Hindawi
Advances in Human-Computer Interaction
Volume 2023, Article ID 7398538, 11 pages
https://doi.org/10.1155/2023/7398538

Research Article

Deep Learning Methods for Arabic Autoencoder Speech Recognition System for Electro-Larynx Device

Zinah J. Mohammed Ameen and Abdulkareem Abdulrahman Kadhim

College of Information Engineering, Al-Nahrain University, Baghdad, Iraq

Correspondence should be addressed to Zinah J. Mohammed Ameen; zinahameen83@yahoo.com

Received 8 November 2022; Revised 29 December 2022; Accepted 28 January 2023; Published 28 February 2023

Academic Editor: Christos Troussas

Recent advances in speech recognition have achieved remarkable performance comparable with human transcribers' abilities. But this significant performance is not the same for all the spoken languages. The Arabic language is one of them. Arabic speech recognition is bounded to the lack of suitable datasets. Artificial intelligence algorithms have shown promising capabilities for Arabic speech recognition. Arabic is the official language of 22 countries, and it has been estimated that 400 million people speak the Arabic language worldwide. Speech disabilities have been one of the expanding problems in the last decades, even in kids. Some devices can be used to generate speech for those people. One of these devices is the Servox Digital Electro-Larynx (EL). In this research, we developed an autoencoder with a combination of long short-term memory (LSTM) and gated recurrent units (GRU) models to recognize recorded signals from Servox Digital EL Electro-Larynx. The proposed framework consisted of three steps: denoising, feature extraction, and Arabic speech recognition. The experimental results show 95.31% accuracy for Arabic speech recognition with the proposed model. In this research, we evaluated different combinations of LSTM and GRU for constructing the best autoencoder. A rigorous evaluation process indicates better performance with the use of GRU in both encoder and decoder structures. The proposed model achieved a 4.69% word error rate (WER). Experimental results confirm that the proposed model can be used for developing a real-time app to recognize common Arabic spoken words.

1. Introduction

Arabic is the official language of 22 countries worldwide, and it has been estimated that 400 million people speak the Arabic language worldwide [1]. Based on recent developments in the area of artificial intelligence (AI), particularly natural language processing (NLP), many researchers have focused on using AI applications for automatic speech recognition (ASR). Mainly, they investigated morphological analysis, resource building, and machine translation for the Arabic language [2]. Speech and language disorders are a side effect of many diseases nowadays. Due to these side effects, many people cannot talk at all. There are different devices that can be used to generate sounds from the vocal cords (throat) of people who cannot talk at all. One of these devices is the Servox Digital Electro-Larynx (EL) [3]. This device generates a quasi-clear voice for people with disorders problem and helps them communicate with others. Unfortunately, the quality of the generated speech is not good. In order to recognize spoken Arabic speech by this device, an autoencoder with combinations of long-short term memory (LSTM) and gated recurrent units (GRU) is proposed.

Applications of NLP for the Arabic language are wide. Here, we review related Arabic speech recognition applications in recent years. Darwish [4] used a conditional random field model to label Arabic handwriting with English phonemes. They reported 98.5% accuracy for word-level language classification. With the development of new models and the gathering of more datasets, more specific tasks like semantic-level analysis of the Arabic language have been conducted. Duwairi et al. [5] constructed a small Arabic annotated corpus manually. They collected 3026 messages containing Arabize messages transliterated into

Arabic. They applied a support vector machine (SVM) and a naive Bayesian (NB) model to recognize spoken Arabic words. They concluded that NB is far superior for Arabic speech recognition. With the availability of more Arabic language datasets, utilizing deep learning models to conduct Arabic ASR) gained more attention.

Commonly, automatic speech recognition is implemented through two main stages: the extraction of the acoustic features from the incoming signal and the recognition of spoken words. Dendani et al. [6] used a deep autoencoder (DAE) algorithm to increase the performance of proposed ASR models. Their proposed framework consisted of a two-step procedure. In the first step, a complete DAE is trained for denoising, and then a DAE is trained in a supervised manner on the clean speech produced in the previous step. They concluded that their proposed framework created a benchmark denoiser for all Arabic speech recognition models. Recently, more hybrid structures have been proposed to extract comprehensive feature sets from sounds. Eljawad et al. [7] proposed feature extraction techniques in a three phases framework. In the beginning, they proposed a discrete-level removal from the input dataset. Then, they proposed increasing the number of instances in the dataset to 2000 samples for each word. After preprocessing, they extracted features from the input samples with wavelet transform coefficients. Finally, they used multilayer perceptron (MLP) and fuzzy logic to create the recognizer models. They evaluated their proposed model on 250 samples (75 females and 175 males). They reported 94.5% recognition accuracy with MLP and 77.1% with the use of a fuzzy interference system for Arabic classification. They concluded that MLP outperformed Surgeon type fuzzy logic systems for Arabic recognition language. So far, all of the mentioned methods have been developed for ASR in a normal situation. But situations for recording sounds might be stressful, or the collected dataset may contain noise. Hamsa et al. [8] proposed a method for Arabic emotion recognition in stressful and noisy situations. They proposed a model based on novel wavelet packet transform as denoising techniques and a random forest model for emotion recognition. They evaluated their proposed model using Emirati-emphasized Arabic speech. This dataset contains speech signals from 25 male and 25 female native Emirati speakers with ages between 14 to 55 years old. Each speaker uttered 8 common Emirati sentences that are heavily utilized in the United Arab Emirates society. Every speaker expressed eight sentences in each of angry, happy, neutral, sad, fearful, and disgusted emotions 9 times with a span of 2 to 5 seconds. They reported 89.60% accuracy for the recognition of emotion in the Arabic language. They also reported that with the use of the proposed model recognizing fearful and sad emotions can be done with better accuracy than other emotions in the Arabic language. In another research study by Ali et al. [9], they used ML models with ordered Mel Frequency Cepstral Coefficients (MFCC) as feature extraction to detect impostors based on their spoken language. They used two datasets for their work. The first dataset was prominent leaders' speeches, which included audio clips of five country leaders. The second, called the speaker recognition audio dataset, contains audio clips of fifty people. They evaluated various machine learning models, such as random forest (RF), NB, and K nearest neighbor (KNN), to detect imposters. Imposters detection rate of 97.9% is reported in such work. Shahin and Nassif [10] have used a hidden Markov model (HMM) with a combination of MFCC for Arabic speech recognition. They evaluated their proposed structure using a dataset from 50 samples (25 females and 25 males). These samples were gathered from Emirati persons in scenarios such as neutral, shouted, slow, loud, soft, and fast talking. They reported 65.0% accuracy for ASR in stressful conditions. In another research study by Dendani et al. [11] an autoencoder model was proposed for enhancing Arabic language recognition performance. They used the proposed model to restore the original clean speech from noisy datasets. Their proposed model consisted of 5 different hidden layers for speech enhancement. They reached 65.7% for Arabic speech recognition.

Deep learning models have shown promising results for automatic speech recognition (ASR), such as LSTM [12]. LSTM is designed to perform feature extraction on time-series datasets. In [13], Zerari et al. proposed a method based on LSTM for automatic speech recognition. Their aim was to convert natural Arabic voice into computer text as well as perform an action based on instructions given by a human. The proposed structure covered extracting features from input signals using MFCC. A padding structure is then performed to deal with the nonuniformity of the sequence's length. Either LSTM or Gated Recurrent Unit (GRU) models were applied to extract the relationship among the signals, which were then followed by the MLP model for recognition. The work concluded that when using GRU on digital corpus an error rate of 1.15% was observed for voice recognition and 2.89% for TV command recognition. LSTM model with attention layer, in order to extract useful information, is used in [14] to develop deep learning structure for automatic Arabic speech recognition model. At first, the proposed structure performs data processing to determine MFCC features using an LSTM model and an attention layer to eliminate unnecessary extracted data from the last LSTM layer. The proposed model is evaluated using a standard Arabic single speaker corpus and they report a 28.48% word error rate (WER). Alsayadi et al. [15] proposed a combination of convolutional neural network (CNN) with LSTM model for speech recognition. They evaluated the proposed model using a standard Arabic single-speaker corpus. They concluded that with the removal of diacritics out-of-vocabulary, they could reduce WER to 13.52%.

In summary, all of the reviewed articles used either machine learning or deep learning models for ASR. Based on recent work, the most promising result have been reported by using hybrid deep learning models. A summary of the proposed structure is shown in Figure 1. Inspired by the experience of previous researches studies [11, 14, and 15], this work focuses on developing a proposed structure of an Autoencoder for Arabic speech recognition with two of the most popular branches of recurrent neural network (RNN) [16] LSTM, GRU, and their combination. The first part is an encoder that is used for feature extraction, and the second part is a decoder that is used for detection. The achieved

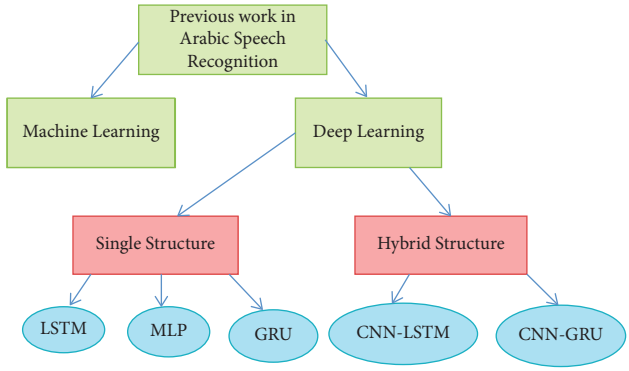

Figure 1: Summary of the existing models of Arabic speech recognition.

accuracy value is 95.04% and 4.09% for the word error rate (WER) to detect the noisy Arabic words that were recorded by the EL device. Experimental results indicate the superiority of GRU as an encoder and decoder for ASR. The proposed model increases the quality of recorded sound and helps to detect and clarify pronounced words better. The proposed structure is composed of two parts.

2. Dataset

Arabic speech datasets for native Iraqi people are considered in this work. The speech is recorded by three groups: two children from age of 9 to 10, three women from age of 34 to 37, and three men from age of 40 to 50. The total number of samples in the dataset is 1040, with 520 representing the pure (normal) samples and the rest being noisy samples that were collected using an EL device. The dataset contains common Arabic expressions like welcome and good morning beside numbers from one to ten. The dataset contains 30 different classes of words in both pure and noisy datasets for 10 persons. Spoken Arabic words in the datasets are indicated in Table 1.

Samples of the used dataset are shown in Figure 2. The length of recorded samples in both cases for the same class of words is not the same. As shown in Figure 2, the difference between normal and noisy samples of the recorded voice is significant.

An example of recorded normal and noisy samples and the difference between them is shown in Figure 3. As shown, the noisy signal for "Ahlan wa Sahlan" words are longer than the normal sample. Also, with the use of an EL device, the locations of announced letters are changed. In order to maintain a proper situation for gathering input datasets, both noisy and normal datasets for each person were recorded in similar situations, and the difference is only the device setting for generating noisy sounds in the dataset.

3. The Proposed Algorithms

3.1. Preprocessing and Feature Extraction. In order to alleviate the effect of noise in recognition tasks, different denoising and filtering techniques are available to convert corrupted speech (as in the EL device case) to quasi-pure

Table 1: The prepared Arabic dataset.

Sentences in Arabic	Corresponding representation in English
أهلا و سهلا	"Ahlan Wasahlan"
عجلة الرا	"Aljaw Haar"
الوداع	"Alwadaa"
أرك غدا	"Arak Gadan"
أين ذهبت	"Ayn Dahabt"
أين أنت	"Ayn Ant"
عطفا أنا يمني	"Enaataf Yamenan"
ثمانية	"Thamanya"
في أمان الله	"Fi Aman Allah"
خمسة	"Khamsa"
أربعة	"Arbaa"
هل أنت مريض؟	"Hal Anta Mareedh"
كيف الحال؟	"Kahifaa Alhaal"
لا تقترب	"La Taqtarib"
لن أبقى	"Lan Akbaal"
ما اسمك؟	"Ma Esmak"
ماذا تعمل؟	"Madha Taamal"
مساء الخير	"Masaa Alkhair"
السادسة السا عة	"Al Saa Al Sadisaa"
متى سوف تأتون؟	"Mata Sawfaa Taatoon"
تسعة	"Tesaa"
واحد	"Wahid"
صباح الخير	"Sabah Alkhair"
السلام عليكم	"Al Salam Alaikum"
سبعة	"Sebaa"
ستة	"Setaa"
عشرة	"Ashraa"
ثلاثة	"Thalatha"
اثنان	"Ethnan"
يوم الجمعة	"Yaum Al Jumaa"

instances that may be followed. Different filtering techniques, such as low path, high path, and middle path filters [17], were tested to see the best arrangement to filter out the noise. Wavelet denoising is also applied to eliminate the noise from the recorded samples [18]. Both Fissuring and Bayes Shrink [19] approaches were considered as denoising techniques to see which one performs better. Based on experimental results using the fissuring approach, it is better with Arabic speech recognition. Fissuring is an approach that is designed to remove additive Gaussian noise with high probability, which tends to result in overly smoothed signals appearance [20]. This approach employs a single universal threshold for all wavelet coefficients.

After the denoising step with the wavelet technique, the next step is feature extraction. MFCC is one of the successful techniques for feature extraction [11, 14, 15]. Experiments revealed that MFCC is a commonly utilized technique, especially for a noisy dataset like the collected speech dataset produced by an EL device. In this work, for each sample, a vector of 128 MFCC features is created, where this value has been approved to get better results in comparison to using 10, 20, 40, 80, 120, and 200 MFCC features. The MFCC feature extraction technique includes windowing the signal, applying the discrete fourier transform (DCT), taking the log of the magnitude, and then warping the frequencies on a Mel scale, followed by applying the inverse discrete cosine

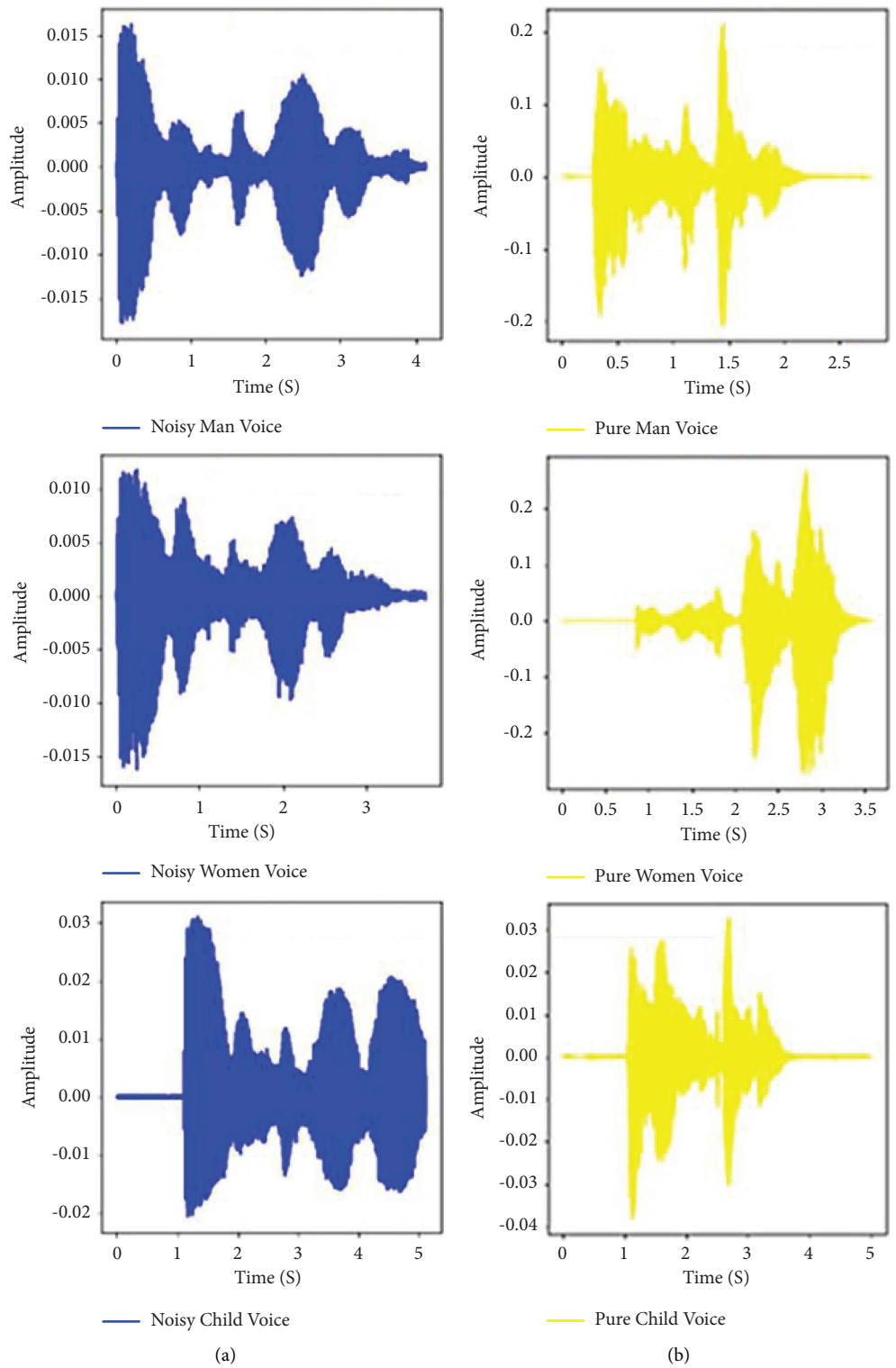

Figure 2: Recorded signals in (a) pure samples and (b) noisy samples.

transform [21]. In a summary, the results of the denoising and feature extraction steps are fed to the proposed deep learning (DL) model for recognizing the words. These are obtained by separating the dataset into training and testing sets. The dataset is divided into 70% for training with 728 samples and 30% for testing with 312 samples.

3.2. Deep Learning. Recurrent neural network (RNN) and their branches have useful applications in time series prediction [22]. RNNs' memory is called recurrent hidden states and gives the RNNs the ability to predict what input is coming next in the sequence of input data. However, due to RNNs' memory limitations, the length

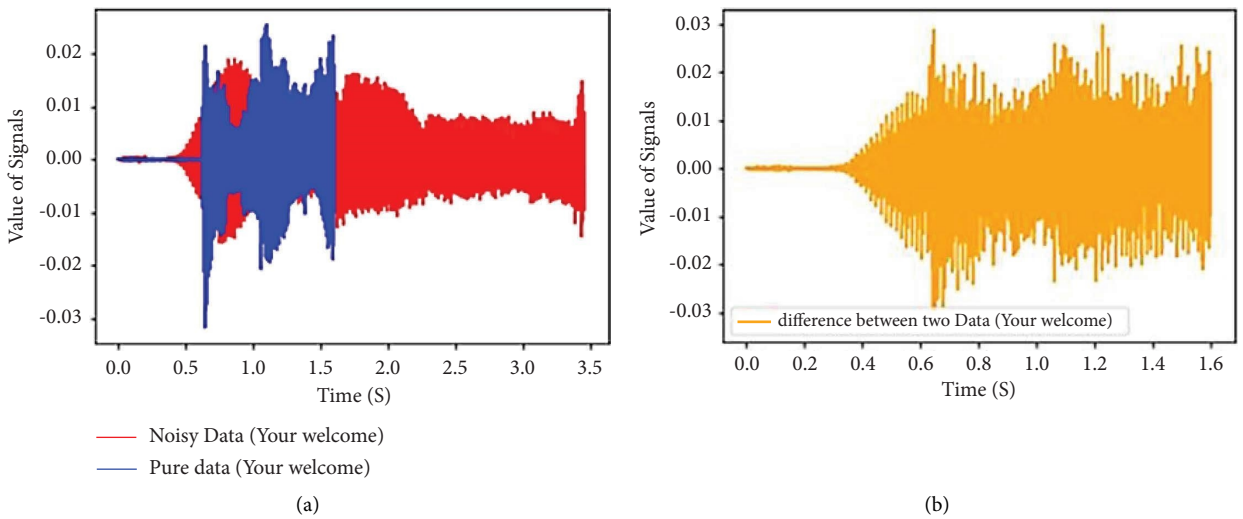

Figure 3: Recorded voices in (a) pure and noisy cases and (b) differences between them.

of the sequential information is limited to only a few steps back. RNN branches are as follows.

3.2.1. Long Short-Term Memory. To overcome the problem of short memory in RNN, various structures like long short-term memory (LSTM) have been developed. LSTM has four gates to not only remember the important part of long-term memory but also to improve the flow of the forward signals in the LSTM unit structure too [23]. The structure of the LSTM is shown in Figure 4.

$$\text{Sigmoid}(x) = \frac{1}{1+e^{x}}, \tag{1}$$

$$\text{Tan}h(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}, \tag{2}$$

$$f_t = \text{Sigmoid}\left(x_t \times \text{w}_f + w_f \times h_{t-1} + b_f\right), \tag{3}$$

$$i_t = \text{Sigmoid}\left(x_t \times w_i + w_i \times h_{t-1} + b_i\right), \tag{4}$$

$$\text{out}_t = \text{Sigmoid}\left(x_t \times w_{\text{out}} + w_{\text{out}} \times h_{t-1} + b_{\text{out}}\right). \tag{5}$$

Equations (1) and (2) illustrate the calculations for sigmoid and hyperbolic tangent activation functions. The procedure for calculating forget and ignore factors is indicated in equations (3) and (4) The output of LSTM cells is shown in equation (5). As shown in Figure 4, the first part of the LSTM unit consists of two different gates. One of them is the forget gate, which uses a forget factor to eliminate part of the input signals. A sigmoid activation function is used to compute the forget factor for the input dataset. By using the sigmoid, the range of the forget factor is between 0 and 1. Thus, it is very hard to get results close to 0 or 1 since the input of forget factor must reach ∞ or $-\infty$. Applying

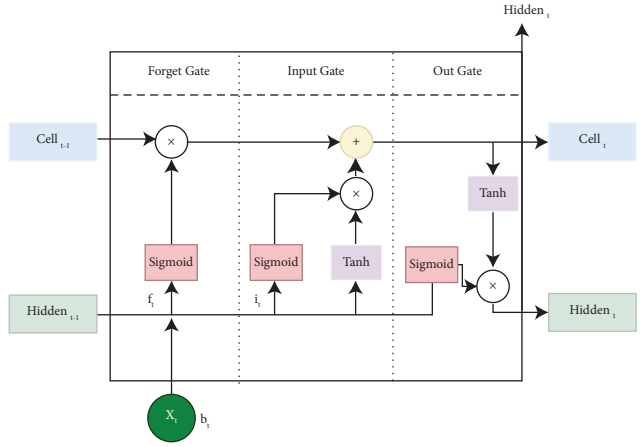

Figure 4: Structure of the LSTM.

sigmoid function would allow the LSTM to forget unnecessary information from the last hidden layer and remember the essential information from the same layer. Ignoring the gate helps to ignore unnecessary information for the next stage. The structure of ignore factor is similar to forget factor, but instead of using short term memory as a hidden layer, it uses input gates as the previous layer to work with. In the learning section, the outputs of the ignore and forget factors with the input will be accumulated together. To increase the performance of the LSTM cell, instead of using the sigmoid function, an absolute tangent hyperbolic function can be used. Utilizing this function, the LSTM unit would extract information from negative values as well as positive values. The tangent hyperbolic function has an output range from −1 to 1.

3.2.2. Gated Recurrent Unit. Gated recurrent unit (GRU) is a kind of gated RNN that is used to solve the common problems of vanishing and exploding gradients in traditional RNNs when learning long-term dependencies [24].

GRU is a variation of LSTM; both have similar designs and, in the case of speech recognition, both produce equally excellent results [25]. The structure of GRU is shown in Figure 5, with the following equations describing the calculation of the main parameters [24]:

$$r_t = \text{Sigmoid}\left(x_t \times w_r + w_r \times h_{t-1} + b_r\right), \quad (6)$$

$$z_t = \text{Sigmoid}\left(x_t \times w_z + w_z \times h_{t-1} + b_z\right), \quad (7)$$

$$\text{Out}_t = \left(1 - z_t\right) \times h_{t-1} + z_t \times (\text{Tan}h \left(x_t \times w + w \times r_t \times h_{t-1} + b\right)). \quad (8)$$

As illustrated in Figure 5, there is an input layer composed of multiple neurons the number of neurons is determined by the size of the feature space. Similarly, the number of neurons in the output layer corresponds to the output space. The hidden layer (s) containing memory cells cover the main functions of the GRU networks. Changes and maintenance of cell status depend on two gates in the cell [24]: a reset gate r_t and an update gate z_t.

The calculation process for the reset gate is shown in equation (6), while the calculation of GRU output is shown in equation (8). The key distinction between vanilla RNNs and GRUs is that the latter supports gating of the hidden state. To solve the vanishing gradient problem of the standard RNN, GRU is used to update and reset the gates. Basically, these two factors decide what information should pass to the output [24]. They can be trained to keep information for a long time depth, without forgetting it or removing information which is irrelevant to the prediction. These gates decide when the information's hidden state should be updated and when this information should be eliminated. Instead of using two gates for remembering and forgetting information, GRU uses the reset gate to set the flow of information [25]. By using one gate instead of two, the calculation for reaching the output is decreased. This leads to a faster convergence rate compared to LSTM.

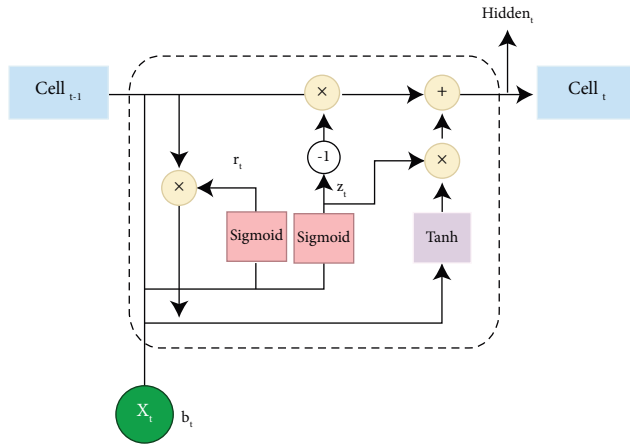

Figure 5: Structure of GRU.

4. Proposed Model

In this section, the proposed model is presented in detail. The model includes two sections: the encoder and the decoder. The merit of using an autoencoder is the process of encoding the input into latent space and then applying these features for recognition. The encoder contains two spots (or layers) for LSTM or GRU and two spots for dropout layers [26]. The dropout layer is used to decrease the chance of overfitting while training the model. The decoder consists of one repeated vector, one layer for GRU or LSTM, another layer for dropout, and at last a dense layer for detection. Model processing starts from the raw input dataset to denoising the signal using the fissuring denoising technique using Symmlet8. The MFCC feature extraction step is performed for the denoised signals. After preprocessing steps, the autoencoder is trained in a supervised manner with the training dataset, and then the proposed model is tested with the noisy dataset for evaluation. Various combinations of memory units in different cell spots are tested and evaluated until the best performance is obtained. The structure that guarantees the best possible outcome is considered the proposed autoencoder architecture. The proposed structure is shown in Figure 6.

Based on the autoencoder structure, we have decided to evaluate different types of activation functions, such as the rectifies linear unit (ReLU), leaky rectifies linear unit (Leaky RLU), and self-normalize linear unit (SLU) [27]. To keep sustainability in our network, the symmetric structure of the encoder and decoder is preserved. So, the same number of units are used for encoder and decoder parts in each layer. Also, the same dropout ratio was applied for the encoder and decoder, too. A flattening layer is used to create the flat vector before the final dense layer. The last layer is used to specify the class of the recognized words. The relationship between the flat and dense layers is fully connected, and there are no dropout factors between them. The size of this vector is equal to the size of the encoder output and decoder input.

5. Experimental Results

Based on achieved results, the use of wavelet denoising filter with the Fissuring (VisuShrink) technique showed superior performance compared to other wavelet filtering techniques. In order to ensure that this technique got the best result for denoising, other structures like unsupervised denoising techniques are tested. Those unsupervised techniques had shown good results with the English language but due to the sophisticated morphology of the Arabic language, their performance is not acceptable. For the feature extraction part, the use of MFCC is the best option as it extracts an even size vector for the input noisy and pure signals. The results of feature extraction for the input normal and noisy dataset are shown in Figure 7. In Figure 7(a), the extracted values of MFCC are stretched along the original signal value. But, due to the use of denoising and feature extraction procedures, the difference between the two signals is decreased significantly. As shown in Figure 7(b), the extracted value for the normal and noisy signals are not the same.

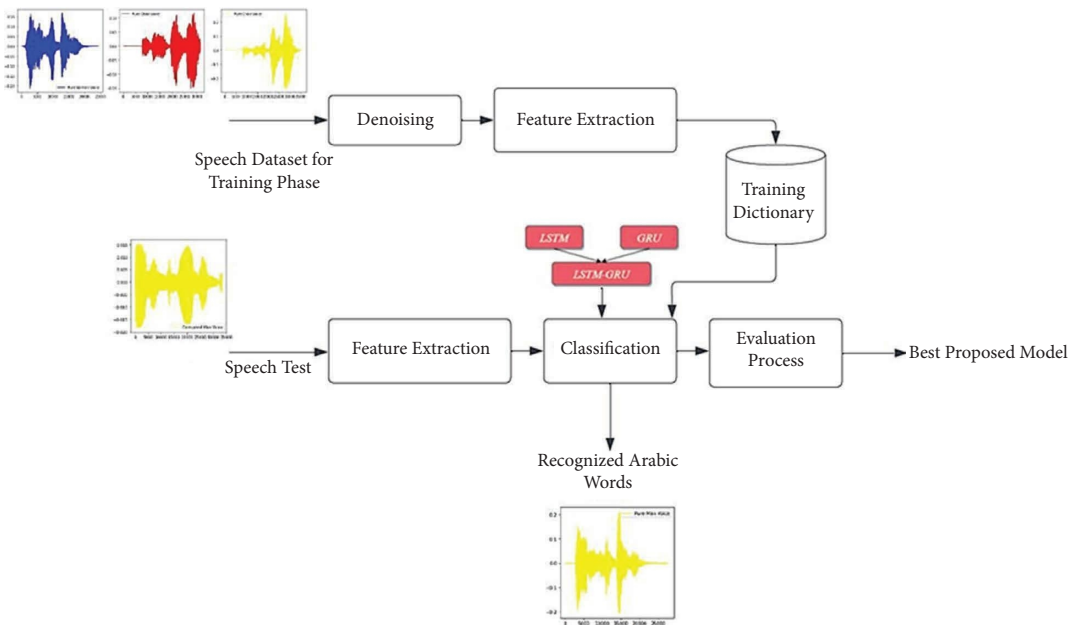

Figure 6: The proposed structure.

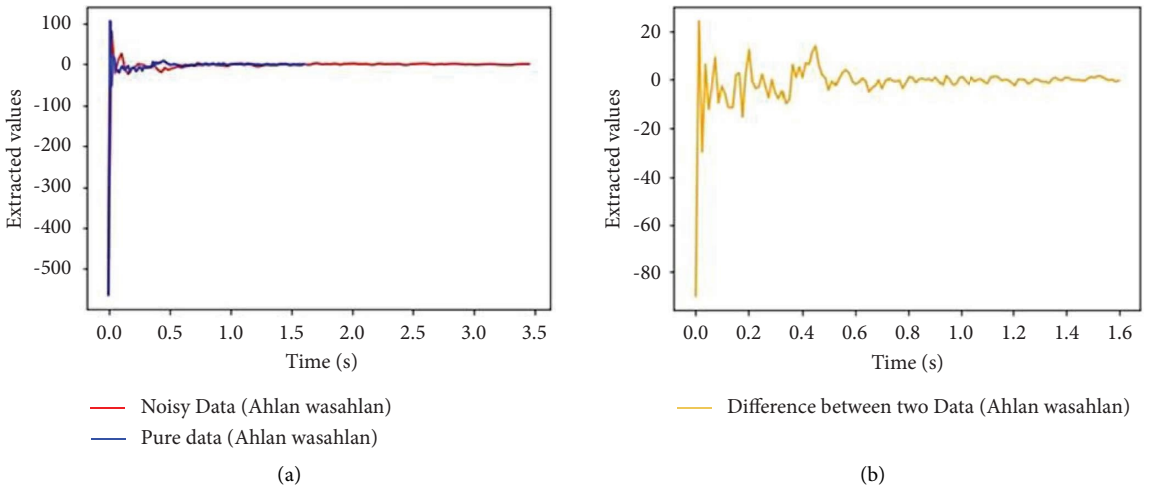

Figure 7: Extracted features in (a) normal and noisy samples and (b) difference between them.

Various groups of memory cells, activation functions, and dropout factors were evaluated to choose the best arrangement for the proposed ASR model. Table 2 shows the obtained results for the best model. As shown in Table 2, an autoencoder is developed with either LSTM or GRU as memory cells in each layer and a combination of both memory units to choose the best architecture. The versatile options for choosing the autoencoder architecture ensure the best possible outcome for the hyperparameters of the model, too. In all situations, the number of memory units and dropout factors on the encoder and decoder sides is the same to preserve a quasi-symmetry in the autoencoder. The result of using this structure for training and testing is shown in Table 3. As it is shown in Table 3, the combination of GRU, and GRU as the memory cells in the autoencoder structure demonstrated superior results to other autoencoders. The other two models have shown good results in the training phase too, but the result on the testing dataset is overfitted. For better observation of the proposed model, the result of training and validation sets is shown in Figure 8. The results of precision for each class of words are shown in Figure 9. As it is shown in Figure 8, the proposed model faced slight overfitting. The weight of the proposed model was saved based on the best result. In Figure 9, the precision of the correctly predicted words is presented. The performance of the proposed model shows that words like "Alwadaa," and numbers like "Arbaa" and "Ashara" got perfect values. Words like" Al Saa Al Sadisaa" and "Ayn dhahabta," which are beginning with the letter "A" have been mistaken for each other. Also, WER is calculated for this model. The result of WER for the proposed model is 4.097%.

Table 2: The best chosen model structure for Arabic speech recognition.

	Different specifications for each model		
Name of each layer	Autoencoder1	Autoencoder2	Autoencoder3
Encoder layer 1 (120 units) dropout	LSTM (120 units) 0.25 rate	GRU (120 units) 0.25 rate	LSTM (120 units) 0.25 rate
Encoder layer 2	LSTM (60 units)	GRU (60 units)	LSTM (120 units)
Dropout	0.25 rate	0.25 rate	0.25 rate
Encoder layer 3	LSTM (120 units)	GRU (120 units)	LSTM (120 units)
Decoder layer 1	LSTM (120 units)	GRU (120 units)	GRU (120 units)
Decoder layer 2	LSTM (60 units)	GRU (60 units)	GRU (60 units)
Dropout	0.25 rate	0.25 rate	0.25 rate
Flatten layer	—	—	—
Recognize layer			
Dense layer	SoftMax (10 neurons)	SoftMax (10 neurons)	SoftMax (10 neurons)

Table 3: Results of Arabic language recognition with various models' architectures.

Model name	Training accuracy (%)	Training loss	Testing accuracy (%)	Testing loss
Autoencoder1	96.58	0.0945	93.66	0.2927
Autoencoder2	96.83	0.0982	95.31	0.1504
Autoencoder3	95.98	0.1154	92.61	0.3672

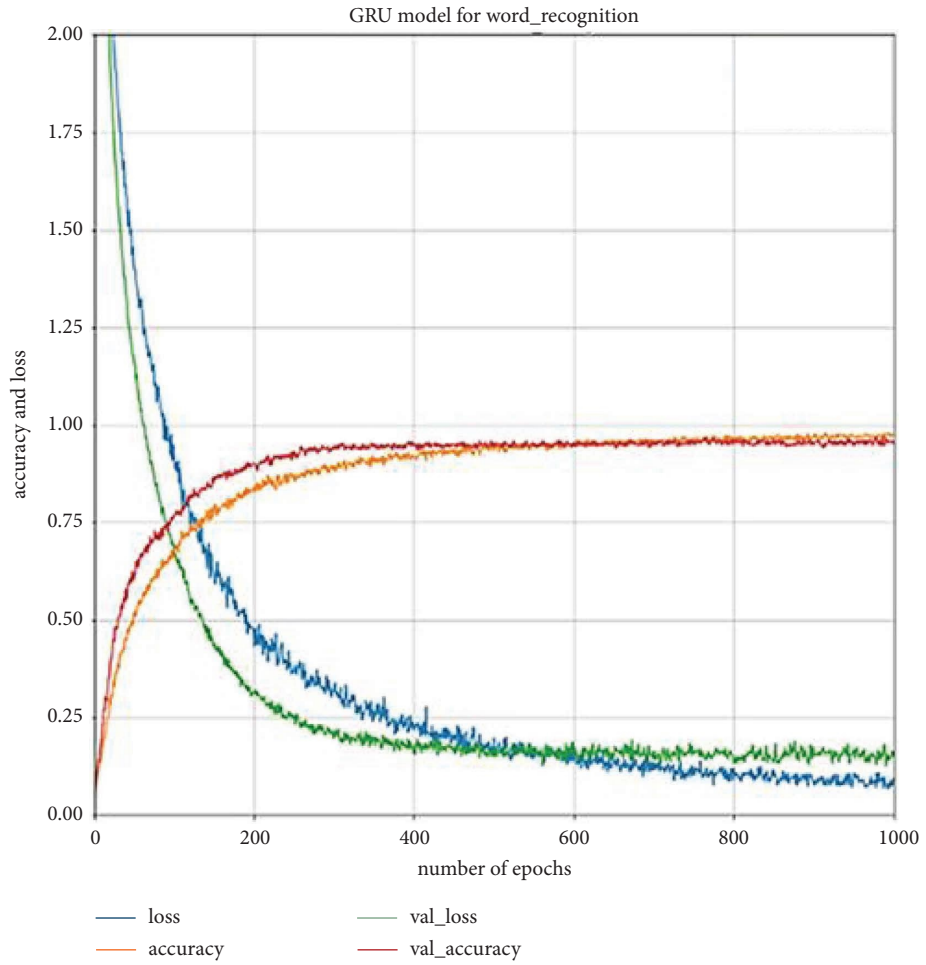

Figure 8: Proposed model accuracy and loss.

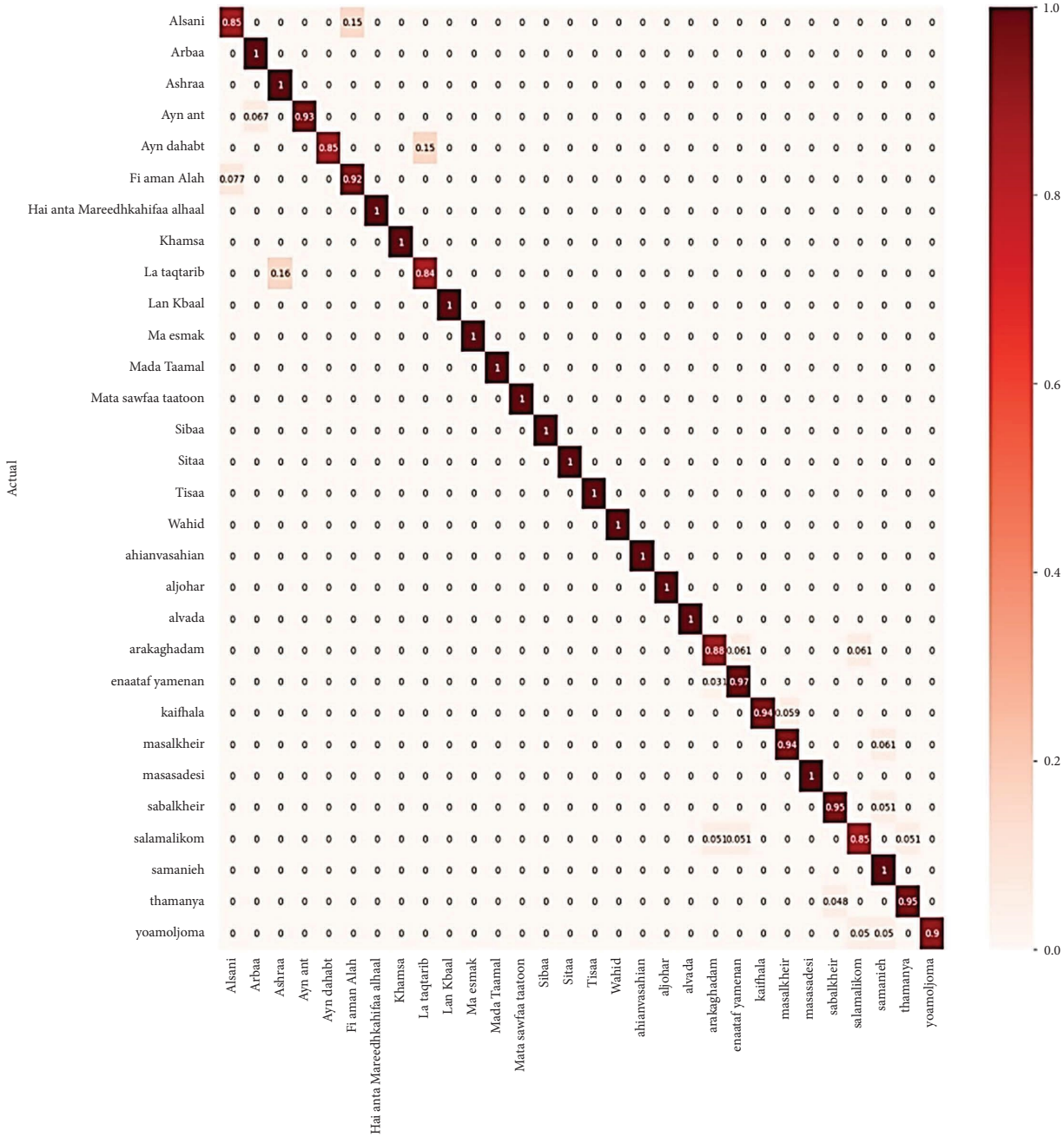

Figure 9: Proposed model confusion matrix.

6. Discussion

The Arabic language is among the six most usable of the world's major languages. The Arabic language is the language of the Qur'an, the holy book of Islam. So, the Arabic language is wildly used among Muslims around the world too. With this variety of usage, there are people with disorders who use the Arabic language for communication. There are different devices that are used to help people with speech disabilities communicate better with others. One of these devices is the Servox Digital Electro-Larynx (EL). The quality of the speech generated with this device is not good enough, especially for the Arabic language. Thus, in this research, we developed an autoencoder model to better recognize spoken Arabic words by this device. First, we managed to gather a proper dataset from males, females, and children from Iraq that speak Arabic language as their mother tongue. Then we utilize the developed framework based on this dataset to recognize spoken words correctly. The proposed structure comprised steps like preprocessing, denoising, and recognition. Based on the nature of the signals, we developed an autoencoder using LSTM, GRU, and a combination of them. The experimental result of the proposed model has shown a 95.31% recognition rate on the

Table 4: Comparison of the proposed model with related works for Arabic language recognition.

Reference	Model's name	Accuracy	Word error rate
Jaber and Abdulbaqi [28]	Autoencoder (CNN)	93%	—
Eljawad et al. [7]	Fuzzy neural network	94.5%	—
Dendani et al. [11]	Autoencoder (MLP)	65.72%	—
Alsayadi et al. [14]	Autoencoder (LSTM)	71.58	28.42%
Alsayadi et al. [15]	CNN-LSTM	—	13.52%
Proposed model	Autoencoder (GRU)	95.31%	4.69%

testing set. In order to justify the proposed model, we have compared the results of this model with similar Arabic speech recognition.

As it is shown in Table 4 the proposed structure got better results in comparison with similar work in the case of recognition accuracy and WER. Also, based on the obtained result in Figure 8 the proposed model has shown 99% accuracy for the recognition of spoken digits in the Arabic language. This result is superior to [13]. Furthermore, the proposed structure is evaluated by noisy signals that are totally corrupted, which confirms the capability of the proposed model for dealing with such signals.

7. Conclusion

This research focused on developing a deep learning model based on a combination of LSTM and GRU memory cells for Arabic speech recognition. First, we managed to gather a proper dataset from males, females, and children from Iraq that speak the Arabic language as their mother tongue. Then we utilize the developed framework based on this dataset to recognize spoken words correctly. The proposed structure comprised steps like preprocessing, denoising, and recognition. Based on the results, the developed Autoencoder using GRU, in both the encoder and decoder side demonstrated the best performance. Experimental results achieved about 95.31% accuracy for Arabic word recognition with a WER 4.097%. This research has introduced one of the practical solutions for Arabic speech recognition based on Servox Digital EL which is used for people with speech disorders towards improving their speaking performance.

Data Availability

The data that support the findings of this study are not openly available and are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, "Arabic natural language processing: an overview," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 5, pp. 497–507, 2021.

[2] A. Shoufan and S. Alameri, "Natural language processing for dialectical Arabic: A survey," in *Proceedings of the Second Workshop on Arabic Natural Language Processing*, Beijing, China, July 2015.

[3] J. Vojtech, M. Chan, B. Shiwani et al., "Surface electromyography–based recognition, synthesis, and perception of prosodic subvocal speech," *Journal of Speech, Language, and Hearing Research*, vol. 64, pp. 1–20, 2021.

[4] K. Darwish, "Arabizi detection and conversion to Arabic," in *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pp. 217–224, Doha, Qatar, October 2014.

[5] R. Duwairi, M. Alfaqeh, M. Wardat, and A. Alrabadi, "Sentiment analysis for Arabizi text," in *Proceedings of the 2016 7th International Conference on Information and Communication Systems (ICICS)*, pp. 127–132, IEEE, Irbid, Jordan, April 2016.

[6] B. Dendani, H. Bahi, and T. Sari, "Self-supervised speech enhancement for Arabic speech recognition in real-world environments," *Traitement du Signal*, vol. 38, no. 2, pp. 349–358, 2021.

[7] L. Eljawad, R. Aljamaeen, M. Alsmadi et al., "Arabic voice recognition using fuzzy logic and neural network," *International Journal of Applied Engineering Research*, vol. 14, no. 3, pp. 651–662, 2019.

[8] S. Hamsa, I. Shahin, Y. Iraqi, and N. Werghi, "Emotion recognition from speech using wavelet packet transform cochlear filter bank and random forest classifier," *IEEE Access*, vol. 8, pp. 96994–97006, 2020.

[9] A. Ali, H. S. Abdullah, and M. Fadhil, "Voice recognition system using machine learning techniques," *Materials Today Proceedings*, 2021.

[10] I. Shahin and A. Nassif, "Emirati-accented speaker identification in stressful talking conditions," in *Proceedings of the 2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, pp. 1–6, IEEE, Ras Al Khaimah, UAE, November 2019.

[11] B. Dendani, H. Bahi, and T. Sari, "Speech enhancement based on deep AutoEncoder for remote Arabic speech recognition,"

in *International Conference on Image and Signal Processing*, pp. 221–229, Springer, New York, NY, USA, 2020.

[12] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Article ID 132306, 2020.

[13] N. Zerari, S. Abdelhamid, H. Bouzgou, and C. Raymond, "Bidirectional deep architecture for Arabic speech recognition," *Open Computer Science*, vol. 9, no. 1, pp. 92–102, 2019.

[14] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, and Z. T. Fayed, "Arabic speech recognition using end-to-end deep learning," *IET Signal Processing*, vol. 15, no. 8, pp. 521–534, 2021.

[15] H. A. Alsayadi, A. A. Abdelhamid, I. Hegazy, and Z. T. Fayed, "Non-diacritized Arabic speech recognition based on CNN-LSTM and attention-based models," *Journal of Intelligent and Fuzzy Systems*, vol. 41, no. 6, pp. 6207–6219, 2021.

[16] Y. Tai, H. He, W. Zhang, and Y. Jia, "Automatic generation of review content in specific domain of social network based on RNN," in *Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, pp. 601–608, IEEE, Guangzhou, China, June 2018.

[17] Y. C. Lien, E. A. M. Klumperink, B. Tenbroek, J. Strange, and B. Nauta, "Enhanced-selectivityhigh-linearitylow-noisemixer-first receiver with complex Pole pair due to capacitive positive feedback," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 5, pp. 1348–1360, 2018.

[18] J. Tang, S. Zhou, and C. Pan, "A denoising algorithm for partial discharge measurement based on the combination of wavelet threshold and total variation theory," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 3428–3441, 2020.

[19] F. Bayer, A. Kozakevicius, and R. Cintra, "An iterative wavelet threshold for signal denoising," *Signal Processing*, vol. 162, pp. 10–20, 2019.

[20] P. Ravisankar, "Underwater acoustic image denoising using stationary wavelet transform and various shrinkage functions," *Electronic Letters on Computer Vision and Image Analysis*, vol. 20, no. 2, 2021.

[21] H. Elharati, M. Alshaari, and V. K¨epuska, "Arabic speech recognition system based on MFCC and HMMs," *Journal of Computer and Communications*, vol. 8, no. 3, pp. 28–34, 2020.

[22] S. Selvin, R. Vinayakumar, E. Gopalakrishnan, V. Menon, and K. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proceedings of the 2017 international conference on advances in computing, communications and informatics (ICACCI)*, pp. 1643–1647, IEEE, Udupi, India, September 2017.

[23] W. Zhang, W. Guo, X. Liu et al., "LSTM-based analysis of industrial IoT equipment," *IEEE Access*, vol. 6, pp. 23551–23560, April 2018.

[24] G. Shena, I. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep learning with gated recurrent unit networks for financial sequence predictions," *Procedia Computer Science*, Elsevier, vol. 131, pp 895–903, 2018.

[25] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU neural network performance comparison study: taking Yelp review dataset as an example," in *Proceedings of the 2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, pp. 98–101, IEEE, Shanghai, China, June 2020.

[26] C. Wei, S. Kakade, and T. Ma, "The implicit and explicit regularization effects of dropout," in *Proceedings of the International Conference on Machine Learning*, pp. 10181–10192, PMLR, New York, NY, USA, July 2020.

[27] K. Eckle and J. Schmidt-Hieber, "A comparison of deep networks with ReLU activation function and linear spline-type methods," *Neural Networks*, vol. 110, pp. 232–242, 2019.

[28] H. Q. Jaber and H. A. Abdulbaqi, "Real time Arabic speech recognition based on convolution neural network," *Journal of Information and Optimization Sciences*, vol. 42, no. 7, pp. 1657–1663, 2021.