

Machine learning for Arabic phonemes recognition using electrolarynx speech

Zinah Jaffar Mohammed Ameen, Abdulkareem Abdulrahman Kadhim

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

Article Info

Article history:

Received Jan 21, 2022

Revised Aug 3, 2022

Accepted Aug 28, 2022

Keywords:

Electrolarynx speech
Machine learning
Mel frequency cepstral coefficients
Performance evaluation

ABSTRACT

Automatic speech recognition system is one of the essential ways of interaction with machines. Interests in speech based intelligent systems have grown in the past few decades. Therefore, there is a need to develop more efficient methods for human speech recognition to ensure the reliability of communication between individuals and machines. This paper is concerned with Arabic phoneme recognition of electrolarynx device. Electrolarynx is a device used by cancer patients having vocal laryngeal cords removed. Speech recognition here is considered to find the preferred machine learning model that can classify phonemes produced by electrolarynx device. The phonemes recognition employs different machine learning schemes, including convolutional neural network, recurrent neural network, artificial neural network (ANN), random forest, extreme gradient boosting (XGBoost), and long short-term memory. Modern standard Arabic is utilized for testing and training phases of the recognition system. The dataset covers both an ordinary speech and electrolarynx device speech recorded by the same person. Mel frequency cepstral coefficients are considered as speech features. The results show that the ANN machine learning method outperformed other methods with an accuracy rate of 75%, a precision value of 77%, and a phoneme error rate (PER) of 21.85%.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Zinah Jaffar Mohammed Ameen
College of Information Engineering, Al-Nahrain University
Baghdad, Iraq
Email: Zinah.jaffar@coie-nahrain.edu.iq

1. INTRODUCTION

Speech recognition is a technique of converting spoken words into writing. Every spoken word is a composition of the most basic symbols of a particular language. Phonemes are the smallest units of each language. The method of recognizing simple units of phonemes is critical for developing speech recognition systems [1]. The language model and the acoustic model are two main components of each speech recognition system. A phoneme recognizer's accuracy is crucial to the acoustic model's accuracy [2]. In order to avoid large vocabulary word size, phoneme-based speech recognition is used, because words may be generated by combining the phonemes of the language. Due to the finite number of phonemes in each language, the process requires a substantial amount of training data compared to word-based models. Lowering complexity of the system allows the use of neural network (NN) often used in speech recognition systems. A neural network is a kind of machine learning technique which is based on the human nerve system and brain structure [3]. Machine-learning methods have recently encountered increasing attention due to their structure which is able to extract robust latent characteristics that allow various recognition algorithms to show generalization in a variety of applications.

Pipiras *et al.* [4] proposed a Lithuanian language phonemes recognition system. The system used deep learning techniques to recognize spoken words based on their phoneme sequences. Encoder/decoder model for feed-forward beside recurrent neural network (RNN) architecture were used and the performance was measured in isolated speech recognition tasks, yielding an overall recognition accuracy of 99.3% as well as long-phrase recognition tasks with an accuracy of 99.2%. The given rates were for pure Lithuanian phoneme sequences, without considering any added noise or disturbances.

A hidden Markovian model (HMM) with deep neural networks was presented in [5]. Using mel-frequency cepstral coefficients (MFCC) features, 96% recognition accuracy was achieved for subset samples from Lithuanian native speakers' database for normal dataset only. The convolutional neural network (CNN) provided 2D audio features mapped from chromatograms, spectrograms, and mel-scale cepstograms, producing a 99% f-score for a small dataset [6]. Attention and careful parameter selection to guarantee accurate supervised learning are necessary to create robust models with a small normal dataset.

CNN, as a feature extractor with support vector machine classifier, is considered by Glackin *et al.* [7]. The achieved frame error rate (FER) was 28%. It should be noted that the FER is a more precise metric because it shows the capability of the acoustic model. The Texas Instruments/Massachusetts Institute of Technology (TIMIT) speech corpus was utilized for labeled spectrogram patches in the CNN-based acoustic model for training. Experiments were applied using noisy TIMIT (NTIMIT) which contains noise from various telephone networks. The recognition rate results of NTIMIT were about 10% less than those for TIMIT.

HMM model with MFCC feature extraction was applied by Mouaz *et al.* [8] to a Moroccan dialect speech. The accuracy of HMM speech recognition system was about 90% for normal Moroccan speech. Deep belief network HMM (DBN-HMM) for phoneme recognition was implemented in [9]. The DBN-HMM experiment on TIMIT exploited information embedded in the underlying structure more effectively, resulting in a reduced phoneme error rate (PER) ratio of 22.8%. Pradeep and Rao [10] used "Kannada" an ancient Indian language with speech corpus collection of continuous utterances recorded in three modes: read mode, lecture mode, and conversation mode. These are compared according to recognition baselines for HMM with Gaussian mixture model (GMM), HMM artificial neural network (HMM-ANN), and HMM with deep neural network (DNN). According to PER results, the HMM-DNN baseline outperformed the HMM-GMM and HMM-ANN baselines by 7% and 8%, respectively.

The effect of Arabic phonemes on the performance of speaker recognition systems was investigated by Alsulaiman *et al.* [11]. The recognition rates for Arabic vowels were all above 80%, whereas the recognition rates for consonants ranged from 14% to 94%, with the latter being achieved by a pharyngeal consonant followed by two nasal phonemes, which achieved recognition rates above 80%. Four more consonants had recognition rates ranging from 70% to 80%. Alsharhan and Ramsay [12] introduced data properties that may affect the performance of Arabic automatic recognition system (ASRS). The experimental results demonstrated that developing gender and dialect-specific models resulted in a significant decrease in word error rate (WER). Using 25-dimension MFCCs, eliminating poor quality recordings, and employing a grapheme-based dictionary resulted in a total reduction in WER between 3.24% to 5.35%. In [13], an ASRS for the Amazigh language was developed. This system was built on the top of an open-source Carnegie Mellon University (CMU) Sphinx-4. The essential components of the ASRS were feature extraction, acoustic modeling, pronunciation, and HMM modeling. The size of the dataset was 2970 words, and the accuracy obtained was 88%. Applying modular HMM-DNN ASRS, and human speech recognition on the Arabic language and its dialects is introduced in [14]. Performance evaluation based on multi genre broadcast (MGB), which is an evaluation of speech recognition with 17 classes' Arabic dialect identification using YouTube recordings MGB2, MGB3, and MGB5 challenges resulted in 12.5%, 27.5%, 33.8% WER, respectively.

This paper considers machine learning models for Arabic phonemes recognition. Arabic is a complex morphological and consonantal language with difficulty to determine the location of the vowels, which can convey different meanings and complex morphological structure. The applied data sets are prepared using 8 individuals for the 27 Arabic phonemes recorded using normal speech and that produced by an electrolarynx (EL) device. The applied models are ANN, CNN, RNN, random forest (RF), extreme gradient boosting (XGBoost), long short-term memory (LSTM), and hybrid arrangements for the classification task. Unlike the previously mentioned studies, where only a pure speech dataset is considered, the work here considers both pure (normal) speech and that produced by EL device which is usually contaminated with noise. The main contribution here is the selection of the preferred machine learning models for Arabic phonemes recognition for the mentioned datasets. The remaining parts of this paper are organized as follows: section 2 provides a presentation of related machine learning algorithms used in the tests, and section 3 presents the research methods. Section 4 discusses the results of the proposed Arabic phonemes model. Finally, the concluding remarks are given in section 5.

2. RELATED MACHINE LEARNING ALGORITHMS

The term machine learning (ML) refers to a collection of methodologies or algorithms that enable computers to automate data-driven model programming and develop models employing systematically detecting patterns in statistically significant data [15]. There are three categories of ML: supervised, unsupervised, and reinforcement. The basic concepts of the applied supervised ML algorithms used in this paper for training and implementing an Arabic phonemes recognition system are described briefly in the next subsections.

2.1. Artificial neural network

The artificial neural network (ANN) is a non-parametric prediction tool for pattern classification applications, including speech recognition [16]. ANNs extract complex patterns from data and detect complicated trends for people or other computational approaches [17]. As a result, ANN is capable of modeling both complicated and multi-complex problems [18]. Initially, the number of layers and the activation functions are determined according to the complexity of the problem to be solved [19]. The multi-layer perceptron (MLP) is one of the most widely utilized ANN architecture for pattern classification. It has been employed in a variety of voice synthesis and recognition schemes [17]. Figure 1 shows the structure of an ANN, which consists of two layers: the hidden layer and the output layer. The output layer value is expressed via (1) [17],

$$Op_K = f^*(\sum_{j=0}^M W2_{kj} \times f(\sum_{i=0}^N W1_{ji} \times Ip_i)) \quad (1)$$

where Op_K is ANN output, M is the number of output elements, Ip_i is the input data, N is the number of input attributes, $W1_{kj}$ is the first layer weight, $W2_{ji}$ is the second layer weight, and f and f^* represent the applied activation function for each layer.

One of the weight adjustment methods applied in ANN is back propagation which is necessary to reduce error using gradient descent algorithm [17] by adjusting the weights according to the partial derivative of error concerning each weight [3]. Thus, the actual output becomes closer to the target output due to error minimization for each output neuron and the whole network [15]. The main families of ANN are:

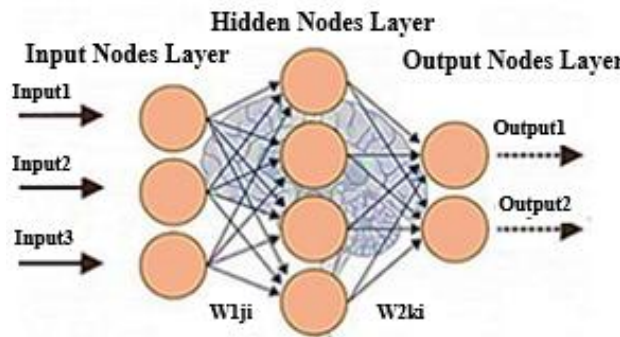


Figure 1. Artificial neural network structure [17]

2.1.1. Convolutional neural network

Convolutional neural network (CNN) is a powerful family of neural networks designed precisely for image processing applications containing convolutional layers. CNNs are regularized fully connected networks in which each neuron in one layer is connected to all neurons in the following layer [20]. Convolution, pooling, and fully linked layers are the three primary layers of CNN. The first layer is the convolution layer utilized to extract the features from an input. The pooling layer is the second layer used to reduce the number of parameters. Different pooling techniques are available, usually based on the requirement. The most commonly used one is max pooling, which only considers the highest concentrated element of the obtained feature map [21]. Sub-sampling is frequently achieved using max/mean pooling or local averaging filters [3]. After flattening the generated feature map, the 1D array is sent into a fully connected network [21]. The final layer of CNN handles the actual classifications. Multiple series of sub-sampling and weight-sharing convolution layers can be used to build deep CNN [3]. The basic CNN architecture with its main layers is displayed in Figure 2.

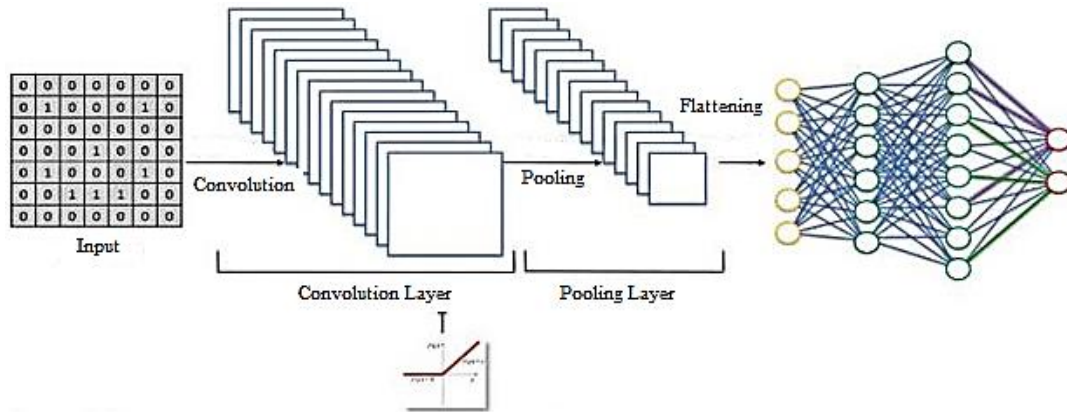


Figure 2. Convolutional neural network architecture [21]

2.1.2. Recurrent neural network

Recurrent neural network (RNN) is an ANN with cyclic connections that are more powerful tool for modeling sequence of data compared to an ordinary ANN [17]. In a regular MLP, each layer has its weights and biases. This ensures that the neuron remembers the existing state, and based on this state, the following output is generated [21]. RNN has a memory which determines future predictions. As a result, anticipating any word in a sentence necessitates knowledge of the previously processed words [22]. The RNN algorithm is depicted in Figure 3. As shown in (2) calculates the hidden state S_t which stores the data of the previous steps:

$$S_t = f(U \times x_t + W \times x_{t-1}) \tag{2}$$

where x_{t-1} is the previous input, x_t is the present input at time t , U and W are the hidden layer weights, and V is the output layer weight. Based on the memory at time t , the output step is calculated as in (3) [23].

$$O_t = softmax(V \times S_t) \tag{3}$$

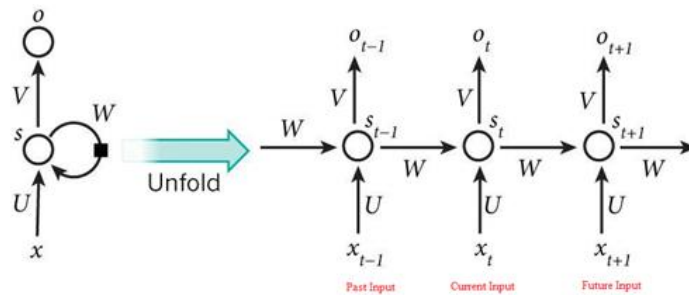


Figure 3. RNN algorithm

LSTM network is a type of RNN that uses a combination of specialized and standard units. LSTM can remember prior states and be trained to do tasks requiring memory or awareness of current states. LSTM partially overcomes the significant weakness of RNN, especially the problem of vanishing gradients [23]. As depicted in Figure 4, a memory cell is a component of LSTM units which is able to store information for a long time [3]. A memory state can add or remove any information as needed using gates. Input, output, and forget gates are the three types of gates considered in RNN. These gates have the ability to protect or control memory state. The LSTM network is made of a sigmoid function, which is a type of activation function. This squashing function restricts the output to a range between zero and one, making these functions useful in predicting probabilities [23]. In Figure 4, the value of the time step is denoted by the subscript t and the memory cell, input, and output values are represented by c , x , h . Moreover, $\sigma()$ represents the sigmoid function while the hyperbolic tangent function is defined by $tanh$. The gates' computations are given in (4) to (8) [3].

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (4)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (5)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (6)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (7)$$

$$h_t = o_t \otimes \tanh(c_t) \quad (8)$$

where W , U , and b stand for input weights, recurrent output weights, and bias, and the forget, input, and output gates are denoted by f , i , and o , respectively. The ANN algorithm described above is closely connected to deep learning [17]. A deep neural network is made of multiple layers of nodes. To address problems in a variety of areas or used cases, many designs have been devised. CNN is extensively employed in image recognition and computer vision, while RNN is commonly used in forecasting and time series problems [3].

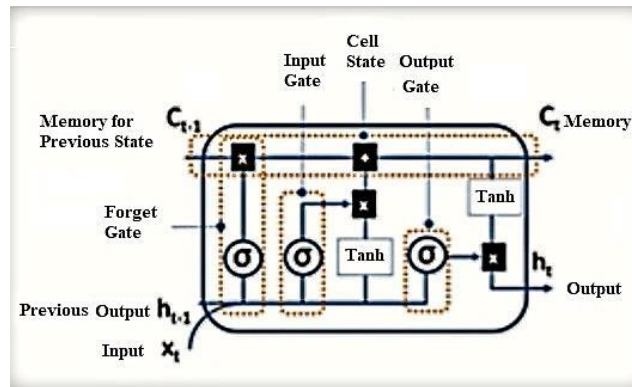


Figure 4. LSTM structure

2.2. Random forest classifier

Decision tree algorithms are the most widely used ML approaches. They are employed to represent a wide range of data classification problems [17]. Random forest classification (RFC) is a supervised classification technique in ML that uses decision trees. A decision tree method creates a tree-like model of the dataset, with each node being further divided. An RFC is a collection of de-correlated, unbiased, and unrelated decision trees and hence is called a random forest. It is based on integrating two basic principles: each decision tree is trained using only a portion of the training samples and must make its prediction using only a random subset of the whole features [22]. The two most significant concepts that are widely used for this task are Gini and entropy values. Gini is utilized to calculate the impurity, while entropy is used to calculate the node's information gain. The formula for calculating the impurity from the Gini value is defined as (9) [24]:

$$G = 1 - \sum_{t=0}^1 P t^2 \quad (9)$$

where $P t$ represents the relative frequency of the class in the dataset and c represents the number of classes. To split the node, the least Gini impurity is chosen. At the leaf nodes, a perfect split would result in a Gini score of 0. Similarly, the goal of entropy is to split at the node that gives the maximum information gain [24]. The RFC method also has advantages over other methods due to its suitability for classifying high-dimensional data [25].

2.3. Extreme gradient boosting classifier

Extreme gradient boosting (XGBoost) is a sequential tree-building algorithm implemented by parallelization [26]. It runs faster than any other model, and it is famous for its scalability in all scenarios. There are many different boosting algorithms like parallel boosting, regression tree boosting, and stochastic

gradient boosting, but XGB is one of the best boosting algorithms [23]. In XGB, interchangeable nature of loops determines the foundation of building algorithm. The exterior loop keeps track of the tree count, while the inside loop calculates the features. Loops can be swapped out, and this action improves run time performance [27]. Parallel threads are used to scan, initialize, and sort all the instances, globally. The algorithm performance is improved by switching loops. The overheads of parallelization in the computation are offset. The negative loss criterion at the split point determines whether or not the tree at the node will split [26].

3. RESEARCH METHODS

3.1. The considered Arabic phoneme model

Arabic is one of the oldest languages in the world, and it is the fourth most widely spoken language worldwide [28]. Several researchers have addressed the development of an Arabic speech recognition system. The Arabic phoneme set used in the present work is the small modern standard Arabic (MSA) speech corpus shown in Table 1. In the table, each phoneme is listed with its corresponding English symbol. This corpus is considered as the primary reference for Arabic speech recognition systems [29]. In comparison to English, Arabic has fewer vowels, and it possesses only three long and short vowels. The articulation refers to the influence of emphatic phonemes on adjacent phonemes, particularly vowels. In nearby segments, emphatic consonants cause a significant backing (i.e., sliding the tongue back during articulation) gesture, which happens mainly for adjacent vowels. This effect can be felt throughout full syllables as well as across syllable boundaries [28]. Using V symbol for short or long vowels and C for consonants, CV, CVC, and CVCC are the syllable types allowed in Arabic. In the third type of Arabic syllable, the indicated vowel can only be short. All Arabic syllables must have at least one vowel, and all Arabic utterances must begin with a consonant [29].

Table 1. Modern standard Arabic phonemes list

Phoneme Type	Voiced	Emphatication	Bilabial	Labi-dental	Inter-dental	Dental-alveolar	Lab-velar	Palatal	Velar	Uvular	Pharyngeal	Glottal
Stop	Yes	Yes				ض/Dh/						
	No	No	ب/B/			د/D/						
Fricative	Yes	Yes				ط/TA/				ك/K/	ق/Q/	ء/AH/
	No	No				ت/T/						
Nasal	Yes	Yes			ظ/DH/							
	No	No			ذ/Tha/	ز/Z/			غ/G/		ع/C/	
Lateral/Trill	Yes	Yes				ص/SA/						
	No	No		ف/F/	ث/TH/	س/S/		ش/SH/	خ/KH/	ح/H/	ه/HA/	
Semi-Vowels	Yes	No	م/M/						ن/N/			
	No	No							ر/R/			
Affricate	Yes	No							ل/L/			
	Yes	No						و/W/	ي/Y/			
												ج/J/

3.2. The proposed Arabic phoneme recognition model

In this section the proposed Arabic phonemes model is discussed in detail. This model is comprised of four sections. In the first part the dataset is prepared, then MFCC features are extracted. Followed by training and testing phases for the related ML algorithms which would be evaluated with the proposed performance measure. The proposed Arabic phoneme's structure is shown in Figure 5. This structure involves the following processing steps.

3.2.1. The preparation of data set

The recognition system here used phoneme classes of 27 Arabic phoneme categories as shown in Table 1. The data set was gathered for two cases: i) the normal speech and ii) EL produced speech. The dataset consists of 63 instances for each phoneme class. The tests involved eight individuals (three males and five females). The recorded speech is sampled at 48 kHz sampling frequency covering one-second utterance. To have an idea about the quality of the speech produced by EL, Figures 6 and 7 show the time waveform of normal phoneme example /ع/ and its EL produced version.

The processed dataset used in NN is divided into three subsets. The first is the training set that is used to update weights and biases according to output values of NN and the target output. The second set is utilized for validation by measuring the NN generalization to stop training before overfitting occurs. The last set is the testing one which is considered as an independent measure of NN performance using random indices for NN future prediction. In general, the training set makes up approximately 70% of the entire dataset, while the validation and testing sets are about 30%. The experiments are conducted on the prepared

dataset of 1,709 samples (given by 27 phoneme classes multiply by 63 instances plus 8 added samples for dataset structuring purposes) randomly divided into a 70% training set (1,196 samples) and 30% testing set (513 samples).

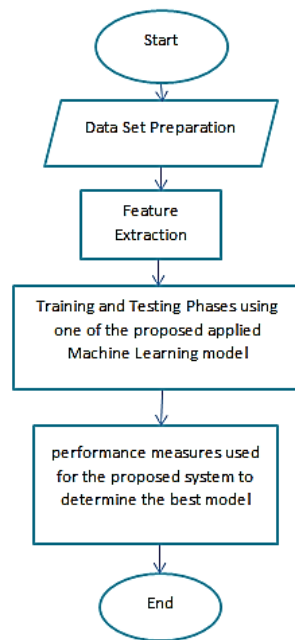


Figure 5. Proposed Arabic phonemes model

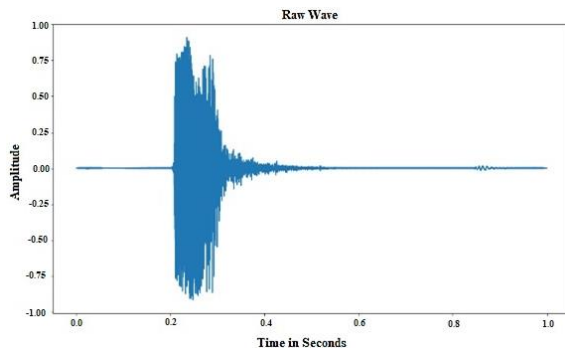


Figure 6. Normal /ع/ phoneme time waveform

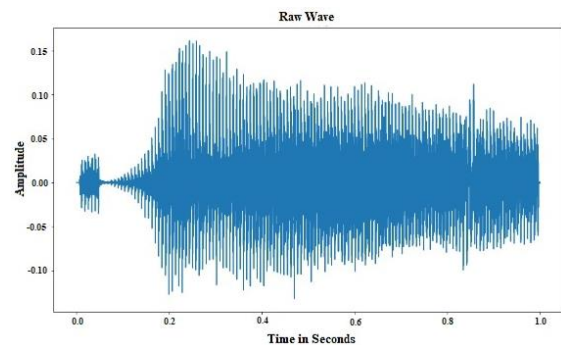


Figure 7. EL produced /ع/ phoneme time waveform

3.2.2. Extraction of features

Because the input data provided to the system is too large to be processed and highly redundant, feature extraction is considered as a dimensionality reduction procedure. Several methods have been designed to extract the features for speech recognition [30]. Experiments revealed that MFCC is a commonly utilized technique, especially for noisy datasets like the collected speech dataset produced by EL device. In this work, for each phoneme, a vector of 40 MFCC features is created, where this value has been approved to get better results in comparison to using 10, 20, 80, 120, and 200 MFCC features. Figure 8 [31] shows the main steps for calculating MFCC coefficients. These are found by taking the fast Fourier transform for a windowed portion of the speech signal. The power of the spectrum obtained is then mapped into Mel scale using overlapping triangular windows. This is followed by taking the logarithm of the power at each of the Mel frequencies. Then, discrete cosine transform is applied to the sequence of the Mel log powers. The MFCCs are the amplitudes of the resulting spectrum. A sample of the extracted MFCC features for normal and EL speech (27×40) is shown in Tables 2 and 3. The extracted MFCC features are then classified into 27 classes that correspond to different Arabic phonemes.

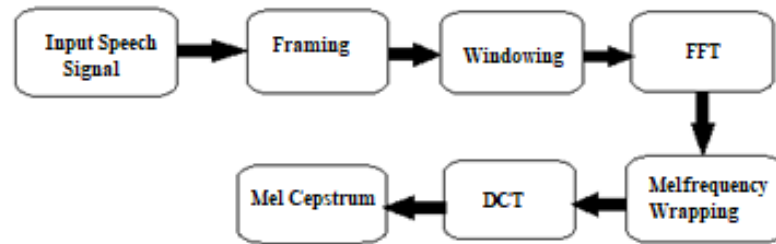


Figure 8. Features extraction using MFCC [31]

Table 2. Sample of extracted MFCC features for normal speech dataset

26	-12.93514	-12.78101	12.07424	-7.64229	-12.93514
27	7.55225	4.16052	-1.61571	2.12973	7.55225
28	1.16899	-2.67745	-2.72830	-1.93349	1.16899
29	-2.05475	1.94528	-4.33331	0.08896	-2.05475
30	-8.37376	-6.05577	-7.80390	-9.49206	-8.37376
31	-1.49524	-1.11541	0.88058	-2.48971	-1.49524
32	3.32171	-1.50352	3.74667	1.33721	3.32171
33	-0.23734	4.24566	2.51809	5.39290	-0.23734
34	-3.77050	-5.91520	0.18575	0.07774	-3.77050
35	0.68559	-0.25562	0.93503	-1.93664	0.68559
36	-5.76186	-7.02887	-2.88118	-1.22070	-5.76186
37	-0.40331	1.07401	-2.93137	-0.67019	-0.40331
38	-1.35721	-1.81024	-4.48035	-6.65883	-1.18336
39	4.83753	3.82572	2.09323	3.92759	4.83753

Table 3. Sample of extracted MFCC features for EL based data set

26	-1.49215	-0.87398	-3.29879	3.49641	-3.67335
27	-0.50662	0.51021	-1.23373	7.81420	-3.96603
28	-3.43310	-1.29136	-2.49257	2.68149	-4.01129
29	0.29979	0.71781	-0.36492	0.56282	-1.29038
30	0.91328	4.22358	1.86276	4.11376	-2.14616
31	3.44832	4.78779	1.82512	2.65514	2.65741
32	2.14013	0.86060	-0.17076	0.88931	4.81634
33	1.38956	0.95144	3.22962	1.28262	6.50467
34	-1.66049	-1.62705	0.38876	0.09730	7.19661
35	-2.83102	-2.96715	-2.74466	-4.66885	1.15356
36	-3.75353	-2.21525	-1.15170	-1.76901	-3.82262
37	-2.21951	-0.36873	-0.25384	0.69761	-4.12909
38	-1.35721	-0.43995	-0.90816	-0.34871	-2.74773
39	-0.17302	1.39558	2.53936	1.34547	0.21172

3.2.3. Data set training and testing

Like any other pattern recognition systems, the process of performing speaker recognition consists of two phases namely: training and testing. In the training phase, a database of the extracted features from the whole dataset is created. These features are used to train the proposed machine learning algorithms applied in this paper. The applied machine learning system models are ANN, CNN, RNN, RF, XGB, and LSTM either independently or in a hybrid model for the classification task. The applied hybrid models are CNN-LSTM, CNN-RF, and CNN-XGB, whereas in the testing phase, features are extracted from every input signal and a feature matching process is performed to decide whether these features belong to a previously created database or not. The behavior of each machine learning method is evaluated based on specific parameters for its performance measurements. The applied parameters for each model were varied many times to ensure a fair comparison and to get the best results of the given method. The ANN model is trained with 40 neurons in the input layer, 256 neurons in the hidden layer, and 27 neurons in the output layer. Exponential linear unit (ELU) and SoftMax activation functions [32] with a learning rate of 0.001 are used with 1,500 epochs. The batch size value is 32, while the dropout value is 0.5 for the input and 0.75 for the output layer. The regularizer value of 0.2 to face the overfitting issue is used.

For the XGB classifier, the applied parameters are as follows; the number of estimators is 102, the maximum depth is 5, and the learning rate is 0.7. For RFC, the involved parameters are: 73 estimators, maximum depth of 90, minimum samples leaf is 1, and the minimum samples split is 2.

For the CNN model, the system is trained using one convolution layer, one maximum pooling layer, and one flatten layer. The input layer is 40 neurons, 2 hidden layers of 256 neurons for each layer, and an output layer of 27 neurons. The applied activation functions are rectified linear unit (ReLU), ELU, and SoftMax [33], while the batch size is 64. The dropout value is 0.6, and the regularizer is 0.02.

In the LSTM model, the system is trained to apply one input layer of 40 neurons, one hidden layer of 256 neurons with one flatten layer and the output layer of 27 neurons. The dropout values are 0.2, 0.25, and 0.6. The applied activation functions are ELU and SoftMax [32].

For the hybrid CNN-XGB model, the system is trained using 2 convolution layers, 2 maximum pooling layers, and 1 flatten layer. The input layer has 40 neurons, while the number of estimators is 40, the maximum depth is 40, and the learning rate is 0.8. The CNN-RFC model is trained using two convolutional layers, two maximum pooling layers, and one flatten layer. The input layer has 40 neurons, while the parameters of RFC classifier are: the number of estimators is 31, maximum depth is 20, minimum samples leaf is 1, and the minimum samples split is 2.

3.2.4. Performance measures

The performance of the Arabic phoneme recognition system is evaluated for each machine learning method considered in the work. After feature extraction, the model is implemented to produce output in the form of a class. The accuracy rate is considered as the main classification performance measure here. Further, receiver operating characteristics (ROC) curve, phoneme error rate (PER), confusion matrix, and precision are also considered as performance measures.

The accuracy or recognition rate represents the ratio of the correctly predicted samples relative to the total number of instances present in the data set. The following relation determines the accuracy rate (A_c) [25]:

$$A_c = \frac{TP+TN}{TP+TN+FP+FN} \times 100\% \quad (10)$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

ROC metric is used to assess a classifier's output quality. The ROC curve gives an indication to the fraction of correct predictions for the positive class (number of false positives) on the x-axis versus the errors for the negative class (number of true positives) on the y-axis [34]. Each classifier uses a single point on the ROC to represent the false positive (FP) and true positive (TP) pairs. The upper left corner point in the ROC curve represents a perfect or ideal classification [25].

ROC probability curve represents an insensitive to class distribution, if the proportion of positive to negative instances changes, the ROC curve will not change. There is a simple relationship to determine A_c based on ROC:

$$A_c = TRP \times POS + NEG - NEG \times FPR \quad (11)$$

where TRP is the true positive rate, FPR is the false positive rate taken from ROC, POS is fraction of correct predictions for the positive class, and NEG is fraction of negative classes.

The confusion matrix gives valuable information in comparing actual and predicted classes to evaluate the classifier's performance. Four categories covered by this matrix as shown in Figure 9. TPs and TNs represent the predicted and actual phoneme classes, while FPs and FNs exist when the prediction does not match with the actual phoneme classes. The predicted class in FN is negative, but the actual class is positive. Similarly, the class in FP is positive, but the actual class is negative [25]. Two main values used for performance evaluation as depicted in Table 4, could be calculated from the confusion matrix in (10) and precision as in (12).

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

Table 4. Simple confusion matrix

		Predicted Labels			Precision Accuracy
		False True	True Negative (TN) False Negative (FN)	False Positive (FP) True Positive (TP)	
Actual Labels	False				
	True				

In the field of speech recognition and processing, the error rate is a widely used measure. The error rate here is PER using a phoneme-level recognition system, and it is calculated using (13) and (14) [33],

$$PER = \left(\frac{ERR}{N} \times 100\% \right) \tag{13}$$

$$ERR = 1 - A_c \tag{14}$$

where N represents the total number of reference labels or classes.

4. RESULTS AND DISCUSSION

This section summarizes the tests that were carried out to find the suitable ML algorithm for phonemes classifying tasks among the tested methods and discuss the results. The results confirm that ANN model is the most appropriate choice for Arabic phonemes recognition. Furthermore, the applied models were evaluated depending on noisy signals that are totally corrupted that specify the capability of proposed model for dealing with such signals.

4.1. Recognition results

Figure 9 shows the accuracy results for training and testing phases obtained by the proposed methods for Arabic phonemes recognition depending on normal and EL based produced speech. As indicated in the figure, the ANN model outperforms other models with a 73.23% testing accuracy rate. ANN is a comparatively lightweight way of solving data classification problems, especially for limited datasets conditions.

The training learning curve is calculated from the training dataset. It gives an idea of successful learning of the method, while the validation learning curve calculated from a hold-out validation dataset, gives an idea on how well the model in general. It is also common to create learning curves for optimization tasks according to cross-entropy loss and the model performance is evaluated using classification accuracy. Figure 10 shows the ANN model learning curves as the training accuracy grows. The validation accuracy increases at first, then begins to fall after a given number of epochs (increase or decrease) due to the overfitting effect. This has been dealt with using regularization during the training and adding dropout layers. The loss measures the model error, decreasing as the training progresses, indicating better model performance.

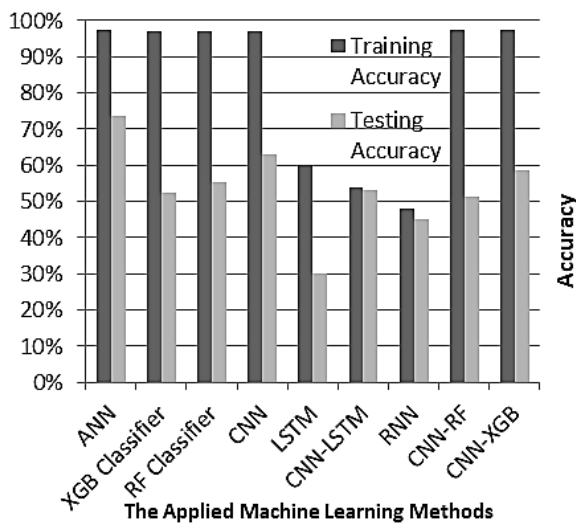


Figure 9. Accuracy results of different methods

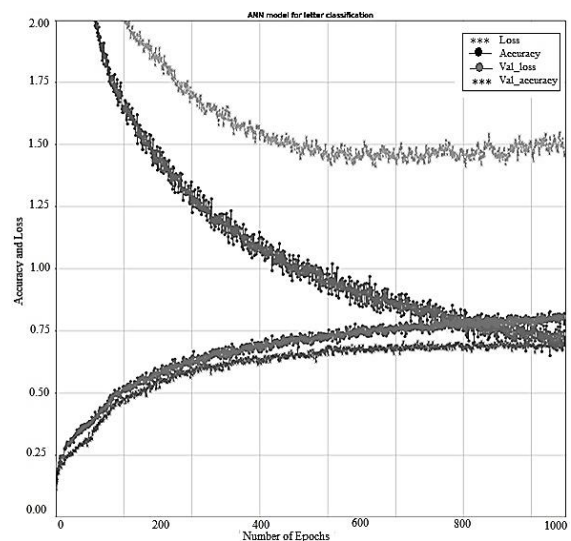


Figure 10. Learning curves for ANN model

The higher the ROC value is, the better the model will be. Figure 11 shows the ROC results for each class for the ANN network. As shown in Figure 12, the confusion matrix for the ANN network, TPs are represented on the matrix's diagonal. It is approved that most of the phonemes were classified correctly, and the letter /ش/ of class 20 got the higher value among the other letters. The value of precision for the ANN model value is 77%, while the value of PER is 21.85, calculated as calculated depending on the confusion matrix Figure 12 expressed as:

$$(1 - 0.81) + (1 - 0.83) + (1 - 0.58) + (1 - 0.76) + \dots (1 - 0.82) = 5.88$$

$$PER = 5.88/27 = 21.8$$

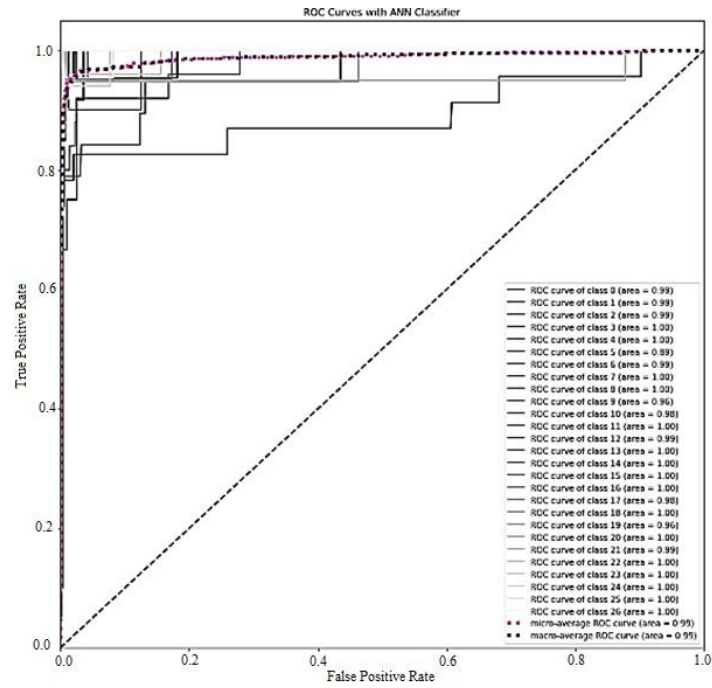


Figure 11. ROC curve

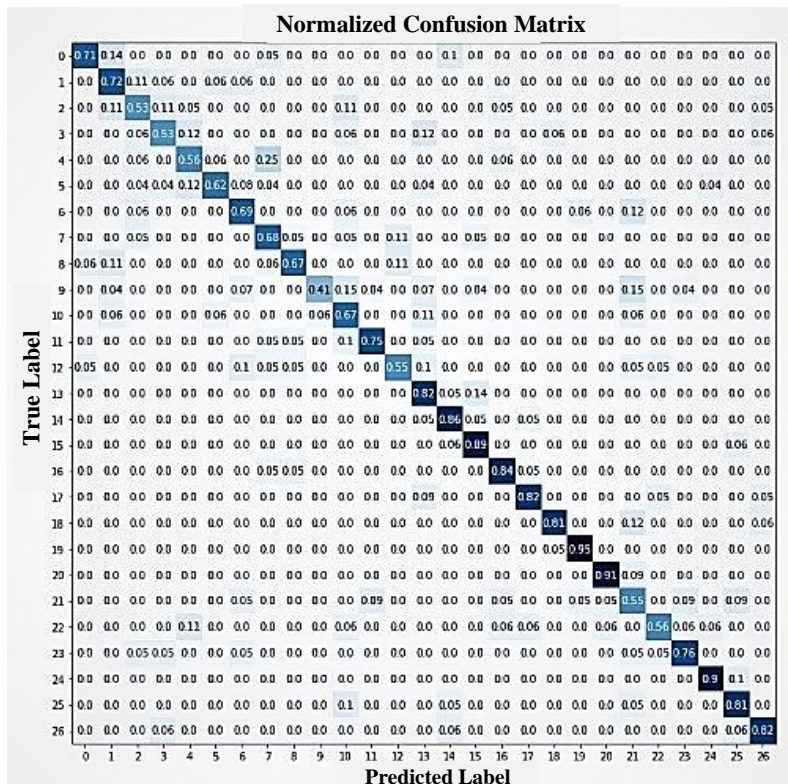


Figure 12. The ANN model confusion matrix

5. CONCLUSION

This work aims to adopt the most promising machine learning model to detect the produced speech by the electrolarynx device. Automatic speech recognition has recently emerged as a significant research subject in human-computer interaction. The work in this paper investigates the most useful machine learning techniques for Arabic phoneme recognition. The applied machine learning system models are ANN, CNN, RNN, RF, XGB, and LSTM either independently or in a hybrid model for the classification task. The applied hybrid models are CNN-LSTM, CNN-RF, and CNN-XGB. According to Table 1, the experiments have been conducted on the prepared 63 instances for the 27 classes of Arabic phoneme categories. The dataset has been gathered in two stages: the normal speech and EL-produced speech. The performance results show that ANN outperforms the other applied models with the value of 97.49% training accuracy rate and 73.23% testing accuracy rate with an achieved PER of 21.85, and the precision value is 77%. Due to their robust pattern recognition and classification skills, ANNs have delivered remarkable results. In terms of performance metrics, the behaviors and performances of ANN training algorithms have been compared.





REFERENCES

- [1] S. Bhatt, A. Dev, and A. Jain, "Confusion analysis in phoneme based speech recognition in Hindi," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 4213–4238, Oct. 2020, doi: 10.1007/s12652-020-01703-x.
- [2] K. M. O. Nahar, M. Abu Shquier, W. G. Al-Khatib, H. Al-Muhtaseb, and M. Elshafei, "Arabic phonemes recognition using hybrid LVQ/HMM model for continuous speech recognition," *International Journal of Speech Technology*, vol. 19, no. 3, pp. 495–508, Sep. 2016, doi: 10.1007/s10772-016-9337-5.
- [3] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019, doi: 10.1109/ACCESS.2019.2912200.
- [4] L. Pipiras, R. Maskeliūnas, and R. Damaševičius, "Lithuanian speech recognition using purely phonetic deep learning," *Computers*, vol. 8, no. 4, Oct. 2019, doi: 10.3390/computers8040076.
- [5] L. Dovydaitis and V. Rudžionis, "Identifying Lithuanian native speakers using voice recognition," in *Business Information Systems Workshops*, 2017, pp. 79–84.
- [6] G. Korvel, P. Treigys, G. Tamulevičius, J. Bernatavičienė, and B. Kostek, "Analysis of 2D feature spaces for deep learning-based speech recognition," *Journal of the Audio Engineering Society*, vol. 66, no. 12, pp. 1072–1081, Dec. 2018, doi: 10.17743/jaes.2018.0066.
- [7] C. Glackin, J. Wall, G. Chollet, N. Dugan, and N. Cannings, "Convolutional neural networks for phoneme recognition," in *Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods*, 2018, pp. 190–195, doi: 10.5220/0006653001900195.
- [8] B. Mouaz, B. H. Abderrahim, and E. Abdelmajid, "Speech recognition of Moroccan dialect using hidden Markov models," *Procedia Computer Science*, vol. 151, no. 1, pp. 985–991, 2019, doi: 10.1016/j.procs.2019.04.138.
- [9] Y. Xie, C.-R. Zou, R.-Y. Liang, and H.-W. Tao, "Phoneme recognition based on deep belief network," in *2016 International Conference on Information System and Artificial Intelligence (ISAI)*, Jun. 2016, pp. 352–355, doi: 10.1109/ISAI.2016.0081.
- [10] R. Pradeep and K. S. Rao, "Deep neural networks for Kannada phoneme recognition," in *2016 Ninth International Conference on Contemporary Computing (IC3)*, Aug. 2016, pp. 1–6, doi: 10.1109/IC3.2016.7880202.
- [11] M. Alsulaiman, A. Mahmood, and G. Muhammad, "Speaker recognition based on Arabic phonemes," *Speech Communication*, vol. 86, pp. 42–51, Feb. 2017, doi: 10.1016/j.specom.2016.11.004.
- [12] E. Alsharhan and A. Ramsay, "Investigating the effects of gender, dialect, and training size on the performance of Arabic speech recognition," *Language Resources and Evaluation*, vol. 54, no. 4, pp. 975–998, Dec. 2020, doi: 10.1007/s10579-020-09505-5.
- [13] M. Telmem and Y. Ghanou, "Amazigh speech recognition system based on CMUSphinx," in *Innovations in Smart Cities and Applications*, 2018, pp. 397–410.
- [14] A. Hussein, S. Watanabe, and A. Ali, "Arabic speech recognition by end-to-end, modular systems and human," *Computer Speech & Language*, vol. 71, Jan. 2022, doi: 10.1016/j.csl.2021.101272.
- [15] D. Povey *et al.*, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech 2016*, Sep. 2016, pp. 2751–2755, doi: 10.21437/Interspeech.2016-595.
- [16] M. Sarma and K. K. Sarma, "An ANN based approach to recognize initial phonemes of spoken words of Assamese language," *Applied Soft Computing*, vol. 13, no. 5, pp. 2281–2291, May 2013, doi: 10.1016/j.asoc.2013.01.004.
- [17] J. D. Pineda-Jaramillo, "A review of machine learning (ML) algorithms used for modeling travel mode choice," *DYNA*, vol. 86, no. 211, pp. 32–41, Oct. 2019, doi: 10.15446/dyna.v86n211.79743.
- [18] O. I. Abiodun *et al.*, "Comprehensive review of artificial neural network applications to pattern recognition," *IEEE Access*, vol. 7, pp. 158820–158846, 2019, doi: 10.1109/ACCESS.2019.2945545.
- [19] A. Farizawani, M. Puteh, Y. Marina, and A. Rivaie, "A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches," *Journal of Physics: Conference Series*, vol. 1529, no. 2, Apr. 2020, doi: 10.1088/1742-6596/1529/2/022040.
- [20] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv:2106.11342*, Jun. 2021.
- [21] S. K. Shetty and A. Siddiqua, "Deep learning algorithms and applications in computer vision," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 7, pp. 195–201, Jul. 2019, doi: 10.26438/ijcse/v7i7.195201.
- [22] M. S. Rao, G. B. Lakshmi, P. Gowri, and K. B. Chowdary, "Random forest based automatic speaker recognition system," *The International journal of analytical and experimental modal analysis*, vol. XII, no. IV, pp. 526–535, 2020.
- [23] A. Amberkar, P. Awasarmol, G. Deshmukh, and P. Dave, "Speech recognition using recurrent neural networks," in *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, Mar. 2018, pp. 1–4, doi: 10.1109/ICCTCT.2018.8551185.
- [24] K. Nugroho, "Javanese gender speech recognition based on machine learning using random forest and neural network," *SISFORMA*, vol. 6, no. 2, Feb. 2020, doi: 10.24167/sisforma.v6i2.2402.
- [25] H. Bahuleyan, "Music genre classification using machine learning techniques," *arXiv:1804.01149*, Apr. 2018.
- [26] A. Dahiya, "Audio instruments identification using CNN and XGBoost," National College of Ireland, Ireland, 2019.
- [27] V. Osadchyy and R. V. Skuratovskii, "Big data classification for the analysis MEL scale features using KNN parameterization,"





- International Journal of Circuits, Systems and Signal Processing*, vol. 14, pp. 978–989, Dec. 2020, doi: 10.46300/9106.2020.14.125.
- [28] Y. A. Alotaibi, S.-A. Selouani, and W. Cichocki, “Investigating emphatic consonants in foreign accented Arabic,” *Journal of King Saud University - Computer and Information Sciences*, vol. 21, pp. 13–25, 2009, doi: 10.1016/S1319-1578(09)80002-5.
- [29] K. M. O. Nahar, M. Elshafei, W. G. Al-Khatib, H. Al-Muhtaseb, and M. M. Alghamdi, “Statistical analysis of Arabic phonemes used in Arabic speech recognition,” in *Neural Information Processing*, 2012, pp. 533–542.
- [30] D. Prabakaran and R. Shyamala, “A review on performance of voice feature extraction techniques,” in *2019 3rd International Conference on Computing and Communications Technologies (ICCCCT)*, Feb. 2019, pp. 221–231, doi: 10.1109/ICCCCT2.2019.8824988.
- [31] D. Prabakaran and S. Sriuppili, “Speech processing: MFCC based feature extraction techniques-an investigation,” *Journal of Physics: Conference Series*, vol. 1717, no. 1, Jan. 2021, doi: 10.1088/1742-6596/1717/1/012009.
- [32] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation functions: Comparison of trends in practice and research for deep learning,” *arXiv:1811.03378*, Nov. 2018.
- [33] A. Ali and S. Renals, “Word error rate estimation for speech recognition: e-WER,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2018, pp. 20–24, doi: 10.18653/v1/P18-2004.
- [34] Z. Comert and A. Kocamaz, “A study of artificial neural network training algorithms for classification of cardiocography signals,” *Bitlis Eren University Journal of Science and Technology*, vol. 7, no. 2, pp. 93–103, Dec. 2017, doi: 10.17678/beuscitech.338085.

BIOGRAPHIES OF AUTHORS



Zinah Jaffar Mohammed Ameen     received the B.Sc. degree in 2005 in Software Engineering from University of Technology/Baghdad and received her M.Sc. degree in Information Engineering from Al Nahrain University, Baghdad, Iraq in 2011. She is currently pursuing her Ph.D. degree in Information and Communication Engineering department, College of Information Eng., Al-Nahrain University. She is an assistant lecturer in Computer Engineering department, University of Technology, Baghdad, Iraq. Her research interests include computer networks, information technology, data mining, and speech processing. He can be contacted at Zinah.jaffar@coie-nahrain.edu.iq.



Abdulkareem Abdulrahman Kadhim     was born in Baghdad, Iraq, in 1958. He received his B.Sc. degree in Electrical and Electronics Engineering in 1981 from MEC, Iraq, and M.Sc. and Ph.D. degrees from Loughborough University of Technology, UK, in 1984 and 1989, respectively, in Digital Communication Systems. He is an IEEE Senior Member and Member of ACM. Currently, he is a professor of Digital Communications in the College of Information Engineering, Al-Nahrain University, Iraq. He has published 68 papers in international and national journals and scientific conferences. He successfully supervised 11 Ph.D. dissertations and 62 M.Sc. theses. His research interests include modern error correction codes for next generation networks, detection of coded and modulated signals, low complexity decoders, millimeter wave channel modeling, network coding, software defined environments, and efficient routing for wireless sensor networks (WSNs) and IoT networks. He can be contacted at abdulcareem.a@coie-nahrain.edu.iq.