Development of a Deep Learning-based Arabic Speech Recognition System for Automatons

Abdulrahman Alahmadi
Department of Computer Science and Information, Taibah University, Medinah, Saudi Arabia
aahmadi@taibahu.edu.sa

Ahmed Alahmadi
Department of Computer Science and Information, Taibah University, Medinah, Saudi Arabia
aahmadio@taibahu.edu.sa

Eman Alduweib
Department of Computer Science, University of Petra, Amman, Jordan
eman.alduweib@uop.edu.jo

Waseem Alromema
Department of Computer Science and Information, Taibah University, Medinah, Saudi Arabia
wromema@taibahu.edu.sa (corresponding author)

Bakil Ahmed
Information Technology Department, Engineering and Information Technology Faculty, Al-Qalam University, Yemen
bakil_ahmed@quni.edu.ye

Received: 7 August 2024 | Revised: 12 September 2024, 28 September 2024, and 12 October 2024 | Accepted: 16 October 2024

 DOI: https://doi.org/10.48084/etasr.8661

ABSTRACT

The latest developments in voice recognition have achieved amazing results that are on par with those of human transcribers. However, this significant efficiency may not apply to all languages, nor Arabic. Arabic is the native language of 22 countries and is spoken by approximately 400 million individuals. Verbal difficulties have become a growing problem in recent decades, especially among children, and data samples on Arabic phonetic recognition are limited. For Arabic pronunciation, Artificial Intelligence (AI) techniques show encouraging results. Some devices, such as the Servox Digital Electro-Larynx (EL), can produce voice for such individuals. This study presents a Deep Learning-based Arabic speech recognition system for automatons to recognize captured sounds from the Servox Digital EL. The proposed system employs an autoencoder using a mix of Long-Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models. The proposed approach has three main stages: de-noising, feature extraction, and Arabic pronunciation. The experimental findings demonstrate that the proposed model was 95.31% accurate for Arabic speech recognition. The evaluation shows that the use of GRU in both the encoding and decoding structures improves efficiency. The proposed model had a Word Error Rate (WER) of 4.69%. The test results demonstrate that the proposed model can be used to create a real-time application to recognize commonly spoken Arabic words.

Keywords-deep learning; electro-larynx; arabic speech recognition; long-short-term-memory; voice recognition

I. INTRODUCTION

Arabic is the native language of 22 countries and is spoken by approximately 400 million people worldwide [1]. Several studies have focused on employing algorithms for Automatic Voice Recognition (AVR), inspired by recent advances in the field of Artificial Intelligence (AI) and Natural Language Processing (NLP), especially in the morphological examination, resource creation, and computational translation of Arabic dialects [2]. As speech impairments can be caused by several diseases or disorders, many people are unable to speak correctly and clearly. However, various technologies can be

used to help them produce voices from their vocal membranes (throat). The Servox Digital Electro-Larynx (EL) [3] is one such device, which employs a technology that produces quasi-clear speech and assists in communication. However, since its output audio has poor quality, an autoencoder based on LSTM and GRU was proposed to recognize Arabic speech through this tool. Samples were generated in the United Arab Emirates, including indifferent, gradual screams, loud, quiet, and quick talk. In challenging environments, this model achieved a speech recognition accuracy of 65.0% [4].

In [4], a holistic approach was proposed for the recognition of words in Urdu using deep neural networks. In [5], a tiny Arabic-commented library was handcrafted by collecting 3026 communications that were transcribed into Arabic. Automatic speech detection is often accomplished in two stages: the collection of acoustic information and its encoding. In [6], the efficiency of AVR algorithms was improved by employing an autoencoder technique. More mixed frameworks are now offered to extract extensive arrays of features through audio. The studies in [7-8] used multiphase feature extraction. In [7], discrete levels from the source dataset for every word up to 2000 occurrences were removed. The recognition models used Multilayer Perception (MLP) and fuzzy logic, achieving 94.5% and 77.1% accuracy, respectively. Neural networks with arranged Mel Frequency Cepstral Coefficients (MFCC) have been employed to extract features to recognize counterfeits by looking at their language. In [9], a Hidden Markov Model with MFCC was used for Hebrew speech detection on a dataset of 50 samples (25 females and 25 males).

In [10], a clear speech recovery strategy was initially used on noisy samples to enhance speech. This approach used an autoencoder approach with five masked layers to improve Arabic recognition, achieving 65.7% accuracy. For Arabic speech identification, deep learning models including LSTM have shown promising results [11]. LSTM is intended to obtain features from time-sequence datasets. In [12, 13] LSTM-based techniques were proposed for automatic voice identification. LSTM or GRU examples were used, followed by MLP for classification. In [13], an LSTM model with an attention layer was used to develop an autonomous Arabic pronunciation recognition system. In [14], a Convolutional Neural Network (CNN) was combined with an LSTM model on a conventional Arabic single-speaker dataset. This study showed that by removing symbols from the terms, the Word Error Ratio (WER) can be decreased to 13.52%.

In summary, artificial intelligence and deep learning algorithms are being mainly utilized for AVR. According to recent studies, adopting mixed algorithms for deep learning yields promising results. The studies in [10, 13, 14] focused on developing an autoencoder design for Arabic speech detection, combining two of the most prominent segments of RNNs, LSTM and GRUs [15].

The proposed system has two parts: an encoder for extracting features and a decoder for classification. The proposed model achieved 95.31% accuracy while the corresponding WER was 4.69%. Research findings have shown that GRU is preferable as an AVR decoder and encoding algorithm. The proposed model improves the quality of audio recordings and assists in the identification and articulation of spoken syllables. Different approaches have been proposed for AVR, as stated above. However, audio conditions might be difficult, and the acquired dataset could include noise. In [8], an approach to recognizing Arabic emotions under stressful and noisy conditions was presented, based on Fourier transformation and RF. This approach was tested with Emirati-accented Arabic speech on a dataset with sounds of 25 males and 25 females with ages ranging from 14 to 55 years. Each participant delivered eight widely used Emirati statements, with durations of 2 to 5 seconds. Each participant articulated eight statements in all of the following emotions: frustrated, joyful, indifferent, sad, frightened, and disgusted. This approach achieved 89.60% accuracy in recognizing emotions in the Arabic dialect, showing its ability to recognize fear and depression with much greater precision than alternative emotions.

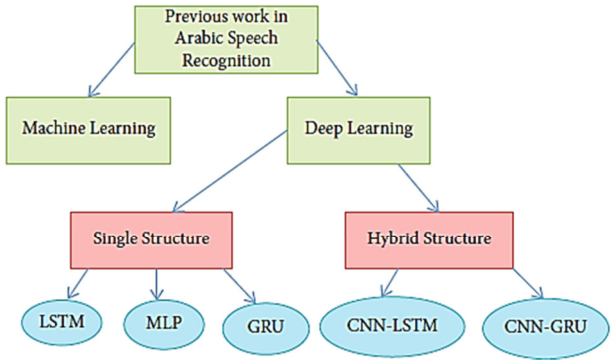

Fig. 1. Overview of current Arabic speech identification methods.

This study presents a novel deep-learning algorithm for Arabic speech recognition by investigating both LSTM and GRU models. The proposed approach is based on three main stages: de-noising, feature extraction, and Arabic pronunciation.

II. THE PROPOSED APPROACH

A. *Dataset*

This study used Arabic speech samples from two children aged 9 to 10, three women aged 34 to 37, and three men aged 40 to 50. The dataset contains 1040 samples, 520 of which are pure (normal) examples while the others are chaotic data obtained through an EL device. The dataset includes popular Arabic statements such as hello and good morning, as well as numerals ranging from one to 10. The dataset includes 30 different kinds of expression in both clean and chaotic datasets. Table I shows the usage of Arabic words in both datasets. Figure 2 illustrates some examples of the dataset used. The duration of the collected samples varied in both cases for the same category of words. Signals were collected in (a) chaotic samples as well as (b) pristine samples. The distinction between regular and noisy speech samples is pronounced.

Figure 3 depicts a case study of regular and chaotic data, as well as the distinctions between them. The chaotic signal (a) for the "Ahlan wa Sahlan / you are welcome" phrase was found to

be longer than the usual average. The placements of the indicated letters (b) are also modified when an EL device is used. To ensure a proper environment for collecting the input datasets, both the noise and normal datasets were collected in similar circumstances for everyone in the group, and the difference is simply the instrument configuration for providing noisy samples in the data file.

TABLE I. THE GENERATED ARABIC DATASET.

Arabic Phrases	Equivalent in English pronunciation	English phrase
أهلًا وسهلًا	"Ahln Wsahlan"	Welcome
الجو حار	"Algaw Har"	It's hot
الوداع	"Alwada"	Goodbye
أراك غدًا	"Ark Gdan"	See you tomorrow
أين ذهبت	"Ayn Dthahabt"	Where did you go
أين أنت	"Ain Ant"	Where are you
انعطف يمينًا	"Enataf Yamen"	Turn right
ثمانية	"thamania"	Eight
في أمان الله	"Fi Amaan Alah"	Goodbye
خمسة	"Khmsa"	Five
أربعة	"Arba"	Four
هل انت مريض	"Hal Ant Mared"	Are you sick
كيف الحال	"Kahifa Alhal"	How are you
لا تقترب	"La Taqtareb"	Don't come near
لن اقبل	"Lan Akbal"	I won't accept
ما اسمك	"Ma Esmaak"	What's your name
ماذا تعمل	"Madha Tamal"	What do you do
مساء الخير	"Masa Alkhaer"	Good evening
الساعه السادسه	"Al Saa Al Sadisa"	Six o'clock
متي سوف تاتون	"Mata Sawfa Tatoon"	When will you come
تسعه	"Tesa"	Nine
واحد	"Wahed"	One
صباح الخير	"Sabaah Alkhaer"	Good Morning
السلام عليكم	"Al Salam Alaykum	Peace be upon you
سبعه	"Seba"	Seven
سته	"Seta"	Six
عشره	"Ashra"	Ten
ثلاثه	"Thlatha	Three
اثنان	"Ethenan"	two

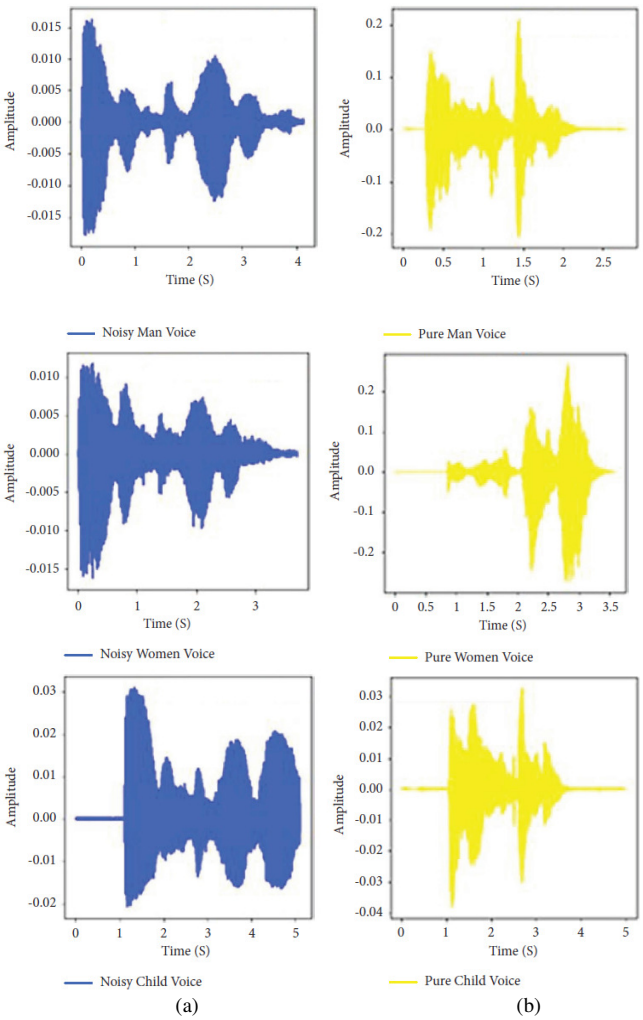

Fig. 2. Signals collected in (a) chaotic and (b) pristine samples.

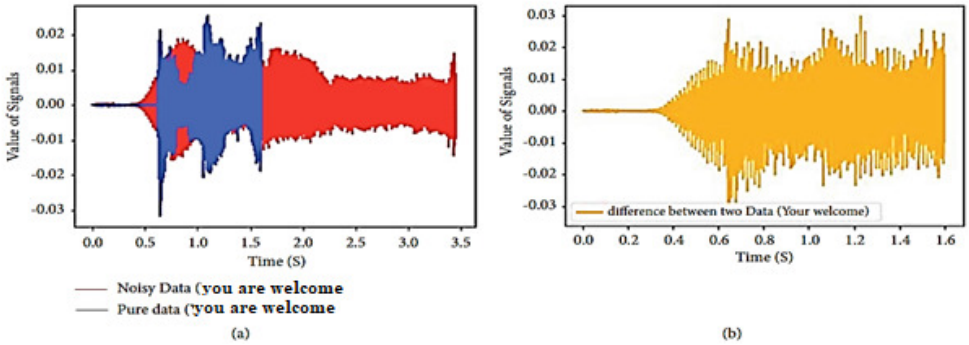

Fig. 3. Variations in captured sounds in (a) quiet and (b) loud examples.

B. *Preprocessing and Feature Extraction*

Diverse blurring and processing algorithms can be used to transform damaged audio (as in the EL device case) to quasi-pure examples and minimize the effect of noise in identification tasks. Diverse processing strategies, including low, high, and intermediate routes [16], were examined to determine the best configuration for processing the sounds. Wavelet elimination is frequently used to remove noise from captured recordings [17]. Fissuring and Bayesian inversion [18] methods were examined as noise reduction strategies to determine which performed better. According to the test results, fissuring was better for Arabic speech recognition.

Fissuring is a method to remove cumulative Gaussian distortion, leading to the emergence of smoothed waveforms [19]. For all wave coefficients, this method applies a single global cutoff. After the wavelet blurring stage, feature extraction comes. MFCC is an effective feature extractor [10, 13, 14]. Previous studies have shown that MFCC is an effective technique, particularly for noisy data such as speech samples collected using an EL device. In this study, a vector of 128 MFCC features was used for each sample, as it outperformed employing 10, 20, 40, 80, 120, and 200 MFCC features. The MFCC method includes the Differential Fourier Transformation (DCT), calculating the logarithmic average of the size, bending the frequency bands on a Mel scale, and subsequently executing the opposite discrete cosine transformation [20]. The data after de-noising were incorporated into the proposed deep learning algorithm for word recognition. The data were divided into training and testing sets, with 70% (719 samples) and 30% (314 samples).

C. Deep Learning

Periodic forecasting can benefit from RNNs, including its branches [21]. The storage capacity of RNNs is known as recurring hidden states, which allows the RNNs to predict what information will come next in a series of data being entered. However, as RNNs are given limited memory, the duration of the ordered data is restricted to a few cycles.

D. LSTM

Several frameworks, such as LSTM, have been proposed to address the issue of short-term storage in RNN. The LSTM architecture comprises four gate structures that not only preserve the crucial element of long-term memory but also increase the efficiency of the downstream signals [22].

The computations for sigmoid and tangential hyperbolic activation formulas are shown in (1) and (2). The formulas (3) and (4) explain the technique for determining memory and forget variables, whereas (5) shows the results of LSTM cells.

$$Sigmoid(x) = \frac{1}{1+e^{x}} \quad (1)$$

$$Tanh(x) = \frac{e^{x}-e^{x}}{e^{x}+e^{x}} \quad (2)$$

$$f_t = Sigmoid\ (x_t \times W_f + w_f \times h_t - 1 + b_f) \quad (3)$$

$$i_t = Sigmoid\ (x_t \times W_i + w_i \times h_{t-1} + b_i) \quad (4)$$

$$out_t = Sigmoid\ (x_t \times W_{out} + w_{out} \times h_{t-1} + b_{out}) \quad (5)$$

where f_t is the forget gate's output at a particular time step t, x_t is the input at time t, W_f is the weight matrix associated with the forget gate, w_f is the weight vector associated with the previous hidden state (h_{t-1}), h_{t-1} is the hidden state of the LSTM from the previous time step $t-1$, b_f is the bias term for the forget gate, i_t is the input gate's output at a particular time step t, W_i is the weight matrix associated with the input gate, w_i is the weight vector associated with the previous hidden state h_{t-1}, b_i is the bias term for the input gate, out_t is the final output at time t, W_{out} is the weight matrix that connects the input vector at the current time step to the output, w_{out} is the weight vector associated with the reset gate and hidden state, and b_{out} is the bias vector that is added to the computation.

The initial element of the LSTM unit is made up of two distinct gates. For example, the memory entry employs a forget factor to delete a portion of the incoming signals. The forget factor for the input dataset is calculated using an activation function with a sigmoid coefficient. The forgetting element ranges from 0 to 1 when using the sigmoid. As a result, getting results near 0 or 1 is extremely difficult because the forgotten effect input has to equal ∞ or -∞. Using a sigmoid function could enable the LSTM to lose superfluous details from the previous layer that was hidden while remembering essential details from this level.

Avoiding the entry point helps to ignore irrelevant information for the next stage. The layout of the disregard component is similar to that of the forget factor, which is opposed to employing immediate recall as a hidden layer, and employs input like the prior level. The forgetting element ranges from 0 to 1 when utilizing the sigmoid. Therefore, achieving results close to 0 or 1 is a big challenge, as the forgotten effect input must be equal to either. Using a sigmoid function allows LSTM networks to remove unnecessary information from the previous hidden layer while retaining important details from the current one.

E. Gated Recurrent Uni (GRU)t

GRU is a type of controlled RNN that is used to address the typical issue of vanishing gradients in standard RNNs while acquiring a dependence on time [23]. GRU has an architecture similar to LSTM, providing similarly satisfactory results in understanding speech [24]. The following formulas define the key variables [25].

GRU has a data level made up of numerous cells. The extent of the domain of features determines the total amount of neurons. Comparably, the resultant layer's amount of neurons correlates to the outputting area. Its major functions are covered by the hidden layer(s) comprising the cells that store information. The cell's internal status is changed and maintained by a pair of gates [23], the resetting gateway r_t and a modification gate z_t.

Equation (6) formulates the estimation method associated with the disable gate, (7) depicts the technique to determine the remember and disregard variables, whereas (8) depicts the approach to calculate the GRU return.

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

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

$$out_t = (1 - z_t) \times h_{t-1} + z_t \times (Tanh((x_t \times W + w_{out} \times r_t \times h_{t-1} + b_{out})) \quad (8)$$

where x_t is the input at time t, W_r is the weight parameter matrix, w_r is the weight vector associated with the previous hidden state h_{t-1}, b_r is the bias term, r_t is the resetting gate and z_t is the modification gate, $tanh$ is the nonlinear activation function, and out_t is the final output at time t.

The main difference between vanilla RNNs and GRUs is that the former allows for filtering on the buried state. GRU

uses update and reset gates to adress the problem of vanishing gradients. Essentially, both of these settings determine which data should be used to send results [23]. They can be taught to remember data for lengthy periods of time, either discarding them or eliminating data that are unnecessary for accurate forecasting.

These systems determine how the hidden status of the information should be changed or removed. GRU uses the reset signal to set data storage rather than a pair of gates to store or delete them [24]. The process of calculating to achieve the output is reduced by employing one gate rather than two. When juxtaposed with LSTM, this results in a more rapid convergence rate.

III. EVALUATION

The design of the model consists of two parts: the encoder and the decoder. The advantage of utilizing an autoencoder is that it encodes the input into a hidden space before it applies these features for identification. The encoding device has two LSTM or GRU hotspots and two layers for dropout [25]. The dropout component is designed to reduce the likelihood of overfitting while training. A decoder is made up of one repeating vector, one GRU or LSTM layer, another dropout layer, and finally a dense detecting layer. Model processing begins with the initial input data and progresses to eliminating the signal with the Symmlet8 whitening algorithm. Demised data are subject to the MFCC feature extraction stage. Following the preparation procedures, the autoencoder is trained using the training set and then evaluated using unprocessed data. Different storage configurations in different cell locations were tested and reviewed until the highest efficiency was achieved. The proposed autoencoder design ensures an excellent result. Figure 4 depicts the proposed structure design.

Various activation functions were examined, including Rectified Linear Unit (ReLU), Leaking Rectified Linear Unit (Leaky RLU), and Self-normalized Linear Unit (SLU) [26, 27]. The asymmetric arrangement of both the decoder and the encoder was kept to ensure system longevity. As a result, the encoder and decoder sections of each layer employ an identical amount of units. The same loss ratio was used for the encoder and the decoder. Following the anatomical thick level, a patenting level was employed to form the Pat vector. The final layer determines the class of recognized words. There are no dropout elements in the interaction between the thin and thick layers. The resulting vector has the same length as the encoder result and decoder feed.

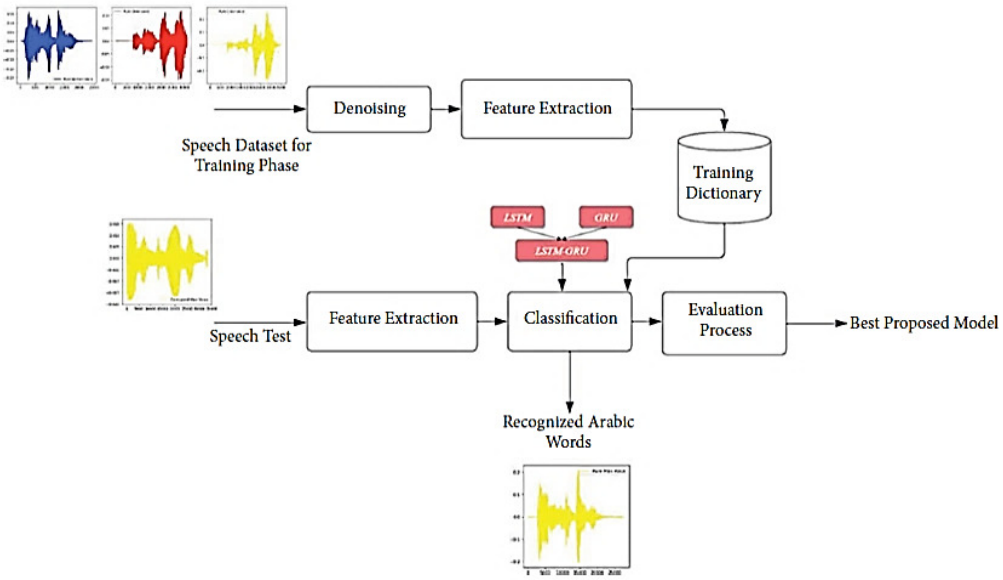

Fig. 4. The proposed design.

IV. RESULTS AND DISCUSSION

The results of the proposed framework were evaluated in comparison with similar Arabic transcription models. As illustrated in Table IV, the proposed framework outperformed comparable models in terms of detection precision and WER. Unregulated blurring methods have shown excellent results in English, but their effectiveness in Arabic is poor due to its complex morphology. The usage of MFCC for separating features is the best choice since it extracts an even-sized vector from both noisy and clean signals. Figure 7 shows the characteristics and collection results for the normal and chaotic datasets. Figure 7(a) shows the retrieved MFCC values extended over the initial signal intensity. However, the disparity between the two sources is significantly reduced because of the application of blurring and feature extraction processes. The number of pristine and noisy signals retrieved is not the same, as illustrated in Figure 7(b). To select the optimal configuration in the proposed AVR model, different combinations of cells with memories, activation mechanisms, and loss factors were tested.

TABLE II. OPTIMAL MODEL FRAMEWORKS FOR ARAB SPEECH DETECTION.

Different characteristics for every structure			
Name/ Levels	**Autoencoder 1**	**Autoencoder 2**	**Autoencoder3**
Encoder level 1 (120 units) dropout	LSTM (120 units) 0.25 rate	GRU (120 units) 0.25 rate	LSTM (120 units) 0.25 rate
Encoder level 2	LSTM (60 units)	GRU (60 units)	LSTM (120 units)
Dropout	0.25 rate	0.25 rate	0.25 rate
Encoder level 3	LSTM (120 units)	GRU (120 units)	LSTM (120 units)
Decoder level 1	LSTM (120 units)	GRU (120 units)	GRU (120 units)
Decoder level 2	LSTM (60 units)	GRU (60 units)	GRU (60 units)
Dropout	0.25 rate	0.25 rate	0.25 rate
Flatten level	-	-	-
Recognize level	-	-	-
Dense level	SoftMax (10 neurons)	SoftMax (10 neurons)	SoftMax (10 neurons)

Table II demonstrates the autoencoder designs that use LSTM or GRU memory cells at all levels and an amalgamation of both to select the optimal design. The various options for selecting the autoencoder design ensure the greatest chance of success for the algorithm's hyperparameters. To maintain the autoencoder's quasi-symmetry, the total amount of storage units and ejection ratios on both the decoder and encoder ends must remain equal in all cases. Table III shows the results of adopting this framework for training and testing.

TABLE III. RESULTS ON ARABIC RECOGNITION

Structure	**Training accuracy**	**Training loss**	**Testing accuracy**	**Testing loss**
Autoencoder1	96.51	0.8491	92.88	0.2398
Autoencoder2	96.58	0.0921	95.41	0.1436
Autencoder3	95.87	0.1213	92.56	0.3568

According to Table III, embedding GRU memory cells within the autoencoder achieved superior performance compared to the other autoencoders. While the remaining two models also showed commendable results during the training phase, their performance on the test dataset was less. It is important to analyze the roles of each layer involved, as this will help to understand how the GRU components contribute to improved results compared to alternative methods.

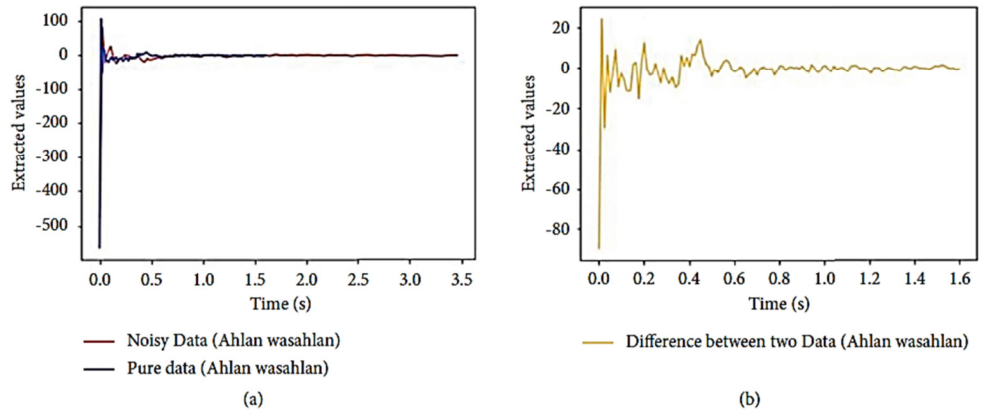

Fig. 5. Derived features from regular and chaotic data in (a), and the difference among them (b).

TABLE IV. PERFORMANCE COMPARISON OF THE PROPOSED WITH PREVIOUS METHODS ON ARABIC SPEECH RECOGNITION

Reference	**Model Name**	**Accuracy**	**WER**
[28]	Autoencoder (CNN)	93%	—
[7]	Fuzzy neural network	94.5%	—
[10]	Autoencoder (MLP)	65.72%	—
[13]	Autoencoder (LSTM)	71.58%	28.42%
[14]	CNN-LSTM	93%	13.52%
Proposed	**Autoencoder (RUG)**	**95.31%**	**4.69%**

Figure 6 shows the results of the validation and training sets for improved comprehension of the proposed approach. As can be observed, the proposed framework had some overfitting. The size of the proposed framework was reduced according to the best results. Figure 7 shows the confusion matrix for every word category.

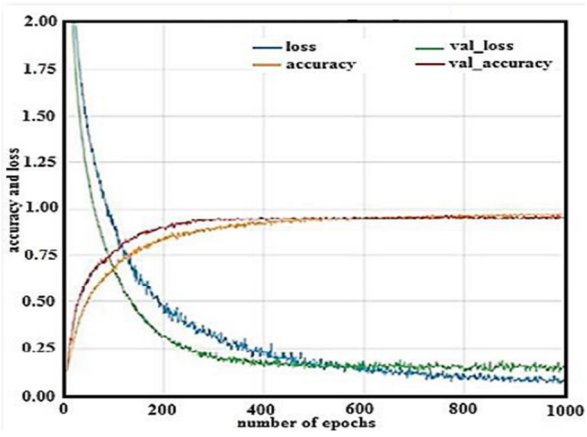

Fig. 6. The proposed algorithm's precision and loss.

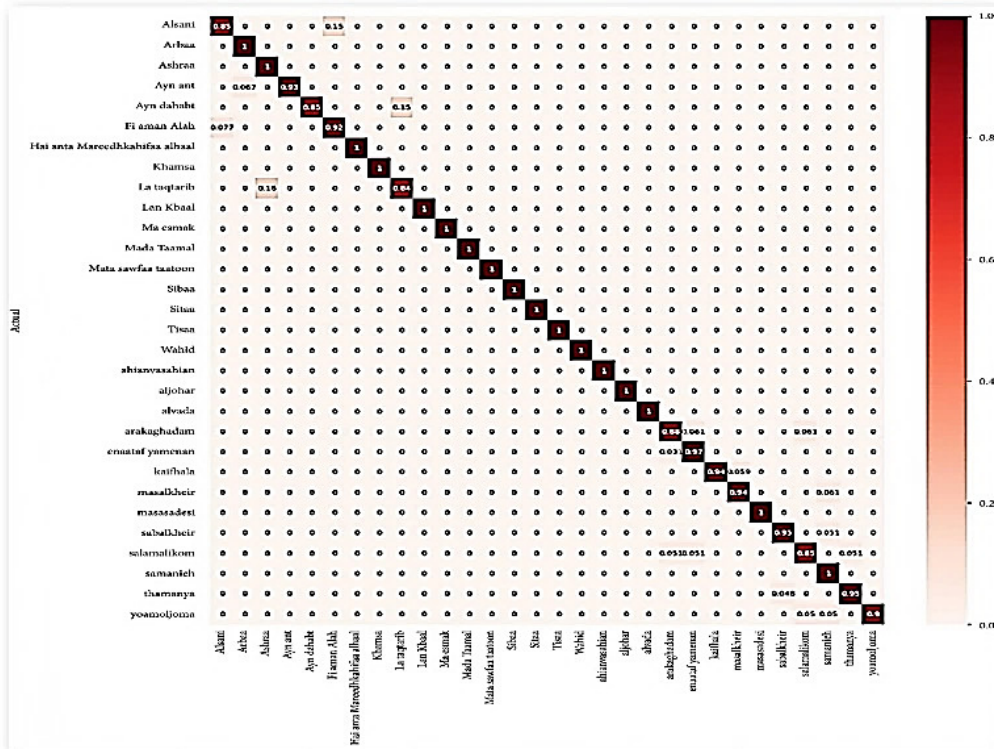

Fig. 7. Generalized confusion matrix of the proposed algorithm.

The confusion matrix in Figure 7 shows that nouns such as "Alwadaa" and digits such as "Arba" and "Ashara" received ideal ratings. Phrases that start with the word "A," such as "Al Saa Al Sadisaa" and "Ayn dhahbta," were mixed up. The WER for the proposed algorithm was 4.69%. Furthermore, according to the results in Figure 7, the proposed algorithm exhibited a precision of 99% in recognizing audible digits, outperforming the model proposed in [12]. In addition, the proposed framework was tested using noisy signals, confirming its ability to handle such data.

V. CONCLUSION

As Arabic is widely used around the world, being one of the six most widely used languages, many people with disabilities interact in Arabic. There are several technologies available to help people with speech difficulties communicate more effectively with others. The Servox Digital EL is one such device. The standard of language created by this device is inadequate for Arabic. As a result, this study presented an automatic encoding framework to help this device better recognize audible Arabic phrases. Initially, an appropriate dataset was collected from men, women, and children who speak Arabic as their primary language. A novel deep-learning algorithm was presented for Arabic speech recognition, using LSTM and GRU models. Then, the proposed framework was trained and tested on the collected dataset to recognize spoken words. The proposed approach has three main stages: denoising, feature extraction, and Arabic pronunciation. The experimental results showed that the proposed autoencoder that employs GRUs had excellent performance on both the encoder and decoder sides for Arabic word identification, with 95.31% accuracy and 4.69% WER. This study offers a practical method for Arabic pronunciation using the Servox Digital EL, which is used for patients with speech difficulties/disabilities to improve their ability to communicate.

ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education, Saudi Arabia for funding this research work through project number 445-9-440.

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, Jun. 2021, https://doi.org/10.1016/j.jksuci.2019.02.006.

[2] A. Shoufan and S. Al-Ameri, "Natural language processing for dialectical arabic: A survey," in 2nd Workshop on Arabic Natural Language Processing, ANLP 2015 - held at 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015 - Proceedings, Beijing, China, 2015, pp. 36–48.

[3] J. M. Vojtech *et al.*, "Surface Electromyography–Based Recognition, Synthesis, and Perception of Prosodic Subvocal Speech," *Journal of Speech, Language, and Hearing Research*, vol. 64, no. 6S, pp. 2134–2153, Jun. 2021, https://doi.org/10.1044/2021_JSLHR-20-00257.

[4] H. R. Khan, M. A. Hasan, M. Kazmi, N. Fayyaz, H. Khalid, and S. A. Qazi, "A Holistic Approach to Urdu Language Word Recognition using Deep Neural Networks," *Engineering, Technology & Applied Science Research*, vol. 11, no. 3, pp. 7140–7145, Jun. 2021, https://doi.org/10.48084/etasr.4143.

[5] Z. Ellaky, F. Benabbou, and S. Ouahabi, "Systematic Literature Review of Social Media Bots Detection Systems," *Journal of King Saud University - Computer and Information Sciences*, vol. 35, no. 5, May 2023, Art. no. 101551, https://doi.org/10.1016/j.jksuci.2023.04.004.

[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, Apr. 2021, https://doi.org/10.18280/ts.380212.

[7] L. Eljawad *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, https://doi.org/10.1109/ACCESS.2020.2991811.

[9] I. Shahin and A. B. Nassif, "Emirati-Accented Speaker Identification in Stressful Talking Conditions," in *2019 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, United Arab Emirates, Nov. 2019, pp. 1–6, https://doi.org/10.1109/ICECTA48151.2019.8959731.

[10] B. Dendani, H. Bahi, and T. Sari, "Speech Enhancement Based on Deep AutoEncoder for Remote Arabic Speech Recognition," in *Image and Signal Processing*, Marrakesh, Morocco, 2020, pp. 221–229, https://doi.org/10.1007/978-3-030-51935-3_24.

[11] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, Mar. 2020, Art. no. 132306, https://doi.org/10.1016/j.physd.2019.132306.

[12] 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, Jan. 2019, https://doi.org/10.1515/comp-2019-0004.

[13] 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, https://doi.org/10.1049/sil2.12057.

[14] 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 & Fuzzy Systems*, vol. 41, no. 6, pp. 6207–6219, Jan. 2021, https://doi.org/10.3233/JIFS-202841.

[15] Y. Tai, H. He, W. Zhang, and Y. Jia, "Automatic Generation of Review Content in Specific Domain of Social Network Based on RNN," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, Guangzhou, China, Jul. 2018, pp. 601–608, https://doi.org/10.1109/DSC.2018.00096.

[16] Y. C. Lien, E. A. M. Klumperink, B. Tenbroek, J. Strange, and B. Nauta, "Enhanced-Selectivity High-Linearity Low-Noise Mixer-First Receiver With Complex Pole Pair Due to Capacitive Positive Feedback," *IEEE Journal of Solid-State Circuits*, vol. 53, no. 5, pp. 1348–1360, Feb. 2018, https://doi.org/10.1109/JSSC.2018.2791490.

[17] 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, Jun. 2020, https://doi.org/10.1109/TIM.2019.2938905.

[18] F. M. Bayer, A. J. Kozakevicius, and R. J. Cintra, "An iterative wavelet threshold for signal denoising," *Signal Processing*, vol. 162, pp. 10–20, Sep. 2019, https://doi.org/10.1016/j.sigpro.2019.04.005.

[19] P. Ravisankar, "Underwater Acoustic Image Denoising Using Stationary Wavelet Transform and Various Shrinkage Functions," *ELCVIA. Electronic letters on computer vision and image analysis*, vol. 20, no. 2, pp. 38–50, 2021, https://doi.org/10.5565/rev/elcvia.1360.

[20] H. A. Elharati, M. Alshaari, and V. Z. Këpuska, "Arabic Speech Recognition System Based on MFCC and HMMs," *Journal of Computer and Communications*, vol. 8, no. 3, pp. 28–34, Mar. 2020, https://doi.org/10.4236/jcc.2020.83003.

[21] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Udupi, India, Sep. 2017, pp. 1643–1647, https://doi.org/10.1109/ICACCI.2017.8126078.

[22] W. Zhang *et al.*, "LSTM-Based Analysis of Industrial IoT Equipment," *IEEE Access*, vol. 6, pp. 23551–23560, 2018, https://doi.org/10.1109/ACCESS.2018.2825538.

[23] G. Shen, Q. Tan, H. Zhang, P. Zeng, and J. Xu, "Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions," *Procedia Computer Science*, vol. 131, pp. 895–903, Jan. 2018, https://doi.org/10.1016/j.procs.2018.04.298.

[24] S. Yang, X. Yu, and Y. Zhou, "LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example," in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, Shanghai, China, Jun. 2020, pp. 98–101, https://doi.org/10.1109/IWECAI50956.2020.00027.

[25] C. Wei, S. Kakade, and T. Ma, "The Implicit and Explicit Regularization Effects of Dropout," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 10181–10192.

[26] 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, Feb. 2019, https://doi.org/10.1016/j.neunet.2018.11.005.

[27] W. Helali, Z. Hajaiej, and A. Cherif, "Real Time Speech Recognition based on PWP Thresholding and MFCC using SVM," *Engineering, Technology & Applied Science Research*, vol. 10, no. 5, pp. 6204–6208, Oct. 2020, https://doi.org/10.48084/etasr.3759.

[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, Oct. 2021, https://doi.org/10.1080/02522667.2021.1967593.