Journal of Healthcare Informatics Research (2023) 7:104–124
https://doi.org/10.1007/s41666-023-00130-9

RESEARCH ARTICLE

Early Diagnosis of Neurodegenerative Diseases Using CNN-LSTM and Wavelet Transform

Elmira Amooei[1] · Arash Sharifi[1] · Mohammad Manthouri[2]

Received: 5 May 2022 / Revised: 9 October 2022 / Accepted: 3 February 2023 /
Published online: 13 February 2023

Abstract

Early diagnosis of neurodegenerative diseases has always been a major challenge that physicians and medical practitioners face. Therefore, using any method or device that helps with prognostics is of great importance. In recent years, deep neural networks have become popular in medical fields, and the reason is that these networks can help diagnose diseases quickly and precisely. In this research, two novel models based on a CNN-LSTM network are introduced. The main goal is to classify three neurodegenerative diseases, including ALS, Parkinson's disease, and Huntington's disease, from one another and from healthy control patients using the gait signals, which are transformed into spectrogram images. In the first model, the spectrogram images derived from the gait signals are fed into a CNN-LSTM network directly. This model achieved 99.42% accuracy. In the second model, the same input data was used to be classified using a CNN-LSTM network, which uses wavelet transform as a feature extractor before the LSTM unit. During the experiments with the second model, the detail sub-bands were eliminated one by one, and the classification results were compared. Comparing these two models has shown that using the wavelet transform and, in particular, the approximation sub-bands can result in a lighter and faster prognosis with nearly 103 times fewer training parameters overall. The classification result using only approximation sub-bands was 95.37%, using three sub-bands was 94.04%, and including all sub-bands was 94.53%, which is remarkable.

Keywords Neurodegenerative diseases · Wavelet transform · CNN-LSTM · Spectrogram · Deep neural networks

✉ Arash Sharifi
a.sharifi@srbiau.ac.ir

[1] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[2] Department of Electrical and Electronic Engineering, Shahed University, Tehran, Iran

1 Introduction

Neurodegenerative diseases (NDs) are diseases that affect the nervous system of the brain, cause the neurons and their connections to die out [1] gradually, and result in multiple issues such as speech disorders, mood swings, memory loss, and more importantly difficulty in movements of limbs or other motor imagery organs [2]. NDs are divided into some categories based on the area of the brain which is mainly affected, the age group in which the diseases are most common, and whether the disease stems from genetics or not. Some widespread examples of such diseases are Parkinson's disease (PD), Huntington's disease (HD), amyotrophic lateral sclerosis (ALS), etc.; all the mentioned NDs have two main symptoms in common when they first appear in a person, gait abnormality which generally results in walking disabilities, difficulty performing daily tasks such as eating or holding on to objects, and speech impediment [3]. Since the symptoms at an early stage of NDs are not easily distinguishable or, in some cases, even indiscernible from one another, they require multiple scans such as magnetic resonance imaging (MRI) or positron emission tomography (PET) and some invasive measures to be prognosticated. Using these traditional diagnostics usually takes a really long time and with the aid of a team of physicians can result in a final diagnostic that is not only time-consuming, but also erroneous. But machine learning (ML) techniques have been shown to be helpful in cases like this. It is because ML techniques can take a minimum amount of data, mainly derived from non-invasive medical tests such as electroencephalogram (EEG) signal of the patient as input and accurately map this data to the disease. In recent years, ML and, more specifically, deep learning (DL) techniques have become popular in the field of healthcare and have proven helpful as primary assistants in such cases to physicians. DL techniques do the classification—specifically in the case of NDs—through different kinds of data such as brain images, gait signals such as walking or standing signals, and EEG of the brain.

Time series derived from gait signals, for example, sensors attached to shoes that track walking and standing or EEG signals, are 1D time series that can be used to classify NDs. Some more popular DL methods in recent years that have proven to be precise in this case are convolutional neural nets (CNN) and long short-term memory (LSTM). These networks have the power to classify the input data using their feature extraction abilities, and that is the main reason for their popularity. CNN is used to both extract features and classify 3D input, such as MRI scans or any other images. CNN's additional strength is that it can easily handle high dimensionality of the input which is of great help especially when it comes to complex datasets. LSTM nets are recurrent neural networks (RNNs) that have the ability to work with long-term dependencies on data sequences.

In diagnostics of NDs, the data derived from early stages cannot be easily interpreted. Because the symptoms are not apparent enough yet, even if they are, they can be challenging to distinguish one from another to human eyes. Therefore, one of the best networks that work on this matter is a combination of CNN and LSTM, and that is why more and more attention is drawn to them. To boost

the ability of such networks, there are other feature extraction techniques in place, one of which is wavelet transform. Typically, this stage takes place before the data is fed into the whole CNN-LSTM. However, this can be problematic and affect the accuracy of diagnostics. But the method proposed in this paper has gone further and used wavelet between two nets, which is described in details further in this paper and solved the problem mentioned.

CNNs work with 2D (gray scale or B/W) images as well as RGB or 3D ones. Depending on the case being studied, one of these inputs can be used. In this paper, RGB images were worked with. The challenge though is to transform 1D signal into 3D image in order for CNN to be trained. There are some techniques available in this regard, one of which is spectrogram transformation which will be explained in the following sections.

The rest of this paper is organized as follows: Section 2 refers to previous works. Section 3 explains the basic concepts of data preprocessing and deep neural networks. The proposed method is presented in Section 4. The experimental results and conclusion are presented in Sections 5 and 6, respectively.

2 Related Works

Some published papers have studied different ML and DL techniques on gait datasets or MRI scans related to NDs. The application of using time series and DNNs has been explored more in recent years. Some of these works are referred to in this section. Time series are 1D input, and CNNs require 2- or 3-dimensional data; there needs to be a transformation technique to preprocess the signals into appropriate input. Different transformations have been experimented with in the following papers.

Zhao et al. [4] used Physionet [5] which includes gait signals to design a dual-channel LSTM network, one channel for force and the other for time features. They conducted three experiments on this architecture: first, to classify ND patients from HC ones, so binary classification was worked on; the second experiment is to classify the ND patients into three categories of PD, HD, and ALS; and finally the third experiment was to balance the designed network and find the optimum parameters. They have gained excellent classification results based on their innovative architecture. Since this paper conducted the proposed experiments on the same dataset, the results are later compared to the work of this, based on their accuracy, and the numbers of training parameters (which is obtained through a simulation based on Zhao's work).

Li et al. [6] also used Physionet dataset and designed a network which is called CLNet to classify the signals into four classes. They used a moving window on the gate signals to resample them and prepare them as appropriate input shape for CNN. Their CLNet is a CNN-LSTM network. CLNet was provided with resampled and normalized data. A similar normalization is used in the current paper. LSTM acted as the classifier on the CNN's extracted features. The base model is similar to model-1 of the at-hand paper. However, the two papers had different preprocessing techniques. This architecture achieved a high classification result. The results of this paper are also compared to the work of the at-hand paper.

Petrosian et al. [7] explored the ability of their designed RNN combined with wavelet transform in order to classify AD and HC patients using EEG signals. Wavelet transform (unlike the method that will be proposed in this paper) is used in the preprocessing stage and before feeding the data into the network. It is explained later why this is not the ideal preprocessing technique for the data of the at-hand paper. After several experiments on raw EEG and wavelet decomposed one, it was discovered that a 4th level wavelet on the low pass sub-bands achieved the best results. This was observed through training their RNN on each sub-bands and comparing classification results.

Oh et al. [8] took EEG signals from healthy controls and PD patients while in resting state. A 13-layer 1D CNN network was designed, and raw signal was fed into it directly in order to identify which category the data belongs to. Since the data was raw and no preprocessing was included, the CNN needed to be deep, which made the whole training costly. In a real-life application, unlike the work that will be presented in this manuscript, lots of time, RAM (random access memory), CPU, and/or GPU are needed.

Ruffini et al. [9] designed a DNN to classify HC and PD patients based on the spectrograms derived from their awake-resting stage EEG signals. Their RNN architecture consists of stacked LSTM and GRU cells. To generate the spectrograms, EEG signals were transformed using fast Fourier transform (FFT), and then a hamming window was applied. Compared to the work that will be presented in this article, similar spectrogram techniques are used on signal, although the parameters will differ.

Gaowei et al. [10] used ECG signals and designed a 1D CNN-LSTM classification model. The data was preprocessed during the acquisition, and the only preprocess conducted in this study was normalization. The data was fed into the CNN including 4 conv layers and one pooling layer. The results were then given to a fully connected layer. Two LSTM layers were placed after this layer, and the model was wrapped up using three more fully connected layers and a final softmax layer. The classification results were conducted in a binary model and an all-in model.

3 Preliminaries

In this section, some basic concepts of methods that are used in this study are explained, including the implemented preprocessing techniques, what they are and how they are used, and the architecture of CNN and LSTM and how they are modified to suit the data.

3.1 Spectrogram Transform

A variety of methods can be used in order to transform signals into images. Spectrogram [11] was chosen as a transformation method in this paper. Spectrogram is a transformation which represents the spectrum of the signal. It is a well-known transformation that is used very often, especially when gait time series are going to be trained with.

How this transformation works is that it takes the time series signals and returns the power spectra accordingly. From each spectrogram image, one can see the alterations in frequency aligned with time, so the more the energy at a particular point in time, the higher the frequency. The horizontal axis shows the times, and the vertical axis shows the frequency. The main signal for four classes is illustrated in Fig. 1, and the spectrogram images of four of the samples in this study are shown in Fig. 2. Before the transformation, the two channels of each signal are stacked. (The data which is used in this study is the gait signals acquired from 4 classes, and in-depth information on the dataset is given in Section 4.1.)

Each expanded time series has undergone the spectrogram transformation in which the window was meticulously chosen after some experimentation. Amongst

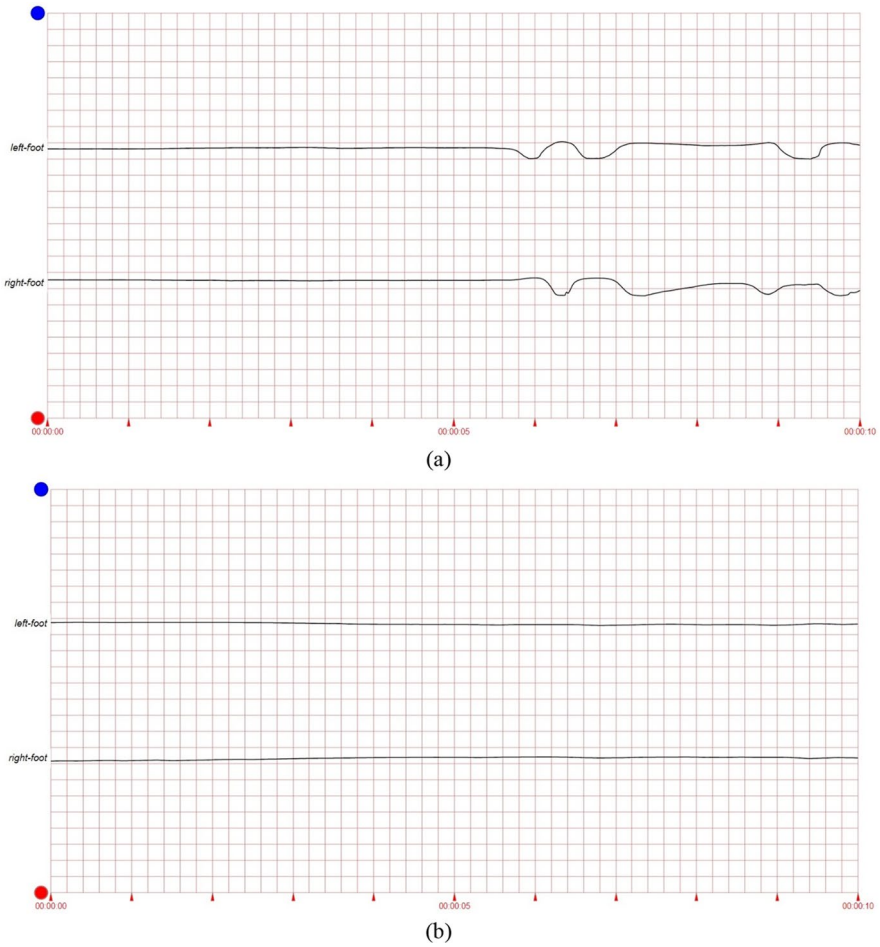
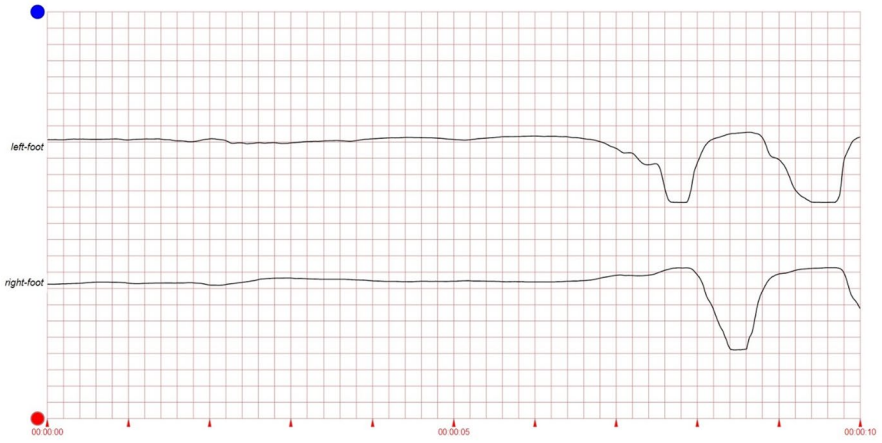


Fig. 1 Original signal from top to bottom for the first 10 s of each class. **a** ALS, **b** control, **c** HD, **d** PD [5]

(c)



(d)

Fig. 1 (continued)

Hamming [12], Hanning [13], Blackman [13], and Kaiser window [14], Kaiser window showed better results in the classification process.

The Kaiser window is defined [14] as can be seen in Eqs. (1) and (2):

$$w(n) = I_0\left(\beta\sqrt{1-\frac{4n^2}{(M-1)^2}}\right)\Big/ I_0(\beta) \tag{1}$$

where n is defined as followed:

$$-\frac{M-1}{2} \leq n \leq \frac{M-1}{2},\ \beta = 8.5 \tag{2}$$

Generally put, as the β parameter increases, the window gets narrower. To start the process, a large β was selected, and it gradually decreased until eventually $\beta = 8.5$ which is nearly as narrow as Blackman window. Furthermore, it is important

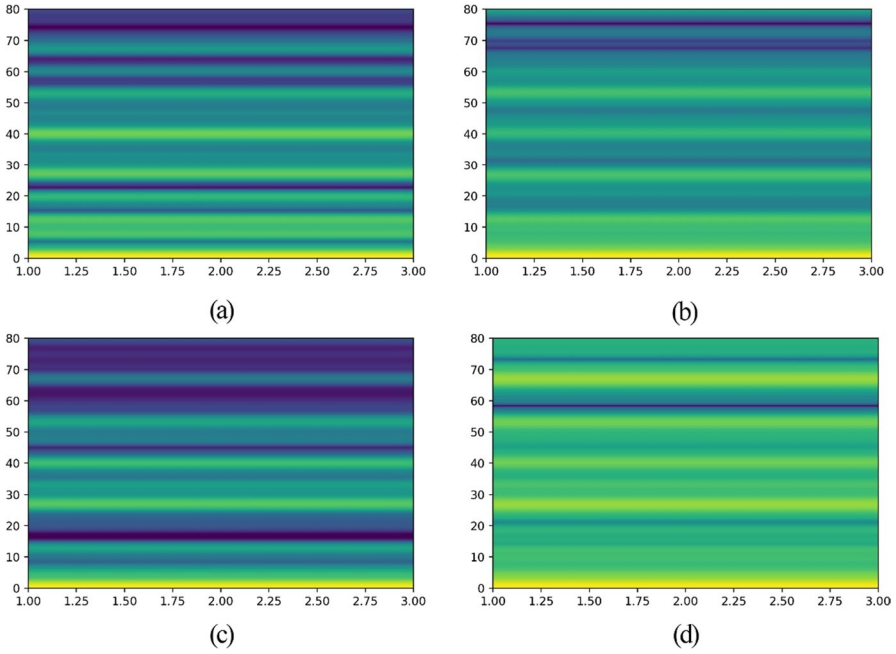


Fig. 2 Spectrogram images of the first 10 s of the first subjects of each class. **a** ALS, **b** control, **c** HD, **d** PD. (Darker colors has higher energy levels.)

to have large enough number of samples in each window because when the window gets narrower, the spikes in the signals will result in NaNs.

The spectrogram is calculated on each segment using as written in Eq. (3) [15].

$$X(T,\omega) = \int_{-\infty}^{+\infty} x(n)w(T-n)e^{-i\omega n}dn \tag{3}$$

where $w(n)$ is the window function as explained above, $x(n)$ is the signal to be transformed, ω is the frequency, and $X(T,\omega)$ is the Fourier transformation of $x(n)w(T-n)$.

3.2 Wavelet Transform

Each complex waveform can be seen as a sum of shorter and simpler waveforms. Wavelet transform [16] is a representation of a signal as a set of smaller waves called wavelets in frequency domain. Applying this transformation on images will result in sub-bands of images including the approximation and detail sub-bands. These sub-bands are obtained when the data is passed through low-pass and high-pass filter.

Mathematically speaking, a wavelet is an orthogonal function. A mother-wavelet is the general function that is multiplied by the samples in the window and results in wavelet coefficients. There are multiple mother wavelets available. The

most common family is called Daubechies [16]. The strength of this wavelet family is localization which is useful for the dataset and process used in this paper. The wavelet function is calculated recursively in Daubechies formulations, as shown in Eq. (4).

$$x(t) = \sum_{m\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} \langle x, \psi_{m,n} \rangle . \psi_{m,n}(t) \tag{4}$$

where $\psi_{m,n}$ is the mother wavelet function as is calculated as given in Eq. (5).

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a^m}} \psi(\frac{t - nb}{a^m}) \tag{5}$$

To achieve these sub-bands, discrete wavelet transform (DWT) [17] is used in this study because the nature of the deep networks that are used for the classification phase is discrete. DWT consists of these filters and divides the input signal into high and low frequency spectrum subsequences as desired. The higher the level of decomposition, the more sub-bands are derived from the input. A level 3 decomposition of a DWT is shown in Fig. 3.

The first Daubechies mother wavelet is db1. It is a basic decomposition. As the decomposition goes higher in level, the more approximation and detail subsequences are provided. In general, the approximation contains useful information of the input, so the main energy of the signal is centered in this subsequence. The detail sub-band usually includes less energy such as noise. Therefore, by eliminating the detail sub-band, the original input can be reconstructed using the approximation or high sub-band. The reconstructed input in this manner loses the least energy compared to its original form. As a result, if one was to reconstruct the input using only the approximation sub-bands, the reconstructed input could illustrate either nearly the original input or an input so similar to the original that the lost energy could be neglected.

3.3 CNN Architecture

CNN [18], also known as ConvNet, is a Deep Neural Net that is made of an input layer, at least one hidden layer, and an output layer. The input of each 2D convolutional layer

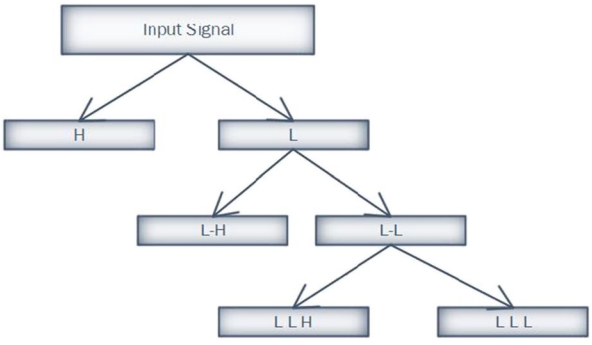


Fig. 3 Discrete wavelet transform decomposition to 3 levels

which is used to classify image data is a 4D tensor as follows: the first dimension is the number of batches, the second dimension is image height, the third one is image width, and the last dimension is image color channels.

Each convolutional cell is a moving window known as filter or kernel. This filter moves through the image starting from the top left corner and calculates the convolution of the contents and then slides one pixel towards the right all the way to the bottom right corner of the image. During this process, the features with less significance reach nearly zero since the convolution gets lower and lower. The result of the convolution calculation is shown in Eqs. (6) and (7):

$$x_j^l = f\left(\sum_{i \in M_j} x_j^{l-1} * k_{ij}^l + b_j^l\right) \tag{6}$$

$$f(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

where x is the output of the convolution, k stands for kernel, l is the number of output layers, i is how many stride the kernel moves every step, the feature map is M, the bias is b, and finally f is the activation function.

After the convolutional layer, pooling layer is normally placed. The pooling layer down-samples the feature map in order to reduce the dimensions. This layer removes the values which are nearly zeros because these features do not play an important role in the feature selection process. Therefore, the output of every pooling layer is the most important feature. Obviously, based on how this works, the dimensionality of the input also decreases. The output of this layer is shown in Eq. (8):

$$x_j^l = f\left(\beta_j^l downsample\left(x_j^{l-1}\right) + b_j^l\right) \tag{8}$$

where x is the pooling layer's output, the function of the pooling layer is *downsample* and its bias is β, and the activation function and the bias are f and b, respectively. A general representation of CNN model is presented in Fig. 4.

3.4 LSTM Architecture

Each LSTM [19] cell includes a memory cell, an input gait, a forget gait, and an output gait as can be seen in Fig. 5.

The input at time t is let in through the input gait. The weights for this input are then updated based on the input at time $t-1$ which is stored in the memory cell. Equations (9) and (10) show how it is calculated in the forward pass phase.

$$\bar{i}^t = W_i x^t + R_i y^{t-1} + p_i \odot C^{t-1} + b_i \tag{9}$$

$$i^t = \sigma(\bar{i}^t) \tag{10}$$

The forget gate decides if the data at time t needs to be forgotten and discarded or to be sent out through the output gait. This is done through Eqs. (11) and (12) [11]:

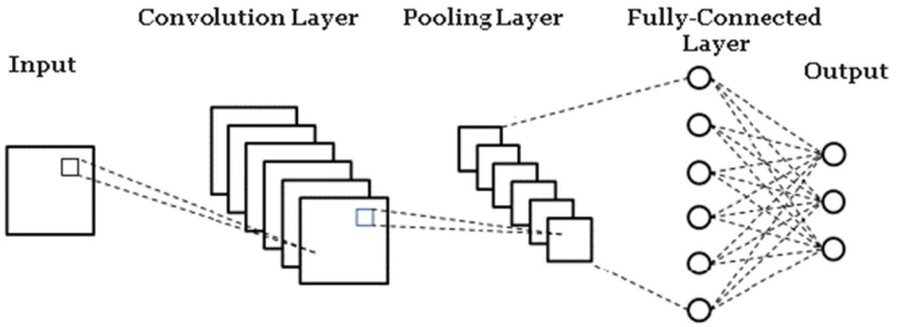


Fig. 4 Basic CNN layers

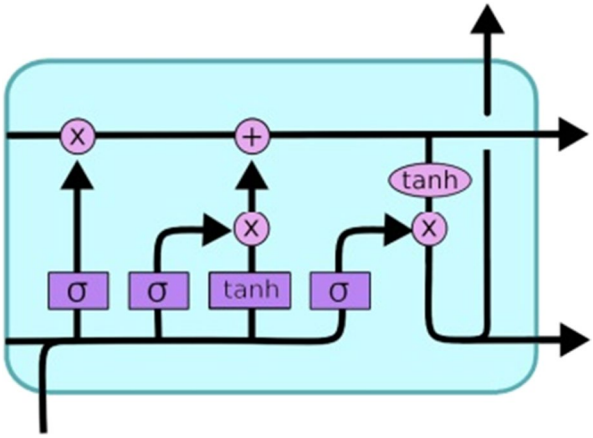


Fig. 5 Basic LSMT cell structure [20]

$$\vec{f}^t = W_f x^t + R_f y^{t-1} + p_f \odot C^{t-1} + b_f \tag{11}$$

$$f^t = \sigma(\vec{f}^t) \tag{12}$$

The output gait decides what should be filtered as an output based on the state of the LSTM cell. Based on Eqs. (13) and (14), this will be decided.

$$\vec{o}^t = W_o x^t + R_o y^{t-1} + p_o \odot C^t + b_o \tag{13}$$

$$o^t = \sigma(\vec{o}^t) \tag{14}$$

In Eqs. (9) to (14):

- $W_i, W_f, W_o \in \mathbb{R}^{N*M}$ are the input weights.
- $R_i, R_f, R_o \in \mathbb{R}^{N*M}$ are the recurrent weights.
- $p_i, p_f, p_o \in \mathbb{R}^N$ are the peephole Wight.

- $b_i, b_f, b_o \in \mathbb{R}^N$ are the biases.

3.5 CNN-LSTM Architecture

A CNN-LSTM network [21] is a combination of CNN and LSTM cells. It is mainly used for image or video processing. It has the advantages of feature extraction of CNN and the ability to handle long-term dependencies of LSTM. In this study, CNN-LSTM architecture is used in 2 models. In both models, the CNN acts as the feature extractor, and the LSTM has the role of the classifier. The difference between the two models is that in the second model, an additional feature extractor, the wavelet transform, is used in place of the two. This whole architecture is explained in detail in Section 4.

As defined earlier in this section, CNN extracts the features from input images; the output is a feature map matrix. When it is fed into the LSMT, the time factor (which is essential in this type of disease classification as the gait signals demonstrate the type of disease each subject is suffering from) is also considered in the classification. Therefore, the classification is much more accurate.

4 Proposed Method

In this section, the proposed method is explained. First, the data which is used in this study is described, and then preprocessing and how it is done is defined. Afterwards, the two models are explained.

4.1 Data Acquirement

The Physionet dataset, which is used in this study, includes 64 gait-signal recordings acquired from 4 groups of people: 15 Parkinson patients, 20 Huntington patients, 13 ALS patients, and 16 healthy control people. The subjects did not have any other motor imagery diseases affecting their walking ability. They were asked to walk on a flat surface of a 77-m distance wearing shoes that had force-sensitive sensors installed on the soles, for approximately 5 min or 300 s. The first 50 s of each signal are removed due to acquisition artefacts and multiple zeros. The time series acquired included 13 features as described in Table 1. The first 2 min of one of ALS samples are illustrated in Fig. 6.

There are uneven time series recordings, so some signals are a few seconds longer than the rest, but this was not an issue in this paper. As soon as the sole of the foot touches the floor, the sensor is activated, and the pressure is measured. As soon as the sole detaches from the ground, the sensor is deactivated. The activated and deactivated durations are recorded and are known as stance and swing, respectively. Stride is the stance and swing times added together. Double-support refers to the amount of time that both soles of the feet are touching the ground. The subjects are chosen using clinical questionnaires. They did not use any aids, such as a wheelchair, to perform their daily

Table 1 Physionet data description based on the features [5]

COLUMN	CONTENT
1	Elapsed time (sec)
2	Left stride interval (sec)
3	Right stride interval (sec)
4	Left swing interval (sec)
5	Right swing interval (sec)
6	Left swing interval (% of stride)
7	Right swing interval (% of stride)
8	Left stance interval (sec)
9	Right stance interval (sec)
10	Left stance interval (% of stride)
11	Right stance interval (% of stride)
12	Double support interval (sec)
13	Double support interval (% of stride)

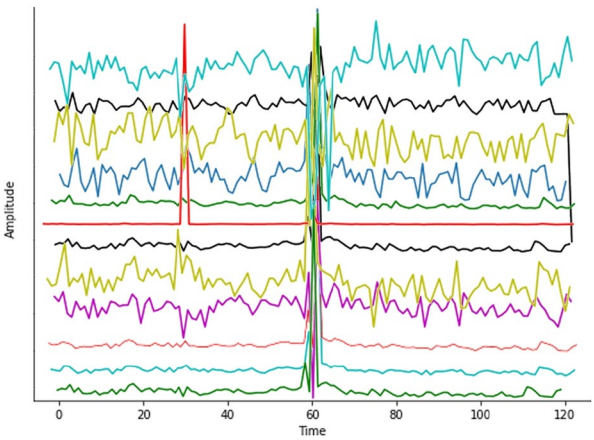


Fig. 6 Representation of one of the asl case's gait signal with 12 features in 120 s. (From bottom to top: left stride interval (sec), double support interval (sec), left stance interval (% of stride), left swing interval (% of stride), right stance interval (sec), left swing interval (sec), right stride interval (sec), right stance interval (% of stride), double support interval (% of stride), right swing interval (sec), left stance interval (sec), right swing interval (% of stride))

tasks outside of the experiment. A brief description of the subjects is given in Table 2.

4.2 Data Preprocessing

One basic and crucial step in every machine learning problem-solving is preprocessing the data. This step is done to ensure that the data is normalized and noise-free so that the maximum performance can be achieved by the models. Preprocessing also helps to transform the data into an acceptable format for the training models. Furthermore, if the number of training samples is lower than needed, some data augmentation techniques can be applied to the input. All this is explained in the following sections.

Table 2 Subject statistics of Physionet dataset [5]

Statistical parameters	CO (Mean ± STD)	ALS	HD	PD
Age (year)	39.3 ± 18.5	55.6 ± 12.8	47.4 ± 12.5	66.8 ± 10.9
Height (m)	1.83 ± 0.08	1.74 ± 0.10	1.84 ± 0.09	1.87 ± 0.15
Wight (kg)	66.81 ± 11.08	77.11 ± 21.15	73.47 ± 16.23	75.07 ± 16.9
Gait speed	1.35 ± 0.16	1.05 ± 0.22	1.15 ± 0.35	1.0 ± 0.2
Disease severity	0	18.3 ± 17.8	6.79 ± 3.9	2.8 ± 0.86

4.2.1 Normalization and Data Augmentation

The first step of the preprocessing was to normalize the data using MinMaxScaler. Normalization helps with achieving uniformity that leads to having accurate classification results. Note that the outliers are filled using median filter after acquisition. Therefore, the dataset is clean enough to be standardized.

MinMaxScaler is a standard transformation that takes each individual feature and transforms it into a given range which by default is between zero and one. The default range is used in this paper. The calculation is shown in Eq. 15.

$$X_{std} = \left(\left(x - X_{min}\right) \Big/ \left(X_{max} - X_{min}\right) \right) \tag{*15}$$

where x is the data point that is going to be normalized. The range is decided based on X_{min} and X_{max} since they are the minimum and maximum on the matrix, respectively.

The next step in preprocessing the dataset is to do data augmentation since there are not enough data samples available and DNNs are trained better on large amounts of data samples. Therefore, a moving window is designed which is slid onto each original sample; with the help of spectrogram transform, the data inside this window is transformed into spectrum RGB image. As a result, the one-dimensional time series is converted into a three-dimensional image that the CNN can easily work with. Then the window is slid for 1 s, and the process repeats. The moving window is demonstrated in Fig. 7.

4.2.2 Spectrogram Transform

After the dataset was normalized, data augmentation was needed as mentioned earlier. As explained in Section 3.1, Kaiser window was chosen as the transformation window. Initially, 64 data samples were available with a left and a right channel, through the moving window used as the data augmentation, 7271 samples were produced. The spectrogram window had 50% overlapping samples which is generally the best portion. The RGB images which were stored had same height and length equal to 32 pixels. All the parameters of spectrogram transform were chosen after several trials and comparisons amongst the classification results.

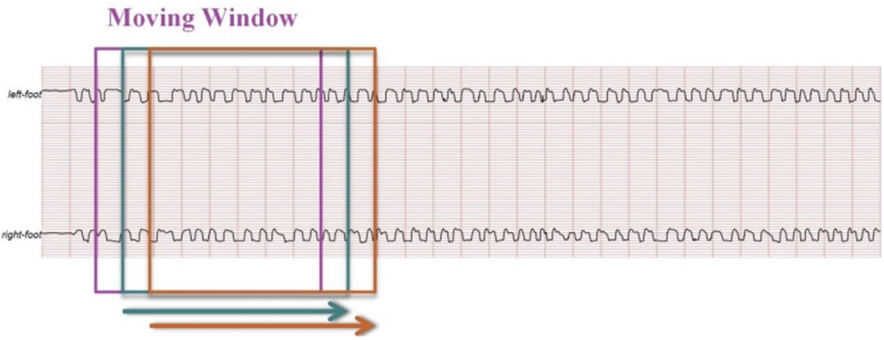


Fig. 7 The moving window defined in order to do data augmentation

4.3 Proposed Models

This section is dedicated to two main models studied in this paper and how they are used. The first model which is called model-1 is a basic CNN-LSTM which is tuned for the preprocessed data. The second model called model-2 is a state-of-the-art CNN-LSTM model which is coupled with wavelet transform. How they work is mentioned in Sections 4.3.1 and 4.3.2, respectively.

4.3.1 Model-1:+65+36 Pure CNN-LSTM

In the first experiment, the spectrogram images were fed into a CNN-LSTM network. The general structure of the network can be seen in Fig. 8.

The input data here was a 3-dimensional RGB image with a width and height of 32 pixels. These input images are fed into the layers of CNN that act as the feature extractor.

Three convolution layers were put in place. The output of this section was flattened and then fed into the LSTM unit which included 2 LSTM cells. Afterwards, the batch normalization layer that works as a standardization layer and normalizes the mini-batches from the previous layer was set. The way this layer helps with the whole process is that fewer training epochs will be needed as mini-batches are standardized. It is often recommended not to use batch normalization, and dropout layer is the same network architecture, since the dropout layer drops out the nodes randomly, and the statistics (standard deviation and mean) used in the normalization layer may become noisy when randomly removed. However, experimentation revealed that this was not the case in this model and with this preprocessed data, as removing either one resulted in lower results when compared to having them in the specified order. Finally, the fully connected layer decides the class where each input belongs to. The structure of the model can be seen in Table 3, and the results of this scenario can be found in Table 5.

4.3.2 Model-2: CNN-LSTM Coupled with Wavelet Transform

During the second experiment which was the main goal of this article, the same architecture was initially used as illustrated in Fig. 9. The main difference is that

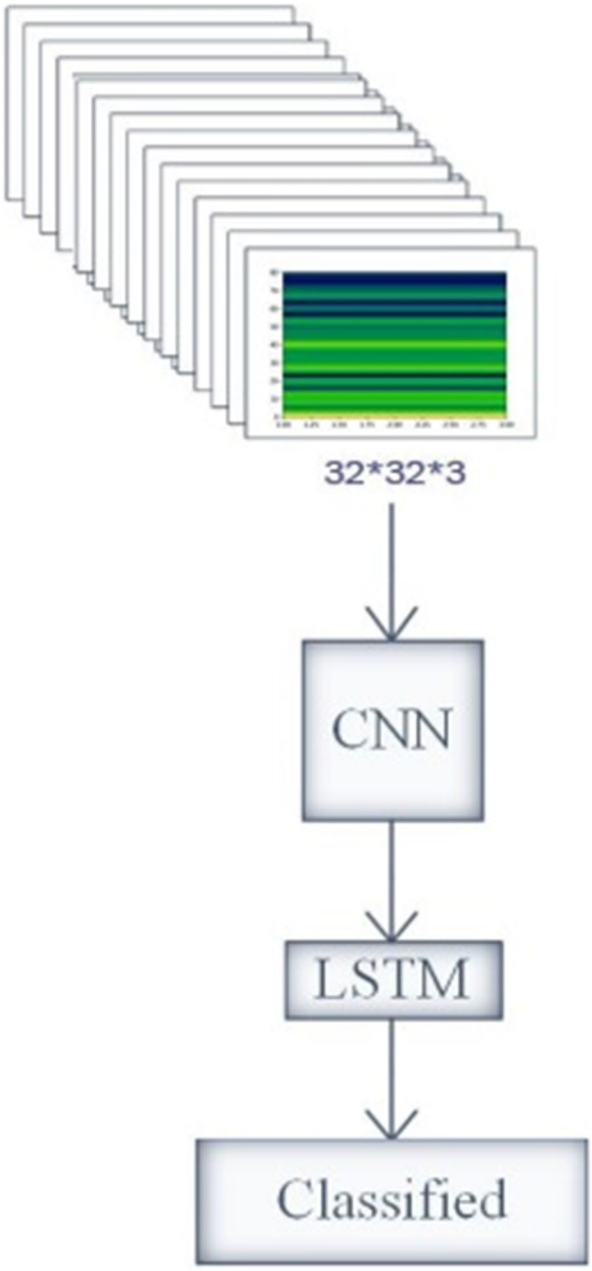


Fig. 8 CNN-LSTM architecture used in model-1

after the convolutional layers, wavelet transform is utilized to act as a feature extraction unit before the classification unit consisting of parallel LSTM cells.

As described in Section 4.3.2, DWT is used to decompose the given input into its sub-bands. Various Daubechies mother wavelets were tested during this experiment

Table 3 Model-1's structure

-	PARAMETERS
Number of conv layers	5
Pooling layer	MaxPooling2D
Optimizer	SGD
Early stop patience	8
Number of LSTM layers	2
Dropout rate	0.5

as described priorly. The experiment included three trials as mentioned earlier; in the first run, all 4 wavelet sub-bands were used (model-2A); in the second run, the high-high sub-band was eliminated, and the experiment was conducted again (model-2B), and finally, all high sub-bands were removed, and the experiment was done using only the approximation bands (model-2C). In all of these trials, 4 mother wavelets were involved, and the results of these 12 experiments were compared.

The major difference between the method proposed in this paper and all the other work done so far is that normally the wavelet transform is used as a pre-processing step before the data is fed into the network. However, in this paper, wavelet is used between the two networks. Therefore, the main data is given to CNN to extract the features first. And then the wavelet transform is placed

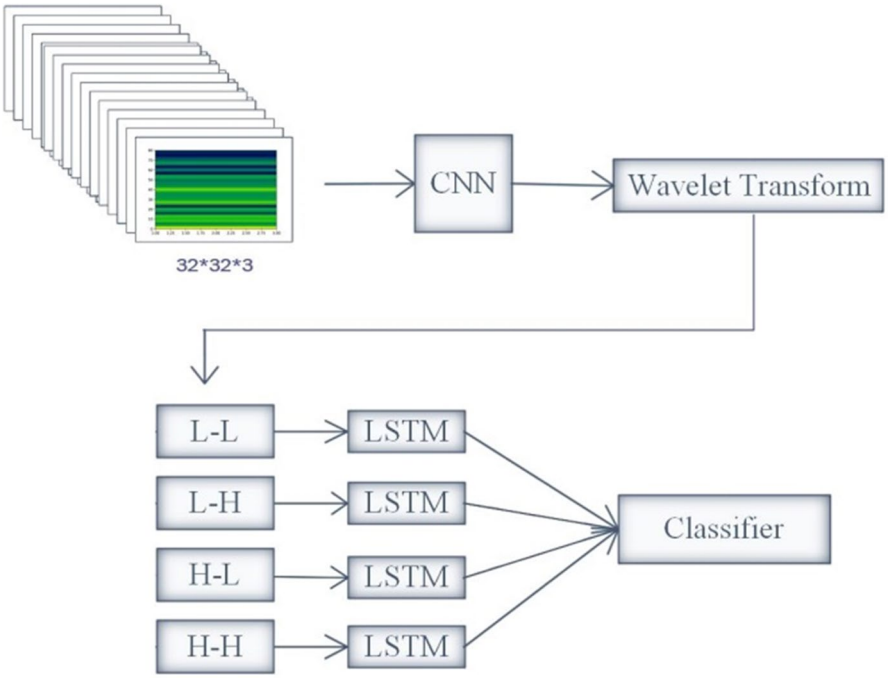


Fig. 9 CNN-LSTM used in model-2 processing the wavelet decomposed images

after CNN, so the output of the CNN was wavelet's input, and different kinds of features were extracted; this is absolutely different from the previous studies because when the wavelet is used in the preprocessing stage, the quality and definition of the initial images which matters the most when working with medical subjects and data drop. Since the slightest details matter in such images, in this study, the wavelet was used after CNN because the output of CNN is a feature map, and what wavelet does is to select the more important features from this map which helps to reduce the number of learning parameters. This results in a lighter-weight architecture and, therefore, a faster prognosis. The classification results in this manner are trustworthy and can be applied to real-life cases as the actual features that weigh the most are selected from the input data. The structure of this model can be seen in Table 4.

5 Simulation Results

This section contains the results of this method and architecture. A brief description of how the models are evaluated is given. Then the results each model obtained are mentioned in Tables 4 and 5.

Standard evaluation criteria are used as measures including accuracy, precision, and F1-score. The accuracy determines if the deep model is able to correctly discern the class each sample belongs to or not. Precision determines the number of related samples amongst all the retrieved samples. F-measure or F1 score returns a number showing the harmonic mean between precision and recall.

After each training epoch, the accuracy, precision, and recall of the training and validation models were stored, and the best model was chosen based on the highest validation accuracy. The training was halted when the validation accuracy did not improve after 8 consecutive epochs using early stop.

In the first model, the data is first preprocessed as described earlier. It was given to CNN-LSTM afterwards. This serialized classification resulted in remarkable classification outcome considering the number of training parameters, which was the least possible. As a measure, some pre-trained networks with added LSTM layers were trained on the same data, and the test results are compared in Table 5.

Table 4 Model-2's structure

-	PARAMETERS
Number of conv layers	5
Pooling layer	MaxPooling2D
Optimizer	SGD
Early stop patience	8
Wavelet type	DWT
Mother wavelets	Db2, db4, db8, db12
Number of LSTM layers	2
Dropout rate	0.5

Table 5 CNN-LSTM model-1 results in comparison with 3 pre-trained networks with added LSTM layers

NETWORK	#TRAINABLE pARAMETERS	ACCURACY (%)	F1
AlexNet-LSTM	62,391,276	99.68%	0.9967
Dense121-LSMT	8,647,416	80.11%	0.6025
MobileNet-LSTM	5,882,232	81.15%	0.7650
Model-1 CNN-LSTM (proposed method)	250,100	99.42%	0.9948

The results of the second model can be seen in Table 6. In this table, the first trial called model-2A which had all wavelet sub-bands included in the experiment, the second trial called model-2B with the H–H sub-band removed, and model-3C which is the third trial with only the approximation layers can be seen.

Four mother wavelets, db2, db4, db8, and db16, were tested. DWT level 2 was chosen as the best result was given based on all the trials and errors. The experiment was conducted three times. Firstly, all 4 sub-bands were used in the classification process. Each sub-band was given to an LSTM unit. Classification was done separately for each sub-band. Then the results of the LSTM units were concatenated in order to assign the given input to its predicted class.

Based on the results shown in Table 6, the high-high sub-band was removed next, and the experiments were conducted again. There were fewer training parameters expected here. Three sub-bands (L-L, L–H, and H–L) were fed into three LSTM cells, and the final classification result was concatenated. The idea behind this was that the H–H sub-band will probably include the lowest energy; therefore, better classification results based on the kind of the mother wavelet can be obtained. This process was also done using all the listed wavelets. The results are given in Table 6.

Afterwards, the H–L sub-band was also removed, and the experiment was conducted using only the approximation sub-bands and two LSTM cells. In this phase, the minimum training parameter and fastest training pace were expected. The results proved to be remarkable since the accuracy and F1 measure improved in comparison with the other experiments. Furthermore, the number of training

Table 6 Results of the model-2 using all 3 trials (model-2A, model-2B, and model-2C)

MOTHER wAVELET	MODEL-2A		MODEL-2B		MODEL-2C	
	#Trainable parameters 19,764		#Trainable parameters 15,780		#Trainable parameters 13,892	
	Accuracy (%)	F1	Accuracy (%)	F1	Accuracy (%)	F1
db2	83.20	0.7720	89.07	0.7516	91.06	0.8633
db4	94.53	0.9351	92.75	0.9277	95.37	0.9530
db8	92.30	0.9218	94.04	0.9383	95.01	0.9269
db16	76.78	0.6443	79.97	0.6844	82.92	0.7584

parameters drastically decreased as expected. Consequently, the learning process was completed faster on average, compared to the other wavelets and trials.

When this step was completed successfully, the goal was to meticulously alter the hyper-parameters in order to achieve the best results and have the lowest number of training parameters and the shortest training process at the same time.

The results of similar papers that used the Physionet dataset and deployed different preprocessing and deep architectures are compared with the proposed method in this paper. Note that the number of training parameters was not specified in the original papers and the numbers reported in Table 7 are an approximation gained from a simulation that was conducted based on their proposed models using our preprocessed database.

Taking the number of training parameters in mind, the accuracy and F1 score that this paper's proposed model achieved in comparison with other proposed models are pretty promising. Amongst all three results of the second model including model-2A, model-2B, and model-2C, the least training parameters and the best F1 score and accuracy were achieved when mother wavelet db4 with only the approximation sub-bands were trained with. This model can be chosen as the best one.

Having the numbers obtained from Table 6, it can be seen that the proposed model of this paper and its chosen model (model-2C) have the least number of training parameters of all other models proposed by other papers on the same dataset while having nearly the same accuracy. Considering the fact that the proposed model, as explained in Section 4.3.2, can be used in real-life cases to prognose NDs.

6 Discussion

In this paper, the main goal was to be able to do prognostics on ND patients in a way that needed the simplest form of data and cheap computing power. As presented in Table 7, this goal has been achieved. The main problem with prognostics is data because physicians need it from different machines. This study takes only gait signals and does the classification on that. The main problem faced in this area was the lack of sufficient data. Therefore, data augmentation was used. For sure, using more real-life data can improve the whole technique. The results demonstrated that this method can outperform other methods that used the same dataset with different CNN or LSTM architectures. Therefore, using the wavelet transform

Table 7 Final results of the second scenario in comparison with other studies on the same dataset

METHOD	ACCURACY (%)	F1	# PARAMETERS
CLNet [4]	99.50	0.99	1,352,436
Dual channel LSTM [3]	95.67	-	399,364
Proposed method (model-1)	99.42	0.9948	20,100
Proposed method (model-2A, mother wavelet db4)	94.53	0.9351	19,764
Proposed method (model-2B, mother wavelet db8)	94.04	0.9383	15,780
Proposed method (model-2C, mother wavelet db4)	95.37	0.9530	13,892

as a feature extractor after CNN layers was performing well as expected. What is more, using wavelet did not have the neglecting effect of losing valuable features like when it was used before CNN. The results show that the number of training parameters decreased greatly. It shows that the computational power that is needed can be minimized to a minimum. This supports the idea that this method can be applied as a reliable prognostic technique in real life. However, the main loophole that remains is the data. Further suggestions on this are made in Section 7.

7 Conclusion and Future Works

This paper presented a novel diagnostics technique based on two CNN-LSTM networks. The first model was a pure CNN-LSTM, and the second model was a CNN-LSTM coupled with wavelet transform to diagnose neurodegenerative diseases in early stages of the disease from gait signals obtained from force sensors that can be placed in the subjects' shoes. The importance of such methods in healthcare is apparent. The novelty of this method in AI is its significance, as wavelet transform has never before been used as a feature extractor between the two CNN and LSTM networks to classify NDs.

Results show that our proposed method can achieve remarkably accurate results on a light-weight DN because of its ability to learn the patterns of each disease in early stages where a physician might have some trouble diagnosing correctly, because of the lack of data, diagnostic equipment, and lack of accuracy in available equipment. It is shown that through the combination of CNN-LSTM and wavelet transform, classification can take place accurately. The other very important aspect is that using this model, we have been able to achieve a high accuracy without needing the usual expensive resources that ML techniques need, such as large RAMs or costly CPUs and GPUs. This also helped with the speed at which the model can do the diagnostics.

To improve this proposed method, other transformations can be used such as scalogram, in order to transform gait signals into spectrum images. Furthermore, other data augmentation methods can aid in improving the accuracy as neural nets are trained best on large numbers of datasets.

Declarations

Conflict of Interest The authors declare no competing interests.

References

1. Galimberti D, Scarpini E. (2018) Neurodegenerative diseases: clinical aspects, molecular genetics and biomarkers, Springer International Publishing, 179–182
2. Hausdorff JM, Lertratanakul A, Cudkowicz ME, Peterson AL, Kaliton D, Goldberger AL. (1985) Dynamic markers of altered gait rhythm in amyotrophic lateral sclerosis. J Appl Physiol. https://doi.org/10.1152/jappl.2000.88.6.2045

3. Hausdorff JM, Mitchell SL, Firtion R, Peng CK, Cudkowicz ME, Wei JY, Goldberger AL. (1985) Altered fractal dynamics of gait: reduced stride-interval correlations with aging and Huntington's disease. J Appl Physiol. https://doi.org/10.1152/jappl.1997.82.1.262
4. Zhao A, Qi L, Dong J, Yu H. (2018) Dual channel LSTM based multi-feature extraction in gait for diagnosis of neurodegenerative diseases. Knowledge-Based Systems. https://doi.org/10.1016/j.knosys.2018.01.004
5. Goldberger AL, Amaral LA, Glass L, Hausdorff JM, Ivanov PC, Mark RG, Mietus JE, Moody GB, Peng CK, Stanley HE. (2000) PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation. https://doi.org/10.1161/01.cir.101.23.e215
6. Ning Z, Li L, Jin X. (2018) "Classification of neurodegenerative diseases based on CNN and LSTM," in 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 82–85. https://doi.org/10.1109/ITME.2018.00029
7. Petrosian AA, Prokhorov DV, Lajara-Nanson W, Schiffer RB. (2001) Recurrent neural network-based approach for early recognition of Alzheimer's disease in EEG. Clin Neurophysiol. 1378–87. https://doi.org/10.1016/s1388-2457(01)00579-x
8. Oh SL, Hagiwara Y, Raghavendra U. (2020) A deep learning approach for Parkinson's disease diagnosis from EEG signals, Neural Comput & Applic. 10927–10933. https://doi.org/10.1007/s00521-018-3689-5
9. Ruffini G, Ibañez D, Castellano M, Dubreuil-Vall L, Soria-Frisch A, Postuma R, Gagnon JF, Montplaisir J. (2019) Deep learning with EEG spectrograms in rapid eye movement behavior disorder, Frontiers in Neurology. https://doi.org/10.3389/fneur.2019.00806
10. Xu G, Ren T, Chen Y, Che W. (2020) (A one-dimensional CNN-LSTM model for epileptic seizure recognition using EEG signal analysis, Frontiers in Neuroscience. https://doi.org/10.3389/fnins.2020.578126
11. Ramos-Aguilar R, Olvera-López JA, Olmos-Pineda I.(2017) Analysis of EEG signal processing techniques based on spectrograms, Research in Computing Science. 151–162. https://doi.org/10.13053/RCS-145-1-12
12. Blackman RB, Tukey JW. (1958) The measurement of power spectra from the Point of View of Communications Engineering — Part I, Dover Publications. 185–282. https://doi.org/10.1002/j.1538-7305.1958.tb03874.x
13. Blackman RB, Tukey JW. (1958) The measurement of power spectra from the Point of View of Communications Engineering — Part II, Dover Publications,485–569. https://doi.org/10.1002/j.1538-7305.1958.tb01530.x
14. Kaiser JF (1966) Digital Filters" – Ch 7 in "Systems analysis by digital computer. John Wiley and Sons, New York, pp 218–285
15. Sejdic E, Djurovic I, Jiang J.(2009) Time–frequency feature representation using energy concentration: an overview of recent advances. Digital Signal Processing. 153–183. https://doi.org/10.1016/j.dsp.2007.12.004
16. Daubechies I. (1990) "The wavelet transform, time-frequency localization and signal analysis," in IEEE Transactions on Information Theory. 961–1005. https://doi.org/10.1109/18.57199
17. Chui C, Heil C. (1992) An introduction to wavelets, computers in physics. 6. 697-. https://doi.org/10.1063/1.4823126
18. Valueva MV, Nagornov NN, Lyakhov PA, Valuev GV, Chervyakov NI (2020) Application of the residue number system to reduce hardware costs of the convolutional neural network implementation, Mathematics and Computers in Simulation. 232–243. https://doi.org/10.1016/j.matcom.2020.04.031
19. Hochreiter S, Schmidhuber J. (1997) Long short-term memory. Neural Comput. 1735–1780. 10.1162 / neco.1997.9.8.1735
20. Gers F.: Understanding LSTM Networks. https://colah.github.io/posts/2015-08-Understanding-LSTMs (2015). Accessed January 2020.
21. Sainath T, Vinyals O, Senior A, Sak H (2015) Convolutional, Long short-term memory, fully connected deep neural networks, Computer Science 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP.2015.7178838