Neural Computing and Applications (2021) 33:9089–9108
https://doi.org/10.1007/s00521-020-05672-2

ORIGINAL ARTICLE

Deep neural network architectures for dysarthric speech analysis and recognition

Brahim Fares Zaidi[1] · Sid Ahmed Selouani[2] · Malika Boudraa[1] · Mohammed Sidi Yakoub[2]

Received: 16 April 2019 / Accepted: 28 December 2020 / Published online: 9 January 2021

Abstract
This paper investigates the ability of deep neural networks (DNNs) to improve the automatic recognition of dysarthric speech through the use of convolutional neural networks (CNNs) and long short-term memory (LSTM) neural networks. Dysarthria is one of the most common speech communication disorders associated with neurological impairments that can drastically reduce the intelligibility of speech. The aim of the present study is twofold. First, it compares three different input features for training and testing dysarthric speech recognition systems. These features are the mel-frequency cepstral coefficients (MFCCs), mel-frequency spectral coefficients (MFSCs), and the perceptual linear prediction features (PLPs). Second, the performance of the CNN- and LSTM-based architectures is compared against a state-of-the-art baseline system based on hidden Markov models (HMMs) and Gaussian mixture models (GMMs) to determine the best dysarthric speech recognizer. Experimental results show that the CNN-based system using perceptual linear prediction features provides a recognition rate that can reach 82%, which constitutes relative improvement of 11% and 32% when compared to the performance of LSTM- and GMM-HMM-based systems, respectively.

Keywords Deep neural network · Convolutional neural network · Long short-term memory · Mel-frequency cepstral coefficient · Mel-frequency spectral coefficient · Perceptual linear prediction · Hidden Markov model · Gaussian mixture models

1 Introduction

Dysarthria is known as a motor speech disorder resulting from the malfunction of the muscles controlling the vocal apparatus [1]. The causes of dysarthria are multiple and include Parkinson's disease, stroke, head trauma, tumors, muscular dystrophies, and cerebral palsy [2, 3]. Dysarthria may affect breathing, phonation, resonance, articulation, and prosody. The consequences are hypernasality and the drastic reduction in speech intelligibility. Vowels may also be distorted in the most severe cases. The range of degradation of intelligibility is wide and depends on the extent of neurological damage.

1.1 Dysarthric speech recognition

Automatic speech recognition (ASR) systems can be very useful for people who are suffering from dysarthria and other speech disabilities. Unfortunately, due to the high variability and distortions in dysarthric speech [4, 5], the automatic recognition of dysarthric spoken words is still a challenging task [6, 7]. These distortions have a negative impact on the production and articulation of phonemes, which leads to a great complexity of their automatic analysis and characterization. For instance, an effortful grunt is often heard at the end of vocalizations; an

✉ Brahim Fares Zaidi
fzaidi@usthb.dz

Sid Ahmed Selouani
sid-ahmed.selouani@umoncton.ca

Malika Boudraa
mk.boudraa@gmail.com

Mohammed Sidi Yakoub
mohammed.sidi.yakoub@umoncton.ca

[1] Laboratory of Speech Communication and Signal Processing (LSCSP), USTHB University, Algiers, Algeria

[2] Laboratory of Research in Human-System Interaction (LARHSI), University of Moncton, Shippagan Campus, Moncton, Canada

excessively low pitch is frequently found, producing a harsh voice. In some cases, phonemes are characterized by pitch breaks in vocalic segments and imprecision of consonants' production. Therefore, the acoustical analysis of dysarthric speech has to deal with many issues related to aberrant voicing, tempo disturbance, unpredictable shifting of formant frequencies in sonorants and utterances where erroneously dropped phonemes are observed.

This complexity was also demonstrated by the acoustic study carried out by Zeigler and von Cramon's which involved ten patients with spastic dysarthria [8]. This study revealed that impaired acceleration of moving articulators increases production time and thus induces slower speech rate. These alterations disrupt or mask the acoustical characteristics that can normally help discriminating between phonemes, which makes the dysarthric speech recognition a more complex process.

Thus, there has recently been a trend toward the creation of tailored ASR systems for people with dysarthria [6, 9–11]. Indeed, the best results for dysarthria speech recognition have been provided by isolated word ASR models and conventional ASR algorithms, such as artificial neural networks (ANNs) [12], but an effective ASR system requires the ability to recognize continuous speech [13, 14]. Recently, some research initiatives have been successful in recognizing dysarthric speech with a limited vocabulary. However, currently, a large-vocabulary dysarthric speech recognition system is unavailable.

Most conventional dysarthric speech recognition systems are generally based on statistical approaches such as hidden Markov models (HMMs) that perform the modeling of the sequential structure of speech signals. The HMMs of speech are mainly based on Gaussian mixture models (GMMs) that are considered the best statistical representation of the spectral distributions of speech waveforms.

The probabilistic modeling remains a powerful approach when coupled with flexible time dimension representation of uncertainty. In this context, a Gaussian process regression (GPR) method was proposed to predict the human intention [15]. For some applications such as dysarthric speech synthesis where there is a need to complete partially observable sequences, the GPR method could be useful to improve the synthetic speech naturalness.

Nevertheless, these methods cannot be applied in the recognition of dysarthric speech. Actually, GMM-based modeling is effective when a large quantity of data is used to train a robust model. However, it is not as efficient for dysarthria because the corpora used for training are always small [16].

As an alternative to the statistical approaches and in the context of the considerable progress made by connectionist approaches, numerous configurations based on deep neural networks (DNNs) have been proposed to deal with the inherent complexity of dysarthric speech. Among these advanced configurations, convolutional neural networks (CNNs) [17] and long short-term memory (LSTM) networks [18] have achieved state-of-the-art recognition accuracy in many applications.

1.2 Dysarthric speech processing using CNN-based architectures

Isolated word ASR models and conventional ANN architectures have been widely used to perform dysarthric speech recognition. The authors of [12] identified the best-performing set of mel-frequency cepstral coefficient (MFCCs) parameters to represent dysarthric acoustic features for use in ANN-based ASR. The results show that the speech recognizer trained by the conventional 12-coefficient MFCC features without the use of delta and acceleration features provided the best accuracy, and the proposed speaker-independent ASR recognized the speech of unforeseen dysarthric subjects with a word recognition rate of 68.38%. To improve dysarthric speech identification, the authors in [19] proposed a system using features resulting from the coding of 39 MFCCs by a deep belief network (DBN). The evaluation was performed using the Dysarthric Speech Database for Universal Access Research in both text-dependent and text-independent conditions where an accuracy rate of 97.3% was achieved. Using the same data, a study presented in [20] explored multiple methods for improving a hybrid GMM-DNN-based HMM for dysarthric speech recognition. The experiments were carried out using DNNs with four hidden layers and sigmoid activation functions for the 1024 neurons of each layer; a dropout factor of 0.2 for the first four DNN training epochs was applied. This configuration reduced the average relative word error rate (WER) by 14.12%.

Recently, DNN-based architectures have been proposed to generate artificial samples of dysarthric speech. In [21], artificial dysarthric speech samples were presented to five experienced speech-language pathologists. The authors used CNN-based architectures in both the generator and the discriminator of dysarthric speech. The results reveal that speech-language pathologists identified transformed speech as dysarthric 65% of the time.

In [22], an interpretable objective severity assessment algorithm for dysarthric speech based on DNNs was proposed. An intermediate Darley–Aronson–Brown (DAB) layer containing a priori knowledge provided by speech-language pathologists and neurologists was added to the DNN. The model was trained with a scalar severity label at the output of the network and intermediate labels that describe how atypical the impaired speech was along four perceptual dimensions in the DAB layer. The best performance for severity prediction was 82.6%.

In [23], an automatic detection of dysarthria using extended speech features called centroid formants was presented. The experimental data consisted of 200 speech samples from 10 dysarthric speakers and 200 speech samples from 10 age-matched healthy speakers. The centroid formants enabled an accuracy of 75.6% achieved with just one hidden layer and 10 neurons.

In [24], 39 MFCCs were used as input features for a dysarthric speech recognizer based on a hybrid framework using a generative learning-based data representation and a discriminative learning-based classifier. The authors also proposed the use of example-specific HMMs to obtain log-likelihood scores for dysarthric speech utterances to form a fixed-dimensional score vector representation. The discriminative capabilities of the score vector representation technique were demonstrated, particularly in the case of utterances with very low intelligibility.

In a recent application [25], the authors proposed to rate dysarthric speakers along five perceptual dimensions: severity, nasality, vocal quality, articulatory precision, and prosody on a scale from 1 to 7 (from normal to severely abnormal). They also used the Google ASR engine to calculate the WER of uttered sequences. Based on the obtained results, 32 dysarthric speakers were categorized with respect to the severity of their impairment.

To capture relevant acoustic–phonetic information of impaired speech, numerous studies have investigated different types of features. In this context, mel-frequency spectral coefficients (MFSCs) have been proposed as the basic acoustic features [26]. For the CNN, the authors used 40-dimensional filter bank features to obtain more evolved speaker-independent MFSC features, a linear discriminative analysis transformation for projecting sequences of frames into 40 dimensions, and then a maximum likelihood linear transformation for diagonalizing the matrix and gather the correlations among vectors. For speaker-dependent features, the authors employed a feature-space maximum likelihood linear regression. A comparison of the speech recognition architectures shows that even with a small database, the hybrid DNN-HMM models outperform classical GMM-HMM models according to WER measures.

In another study published in [16], different types of input features used by DNNs were assessed to automatically detect repetition stuttering and nonspeech dysfluencies within dysarthric speech. The authors used the TORGO database, and the results obtained using MFCCs and linear prediction cepstral coefficient (LPCCs) features produced similar recognition accuracies. Repetition stuttering in dysarthric speech and nondysarthric speech was correctly identified with accuracies of 86% and 84%, respectively. Nonspeech sounds were recognized with approximately 75% accuracy in dysarthric speakers.

In [27], a convolutive bottleneck network, which is an extension of a CNN, was proposed to extract disorder-specific features. A convolutive bottleneck network stacks a bottleneck layer, where the number of units is extremely small compared with the adjacent layers. The database used in their work was the American Broad News corpus. The use of bottleneck features in a convolutive network improved the accuracy from 84.3 to 88.0%.

In the context of speech-to-text systems for clinical applications, multiple speaker-independent ASR systems robust against pathological speech are presented in [28]. The authors investigated the performance of two convolutional neural network architectures: (1) a time–frequency convolutional neural network (TFCNN), which performs time and frequency convolution on the gammatone filterbank features, (2) a fused-feature-map convolutional neural network (FCNN), which uses frequency and time convolution in the acoustic and articulatory space, enabling the joint use of information from acoustic and articulatory space. The authors also compared TFCNN models with standard DNN and CNN models.

Recently, authors in [29] proposed a novel approach that is able to assess dysarthria intelligibility, which correlates strongly with perceptual intelligibility. Their approach requires the patient to speak a limited set of words (no more than 5 words). The system is based on the end-to-end deep speech framework to obtain a string of characters.

1.3 Dysarthric speech processing using recurrent neural network (RNN)-based architectures

A recurrent neural network (RNN) is a category of artificial neural networks with the capacity to exhibit the temporal dynamic behavior of a given input sequence. Its main feature is that connections between nodes form a directed graph along the input time sequence. In the conventional RNN, the training algorithm uses gradient-based backpropagation through time. This configuration has the drawback of slow updating of the network weight. To solve this problem, a new structure LSTM has been introduced. Unlike conventional RNNs, LSTM networks connect their units in a specific way to avoid the problems of vanishing and exploding gradients. This makes them very useful for tasks such as unsegmented speech processing and recognition. The performance of various RNN architectures to train acoustic models for large-vocabulary speech recognition, namely LSTM, conventional RNN, and DNN, was compared in [30]. A distributed training of LSTM-RNNs using asynchronous stochastic gradient descent optimization was proposed [30]. The authors also showed that two-layer deep LSTM-RNNs, where each LSTM layer has a linear recurrent projection layer, can exceed state-of-the-art speech recognition performance. The deep LSTM-RNNs

[31] were extended by introducing gated direct connections between memory cells in adjacent layers. CNN and LSTM networks with very deep structures were investigated. The performances of each method were analyzed and compared with those of the DNNs. The obtained results clearly demonstrated the advantage of the CNN and LSTM techniques in terms of improving ASR accuracy for various tasks [31]. As with DNNs with deeper architectures, deep LSTM-RNNs have been successfully used for speech recognition [32–34].

Based on the RNN with LSTM units, [35] determined whether Mandarin-speaking individuals were afflicted with a form of dysarthria based on samples of syllable pronunciations. Using accuracy and receiver operating classification tasks, the authors evaluated several LSTM network architectures. Their results showed that the LSTMs' ability to leverage temporal information within its input makes for an effective step in the pursuit of accessible dysarthria diagnoses.

Similarly, [36] proposed a machine learning-based method to automatically classify dysarthric speech into intelligible and unintelligible using LSTM neural networks. The classification and training of dysarthric speech were performed using the bidirectional LSTM type of RNNs. The authors adopted a transfer learning approach, where the internal representations are learned by DNN-based ASR models.

Despite the availability of numerous technological solutions and fundamental approaches, the design of robust dysarthric speech recognition systems still faces numerous issues. Dysarthric speech is versatile, is uncertain and remains in many situations intractable to conventional formalism and methods. It is worth mentioning that a very little research has been done to give dysarthric speech recognition systems the required robustness by realizing the potential benefit from the joint optimization of both front-end processing and recognition modeling.

In an attempt to provide new insights into dysarthric speech recognition, a unified approach that aims to provide robustness when the systems are confronted with imprecise and distorted dysarthric speech signal is proposed. Unlike state-of-the-art methods, this approach investigates the benefits that can be derived from DNN-based architectures that jointly optimize the selection of front-end processing, multiple parameters such as framing and training configuration as well as classifier architectures. A comprehensive and holistic analysis of the dysarthric speech recognition process is carried out to provide a theoretical scheme based on the most effective components leading to a usable user interface.

1.4 Objective and contributions

In this paper, the best approaches for automatically recognizing dysarthric speech using DNN-based architectures are investigated. Our goal is to contribute to the research effort that ultimately will open the doors toward the design of personalized assistive speech systems and devices based on robust and effective speech recognition that are still not available for people who live with dysarthria. In this context, the contributions of this paper are as follows:

(i) to propose a new design of a speaker-dependent dysarthric speech recognizer. The proposed system is an important step toward the realization of usable speech-enabled interface for people with dysarthria;

(ii) to assess original DNN-based architectures, providing a benchmark for DNN models on the Nemours publicly available dataset [37];

(iii) to provide a detailed analysis that investigates the ability of acoustic modeling using perception and hearing mechanisms of yielding more robustness to the dysarthric speech recognition system. In this context, the performance of three acoustical analyzers, mel-frequency cepstral coefficients, mel-frequency spectral coefficients and perceptual linear prediction coefficients, is assessed;

(iv) to present a comprehensive investigation of the pre-processing pipeline that leads to the optimal framing and method of training/test while reducing the risk of bias and overfitting.

The remainder of this paper is organized as follows: Section 2 describes the methodology of this work. Section 3 presents the experimental protocol used in the experiments, particularly regarding the different input features and the baseline HMM-GMM system against which the CNN and LSTM systems were compared. The obtained results and related discussions are provided in Sect. 4. Finally, Sect. 5 draws the conclusions and highlights future work.

2 Methodology

2.1 Data

The Nemours database is a collection of 814 short nonsense sentences; 74 sentences are uttered by each of the 11 American male speakers with different degrees of dysarthria. Each sentence has been transcribed at the word and phoneme levels.

To provide input data to the CNN- and LSTM-based systems, we divided each sentence's waveform into its phoneme waveforms. A set of 14,080 waveform files were created (see Table 1) and used in the experiments.

Different subsets were extracted from the new phoneme set using the following splitting techniques [38]: normal subset, threefold top subset, threefold middle subset, and threefold bottom subset.

The threefold technique is widely used for corpus splitting in the context of neural network-based classification. It consists of splitting the corpus into three equal parts, each comprising 33% of the whole corpus. From the three parts, we chose a single part as the test part (upper part, middle part, or lower part) and used the rest for training. Table 2 shows the extracted subsets used throughout the experiments.

2.2 Auditory-based input features

Several acoustic features can be used as input parameters of recognition systems dealing with speech disorders. In our case, we extracted the MFCC [39], MFSC [40], and the perceptual linear prediction (PLP) features [41] from the data subsets and used them to train and test the HMM-GMM baseline system as well as the CNN- and LSTM-based systems. These three acoustical analysis methods perform modeling of perception and hearing mechanisms, which is expected to provide more robustness to the dysarthric speech recognition system. The detailed results obtained by each system using different types of acoustic features are presented in Sect. 3.

2.2.1 Mel-frequency Cepstral coefficients (MFCCs)

MFCCs are frequency-domain features that have demonstrated their effectiveness in speech recognition [39]. They are the most commonly used frame-based features with the assumption that the speech is wide-sense stationary over short frames with time lengths ranging from 10 to 25 ms. The frequency bands are equally spaced on the mel scale, which approximates the human auditory system's response. The extraction process follows the steps illustrated in Fig. 1.

The input vector of each frame is composed of the 13 first static coefficients of the MFCCs, their first derivatives that represent the velocity (ΔMFCCs), and their second derivatives that represent the acceleration (ΔΔMFCCs) to obtain a vector of 39 MFCCs. We used a 25-ms Hamming window with a 10-ms offset.

2.2.2 Mel-frequency spectral coefficients (MFSCs)

The mel-frequency spectral coefficients were extracted before performing the discrete cosine transform (DCT) in the MFCC procedure. As shown in Fig. 1, this led to the log mel-frequency spectral coefficients. The human auditory system's response is approximated by the frequency bands that are equally spaced on the mel scale. Despite the presence of the correlation within the MFSC, these features can be used in a DNN-based configuration since the deep structure will subsequently perform an implicit decorrelation. [40]. In our study, 39 mel filterbanks were used while calculating the MFSC for 16-kHz sampled speech waveforms.

2.2.3 Perceptual linear prediction (PLP)

The goal of the original PLP model developed by Hermansky [41] was to describe the psychophysics of human hearing more accurately in the feature extraction process. In contrast to pure linear predictive analysis of speech, PLP modifies the short-term spectrum of speech by several psychophysically based transformations. In this study, we used 39 PLP coefficients for each frame. Figure 2 illustrates the main steps for calculating the PLP coefficients.

Table 3 shows the configuration of feature extraction used for all experiments.

2.3 HMM-GMM: baseline system

HMM-GMM modeling has been widely used in ASR. In such a probabilistic-based approach, each HMM state is represented by a GMM output probability distribution. Mostly, diagonal covariance Gaussians are used because they are easier to train than full covariance Gaussians of the same size.

Building HMM-GMM for smaller units of sound such as phonemes is a more efficient solution since phonemes are the basic units of speech. The HMM-GMM can be built on using single phonemes (monophone configuration) or context-dependent phoneme models that have individual phonemes linked to their left and right contexts (a triphone configuration). Triphone models are commonly used in ASR. A three-state left-to-right HMM is generally used to build a phoneme model with an incoming state, a middle

Table 1 Nemours database

	Speakers	Sentences uttered by each speaker	Sentence files	Phoneme files
Total	11	74	814	14,080

Table 2 Subsets generated using different splitting techniques

	Corpus split							
	90/10 (%)	80/20 (%)	70/30 (%)	60/40 (%)	50/50 (%)	Threefold top (%)	Threefold middle (%)	Threefold bottom (%)
Training + validation	90	80	70	60	50	67	67	67
Test	10	20	30	40	50	33	33	33

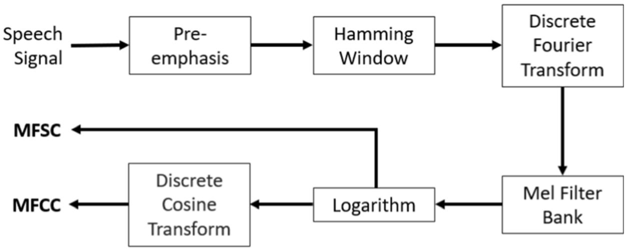


Fig. 1 MFCC and MFSC feature extraction

state, an outgoing state, and dummy start and end nodes. Each state node is associated with an HMM state index, whereas dummy nodes are not related to any acoustic event and are used to mark the two end points of a unit.

HMM mimics a random process that at each time $t \in \{1, \ldots, T\}$ seems to be in one of the N hidden states belonging to the set s and at each next time, left-over in prior state or pass across to different hidden state in accordance with certain transition probabilities. Hidden states are described by features that show in observation sequence O.

We currently specify parameters' number that totally depict the HMM with discrete observations. The hidden state of the HMM at time t is indicated as q_t, and the observation generated at time t is indicated as o_t.

The discrete HMM is typified by:

- Set of hidden states: $s = \{s_1, s_2, \ldots, s_N\},$ (1)
- Observation sequence : $O = \{o_t, t = 1, \overline{T}\},$ (2)
- Initial distribution: $\prod = \{\pi_i = p(q_1 = s_i), i = \overline{1, N}\},$ (3)
- Transition probabilities matrix : $A = \{a_{ij} = p(q_{t+1} = s_j \backslash q_t = s_i), i,j = \overline{1, N}\},$ (4)

Fig. 2 PLP feature extraction

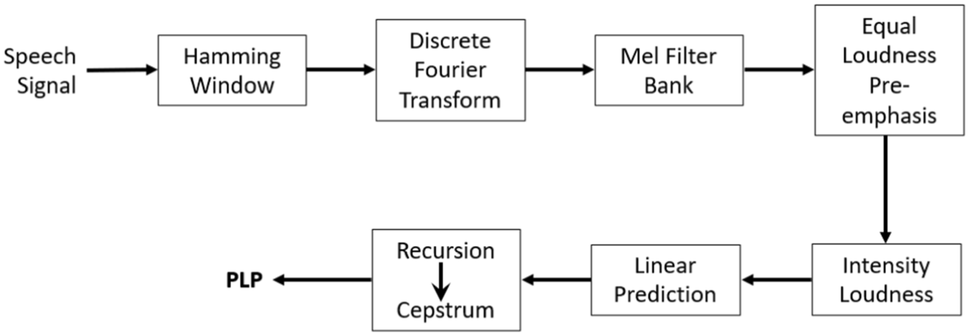


Table 3 Configuration of the different methods of feature extraction used in this study

	MFCC	MFSC	PLP
Hamming window size	25 ms	25 ms	25 ms
Length of offset	10 ms	10 ms	10 ms
Number of cepstral coefficients	12	12	12
Number of final coefficients	39 (MFCC_0_D_A)	39	39 (PLP_0_D_A)
Filter coefficient value	0.97	0.97	0.97
Number of cepstral filters	22	22	22
Number of filterbank channels	26	39	20

- Set of symbols : $\quad V = \{v_1, \ldots, v_M\},$ (5)

  where M is the observation symbols' number per state.
- Observation probabilities matrix :
  $$B = \left\{b_i(m) = p(o_t = v_m \backslash q_t = s_i), i = \overline{1,N}, m = \overline{1,M}\right\}, \tag{6}$$

We indicate an HMM as a triplet:

$$\lambda = (A, B, \pi), \tag{7}$$

In a continuous density HMM-based system, each state is associated with a continuous probability density function. The most effective probability density used in speech recognition is the density of the Gaussian mixture (the GMM part of the scheme) defined as follows:

$$\mathcal{N}\left(O_t; \mu_{jm}; \Phi_{jm}\right) = \frac{1}{\sqrt{(2\pi)^n \left|\Phi_{jm}\right|}} e^{-\frac{1}{2}\left(O_t - \mu_{jm}\right)' \Phi_{jm}^{-1}\left(O_t - \mu_{jm}\right)}, \tag{8}$$

where $\mathcal{N}$ denotes a multivariable Gaussian, where μ is the vector of the mean and Φ is the covariance matrix.

In practice, a mixture of Gaussian densities is used to generate a distribution that is closest to the real distribution of the data. For state j associated with a GMM, the probability of observation O_t is calculated from:

$$b_j(O_t) = \sum_{m=1}^{M} C_{jm} \mathcal{N}(O_t; \mu_{jm}; \Phi_{jm}), \tag{9}$$

where M is the number of components of the mixture and C_{jm} is the mth weight of the Gaussian of the state j.

In HMMs, Gaussian mixture models are used to represent the emission distribution of states. The probability of the vector O_t at each instant t in state j is represented by the following equation:

$$b_j(O_t) = \prod_{S=1}^{S} \left[\sum_{m=1}^{M} C_{jsm} \mathcal{N}(O_{st}; \mu_{jsm}; \Phi_{jsm})\right]^{\gamma_{js}}, \tag{10}$$

where M is the number of components of the mixture and C_{jsm} is the m^{th} weight of the Gaussian of the state j for the source S.

The exponent γ specifies the contribution of each flow to the global distribution by measuring its corresponding distribution. We assume that the value of γ_{js} satisfies the following constraints:

$$0 \leq \gamma_{js} \leq 1 \text{ and } \sum_{s=1}^{S} \gamma_{js} = 1, \tag{11}$$

The classification of a speech is fundamentally based on the value of the observation sequence probability gifted by the model. This value is calculated for the observation sequence and for each competitive HMM. Finally, the sequence is linked to a class, which corresponds to HMM-GMM that have the utmost probability. To calculate the probability of the sequence O, given the model λ the forward–backward algorithm is often used. The first part of the forward–backward algorithm allows calculating the values of forward variables, denoted by:

$$\alpha_t(i) = P(o_1, \ldots, o_t, q_t = s_i \backslash \lambda), t = \overline{1,T}, i = \overline{1,N}, \tag{12}$$

The following calculation steps lead to the determination of forward variables:

(1) $$\text{Initialization-calculate} : \alpha_1(i) = \pi_i b_i(o_1), i = \overline{1,N}, \tag{13}$$

(2) $$\text{induction} : \alpha_{t+1}(i) = b_i(o_{t+1}) \left[\sum_{j=1}^{N} \alpha_t(j) a_{ij}\right], i = \overline{1,N}, t = \overline{1, T-1}, \tag{14}$$

(3) $$\text{termination} : p(O \backslash \lambda) = \sum_{i=1}^{N} \alpha_T(i), \tag{15}$$

The second part of the forward–backward algorithm enables us to calculate the backward variables by:

$$\beta_t(i) = P(o_{t+1}, \ldots, o_T \backslash q_t = s_i, \lambda), t = \overline{1,T}, i = \overline{1,N}, \tag{16}$$

The calculation of backward variables is as follows:

(1) $$\text{initialization} : \beta_T(i) = 1, i = \overline{1,N}, \tag{17}$$

(2) $$\text{induction} : \beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) b_j(o_{t+1}) a_{ij}, i = \overline{1,N}, t = \overline{1, T-1}, \tag{18}$$

(3) $$\text{termination} : p(O \backslash \lambda) = \sum_{i=1}^{N} \beta_1(i), \tag{19}$$

In our experiments and as illustrated in Fig. 3, each phoneme is represented by a 5-state HMM model with two non-emitting states (the 1st and 5th states) and a mixture of 2, 4, 8, or 16 Gaussian distributions (the GMM component). We used three different types of coefficients (MFCCs, MFSCs or PLPs) as inputs to train and test the system.

This HMM-GMM is the baseline system used for comparison with the two proposed systems, namely the CNN- and LSTM-based systems.

2.4 CNN-based system: first proposed system

The principle of a CNN is to perform a convolution operation that produces filtered feature cards stacked on top of each other. A conventional neural network has the following characteristics [17]:

- *A convolutional layer:* This is the basic element of a CNN. The main purpose of this layer is to extract characteristics from the input features. The resulting output of the convolutional layer is given as follows:

$$C(x_{u,v}) = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{m}{2}}^{\frac{m}{2}} f_k(i,j) x_{u-i,v-j}, \tag{20}$$

  where f_k is the filter with a kernel size of $n \times m$ applied to the input x, and $n \times m$ is the number of input connections to each CNN neuron (unit).
- *A pooling layer:* This layer reduces the number of features and makes the learned functions more robust by making them more invariant to changes in scale and orientation. Certain functions are used to reduce subregions, such as taking the average or maximum value. The max-pooling function given below is used by our CNN-based system.

$$M(x_i) = max\left\{x_{i+k,i+l} \middle| |k| \le \frac{m}{2}, |l| \le \frac{m}{2} k, l \in \mathbb{N}\right\}, \tag{21}$$

  where x is the input and m is the size of the filter.
- *A Rectified Linear Unit (ReLU):* The ReLU is an operation that replaces all negative values in the feature map with zero. The goal of ReLU is to introduce nonlinearity into our CNN-based system because most of the data we want our CNN to learn are nonlinear. Other nonlinear functions, such as *tanh* or *sigmoid* functions, can be used, but in most cases, ReLU is more efficient. Given the input x, ReLU uses the activation function $R(x) = max\ (0,\ x)$ to calculate its output.
- *A fully connected layer:* This layer takes all the neurons of the previous layer and connects them to each of its neurons. Adding a fully connected layer is a good method for learning nonlinear combinations of these features. The output of this layer is given by:

$$F(x) = \sigma(W_{l\times k} * x), \tag{22}$$

where σ is the activation function, k is the size of the input x, and l is the number of neurons in the fully connected layer. This results in a matrix W.

- *An output layer:* The output layer is a one-hot vector representing the class of the given input vector. Therefore, its dimensionality is equal to the number of classes. In our work, we used 44 classes. The resulting class for output vector x is represented by:

$$C(x) = \{i | \exists i \forall j \ne i : x_j \le x_i\}, \tag{23}$$

- *A softmax layer:* The error is propagated back over a softmax layer. Let N be the dimension of the input vector. Then, the softmax calculates a mapping such that:

$$S(x) : \mathbb{R}^N \to [0,1]^N, \tag{24}$$

For each component $1 \le j \le N$, the output is calculated as follows:

$$S(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{n} e^{x_i}}, \tag{25}$$

The output of a layer is the input of the next layer, and most of the features learned from the convolution and

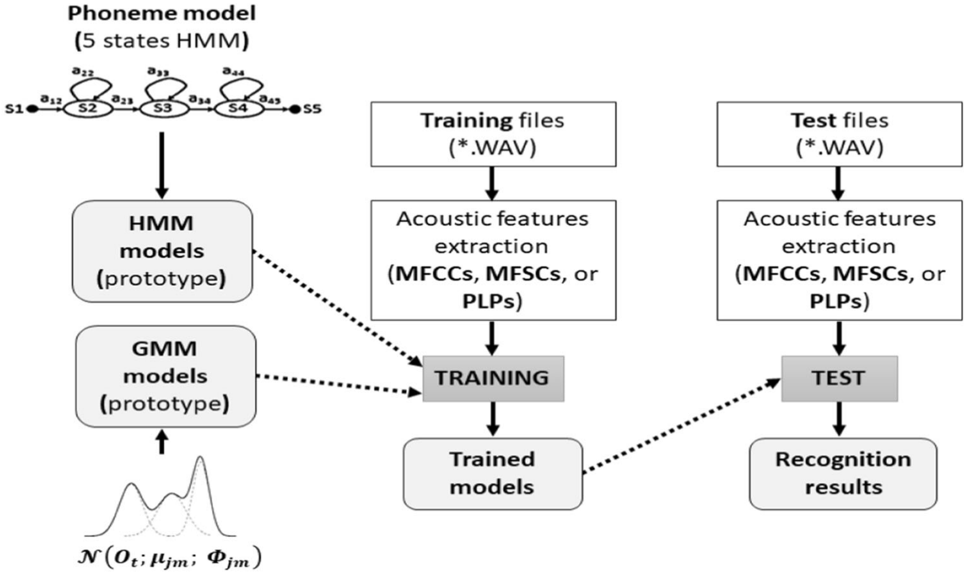


Fig. 3 Baseline system based on HMM-GMM

clustering layers may be good. However, the combinations of these features could be even better.

Conventionally, a CNN consists of several iterations of this succession of layers. One of the advantages of a CNN is its relatively rapid training.

The first proposed and implemented system for this study is the CNN-based system with the architecture depicted in Fig. 4. The input of the CNN is represented by several feature maps that are either MFCCs, MFSCs, or PLPs coefficients. Each input map is represented in two dimensions composed of 112 frames per phoneme and 39 coefficients per frame, corresponding to 4368 features per phoneme.

The implemented CNN consists of a single convolutional layer that allows for the output of 64 activation cards (ReLU), each with a kernel filter of size 39 × 39.

At the output of the convolution layer, we applied max-pooling downsampling layers (2, 2) to reduce the size of the activation cards. The reduced output of the max-pooling layer (64 small activation boards) serves as a dense two-layered input parameter of 500 neurons with a ReLU activation function. The output of the dense hidden layer is connected to the output layer, which represents a single vector. The last layer represents the class of input data (in this case, 44 classes, which represent the 44 phonemes) with a softmax activation function. The dropout regularization method ($p = 0.5$) is only used with dense hidden layers.

2.5 LSTM-based system: second proposed system

An LSTM network is a type of RNN that can learn long-term dependencies between the time steps of sequence data. It consists of a set of recurrently connected subnetworks referred to as memory blocks [18], as illustrated in Fig. 5b. Each memory block contains memory cells to store the temporal state of the network, as well as three multiplicative gate units to control the information flow. The input gate controls the information transmitted from the input activations into the memory cells, and the output gate controls the information transmitted from the memory cells to the rest of the network. Finally, the forget gate adaptively resets the memory of the cell.

An LSTM network computes a mapping from an input sequence $x = (x1, \ldots, xT)$ to an output sequence $y = (y1, \ldots, yT)$ by calculating the network unit activations using the following equations iteratively from $t = 1$ to T:

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i), \tag{26}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c), \tag{27}$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_t + b_o), \tag{28}$$

$$m_t = o_t \odot h(c_t), \tag{29}$$

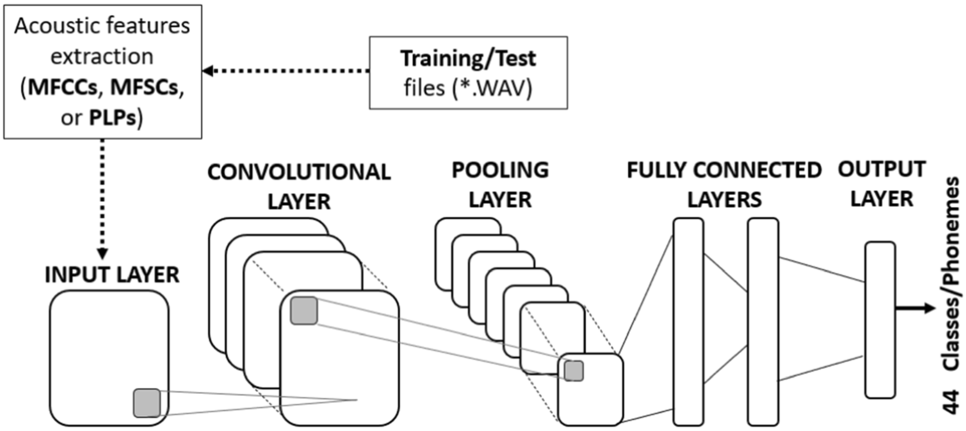
$$y_t = \varnothing\left(W_{ym}m_t + b_y\right), \tag{30}$$

where the W terms represent the weight matrices (e.g., W_{ix} is the matrix of the weights from the input gate to the input); W_{ic}, W_{fc}, and W_{oc} are the diagonal weight matrices for the peephole connections; the b terms denote the bias vectors (b_i is the input gate bias vector); σ is the logistic sigmoid function; i_t, f_t, o_t, and c_t are, respectively, the input gate, forget gate, output gate, and cell activation vectors at step t; m_t is the output of the LSTM layer; $\odot$ is the elementwise product of the vectors; g and h are the cell input and cell output activation functions, respectively; and $\varnothing$ is the network output activation function. The architecture of implemented system is depicted in Fig. 5a.

3 Evaluation setup

The evaluation and comparison experiments were carried out using the Nemours database of pathological voices [37]. We created subsets (see Sect. 2.1) that allowed us to carry out experiments and to draw conclusions regarding

Fig. 4 Implemented CNN-based architecture



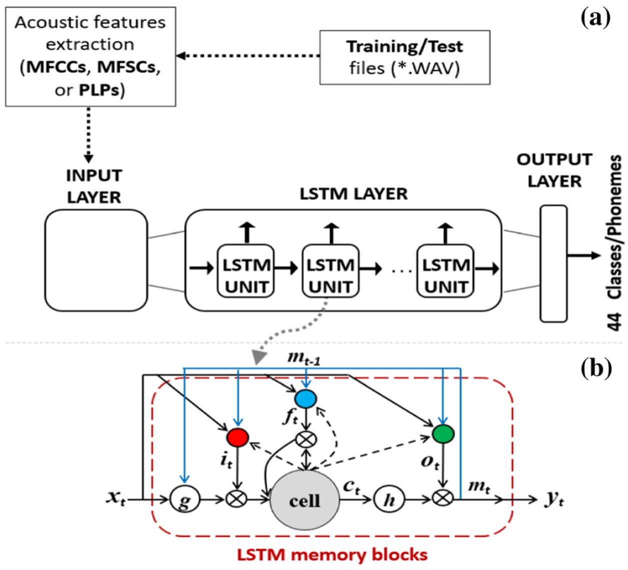


Fig. 5 **a** Implemented LSTM-based architecture, **b** LSTM unit architecture

the effectiveness of deep learning approaches to recognize dysarthric speech.

For comparison and validation purposes, we developed a baseline system based on HMM-GMM that is similar to the system described in [42]. Both implemented CNN- and LSTM-based systems were compared against the HMM-GMM baseline system. The HTK [43] toolkit was used to build the HMM-GMM models. The TensorFlow [44] and Keras [45] tools were used to implement the CNN and LSTM systems. Different architectures using different features were investigated to find the best configurations.

Table 4 describes the hyperparameters used to obtain results on the CNN- and LSTM-based systems.

3.1 Experiments and results

To train the CNN- and LSTM-based systems, we tried several corpus splitting techniques to extract different data subsets from the Nemours database. These subsets were used for training, testing, and validation.

After running the HMM-GMM baseline system as well as the CNN- and LSTM-based systems on the corpus threefold middle using MFCCs, MFSCs, and PLP features, we obtained a global accuracy for each system, as shown in Table 5.

From these results, we noticed that the best recognition rate for all speakers was obtained with the CNN-based system using PLP features and ReLU activation function.

To find the best configuration for the test subsets, multiple experiments were carried out with different data repartitions. Table 6 shows the results obtained for speaker BB after training and testing the CNN-based system of Fig. 4 and the LSTM-based system of Fig. 5a using different subsets with PLP features. The best performance was achieved with the threefold middle configuration.

The accuracies were obtained with standard deviations of $\sigma_{CNN} = 3.53$ and $\sigma_{LSTM} = 2.73$. We noticed that the value of the standard deviation σ_{CNN} was higher than σ_{LSTM}, which means that the values of the results with CNN were more distant from the average compared to the values with LSTM.

The LSTM-based system was trained and tested using an architecture consisting of a single LSTM layer composed of LSTM units. We noticed that an increase in the number of units of the LSTM layer led to an improvement in the recognition rate of dysarthria speech. Table 7 shows the recognition rates for the speaker BB with respect to the LSTM layer size. These rates were obtained with a standard deviation of $\sigma = 3.23$.

3.1.1 Effect of the filter size of the pooling layer

Table 8 shows that the ideal pooling layer filter size for the implemented CNN is 2 × 2 because it yields a better recognition rate for speaker BB. The best recognition rate is 75.27% with a standard deviation of 0.95. These results confirm that the size of the pooling layer has a substantial impact on the performance of the CNN-based system.

3.1.2 Effect of the kernel filter size of the convolutional layer

As shown in Table 9, the performance of the implemented dysarthric speech recognition system is better when the kernel filter size of the convolutional layer is large. These results were obtained with a standard deviation of $\sigma = 1.99$.

From the results in both Tables 8 and 9, we concluded that an ideal configuration of CNN is a 2 × 2 pooling layer filter size and a large kernel filter size for the convolutional layer.

3.1.3 Effect of the CNN's fully connected layer size

In this section, we investigate the effect of the number of neurons and layers of the fully connected layers in the CNN. Table 10 shows that CNN-based system performance was better with two fully connected 500-neuron layers than with a single fully connected layer of 1000 neurons.

3.1.4 Effect of Hamming window size

Table 11 and Figs. 6 and 7 show the impact of the Hamming window size on the automatic recognition rate of dysarthric phonemes with the CNN-based system of

Table 4 Hyperparameters used for experiments using the CNN and LSTM configurations

	CNN	LSTM
Input dimensions	112 frames, 39 coefficients	112 frames, 39 coefficients
Batch size	50	50
Number of epochs	200	200
Validation split	0.01	0.01
Dropout rate	0.5	–
Number of filters	65	–

Adadelta algorithm	Learning rate = 0.001
	Rho = 0.95
	Epsilon = 1e^{-7}
		Learning rate = 0.001
		Rho = 0.95
		Epsilon = 1e^{-7}

Table 5 Global accuracy of phonemes uttered by all speakers using threefold middle corpus using MFCCs, MFSCs, and PLP features

All speakers	CNN-based system	LSTM-based system	GMM-HMM baseline system
Accuracy (%) PLP	**58.94**	45.71	44.13
Accuracy (%) MFSCs	49.75	47.41	34.53
Accuracy (%) MFCCs	58.24	50.78	43.78

Bold indicates the highest accuracy

Table 6 Corpus splitting influence on BB speaker recognition rate with PLP coefficients in CNN and LSTM

	Corpus split							
	90/10	80/20	70/30	60/40	50/50	Threefold top	Threefold middle	Threefold bottom
Training + validation	90%	80%	70%	60%	50%	67%	67%	67%
Test	10%	20%	30%	40%	50%	33%	33%	33%
CNN Acc. (%)	63.93	66.55	68.17	70.46	70.55	72.71	**75.27**	70.58
LSTM Acc. (%)	58.21	58.36	58.91	64.06	60.60	65.03	61.91	63.75

Bold indicates the highest accuracy

Table 7 Influence of the LSTM layer size on the recognition rate of the phonemes uttered by speaker BB

	LSTM layer size			
	100 units	120 units	140 units	160 units
Acc. (%)	59.06	61.91	64.18	**65.25**

Bold indicates the highest accuracy

Table 8 Influence of the filter size of the max-pooling layer in the CNN on the recognition rate of speaker BB

	Filter size of pooling layer (max-pooling)					
	1 × 2	2 × 2	3 × 3	4 × 4	5 × 5	6 × 6
Acc. (%)	72.71	**75.27**	73.56	72.71	73.35	73.13

Bold indicates the highest accuracy

Table 9 Influence of the kernel filter size of the convolutional layer in the CNN on the recognition rate of speaker BB

	Kernel filter size of the convolutional layer		
	6 × 6	39 × 39	112 × 112
Acc. (%)	76.11	75.27	**79.07**

Bold indicates the highest accuracy

speakers BB, BK, and BV. These results show that for speaker BB, who presented a low level of dysarthria severity, the ideal length of the Hamming window was 25 ms.

For the two speakers BK and BV, who presented a high level of dysarthria severity, we noticed that when decreasing the size of the Hamming window (to 15 ms), the performance was better. This is probably due to the characteristics of dysarthric speech phonemes, where abnormal phoneme durations are observed, particularly in the most severe cases.

According to the statistical analysis of the experiments in Table 11 (see Fig. 6), the values of the standard deviations are, $\sigma_{BB} = 1.21$, $\sigma_{BK} = 0.78$ and $\sigma_{BV} = 0.49$.

Table 10 Impact of the number of neurons and layers of the fully connected layer in the CNN on the recognition rate of phonemes uttered by speaker BB

	Fully connected layer sizes					
	Number of layers 1	Neurons by layer 1000	Number of layers 1	Neurons by layer 500	Number of layers 2	Neurons by layer 500
Acc. (%)	72.07		70.36		**75.27**	

Bold indicates the highest accuracy

Table 11 Impact of the Hamming window size on the automatic recognition rate of dysarthric phonemes for speakers BB, BK, and BV using PLP coefficients on the CNN-based system

	Hamming window size			
	15 ms	20 ms	25 ms	30 ms
Speaker BB: Acc. (%)	73.13	73.13	**75.27**	72.49
Speaker BK: Acc. (%)	**43.97**	43.13	42.07	42.92
Speaker BV: Acc. (%)	**53.09**	53.07	52.64	52.03

Bold indicates the highest accuracy

In Fig. 7, we can see that the recognition rate follows a normal distribution because most of the observations were between the mean and mean ± σ and 100% of the observations were between the mean and mean ± 2σ.

3.1.5 Effect of acoustic features

We compared three types of features (see Sect. 2) as inputs to the CNN-based system. Each vector of input features was composed of 39 coefficients. Table 12 compares the three types of features, namely MFCC, MFSC, and PLP coefficients, used as inputs of the CNN-based system. According to the obtained results, the best recognition rate was 80.13%, obtained with the PLP coefficients for speaker FB. In the case of speaker LL, using the MFSCs led to a recognition rate of 54.16%, which represented the lowest result. For all cases, the best results were obtained when the PLP coefficients were used as input features.

Figure 8 describes the statistical analysis of the experiments used to identify the best-performing parameter among these three coefficients: MFCC, MFSC, and PLP coefficients.

3.1.6 Effect of activation functions

We carried out experiments using different DNN activation functions in order to investigate their impact on the performance of the training procedure. A variety of standard activation functions, namely ReLU [46], AReLU [47], and SELU [48], were evaluated. Moreover, in the context of the

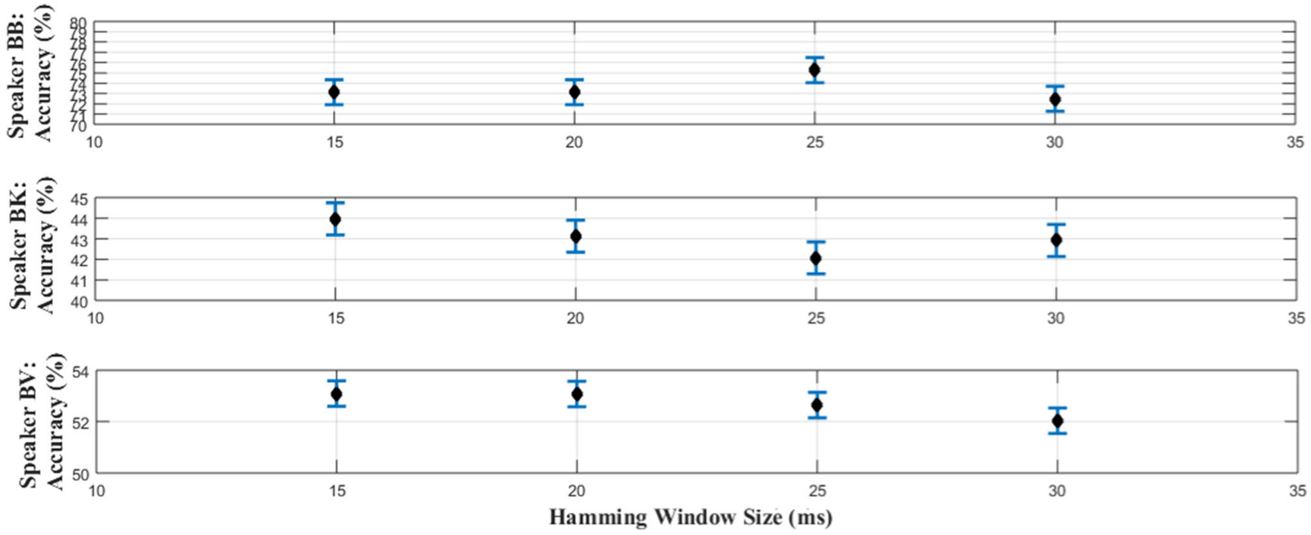


Fig. 6 Statistical analysis of the impact of the Hamming window size on the automatic recognition rate of dysarthric phonemes for speakers BB, BK, and BV using PLP coefficients on the CNN-based system

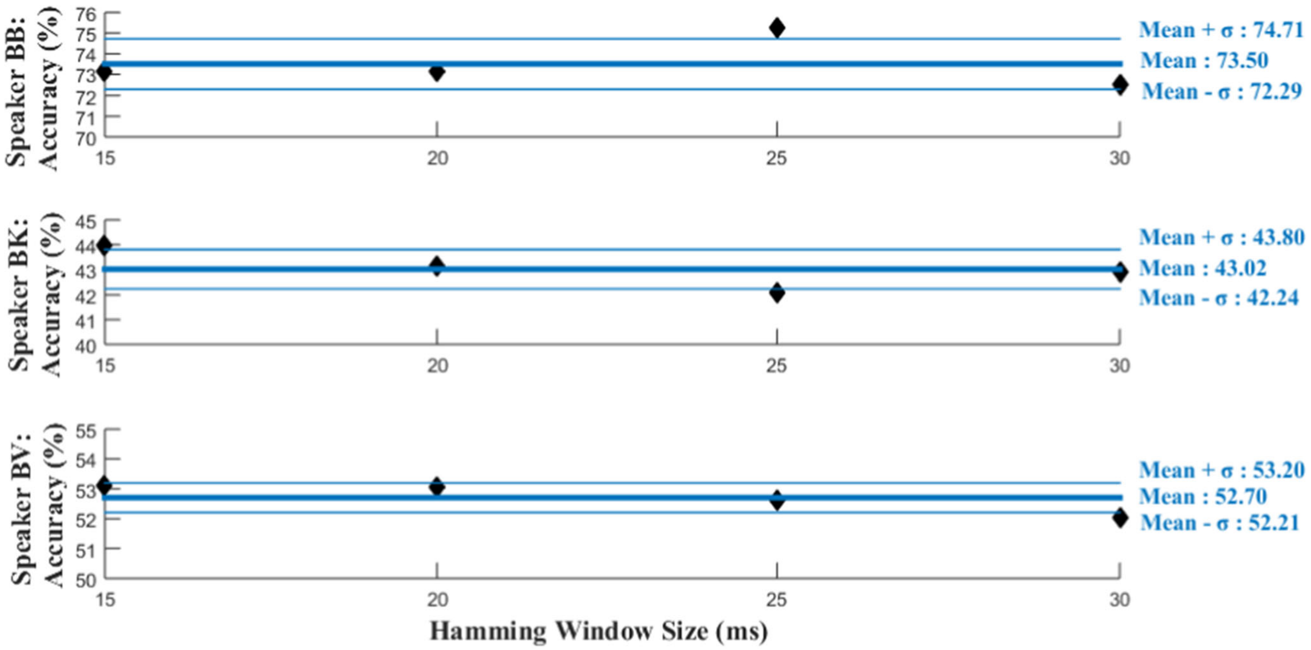


Fig. 7 Statistical analysis of the impact of the Hamming window size on the automatic recognition rate of dysarthric phonemes for speakers BB, BK, and BV using PLP coefficients on the CNN-based system

Table 12 Comparison of the correct recognition rate of the CNN-based system using MFCCs, MFSCs, and PLP coefficients

	Acoustic feature		
	MFCC	MFSC	PLP
Speaker BB: Acc. (%)	74.20	60.75	75.27
Speaker FB: Acc. (%)	78.22	69.77	**80.13**
Speaker LL: Acc. (%)	63.33	54.16	63.54
Speaker MH: Acc. (%)	71.88	60.25	72.73

Bold indicates the highest accuracy

growing and recent interest in polynomial activation functions, we have also introduced and evaluated new functions that we called: Poly1ReLU and Poly2Relu. The description and evaluation of both standard and new functions are given in the following subsections.

(a) *Rectified Linear Units (ReLU)*

The Rectified Linear Units activation function is a piecewise linear function that will output the input directly if it is positive; otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves good performance [46]. The ReLU activation function is defined as:

$$\mathcal{R}(x_i) = \begin{cases} 0, & x_i < 0 \\ x_i, & x_i \geq 0 \end{cases} = \max(0, x_i), \tag{31}$$

where $X = \{x_i\}$ is the input of the current layer.

(b) *Attention-based Rectified Linear Units (AReLU)*

The Attention-based Rectified Linear Units (AReLU) activation function [47] is given by:

$$\begin{aligned} \mathcal{F}(x_i, \alpha, \beta) &= \mathcal{R}(x_i) + \mathcal{L}(x_i, \alpha, \beta) \\ &= \begin{cases} C(\alpha)x_i, & x_i < 0 \\ (1 + \sigma(\beta))x_i, & x_i \geq 0 \end{cases}, \end{aligned} \tag{32}$$

where $\mathcal{R}(x_i)$ is the standard ReLU activation function; $\mathcal{L}(x_i, \alpha, \beta)$ represents the function in elementwise Sign-based attention (ELSA) with a network layer having learnable parameters α and β; $C(.)$ clamps the input variable into [0.01, 0.99]; σ is the sigmoid function. AReLU is expected to amplify positive elements and to suppress negative ones based on the learned scaling parameters β and α.

(c) *Scaled Exponential Linear Unit (SELU)*

The Scaled Exponential Linear Unit (SELU) activation function [48] is defined as:

$$SELU(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \leq 0 \end{cases}, \tag{33}$$

where α and λ are predefined constants with α = 1.67 and λ= 1.05 in our case.

(d) *Poly1ReLU activation function*

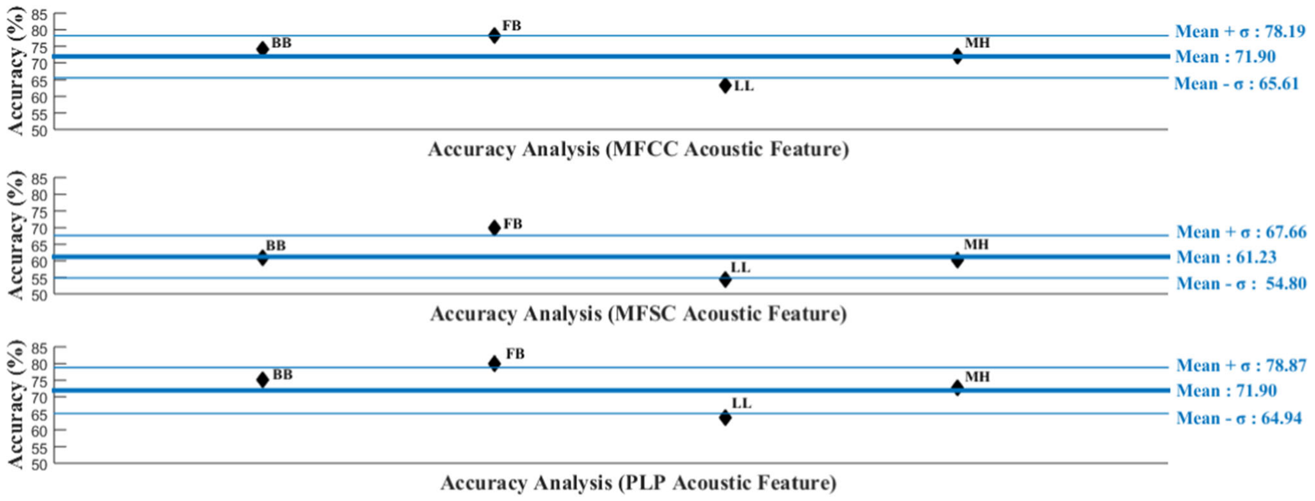


Fig. 8 Statistical analysis of the correct recognition rate of the CNN-based system using MFCCs, MFSCs, and PLP coefficients

The main idea of using polynomial activation functions is to learn nonlinearity and to approximate continuous real values of input data in order to provide the best discriminative model. The proposed Poly1ReLU activation function can be considered as a first-order polynomial activation function and is given by:

$$Poly1ReLU(x_i) = \begin{cases} 0, & x_i < 0 \\ \dfrac{x_i}{MAX(|X|)}, & x_i \geq 0 \end{cases}, \qquad (34)$$

where $X = \{x_i\}$ is the input of the current layer and $MAX(|X|)$ is the maximum of the absolute of X.

(e) ***Poly2ReLU activation function***

We extend polynomial-based activation functions by proposing a second-order polynomial activation function

$$Poly2ReLU(x_i) = \begin{cases} 0, & x_i < 0 \\ \dfrac{(x_i^2 + x_i)}{MAX(|X|)}, & x_i \geq 0 \end{cases}, \qquad (35)$$

The second-order polynomial activation function is expected to have the ability to learn more suitable nonlinearity. However, it is important to mention that the order of the function cannot be increased indefinitely because of the instability due to exploding gradients. Indeed, despite the use of normalization which performs the inputs scaling, the abrupt fluctuations of the high-order polynomials cannot be prevented in the particular case of pathological speech.

Table 13 presents the results obtained by the CNN-based system using different activation functions by maintaining the best features and parameters depicted in the previous sections: threefold middle cross-validation corpus with PLP coefficients. From the obtained results, we notice that the **Poly1ReLU** activation function gives the best results compared to the other activation functions.

3.1.7 Effect of the number of Gaussians and mixture weights on GMM-HMM system

In order to find the best configuration of the GMM-HMM system, an embedded optimization process is carried out within the procedure of probability re-estimation to determine the number of Gaussian mixtures M and the corresponding weighting coefficients c.

The methodology we used to select the number of mixtures consists of repeatedly increasing the number of components by a process of mixture splitting until the desired level of performance is reached. The process increments the number of mixtures step by step and then performs a re-estimation which continues until obtaining a convergence of the estimation probability. Given m mixture components, the probability density function is converted to $m + 1$ mixture components by cloning them through a process where the two resulting mean vectors are perturbed by adding 0.2 standard deviation to one and subtracting the same amount from the other. The re-estimation allows a floor to be set on each individual variance of every mixture, and thus, if any diagonal covariance component falls below the threshold of 0.00001, then the corresponding mixture weight is set to zero. The updating of all parameters is done by using embedded forward–backward algorithm presented in Sect. 2.3. The re-estimation procedure used in this work is provided by the HTK Tools [43].

Table 14 presents the results of the GMM-HMM dysarthric speech recognition for all speakers. The increased number of mixtures was presented with the corresponding

Table 13 Impact of activation functions on the accuracy of dysarthric speech recognition

Activation function	Accuracy (%) PLP by speaker										Global/average
	BB	BK	BV	FB	JF	LL	MH	RK	RL	SC	
ReLU	**75.27**	42.28	53.30	80.13	57.51	**66.10**	72.73	**49.03**	60.68	50.74	60.77
SELU	72.49	43.34	56.01	80.01	57.99	63.11	72.24	44.51	62.27	51.34	60.33
AReLU	75.05	45.87	57.78	81.60	**60.04**	65.45	73.15	46.68	62.57	52.85	62.10
Poly1ReLU	73.13	**47.78**	**58.42**	**82.02**	59.40	65.67	**75.05**	47.34	62.57	**55.60**	**62.69**
Poly2ReLU	71.42	45.24	57.14	80.97	58.98	64.17	74.20	44.69	**63.84**	52.64	61.32

Bold indicates the highest accuracy

sets of optimal weights that were obtained by using the HTK toolkit [43]. These results show that the best recognition rate for all speakers is obtained with a number of components *M* of the mixture Gaussian which is equal to 1. This result is to a certain extent predictable since the limited amount of data available to design speaker-dependent systems are not sparse to the point of requiring multiple mixtures. In this type of systems, the mixture components have little associated training data, and therefore, both the variance and the corresponding mixture weight become very small. The additional mixture components are then deleted, and only one Gaussian is retained to represent the HMM.

3.1.8 Comparison of the CNN- and LSTM-based systems

Table 15 compares the results of the correct recognition rate obtained by the three different recognition systems, CNN, LSTM, and HMM-GMM with optimal number of Gaussians, for each of the ten dysarthric speakers of the Nemours database.

From these results, the best recognition rate of 82.02% was obtained with the CNN-based system. The lowest rates were obtained in the case of speaker BK. This speaker was the most severely impaired by dysarthria. For all cases, the best results were obtained when the CNN-based system was used.

Figure 9 shows the results obtained by the three systems for speaker BB using PLP features for the eight experimental subsets. According to this graph, the best recognition rate was obtained with the CNN-based system using the threefold middle subset.

Figure 10 describes the statistical analysis of experiments used to identify the best-performing system for dysarthric speech. From Figs. 8 and 10, we noticed that the recognition rate followed a normal distribution because the majority of the observations were between the mean and mean ± σ and 100% of the observations were between the mean and mean ± 2σ.

4 Discussion

The results clearly show that the CNN-based system achieves the best recognition rate compared to the HMM-GMM- and LSTM-based systems. Theoretically, the LSTM has a salient feature that makes it a potentially useful configuration to capture time variability and dependencies of factors linked to the impact of dysarthria on the utterances' duration. However, this capacity was not demonstrated by the obtained results. Indeed, the CNN architecture achieved better performance even for the most severe cases where the prosody of speech is disturbed, the speech rate is slowed and timing of phonemes is abnormal. The CNN is thus capable, in the context of speaker-dependent application, of capturing these timing artifacts and can be used as a robust recognizer of dysarthric speech.

As presented in Table 16, the number of convolution layers needed for the CNN is one; an additional convolutional layer did not significantly improve the recognition rate. As for the Kernel filter size and the number of filters used at the input of the convolutional layer, an increase of it led to a better recognition rate. The filter size of the pooling layer, as mentioned in the literature, had a significant impact on the recognition performance. In addition, two fully connected layers composed of 500 neurons each were found to be effective, but no clear rule could be derived for the number of neurons in the fully connected layer.

In terms of acoustic analysis, although there is a superiority of PLPs when used in conjunction with CNNs, the results show that conventional MFCCs remain robust in recognizing dysarthric speech.

An important finding is related to the optimal frame duration. The results show that when the level of severity is

Table 14 Influence of the number of components M of the mixture and weighing coefficients c on the recognition rate for all dysarthric speakers

	The number of components M of the mixture and weighing coefficients W							
	$M = 1$ ($c_1 = 1.0$)	$M = 2$ ($c_1 = 0.4$; $c_2 = 0.6$)	$M = 3$ ($c_1 = 0.2$; $c_2 = 0.4$; $c_3 = 0.4$)	$M = 4$ ($c_1 = 0.2$; $c_2 = 0.2$; $c_3 = 0.2$; $c_4 = 0.4$)	$M = 5$ ($c_1 = 0.2$; $c_2 = 0.2$; $c_3 = 0.2$; $c_4 = 0.2$; $c_5 = 0.2$)	$M = 6$ ($c_1 = 0.2$; $c_2 = 0.2$; $c_3 = 0.2$; $c_4 = 0.2$; $c_5 = 0.1$; $c_6 = 0.1$)	$M = 7$ ($c_1 = 0.2$; $c_2 = 0.2$; $c_3 = 0.2$; $c_4 = 0.1$; $c_5 = 0.1$; $c_6 = 0.1$; $c_7 = 0.1$)	$M = 8$ ($c_1 = 0.2$; $c_2 = 0.2$; $c_3 = 0.1$; $c_4 = 0.1$; $c_5 = 0.1$; $c_6 = 0.1$; $c_7 = 0.1$; $c_8 = 0.1$)
Speaker BB: Acc. (%)	**50.00**	**50.00**	**50.00**	**50.00**	47.10	49.21	49.24	**50.00**
Speaker BK: Acc. (%)	**31.39**	30.11	30.11	30.11	29.45	30.78	30.78	30.11
Speaker BV: Acc. (%)	**46.00**	45.33	45.33	45.33	42.67	45.33	45.33	45.33
Speaker FB: Acc. (%)	**49.31**	49.20	49.20	48.67	47.33	49.23	49.20	48.67
Speaker JF: Acc. (%)	**45.14**	44.00	43.33	41.33	42.00	38.00	42.67	42.67
Speaker LL: Acc. (%)	**46.00**	45.33	44.00	42.67	43.33	43.33	43.33	43.33
Speaker MH: Acc. (%)	**34.67**	34.00	34.00	**34.67**	34.00	33.00	33.00	33.00
Speaker RK: Acc. (%)	**45.33**	**45.33**	42.67	42.00	42.00	41.33	41.33	42.00
Speaker RL: Acc. (%)	**42.00**	**42.00**	40.67	40.00	16.00	35.33	40.00	40.00
Speaker SC: Acc. (%)	**42.67**	40.58	40.00	39.33	34.00	40.00	41.33	40.00

Bold indicates the highest accuracy

high, it is recommended to use a shorter Hamming window. This can be explained by the need to adapt to the distortions induced by impaired speech. This is because a shorter duration allows the system to cope with the rapid changes that characterize the high severity levels of dysarthria.

The threefold cross-validation is found optimal to evaluate the models' ability to generalize on unseen data during the test phase. The experiments confirm that the use of cross-validation results in a less-biased and a reduced overfitting estimate compared to simple train/test split strategy.

Table 15 Comparison of the correct recognition rate of the three systems for all speakers using PLP features and the threefold middle corpus

Speaker/model	HMM-GMM system (%)	CNN-Poly1ReLU-based system (%)	LSTM-based system (%)
BB	50.00	73.13	61.91
BK	31.39	47.78	32.77
BV	46.00	58.42	46.70
FB	49.31	**82.02**	71.46
JF	45.14	59.40	47.99
LL	46.00	65.67	53.73
MH	34.67	75.05	65.54
RK	45.33	47.34	32.30
RL	42.00	62.57	55.39
SC	42.67	55.60	37.84

The best results are in boldface

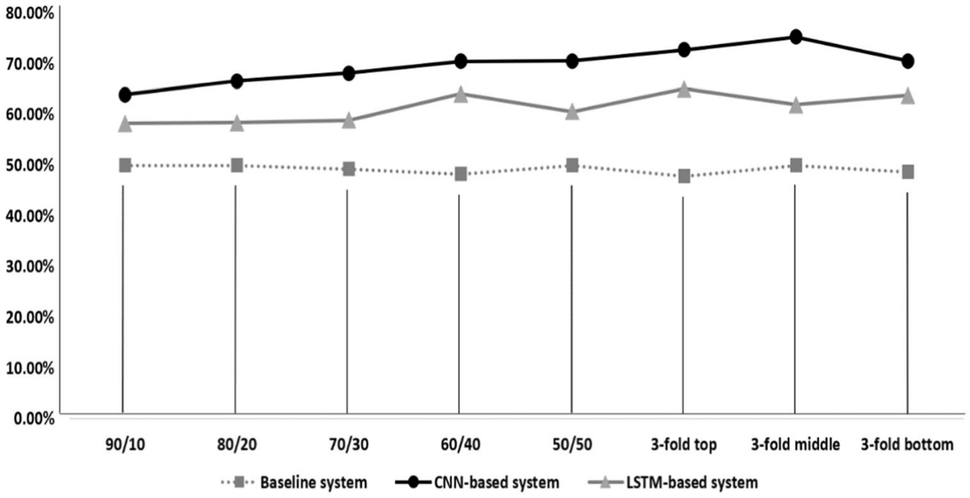


Fig. 9 Comparison results of the CNN-, LSTM-, and HMM-GMM-based systems for speaker BB using PLP features on the eight subsets

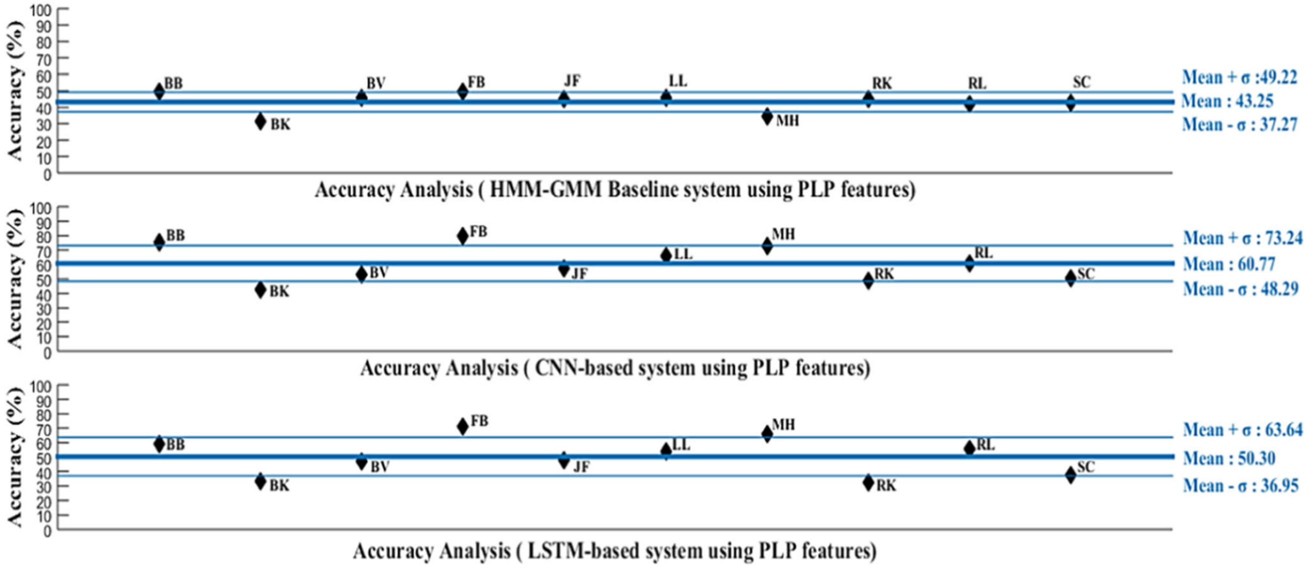


Fig. 10 Statistical analysis of the correct recognition rate of the three systems for all speakers using PLP features and the threefold middle corpus

Table 16 Configurations for CNN and LSTM models

	CNN versus LSTM	
	CNN	LSTM
Number of convolution layers	1	–
Kernel filter size (frequency × time)	112 × 112	–
Pooling size (frequency × time)	2 × 2	–
Number of filters in convolution layer	64	–
Number of LSTM units	–	160
Number of layers	4	1

In the context where there have been several studies on selecting optimal activation function, we have also investigated different types of activation functions considering their potential of improving the performance of DNN systems. ReLU is one of the most widely used activation functions because it is easier to use in CNN training and often achieves satisfactory performance. However, the obtained results showed that in the case of dysarthric speech recognition, it is recommended to replace it by the AReLU or by the polynomial functions we propose. The idea of using polynomial activation functions seems effective and thus provides the best discriminative model.

The Nemours database is used throughout this study. The quantity of samples extracted from this collection of pathological data can be considered insufficient when it comes to train deep learning networks. To face this challenge, we have opted for two strategies that have been found effective when analyzing the outcomes. The first strategy was to design the recognition systems targeted on each dysarthric speaker, which limits the use of a high number of samples in contrast to speaker-independent configurations. The second strategy consisted of using the k-fold cross-validation method to split the training and test data, which is often recommended in the case of small samples.

One of the main characteristics of dysarthria is its extreme inter-speaker variability. The general interpretation of the promising results obtained through this study leads us to conclude that a further work is required in order to perform a kind of prior classification of different varieties of dysarthria and/or their corresponding severity levels before recognition. Therefore, the acoustical and DNN parameters investigated in this study might be seen as not only the best values for recognition, but they may also be associated with a particular type of dysarthria or group of impairment varieties; that is, it may be that they are interpreted as impairment-specific. Studies of other dysarthria or speech impairment varieties are needed to follow up on these issues.

5 Conclusion and future work

In this paper, several solutions have been proposed to bring advancements in using deep learning architectures in the context of dysarthric speech recognition. Two DNN-based architectures, namely CNN and LSTM, have been implemented and compared with a statistical HMM-GMM-based system.

The first contribution made by this study pertains to the design of a robust speaker-dependent dysarthric speech recognition system using a CNN-based model using different activation functions. The performance of the ReLU, AReLU, and SELU standard activation functions has been compared to two polynomial functions Poly1ReLU and Poly2ReLU that we proposed as alternative to the conventional functions.

The CNN configuration using the proposed Poly1ReLU activation function achieved the best recognition rate of 82%, which is obtained by a speaker with a mild severity level of dysarthria. The CNN score represents an improvement of 11% and 32% when compared with the performance of LSTM- and HMM-GMM-based systems, respectively.

The second contribution consisted of presenting the results of a benchmark study that enhanced the understanding of the challenges encountered when creating optimal deep learning models of dysarthric speech. This comprehensive investigation, carried out on the Nemours publicly available dataset, may have an impact on selecting the best architecture of pathological speech processing applications in the future. These results have shown the ability of the speaker-dependent CNN architecture to deal with the most severe cases of dysarthria by capturing the relevant timing artifacts.

The third contribution provided new insights by investigating the ability of acoustic modeling using perception and hearing mechanisms of yielding more robustness to dysarthric speech recognition systems. The performance assessment of three acoustical analyzers using auditory perception modeling, namely the MFCCs, the MFSCs, and the PLP coefficients, was carried out. The results demonstrated competitive performance of the PLP analysis when used in conjunction CNNs.

The fourth contribution consisted of presenting a comprehensive investigation that advances practical knowledge related to the waveform preprocessing in order to improve the performance of deep learning dysarthric speech recognizers. This improvement was reached by optimizing the framing duration and finding the best structure of the datasets that reduces the risk of bias and overfitting.

The main challenge faced by this study remains the availability of a large amount of data needed to train deep learning algorithms in order to reach their full potential. The lack of data affects the speech recognition of impaired speech in more than one way. In the context where we are currently witnessing the change of a paradigm moving from signal processing and expert knowledge modeling into highly data-driven approaches, the mitigation of the risk linked to the lack of appropriate and representative data for training should be addressed.

In future work, we plan to implement sequence-to-sequence architectures that will benefit from transfer learning to integrate the knowledge of pretrained networks on multiple datasets to cope with the limited availability of pathological speech data. Besides this, a data augmentation approach will be used to perform further training of impairment-specific DNN-based dysarthric speech recognizers.

Funding Funding was provided by Natural Sciences and Engineering Research Council of Canada under the reference number RGPIN-2018-05221.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

References

1. Darley FL, Aronson AE, Brown JR (1969) Differential diagnostic patterns of dysarthria. J Speech Hear Res 12(2):246–269. https://doi.org/10.1044/jshr.1202.246
2. Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the 14th international conference on artificial intelligence and statistics, vol 15, pp 315–323, Lauderdale, FL, USA
3. Enderby P (2013) Disorders of communication: dysarthria. Handb Clin Neurol 110:273–281. https://doi.org/10.1016/B978-0-444-52901-5.00022-8
4. Hux K, Rankin-Erickson J, Manasse N, Lauritzen E (2000) Accuracy of three speech recognition systems: case study of dysarthric speech. Augment Altern Commun 16(3):186–196. https://doi.org/10.1080/07434610012331279044
5. Rosengren E (2000) Perceptual analysis of dysarthric speech in the enable project. J TMH-QPSR 41(1):13–18
6. Le Scaon R (2015) Projet 3A: Détection du langage d'un locuteur sur enregistrement audio.
7. Fager SK, Beukelman DR, Jakobs T, Hosom JP (2010) Evaluation of a speech recognition prototype for speakers with moderate and severe dysarthria: a preliminary report. Augment Altern Commun 26(4):267–277. https://doi.org/10.3109/07434618.2010.532508
8. Ziegler W, von Cramon D (1986) Spastic dysarthria after acquired brain injury: an acoustic study. Br J Disord Commun 21(2):173–187. https://doi.org/10.3109/13682828609012275
9. Selouani S, Sidi Yakoub M, O'Shaughnessy D (2009) Alternative speech communication system for persons with severe speech disorders. EURASIP J Adv Signal Process. https://doi.org/10.1155/2009/540409
10. Polur PD, Miller GE (2006) Investigation of an HMM/ANN hybrid structure in pattern recognition application using cepstral analysis of dysarthric (distorted) speech signals. Med Eng Phys 28(8):741–748. https://doi.org/10.1016/j.medengphy.2005.11.002
11. Hasegawa-Johnson M, Gunderson J, Perlman A, Huang T (2006) HMM-based and SVM-based recognition of the speech of talkers with spastic dysarthria. In: 2006 IEEE international conference on acoustics speech and signal processing (ICASSP), vol 3, p III, Toulouse, France. https://doi.org/10.1109/ICASSP.2006.1660840
12. Shahamiri SR, Salim SSB (2014) Artificial neural networks as speech recognisers for dysarthric speech: identifying the best-performing set of MFCC parameters and studying a speaker-independent approach. Adv Eng Inform 28(1):102–110. https://doi.org/10.1016/j.aei.2014.01.001
13. Hermans M, Schrauwen B (2013) Training and analysing deep recurrent neural networks. Adv Neural Inf Process Syst 26:190–198
14. Fager S, Bardach L, Russell S, Higginbotham J (2012) Access to augmentative and alternative communication: new technologies and clinical decision-making. J Pediatr Rehabil Med 5(1):53–61. https://doi.org/10.3233/PRM-2012-0196
15. Das D, Lee CSG (2018) Cross-Scene trajectory level intention inference using gaussian process regression and naïve registration. Department of Electrical and Computer Engineering Technical Reports, Paper 491. https://docs.lib.purdue.edu/ecetr/491/
16. Oue S, Marxer R, Rudzicz F (2015) Automatic dysfluency detection in dysarthric speech using deep belief networks. In: Proceedings of SLPAT 2015: 6th workshop on speech and language processing for assistive technologies (SLPAT), pp 60–64, Dresden, Germany. https://doi.org/10.18653/v1/W15-5111
17. Burkert P, Trier F, Afzal M Z, Dengel A, Liwicki M (2015) Dexpression: deep convolutional neural network for expression recognition. arXiv:1509.05371v1
18. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
19. Farhadipour A, Veisi H, Asgari M, Keyvanrad MA (2018) Dysarthric speaker identification with different degrees of dysarthria severity using deep belief networks. Electron Telecommun Res Inst (ETRI) J 40(5):643–652. https://doi.org/10.4218/etrij.2017-0260
20. Joy NM, Umesh S (2018) Improving acoustic models in TORGO dysarthric speech database. IEEE Trans Neural Syst Rehabil Eng 26(3):637–645. https://doi.org/10.1109/TNSRE.2018.2802914
21. Jiao Y, Tu M, Berisha V, Liss J (2018) Simulating dysarthric speech for training data augmentation in clinical speech applications. In: 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6009–6013, Calgary, AB, Canada. https://doi.org/10.1109/ICASSP.2018.8462290
22. Tu M, Berisha V, Liss J (2017) Interpretable objective assessment of dysarthric speech based on deep neural networks. In: Proceedings of the annual conference of the international speech communication association, INTERSPEECH 2017, pp 1849–1853, Stockholm, Sweden. https://doi.org/10.21437/Interspeech.2017-1222
23. Ijitona T B, Soraghan J J, Lowit A, Di-Caterina G, Yue H (2017) Automatic detection of speech disorder in dysarthria using extended speech feature extraction and neural networks classification. In: IET 3rd international conference on intelligent signal processing (ISP 2017), pp 1–6, London. https://doi.org/10.1049/cp.2017.0360
24. Chandrakala S, Rajeswari N (2017) Representation learning based speech assistive system for persons with dysarthria. IEEE

Trans Neural Syst Rehabil Eng 25(9):1510–1517. https://doi.org/10.1109/TNSRE.2016.2638830

25. Tu M, Wisler A, Berisha V, Liss JM (2016) The relationship between perceptual disturbances in dysarthric speech and automatic speech recognition performance. J Acoust Soc Am 140(5):416–422. https://doi.org/10.1121/1.4967208
26. Espana-Bonet C, Fonollosa JAR (2016) Automatic speech recognition with deep neural networks for impaired speech. In: International conference on advances in speech and language technologies for Iberian languages, IberSPEECH 2016, vol 10077. Springer, Cham, pp 97–107. https://doi.org/10.1007/978-3-319-49169-1_10
27. Nakashika T, Yoshioka T, Takiguchi T, Ariki Y, Duffner S, Garcia C (2014) Convolutive bottleneck network with dropout for dysarthric speech recognition. Trans Mach Learn Artif Intell 2(2):1–15. https://doi.org/10.14738/tmlai.22.150
28. Yılmaz E, Mitra V, Sivaraman G, Franco H (2019) Articulatory and Bottleneck features for speaker-independent ASR of dysarthric speech. Comput Speech Lang 58:319–334. https://doi.org/10.1016/j.csl.2019.05.002
29. Tripathi A, Bhosale S, Kopparapu S K (2020) A novel approach for intelligibility assessment in dysarthric subjects. In: 2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6779–6783, Barcelona, Spain. https://doi.org/10.1109/ICASSP40776.2020.9053339
30. Sak H, Senior A, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In: INTERSPEECH 2014, 15th annual conference of the international speech communication association, pp 338–342, Singapore
31. Zhang Y, Chen G, Yu D, Yaco K, Khudanpur S, Glass J (2016) Highway long short-term memory RNNS for distant speech recognition. In: 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 5755–5759, Shanghai, China. https://doi.org/10.1109/ICASSP.2016.7472780
32. Graves A, Jaitly N, Mohamed A (2013) Hybrid speech recognition with deep bidirectional LSTM. In: 2013 IEEE workshop on automatic speech recognition & understanding (ASRU), pp 273–278, Olomouc, Czech Republic. https://doi.org/10.1109/ASRU.2013.6707742
33. Eyben F, Wöllmer M, Schuller B, Graves A (2009) From speech to letters-using a novel neural network architecture for grapheme based ASR. In: 2009 IEEE workshop on automatic speech recognition and understanding (ASRU), pp. 376–380, Merano, Italy. https://doi.org/10.1109/ASRU.2009.5373257
34. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 6645–6649, Vancouver, BC, Canada. https://doi.org/10.1109/ICASSP.2013.6638947
35. Mayle A, Mou Z, Bunescu R, Mirshekarian S, Xu L, LiuC (2019) Diagnosing dysarthria with long short-term memory networks. In: INTERSPEECH 2019, pp 4514–4518, Graz, Austria. https://doi.org/10.21437/Interspeech.2019-2903
36. Bhat C, Strik H (2020) Automatic assessment of sentence-level dysarthria intelligibility using BLSTM. IEEE J Sel Top Signal Process 14(2):322–330. https://doi.org/10.1109/JSTSP.2020.2967652
37. Menendez-Pidal X, Poliko JB, Peters SM, Leonzio JE, Bunnell HT (1996) The nemours database of dysarthric speech. In*:* Proceeding of 4th international conference on spoken language processing (ICSLP '96), vol 3, pp 1962–1965, Philadelphia, PA, USA. https://doi.org/10.1109/ICSLP.1996.608020
38. Nimbalkar TS, Bogiri N (2016) A novel integrated fragmentation clustering allocation approach for promote web telemedicine database system. Int J Adv Electron Comput Sci (IJAECS) 2(2):1–11
39. Davis SB, Mermelstein P (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. IEEE Trans Acoust Speech Signal Process 28(4):357–366. https://doi.org/10.1109/TASSP.1980.1163420
40. Mohamed A, Hinton G, Penn G (2012) Understanding how deep belief networks perform acoustic modelling. In: 2012 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 4273-4276, Kyoto, Japan. https://doi.org/10.1109/ICASSP.2012.6288863
41. Hermansky H (1990) Perceptual linear predictive (PLP) analysis of speech. J Acoust Soc Am 87(4):1738–1752. https://doi.org/10.1121/1.399423
42. Zaidi B F, Selouani S, Boudraa M, Addou D, SidiYakoub M (2020) Automatic recognition system for dysarthric speech based on MFCC's, PNCC's, JITTER and SHIMMER coefficients. In: Advances in computer vision CVC 2019. Advances in intelligent systems and computing, vol 944. Springer, Cham, pp 500–510. https://doi.org/10.1007/978-3-030-17798-0_40
43. Young S, Evermann G, Gales M, Hain T, Kershaw D, Liu X, Moore G, Odell J, Ollason D, Povey D et al (1995–2015) The HTK book. Cambridge University Engineering Department
44. Alu D, Zoltan E, Stoica IC (2018) Voice based emotion recognition with convolutional neural networks for companion robots. Roman J Inf Sci Technol 20(3):222–241
45. Bhagatpatil MVV, Sardar V (2015) An automatic infants cry detection using linear frequency Cepstrum coefficients (LFCC). Int J Technol Enhanc Emerg Eng Res (IJTEEER) 3(2):29–34
46. Nair V, Hinton G E (2010) Rectified linear units improve restricted Boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML), pp 1–8.
47. Dengsheng C, Jun L, Kai X (2020) AReLU: attention-based rectified linear unit. arXiv:2006.13858v2
48. Klambauer G, Unterthiner T, Mayr A, Hochreiter S (2017) Self-normalizing neural networks. In: Advances in neural information processing systems, pp 971–980