2019**, *111*, 29–43. [CrossRef]
6. Arafa, M.N.M.; Elbarougy, R.; Ewees, A.A.; Behery, G.M. A Dataset for Speech Recognition to Support Arabic Phoneme Pronunciation. *Int. J. Image Graph. Signal Process.* **2018**, *10*, 31–38. [CrossRef]
7. Shareef, S.; Al-Irhayim, Y. Comparison between Features Extraction Techniques for Impairments Arabic Speech. *Al-Rafidain Eng. J.* **2022**, *27*, 190–197. [CrossRef]
8. Keerio, A.; Mitra, B.K.; Birch, P.; Young, R.; Chatwin, C. On preprocessing of speech signals. *World Acad. Sci. Eng. Technol.* **2009**, *35*, 818–824. [CrossRef]
9. Ibrahim, Y.A.; Odiketa, J.C.; Ibiyemi, T.S. Preprocessing technique in automatic speech recognition for human computer interaction: An overview. *Ann. Comput. Sci. Ser.* **2017**, *15*, 186–191.

10. Kaur, M.; Mohta, A. A Review of Deep Learning with Recurrent Neural Network. In Proceedings of the 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 27–29 November 2019; pp. 460–465. [CrossRef]
11. Hassan, A.; Shahin, I.; Alsabek, M.B. COVID-19 Detection System using Recurrent Neural Networks. In Proceedings of the 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Sharjah, United Arab Emirates, 3–5 November 2020. [CrossRef]
12. Nassif, A.B.; Shahin, I.; Attili, I.; Azzeh, M.; Shaalan, K. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* **2019**, *7*, 19143–19165. [CrossRef]
13. Shewalkar, A.; Nyavanandi, D.; Ludwig, S.A. Performance Evaluation of Deep neural networks Applied to Speech Recognition: Rnn, LSTM and GRU. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235–245. [CrossRef]
14. Amberkar, A.; Awasarmol, P.; Deshmukh, G.; Dave, P. Speech Recognition using Recurrent Neural Networks. In Proceedings of the 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India, 1–3 March 2018; pp. 1–4. [CrossRef]
15. Geiger, J.T.; Zhang, Z.; Weninger, F.; Schuller, B.; Rigoll, G. Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling. In Proceedings of the Annual Conference on the International Speech Communication Association (Interspeech 2014), Singapore, 14–18 September 2014; pp. 631–635.
16. Kos, M.; Kačič, Z.; Vlaj, D. Acoustic classification and segmentation using modified spectral roll-off and variance-based features. *Digit. Signal Process.* **2013**, *23*, 659–674. [CrossRef]
17. Shahin, I.; Nassif, A.B.; Bahutair, M. Emirati-accented speaker identification in each of neutral and shouted talking environments. *Int. J. Speech Technol.* **2018**, *21*, 265–278. [CrossRef]
18. Shahin, I. Novel third-order hidden Markov models for speaker identification in shouted talking environments. *Eng. Appl. Artif. Intell.* **2014**, *35*, 316–323. [CrossRef]
19. Shahin, I. Using emotions to identify speakers. In Proceedings of the 5th International Workshop on Signal Processing and Its Applications (WoSPA 2008), Sharjah, United Arab Emirates, 18–20 March 2008.
20. Shahin, I. Identifying Speakers Using Their Emotion Cues. *Int. J. Speech Technol.* **2011**, *14*, 89–98. [CrossRef]
21. Shahin, I.; Nassif, A.B.; Hamsa, S. Novel cascaded Gaussian mixture model-deep neural network classifier for speaker identification in emotional talking environments. *Neural Comput. Appl.* **2020**, *32*, 2575–2587. [CrossRef]
22. Alsabek, M.B.; Shahin, I.; Hassan, A. Studying the Similarity of COVID-19 Sounds based on Correlation Analysis of MFCC. In Proceedings of the 2020 International Conference on Communications, Computing, Cybersecurity, and Informatics (CCCI), Sharjah, United Arab Emirates, 3–5 November 2020. [CrossRef]
23. Ranjan, R.; Thakur, A. Analysis of feature extraction techniques for speech recognition system. *Int. J. Innov. Technol. Explor. Eng.* **2019**, *8*, 197–200.
24. Kinnunen, T.; Li, H. An overview of text-independent speaker recognition: From features to supervectors. *Speech Commun.* **2010**, *52*, 12–40. [CrossRef]
25. Atrey, P.K.; Maddage, N.C.; Kankanhalli, M.S. Audio based event detection for multimedia surveillance. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, Toulouse, France, 14–19 May 2006; Volume 5, pp. 813–816. [CrossRef]
26. Ayoub, B.; Jamal, K.; Arsalane, Z. Gammatone frequency cepstral coefficients for speaker identification over VoIP networks. In Proceedings of the 2016 International Conference on Information Technology for Organizations Development (IT4OD), Fez, Morocco, 30 March–1 April 2016. [CrossRef]
27. Liashchynskyi, P.; Liashchynskyi, P. Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS. *arXiv* **2019**, arXiv:1912.06059.
28. Sokolova, M.; Japkowicz, N.; Szpakowicz, S. Beyond accuracy, F-score and ROC: A family of discriminant measures for performance evaluation. In Proceedings of the 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, 4–8 December 2006; WS-06-06; pp. 24–29. [CrossRef]
29. Bahador, M.; Ahmed, W. The Accuracy of the LSTM Model for Predicting the S&P 500 Index and the Difference between Prediction and Backtesting. Bachelor's Thesis, KTH Royal Institute of Technology, Stockholm, Sweden, 2018; p. 37.
30. Azzouni, A.; Pujolle, G. A long short-term memory recurrent neural network framework for network traffic matrix prediction. *arXiv* **2017**, arXiv:1705.05690.