Computers in Biology and Medicine 115 (2019) 103477

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: http://www.elsevier.com/locate/compbiomed

Bag of Samplings for computer-assisted Parkinson's disease diagnosis based on Recurrent Neural Networks

Luiz C.F. Ribeiro[a], Luis C.S. Afonso[b], João P. Papa[a,*]

[a] *UNESP - São Paulo State University, School of Sciences, Brazil*
[b] *UFSCar - Federal University of São Carlos, Department of Computing, Brazil*

ARTICLE INFO

Keywords:
Parkinson's disease
Handwritten dynamics
Recurrent Neural Networks
Bag of samplings

ABSTRACT

Parkinson's Disease (PD) is a clinical syndrome that affects millions of people worldwide. Although considered as a non-lethal disease, PD shortens the life expectancy of the patients. Many studies have been dedicated to evaluating methods for early-stage PD detection, which includes machine learning techniques that employ, in most cases, motor dysfunctions, such as tremor. This work explores the time dependency in tremor signals collected from handwriting exams. To learn such temporal information, we propose a model based on Bidirectional Gated Recurrent Units along with an attention mechanism. We also introduce the concept of "Bag of Samplings" that computes multiple compact representations of the signals. Experimental results have shown the proposed model is a promising technique with results comparable to some state-of-the-art approaches in the literature.

1. Introduction

Parkinson's Disease (PD) is a clinical syndrome first described by James Parkinson in 1817. Years later, it has been discovered that patients with PD lose cells in the *substantia nigra* and have lower concentrations of dopamine (i.e., neurotransmitter), which plays an essential role in communication among cells. Jankovic [14] mentions the four cardinal features of Parkinson's Disease: tremor at rest, rigidity, akinesia (or bradykinesia), and postural instability, which figure among the motor impairments. However, PD is also characterized by non-motor dysfunctions, which include autonomic dysfunction, cognitive/neurobehavioral disorders, and sensory and sleep abnormalities. Bhat et al. [4] also state that anxiety, depression, fatigue, and sleep disorders are observed prior to the diagnosis of PD.

Although considered as a non-lethal disease, PD shortens the life expectancy of the patients. The diagnose is usually performed using a clinical exam and by a neurologist with expertise in movement analysis. Many works focused on the studies of the motor signs (e.g., freezing of gait, posture analysis, or tremor) to aid the early diagnoses, which are collected by sensors worn by the patient. Ornelas-Vences et al. [18] proposed a fuzzy inference model based on the examiners knowledge for turning rate based on four biomechanical features extracted from sensors worn on the lower limbs. Samá et al. [25] proposed to detect and rate bradykinetic gait through a waist-worn sensor, and MashhadiMalek et al. [17] evaluated the inter-relation of tremor and rigidity in Parkinson's disease.

Tremor is considered as the most common disorder in PD patients and has been one of the most explored characteristics in the literature due to the challenge in capturing subtle tremor features manually. Among the many applications, one can find the work proposed by Rigas et al. [24], which assesses both action and resting tremors based on data collected from accelerometers. The authors employ two parallel Hidden Markov Models to quantify the severity, body posture, and action. Abdulhay et al. [2] explored tremor and gait features acquired during deep brain stimulation. The data is obtained from sensors placed underneath the patient's feet and at the forefinger. Further, they feed machine learning techniques to the automatic PD diagnosis. Deep learning techniques have also been considered in the context of PD diagnosis. The work proposed by Kim et al. [15] differentiate the severity of symptoms using convolutional neural networks.

Visual features were also studied by a few works that used data collected from handwriting exams. Pereira et al. [21,22] applied a handwriting exam comprised of tasks supposed to be nontrivial for PD patients, where they were asked to draw spirals and meanders over a guideline. The drawings were compared against templates, and the differences between them (i.e., the tremors) provide information to feed

[*] Corresponding author.
E-mail address: joao.papa@unesp.br (J.P. Papa).

https://doi.org/10.1016/j.compbiomed.2019.103477
Received 17 August 2019; Received in revised form 26 September 2019; Accepted 29 September 2019
Available online 4 October 2019

machine learning techniques to distinguish the individual as either healthy or PD patient. In a later work, Pereira et al. [20,23] drove their studies to a signal-based approach, which used data collected by sensors during a handwriting exam. A Convolutional Neural Network (CNN) was employed to learn features from image representations of the signals. The images are computed using visual rhythms and generated in different resolutions. The approach outperformed the image-based ones proposed in earlier works [21,22], despite the losses during the mapping from signal to image. Later on, Afonso et al. [3] extended the work proposed by Pereira et al. [23] by mapping the signals into images using recurrence plot technique to further feed a CNN.

A complementary approach to this task consists in exploring raw signals obtained from exams through Recurrent Neural Networks (RNNs). These formulations were designed to explore time dependencies, which are a characteristic inherent to the problem. Therefore, they turn out to be able to learn robust patterns for PD identification. Despite the good results obtained previously, CNNs explore spatial features for classification, whereas RNNs are designed towards leveraging temporal aspects.

Gallicchio et al. [10] employed Deep Echo State Networks for PD identification based on raw signals. Further, other works exploring RNNs for automatic PD identification consider different kinds of data. Che et al. [5], for instance, proposed a deep-RNN model that directly learns patient similarity based on their previous records, and Zhang et al. [29] applied a Long-Short Term Memory-based neural network on clinical records to represent each patient as a multi-dimensional time series for subtype identification.

Despite the recent advances achieved by computer-aided detection and diagnosis systems, there is still room to improve research in automatic PD identification. Moreover, any advance is relevant to early diagnosis. In this work, we study the temporal dependence on signal data by proposing a new model based on Bidirectional Gated Recurrent Units (BiGRUs) along with an attention mechanism to learn temporal information. To the best of our knowledge, this is the first work that proposes the application of such mechanisms in the context of Parkinson's disease identification. Additionally, we also introduce the concept of "Bag of Samplings" (BoS), in which multiple compact representations are generated from a signal, significantly improving our results. Despite being used for PD identification, this approach can be extended to any time-series-based problem.

The remainder of this paper is organized as follows. Section 2 presents the theoretical background, and Section 3 explains the proposed model. Sections 4 and 5 discuss the methodology and experiments, respectively. Finally, Section 6 states conclusions and future works.[1]

2. Theoretical background

This section presents the theoretical background of Recurrent Neural Networks and the Attention Mechanism that were employed in the proposed model. Further, the concept of Bag of Samplings, which consists of representing a signal using different sampling intervals is introduced.

2.1. Recurrent Neural Networks

Recurrent Neural Networks [9] stand for a modification of traditional Neural Networks to cope with sequential problems, in which each sample is represented by a sequence of events (timesteps). While the original model learns how to combine characteristics from each sample in its hidden states individually, this variant also considers features across several timesteps in the input sequence.

More specifically, let $\mathscr{X}_i = (\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(T_i)})$ be a sample composed of T_i timesteps (episodes), each described by vector $\boldsymbol{x}_i^{(t)} \in \mathbb{R}^{d_n}$. For each episode, the RNN computes its hidden vector $\boldsymbol{h}_i^{(t)} \in \mathbb{R}^{d_m}$ as follows:

$$\boldsymbol{h}_i^{(t)} = \tanh\left(W\boldsymbol{x}_i^{(t)} + V\boldsymbol{h}_i^{(t-1)}\right) . \tag{1}$$

where $\boldsymbol{h}_i^{(0)} = 0$, $W \in \mathbb{R}^{n\times m}$, and $V \in \mathbb{R}^{m\times m}$ denote the weight matrices.

Despite being able to combine the most relevant features from each timestep in the sequence, these models are particularly difficult to train due to the recurrence term in Equation (1). Besides, if its values become too much larger or smaller than one, the gradients during the back-propagation step may diverge (explode) or approach zero (vanish).

To cope with such shortcoming, Hochreiter and Schmidhuber [12] proposed the Long-Short Term Memory (LSTM) unit, which consists on several modifications over the traditional RNN at the cost of having significantly more parameters to learn. Further, Cho [6] developed the Gated Recurrent Units (GRUs), which preserve the robustness of the previous model at some extent while figuring fewer parameters to train, making it suitable for smaller datasets.

2.2. Gated Recurrent Units

This type of unit contains two gates that control the amount of information that flows through the recurrent unit, thus preventing the exploding and vanishing gradient problems. Its formulation is presented in Equations (2)–(5):

$$\boldsymbol{r}_i^{(t)} = \sigma\left(W_r\boldsymbol{x}_i^{(t)} + V_r\boldsymbol{h}_i^{(t-1)}\right) , \tag{2}$$

$$\boldsymbol{u}_i^{(t)} = \sigma\left(W_u\boldsymbol{x}_i^{(t)} + V_u\boldsymbol{h}_i^{(t-1)}\right) , \tag{3}$$

$$\tilde{\boldsymbol{h}}_i^{(t)} = \tanh\left(W\boldsymbol{x}_i^{(t)} + \boldsymbol{r}_i^{(t)} \circ V\boldsymbol{h}_i^{(t-1)}\right) , \tag{4}$$

$$\boldsymbol{h}_i^{(t)} = \boldsymbol{u}_i^{(t)} \circ \boldsymbol{h}_i^{(t-1)} + \left(1 - \boldsymbol{u}_i^{(t)}\right) \circ \tilde{\boldsymbol{h}}_i^{(t)} , \tag{5}$$

where $W_r \in \mathbb{R}^{n\times m}$, $W_u \in \mathbb{R}^{n\times m}$, $V_r \in \mathbb{R}^{m\times m}$, $V_u \in \mathbb{R}^{m\times m}$ denote the weight matrices.

Notice that both *reset gate* $\boldsymbol{r}_i^{(t)} \in \mathbb{R}^m$ and *update gate* $\boldsymbol{u}_i^{(t)} \in \mathbb{R}^m$ are computed using the sigmoid function $\sigma(\cdot)$, guaranteeing that their elements lie in the interval $[0, 1]$. Consequently, when these gates interact with other vectors through the Hadamard product '$\circ$', the resulting operation is the copy of the original values if the gate is close to one, or the values are erased otherwise. The reset gate is employed in Equation (4) to compute the intermediate memory of the unit $\tilde{\boldsymbol{h}}_i^{(t)} \in \mathbb{R}^m$ by keeping just a fraction of its previous hidden vector. The update gate is used to produce the current hidden vector as an interpolation of its past value and the current memory in Equation (5). Fig. 1 illustrates such a mechanism.

In their initial conception, the recurrent networks take into account only information from previous timesteps to compute the current hidden state. However, it is possible to consider information from the entire sequence by combining two independent RNNs, regardless of their type. In this case, the first network, say $\overrightarrow{GRU}$, reads the sequence from left to right, while the second, $\overleftarrow{G}\,RU$, reads the sequence backwards. Then, the hidden states from each network are concatenated as follows, forming a Bidirectional GRU (BiGRU) layer:

[1] The source code is available at https://github.com/lzfelix/bag-of-samplings.

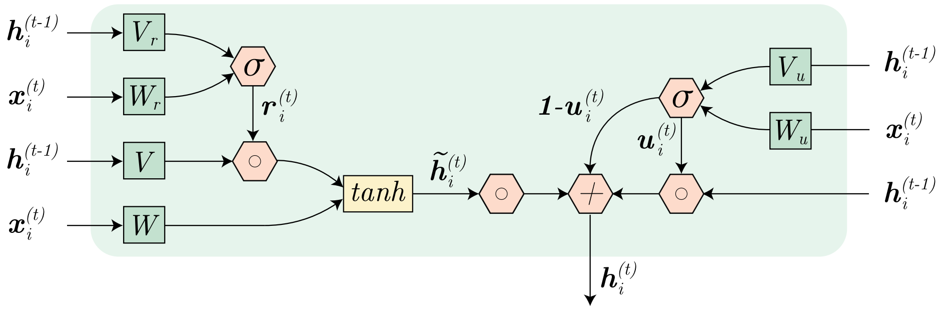


Fig. 1. A standard Gated Recurrent Unit.

$$\overrightarrow{\boldsymbol{h}}_i^{(t)} = \overrightarrow{GRU}\left(\boldsymbol{x}_i^{(t)}, \overrightarrow{\boldsymbol{h}}_i^{(t-1)}\right) \ \forall t \in [1, T_i],$$
$$\overleftarrow{\boldsymbol{h}}_i^{(t)} = \overleftarrow{G}\,RU\left(\boldsymbol{x}_i^{(t)}, \overleftarrow{\boldsymbol{h}}_i^{(t-1)}\right) \ \forall t \in [T_i, 1],$$
$$\overline{\boldsymbol{h}}_i^{(t)} = \left[\overrightarrow{\boldsymbol{h}}_i^{(t)}; \overleftarrow{\boldsymbol{h}}_i^{(t)}\right], \tag{6}$$

where $[;]$ is the concatenation operator and $\overline{\boldsymbol{h}} \in \mathbb{R}^{2m}$.

2.3. Attention mechanism

As described in Equation (6), applying a BiGRU network over some input sequence $\mathscr{X}_i$ yields a sequence of hidden vectors $\mathscr{H}_i = (\overline{\boldsymbol{h}}_i^{(1)}, \overline{\boldsymbol{h}}_i^{(2)}, \dots, \overline{\boldsymbol{h}}_i^{(T_i)})$, where $\overline{\boldsymbol{h}}_i^{(t)}$ contains the most relevant features of the t-th step neighborhood. Nevertheless, it is necessary to map such arbitrary-length sequence to a fixed-length descriptor in order to label its corresponding sample.

Several approaches can be used for such purpose, being the simplest of them to represent $\mathscr{X}_i$ as its last hidden state $\overline{\boldsymbol{h}}_i^{(T_i)}$. Still, as the network reads the sequence, it must forget previous features to store information from the new states, thus characterizing a lossy compression procedure [11]. Therefore, the last hidden vector emphasizes features extracted from the last few timesteps of the sequence. A better procedure consists in employing an Attention Mechanism [28], which computes the sample descriptor as the weighted average of its hidden vectors. Such a procedure allows the model to dynamically learn which vectors are more relevant and to drop irrelevant information. In this formulation, the scalar weights $\alpha_i^{(t)}$ for each timestep are determined as follows:

$$\beta_i^{(t)} = \boldsymbol{q}^T \tanh\left(W_a \overline{\boldsymbol{h}}_i^{(t)}\right),$$
$$\alpha_i^{(t)} = \frac{\exp\left(\beta_i^{(t)}\right)}{\sum_{i=1}^{T} \exp\left(\beta_i^{(t)}\right)}. \tag{7}$$

where $\boldsymbol{q} \in \mathbb{R}^{2m}$ and $W_a \in \mathbb{R}^{2m \times 2m}$ are parameters to be learned. The sample descriptor $\boldsymbol{u}_i \in \mathbb{R}^{2m}$ is computed as follows:

$$\boldsymbol{u}_i = \sum_{i=1}^{T} \alpha_i^{(t)} \overline{\boldsymbol{h}}_i^{(t)}. \tag{8}$$

Finally, the descriptor $\boldsymbol{u}_i$ can be used to label its corresponding input sample $\mathscr{X}_i$. In this case, since we are concerned with a binary classification problem (i.e., healthy individuals vs PD patients), the sigmoid function can be used once again, as presented below:

$$\hat{y}_i = \sigma(\boldsymbol{u}_i) = \frac{1}{1 + \exp(-W_o \boldsymbol{u}_i)}, \tag{9}$$

where $W_o \in \mathbb{R}^{1 \times 2m}$ is the weight matrix concerning the output layer. Finally, the model can be trained using the well-known binary cross-entropy loss.

2.4. Bag of Samplings

Given a dataset of labeled time-series $\{\mathscr{X}_i, y_i\}_{i=1}^N$, one may be interested in designing an RNN-based classifier to predict labels for previously unseen samples. However, if the time-series are too long (i.e., the samples have too many timesteps) it may become difficult to train these models, even by using sophisticated recurrent units as GRUs or LSTMs.

One way to cope with such an issue is to consider only the first few (or last) episodes of each sample or even reading the entire sequence, but backpropagating for a smaller amount of timesteps. Alternatively, we propose to sample the original signal with different fixed intervals of v_j, $j = 1, 2, \dots, V$, thus generating different and reduced representations of the same input signal, each with at most K timesteps. More specifically, given some input signal $\mathscr{X}_i$ and sampling interval v_j, its reduced version is computed as follows:

$$\mathscr{S}_{i,j} = \left(\boldsymbol{x}_i^{(1+t \cdot v_j)}\right)_{t=0}^{K}. \tag{10}$$

Fig. 2 depicts an example on how representations are computed for a signal of length 10 (columns), six channels (rows), and two intervals are considered (i.e., $v_1 = 2$ and $v_2 = 3$) with $K = 4$. Notice that six channels represent our problem, but the approach can be applied to signals of any number of channels, and any length. The interval (stride) defines the timesteps and the parameter K defines how many timesteps must be considered in the final representation. Given the interval v_1 (purple), the representation is comprised of timesteps 1, 3, 5, and 7 since $K = 4$, whereas v_2 (green) gives us timesteps 1, 4, 7, and 10. The boxes with both colors are used in both representations, and the ones in gray are discarded.

The rationale for using multiple samplings of the same sequence is to reduce the chance of missing relevant information and to obtain a compact, but meaningful representation of the input signal. Further, all samplings of the same input signal $\mathscr{X}_i$ are grouped into a Bag of Samplings (BoS) $\mathscr{B}_i = (\mathscr{S}_{i,1}, \mathscr{S}_{i,2}, \dots, \mathscr{S}_{i,V})$, which are in turn presented to the classifier. The BoS role in our work consists only in producing shorter

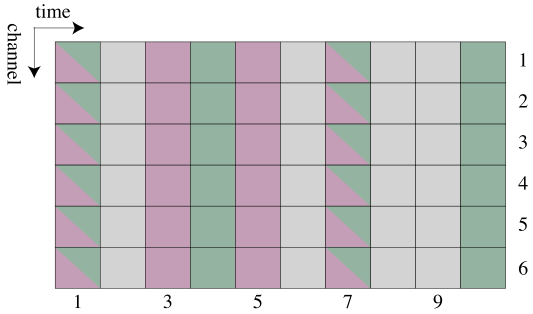


Fig. 2. Signal sampling using two intervals, $v_1 = 2$ (purple) and $v_2 = 3$ (green) with $K = 4$. Gray timesteps are disregarded by the model.

versions of the input signal while preserving, to some extent, its relevant features. Feature extraction and combination, on the other hand, are performed by the proposed model, as discussed in Section 3.

3. Proposed model

The signals employed for automatic PD identification in the experiments are considerably long, making unfeasible the direct application of RNN-based models.[2] Therefore, for each input signal $\mathscr{X}_i$, we first compute its BoS representation $\mathscr{B}_i$ and present it to the model. Given this compact representation, each sampling $\mathscr{S}_{i,j}$ is read independently by the model "input heads" (blue-shaded elements in Fig. 3) using a stack of two BiGRU layers, as depicted in Fig. 3.

Mathematically speaking, such process is described as an extension of Equation (6) for multiple levels of recurrence:

$$\begin{aligned} \boldsymbol{l}_{i,j}^{(t)} &= \text{BiGRU}\left(\boldsymbol{s}_{i,j}^{(t)}, \boldsymbol{l}_{i,j}^{(t-1)}\right) \quad \forall t \in [1, K], \\ \boldsymbol{p}_{i,j}^{(t)} &= \text{BiGRU}\left(\boldsymbol{l}_{i,j}^{(t)}, \boldsymbol{p}_{i,j}^{(t-1)}\right) \quad \forall t \in [1, K], \end{aligned} \tag{11}$$

where $\boldsymbol{l}_{i,j}^{(t)} \in \mathbb{R}^{(2m)}$ and $\boldsymbol{p}_{i,j}^{(t)} \in \mathbb{R}^{(2m)}$ correspond to the BiGRU output in the first and second levels of the stack, respectively, after reading each timestep of the j-th sampling in the i-th bag. The idea is that each of the V "input heads" read the sequence at specific sampling rates, thus becoming specialized in computing a descriptor for each of these representations.

After combining the most relevant characteristics of each timestep in the sequence with its neighbors, a fixed-length descriptor is computed for each sampling interval. Such a step is performed using the Attention Mechanism from Equations (7) and (8), thus generating V attended vectors $\boldsymbol{a}_{i,j} \in \mathbb{R}^{(2m)}$, $i = 1, 2, \ldots, N$, and $j = 1, 2, \ldots, V$.

Further, these vectors are concatenated, forming the descriptor $\boldsymbol{c} \in \mathbb{R}^{(2Vm)}$ for the current bag based on the most relevant features extracted from each sampling of the original signal. Since this representation may be quite large, we map it onto a lower-dimensional vector $z_i \in \mathbb{R}^m$ using a non-linear layer to ease the model training, as follows:

$$z_i = \text{ReLU}(W_z \boldsymbol{c}_i). \tag{12}$$

Finally, the classification is performed using the sigmoid function based on this last representation, as described in Equation (9). We label the input bag as positive (i.e., patient) if $\hat{y}_i \geq 0.5$, and negative (i.e., control group) otherwise. In order to prevent overfitting, we resort to dropout [26] in the GRU recurrent connections and in the input heads with probability p_{drop}, whereas the same technique is applied before Equation (12) with probability $p_{drop}/2$. Additionally, we observed that employing gradient norm clipping [19] improved model convergence.

4. Methodology

This section describes the data and steps performed in the evaluation of the proposed approach.

4.1. Dataset

The proposed model was evaluated in the HandPD dataset [23] comprising images and signals collected from exams performed by potential patients. The exams are comprised of six tasks supposed to be non-trivial to PD patients, being two drawing tasks, as displayed in Fig. 4, and the remainder corresponding to wrist and hand movements. A group of 35 individuals was considered, in which 21 are the healthy control (HC) group with ages ranging from 14 to 79 years old, and 14 are PD patients with ages ranging from 38 to 78 years old. We performed two phases of experiments: first considering only signals from the spirals drawings (Spiral dataset) and then from the meanders task (Meander dataset).

The signals corresponding to each exam in Fig. 4 are depicted in Fig. 5. Under this representation, each exam is composed of six channels (features): microphone (channel 1), finger grip (channel 2), axial pressure of ink refill (channel 3), tilt and acceleration in the X (channel 4), Y (channel 5) and Z (channel 6).

To compare our results with the most recent ones in the literature, we split the dataset in the same proportions used by Afonso et al. [3]. For each exam, the authors trained their models in two procedures: the first one uses half of the data for training and the other half for testing; while in the second this ratio is changed to 75% and 25%, respectively, since they do not fine-tune their hyperparameters. In our case, for the former setup, we use 40% of the data for training, 10% for validation, and 50% for testing, while in the latter the proportions are changed to 65%, 10%, and 25%, respectively.[3]

4.2. Preprocessing

Two preprocessing steps were performed for each set of exams (i.e., spirals and meanders) considering the statistics computed in their corresponding training partitions. First, we computed the lower and upper bounds for each channel as their 5th and 90th percentiles, respectively. Further, all values outside these ranges were replaced by their corresponding boundaries to remove outliers. Following, each channel was standardized individually to have zero mean and unitary standard deviation. Specifically, the value in each timestep for each channel was subtracted from its corresponding channel mean value, and further divided by its standard deviation.

4.3. Experimental setup

Initially, the proposed model hyperparameters were adjusted using the search intervals presented in Table 1 and the values that achieved the best results in the validation set were used for testing. As the model architecture depends on the number of sampling intervals V, we temporarily employed Bags of Samplings formed by signals sampled every 25 and 50 ms. The models were trained using the Adam [16] optimizer with learning rate η and norm clipping ρ. To compute the testing metrics, we trained our model during 30 epochs, but we employed the set of weights from the epoch that achieved the lowest loss in the validation set during training (i.e. early stopping).

After establishing the proper hyperparameter values, we move our attention to finding a set of sampling intervals that are best suited to the PD identification task. In this case, we start with five different sampling intervals: $\{5, 25, 50, 100, 150\}$. The rationale behind using this set of values is that in one extreme we consider almost all the first K timesteps of the input signals, whereas in the other each timestep is considerably far apart from one another in the original sequence.

One may notice that each sampling rate requires allocating a new "input head" in the model, increasing considerably its complexity in terms of parameters to learn. To better analyze this behavior, recall that each "input head" is composed of a BiGRU stack and an attention layer. The output of the latter layer is concatenated with the representation computed by the remaining heads, also increasing the sample descriptor $\boldsymbol{c}_i$ dimensionality by a factor of $2m$.

Turning the model too complex comes at the expense of requiring more data to avoid overfitting. This is an unfeasible option in our case since PD identification datasets are usually small due to the difficulty in collecting this kind of data. Therefore, it is desirable to determine a subset of sampling intervals that achieve the best results while keeping

[2] For the sake of illustration, one of the datasets considered in the experiments contains signals with, on average, $17,914 \pm 11,428$ timesteps.

[3] The splits are available online.

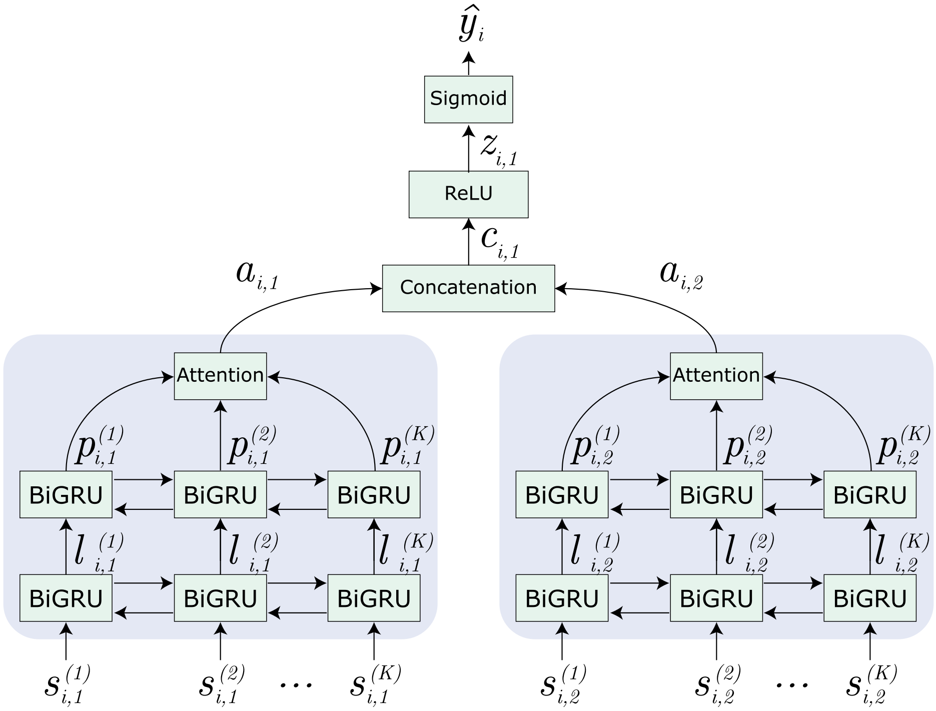


Fig. 3. The proposed model for a bag of two sampling intervals.

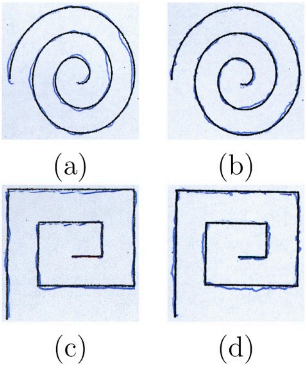


Fig. 4. Drawings samples from spirals: (a) HC and (b) PD patient; as well as meanders: (c) HC and (d) PD patient.

the model complexity at a reasonable level.

Since our set contains only five sampling intervals, its power set is also relatively small with only $2^5 - 1 = 31$ possibilities (we disregard the empty set). Due to the small number of combinations, we train the proposed model considering all possible combinations of sampling intervals and chose the smallest set of intervals that achieve the best results in the validation set. In our experiments, we observed the resolutions $\{25, 50\}$ and $\{5, 25, 50, 100\}$ yielded the best results for the spirals and meanders datasets, respectively.

We compare the results obtained by the proposed model with the most recent findings in the literature from Afonso et al. [3], which employed three deep neural networks (ImageNet, CIFAR-10, and LeNet) to learn representations from recurrence plot data. We also considered three other baselines: two that employ only the first or last K timesteps from each sample $\mathscr{X}_i$ and one that reads K timesteps equally spaced from the input sequence. Further, to statistically compare the obtained results, the training procedure is repeated 20 times, and the measures obtained are compared using the Wilcoxon signed-rank test [27] with $p = 0.05$. Finally, the models were implemented in Python using the Keras[4] [7] library with TensorFlow[5] [1] as backend.

5. Results and discussion

As aforementioned, the experiments were conducted individually by considering the signals from the Spiral and Meander exams in the HandPD dataset. The following sections discuss the results for each case considering accuracy (overall and for each class individually), precision, recall, and F1 results.

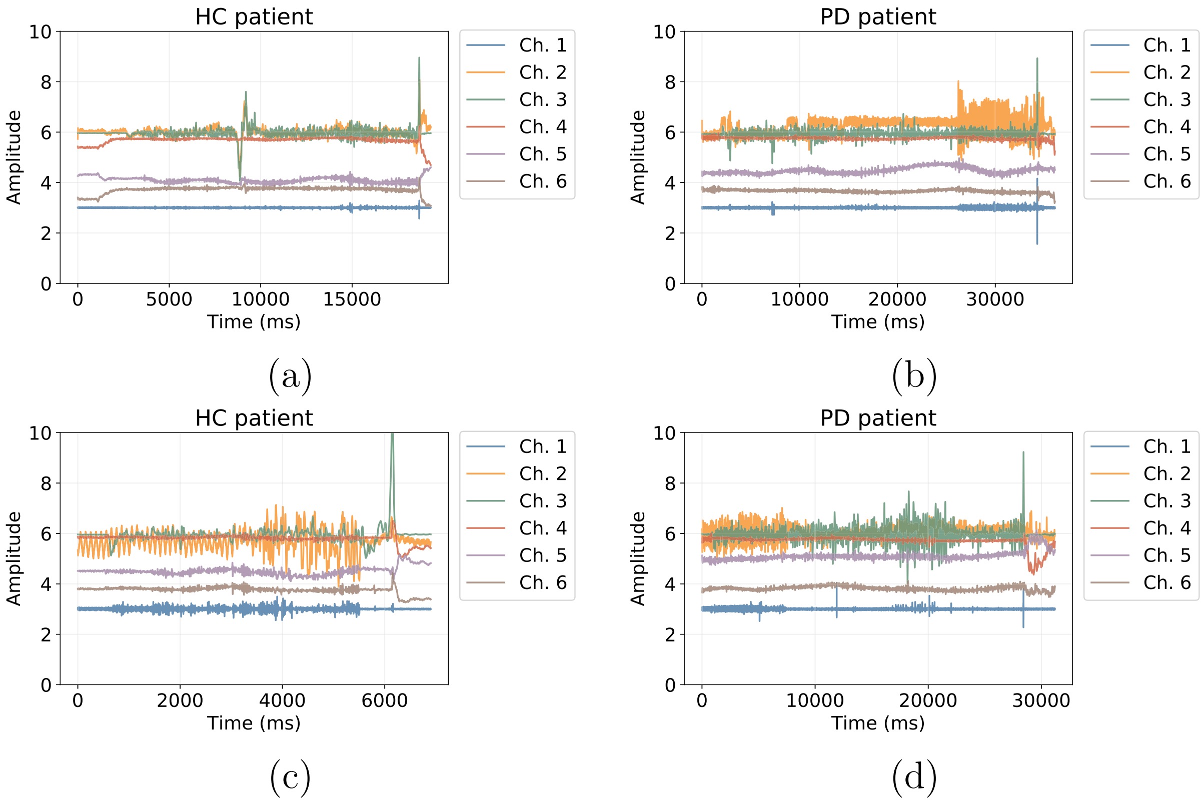
Regarding the proposed model, four different versions are compared: the first relies on the BoS representation; while the other three serve as baselines: the first two consider the first and last K sequence timesteps, respectively, while the last considers K timesteps from the entire signal equally spaced. Since each sample has different lengths, the sampling interval in this case is different for each signal. These models allow to assess the effectiveness of the proposed approach. Further, we consider the best results obtained by Afonso et al. [3] regardless of image resolution, since this parameter does not apply to our experiments. It is also important to highlight that different from their training procedures, our models were trained with 10% less data, which were used for validation purposes and early stopping.

5.1. Spiral dataset

Concerning the overall accuracy results, presented in Table 2, the proposed approach using Bag of Samplings achieved the best results in both training regimes with gains of 1.68 and 1.75, respectively when compared to the state-of-the-art. On the other hand, the first two baselines presented results only better than the Optimum Path Forest (OPF) model. Regarding the equally spaced baseline, despite giving superior

[4] Available at https://keras.io.

[5] vailable at https://www.tensorflow.org/.



(a) (b)

(c) (d)

Fig. 5. Handwriting exam signals: (a) corresponding to Fig. 4a and (b) corresponding to Fig. 4b and (c) corresponding to Fig. 4c, and (d) corresponding to Fig. 4d.

Table 1
Hyperparameters search values.

Hyperparameter	Search interval	Selected value
Batch size	$[8, 32]$	16
RNN Type	{GRU, BiGRU}	BiGRU
# RNN layers	$[1, 3]$	2
# Epochs	$[10, 40]$	30
d_m	$[16, 128]$	64
p_{drop}	$[0, 0.5]$	0.5
η	$[10^{-4}, 10^{-3}]$	$5{\cdot}10^{-4}$
ρ	$[0.5, 3] \cup \infty$	1.0
K	500	500

Table 2
Accuracy results regarding the Spiral dataset.

Approach	50% training	75% training
BoS	**85.38 ± 2.37**	**89.48 ± 3.67**
First 500	67.18 ± 3.01	78.36 ± 2.19
Last 500	69.21 ± 3.59	74.40 ± 3.14
Equally 500	77.34 ± 2.75	83.68 ± 5.09
ImageNet	83.70	87.73
CIFAR-10	81.33	87.60
LeNet	81.43	85.19
OPF	64.90	67.49

metrics, the obtained values are still inferior to our proposed approach and yield a more significant standard deviation. Such observations show that the BoS allows the model to have a broader view of samples for classification, resulting in superior accuracy values with more stable standard deviations.

The results from Table 2 are detailed for each class in Table 3. Under this perspective, it is possible to observe that all versions of our model have outperformed previous results regarding healthy patient identification in both training regimes. Further, the proposed model using the Bag of Samplings approach improved the state of the art by 20.13 and 22.67 using 40% and 65% of the data, respectively.

Table 3
Average HC and PD accuracies regarding the Spiral dataset.

Approach	50% training		75% training	
	HC	PD	HC	PD
BoS	**85.61 ± 3.90**	85.54 ± 3.10	**95.53 ± 4.64**	84.76 ± 4.73
First 500	67.68 ± 2.99	66.65 ± 3.44	80.20 ± 3.74	76.72 ± 2.07
Last 500	66.42 ± 3.38	75.86 ± 6.65	77.70 ± 3.79	71.62 ± 3.47
Equally 500	72.55 ± 3.98	81.75 ± 5.31	83.87 ± 5.14	84.23 ± 6.63
ImageNet	60.36	**94.38**	72.86	**97.86**
CIFAR-10	65.48	90.00	72.14	93.39
LeNet	54.17	91.65	66.43	92.33
OPF	35.24	94.55	40.24	95.54

Regarding precision, recall, and F1 metrics, presented in Table 4, the proposed model presented competitive results, being able to outperform the previous best recall results by 0.053 in the 65% regime. That implies that our model presents a higher true positive rate, at the expense of a small impact in the F1-measure.

5.2. Meander dataset

The performance in this case is similar to the results observed in the Spiral dataset. More specifically, regarding the overall accuracy results presented in Table 5, it is possible to observe the proposed model improved the results in both training regimes by 5.26 and 4.19, respectively, considering the Bag of Samplings representation.

A further analysis based on the results presented in Table 6 shows

Table 4
Precision, Recall and F1 results regarding the Spiral dataset.

Approach	Precision	Recall	F1
50% Training			
BoS	0.855 ± 0.031	0.834 ± 0.054	0.843 ± 0.029
First 500	0.667 ± 0.034	0.615 ± 0.046	0.640 ± 0.036
Last 500	0.759 ± 0.066	0.521 ± 0.078	0.613 ± 0.056
Equally 500	0.816 ± 0.053	0.680 ± 0.079	0.738 ± 0.040
ImageNet	0.944	0.858	**0.894**
CIFAR-10	0.900	**0.870**	0.875
LeNet	0.917	0.842	0.878
OPF	**0.955**	0.822	0.880
75% Training			
BoS	0.848 ± 0.047	**0.955 ± 0.048**	0.897 ± 0.035
First 500	0.767 ± 0.021	0.785 ± 0.052	0.775 ± 0.028
Last 500	0.716 ± 0.035	0.772 ± 0.054	0.742 ± 0.034
Equally 500	0.842 ± 0.066	0.817 ± 0.074	0.827 ± 0.055
ImageNet	**0.979**	0.902	**0.917**
CIFAR-10	0.934	0.899	0.916
LeNet	0.922	0.880	0.901
OPF	0.946	0.825	0.876

Table 5
Accuracy results regarding the Meander dataset.

Approach	50% training	75% training
BoS	**89.29 ± 3.75**	**92.24 ± 2.65**
First 500	61.58 ± 2.91	64.93 ± 4.02
Last 500	69.32 ± 2.75	63.96 ± 3.25
Equally 500	77.34 ± 2.76	83.68 ± 5.00
ImageNet	83.93	88.05
CIFAR-10	81.30	86.36
LeNet	84.03	85.06
OPF	66.61	70.74

Table 6
Average HC and PD accuracies regarding the Meander dataset.

Approach	50% training		75% training	
	HC	PD	HC	PD
BoS	**81.83 ± 4.82**	84.99 ± 4.53	**90.10 ± 3.63**	95.17 ± 2.50
First 500	61.93 ± 2.58	61.67 ± 4.83	63.53 ± 3.50	67.87 ± 5.17
Last 500	68.38 ± 3.10	71.03 ± 3.06	64.96 ± 3.54	63.07 ± 3.89
Equally 500	75.22 ± 3.98	81.75 ± 5.13	83.87 ± 5.14	84.23 ± 6.63
ImageNet	62.98	92.77	73.57	**96.88**
CIFAR-10	66.55	85.54	71.67	92.14
LeNet	68.93	89.66	62.62	93.48
OPF	40.24	**93.75**	46.67	94.82

Table 7
Precision, Recall and F1 results regarding the Meander dataset using 50% of the data for training.

Approach	Precision	Recall	F1
50% Training			
BoS	0.850 ± 0.045	0.779 ± 0.079	0.810 ± 0.050
First 500	0.617 ± 0.048	0.515 ± 0.073	0.558 ± 0.044
Last 500	0.710 ± 0.031	0.596 ± 0.064	0.646 ± 0.044
Equally 500	0.818 ± 0.053	0.680 ± 0.079	0.738 ± 0.040
ImageNet	**0.928**	0.869	0.893
CIFAR-10	0.868	0.874	0.871
LeNet	0.897	**0.885**	**0.891**
OPF	0.964	0.806	0.878
75% Training			
BoS	0.952 ± 0.025	0.883 ± 0.049	0.924 ± 0.031
First 500	0.679 ± 0.052	0.506 ± 0.082	0.576 ± 0.069
Last 500	0.631 ± 0.039	0.595 ± 0.069	0.610 ± 0.045
Equally	0.842 ± 0.066	0.817 ± 0.074	0.827 ± 0.055
ImageNet	**0.969**	**0.904**	0.922
CIFAR-10	0.921	0.896	0.907
LeNet	0.935	0.870	0.901
OPF	0.946	0.841	0.891

again that our model improves healthy patients recognition rates by 12.90 in the smaller dataset and by 16.53 in the larger one. Moreover, these latter results are only slightly worse than the best approach developed by Afonso et al. [3] for PD identification.

In terms of precision, recall and F1 metrics, presented in Table 7, we observed competitive results regarding the larger training set regime, with precision and recall results very competitive with the state-of-the-art. Still, we were able to outperform previous works in terms of F1 results.

5.3. Discussion

The experimental results showed the potential of using data from sensors as a mean to identify PD patients since the accuracy has been increased from work to work. An important consideration is that the proposed model is trained with fewer samples than the models in Ref. [3] because it was implemented a validation set with 10% of the total of samples to optimize the model's hyperparameters and to perform early stopping during training.

As aforementioned, the model is trained with 40% and 65% of the total of samples instead of 50% and 75%. Nonetheless, the proposed approach using Bag of Samplings achieved the best accuracy and sensitivity for training sets with 50% and 75%, whereas showing competitive results for the remaining evaluation metrics in both datasets.

The signal representation plays a vital role in the final results as one can observe in the results concerning only the proposed model. The representation based on Bag of Samplings provides a better generalization of the signal than using either the first or the last 500 timesteps, and a significant gain in any of the evaluation metrics. Moreover, the proposed approach achieved a significant improvement in the classification of HC samples.

One of the difficulties in developing better automatic PD identification approaches regards data scarcity to fit more robust machine learning-based models. To forecast model improvement with more data, which is expensive and difficult to acquire, we trained new versions of the proposed model with gradually less training data.

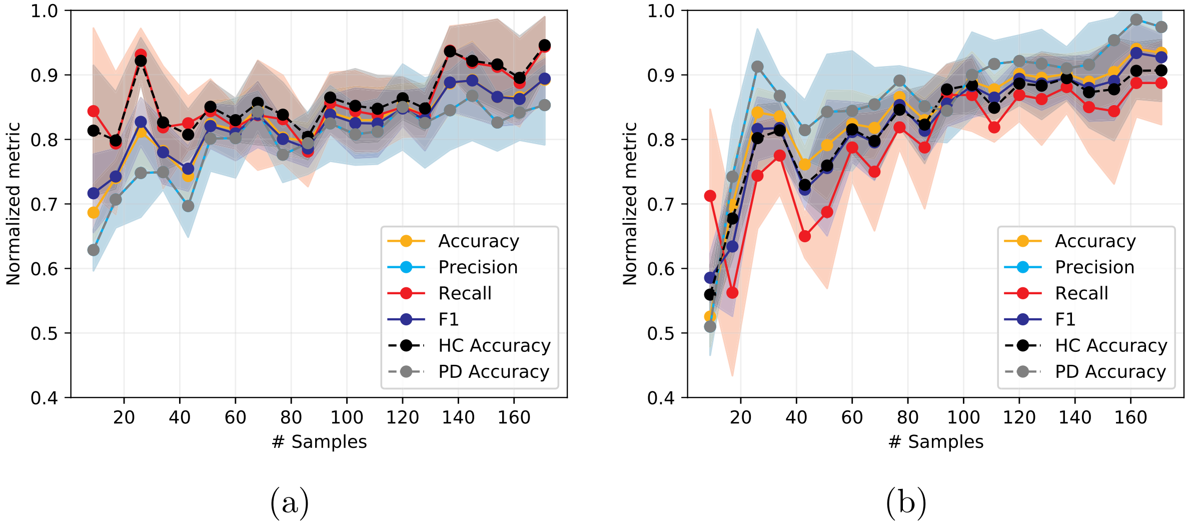
We start fitting the model with only 5% of the training set and increase this value by 5% incrementally until reaching 100%, whereas the testing set is kept fixed across all runs.[6] For each fraction of the dataset, for both exams (meanders and spirals), the models were trained 5 times to compute the metrics' mean and standard deviation, as shown in Fig. 6.

From these results it is possible to observe that the model trained with the Meanders dataset displays a steady improvement. On the other hand, the model trained with the Spiral dataset presents superior results with less training data. Overall, with at least 100 samples, regardless dataset, the proposed model using the BoS representation yields metrics with values superior to 0.8 in all cases. Finally, it is also worth noticing that whereas the Spirals-based model is more prone to identifying healthy individuals, the Meanders-based one is better in identifying PD patients.

6. Conclusions

This work proposed a model based on Bidirectional Gated Recurrent

[6] We employed the 75% training regime, which uses at most 65% of the samples for training, 10% for validation and early stopping and 15% for testing.



(a)

(b)

Fig. 6. Model metrics as a function of the training set size for the (a) Spiral and (b) Meander datasets. Standard deviations are displayed as shaded regions. For display purposes, the accuracy values were divided by 100.

Units along with an attention mechanism to aid the identification of Parkinson's Disease. The model was designed to learn temporal information from signal data collected from handwriting exams. The overall accuracy and healthy patients accuracy outperformed previous state-of-the-art results, despite being trained with fewer samples than the reference models. In the remaining evaluation metrics, the proposed model achieved competitive results and showed to be a promising technique for such a context.

Data representation is important to achieve better results as well. Hence, it was also introduced the concept of Bag of Samplings, which is an approach to compute multiple compact representations from signal data. The representations provided a significant gain over their baselines in all evaluation metrics, and also showing to be a promising approach.

As future works, a different approach could be used to generate the signal samplings presented to the proposed model. For instance, one could learn the best sampling rates as well as its durations using meta-heuristic optimization, or even learn this parameter along with the model training using some Reinforcement Leaning approach. In future experiments, other signal-based datasets could be considered as well, such as the dataset developed by Isenkul et al. [13] and the PaHaW dataset [8].

Acknowledgments

The authors would like to thank FAPESP (São Paulo Research Foundation) grants #2013/07375-0 and #2014/12236-1, as well as CNPq (National Council for Scientific and Technological Development) grants #427968/2018-6, and #307066/2017-7.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, URL: https://www.tensorflow.org/, 2015 (software available from: tensorflow.org).

[2] E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, V. Venkatraman, Gait and tremor investigation using machine learning techniques for the diagnosis of Parkinson disease, Future Gener. Comput. Syst. 83 (2018) 366–373.

[3] L.C. Afonso, G.H. Rosa, C.R. Pereira, S.A. Weber, C. Hook, V.H.C. Albuquerque, J. P. Papa, A recurrence plot-based approach for Parkinson's disease identification, Future Gener. Comput. Syst. 94 (2019) 282–292.

[4] S. Bhat, U.R. Acharya, Y. Hagiwara, N. Dadmehr, H. Adeli, Parkinson's disease: cause factors, measurable indicators, and early diagnosis, Comput. Biol. Med. 102 (2018) 234–241.

[5] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, F. Wang, An RNN Architecture with Dynamic Temporal Matching for Personalized Predictions of Parkinson's Disease, 2017, pp. 198–206.

[6] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: encoder-decoder approaches, in: Proceedings of Syntax, Semantics and Structure in Statistical Translation-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014, pp. 103–111.

[7] F. Chollet, et al., Keras. https://keras.io, 2015.

[8] P. Drotár, J. Mekyska, I. Rektorová, L. Masarová, Z. Smékal, M. Faundez-Zanuy, Decision support framework for Parkinson's disease based on novel handwriting markers, IEEE Trans. Neural Syst. Rehabil. Eng. 23 (3) (2014) 508–516.

[9] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.

[10] C. Gallicchio, A. Micheli, L. Pedrelli, Deep echo state networks for diagnosis of Parkinson's disease, CORR (2018) 06708, abs/1802.

[11] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT press, 2016.

[12] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[13] M. Isenkul, B. Sakar, O. Kursun, Improved spiral test using digitized graphics tablet for monitoring Parkinson's disease, in: Proceedings of the International Conference on e-Health and Telemedicine, 2014, pp. 171–175.

[14] J. Jankovic, Parkinson's disease: clinical features and diagnosis, J. Neurol. Neurosurg. Psychiatry 79 (4) (2008) 368–376.

[15] H.B. Kim, W.W. Lee, A. Kim, H.J. Lee, H.Y. Park, H.S. Jeon, S.K. Kim, B. Jeon, K. S. Park, Wrist sensor-based tremor severity quantification in Parkinson's disease using convolutional neural network, Comput. Biol. Med. 95 (2018) 140–146.

[16] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2014 arXiv preprint arXiv:14126980 2014.

[17] M. MashhadiMalek, F. Towhidkhah, S. Gharibzadeh, V. Daeichin, M.A. Ahmadi-Pajouh, Are rigidity and tremor two sides of the same coin in Parkinson's disease? Comput. Biol. Med. 38 (11) (2008) 1133–1139.

[18] C. Ornelas-Vences, L.P. Sanchez-Fernandez, L.A. Sanchez-Perez, A. Garza-Rodriguez, A. Villegas-Bastida, Fuzzy inference model evaluating turn for Parkinson's disease patients, Comput. Biol. Med. 89 (2017) 379–388.

[19] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013, pp. 1310–1318.

[20] C.R. Pereira, D.R. Pereira, G.H. Rosa, V.H. Albuquerque, S.A. Weber, C. Hook, J. P. Papa, Handwritten dynamics assessment through convolutional neural networks: an application to Parkinson's disease identification, Artif. Intell. Med. 87 (2018) 67–77.

[21] C.R. Pereira, D.R. Pereira, F.A. da Silva, C. Hook, S.A.T. Weber, L.A.M. Pereira, J. P. Papa, A step towards the automated diagnosis of Parkinson's disease: analyzing handwriting movements, in: IEEE 28th International Symposium on Computer-Based Medical Systems, 2015, pp. 171–176.

[22] C.R. Pereira, D.R. Pereira, F.A. Silva, J.P. Masieiro, S.A.T. Weber, C. Hook, J. P. Papa, A new computer vision-based approach to aid the diagnosis of Parkinson's disease, Comput. Methods Progr. Biomed. 136 (2016) 79–88.

[23] C.R. Pereira, S.A.T. Weber, C. Hook, G.H. Rosa, J.P. Papa, Deep learning-aided Parkinson's disease diagnosis from handwritten dynamics, in: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI), 2016, pp. 340–346.

[24] G. Rigas, A.T. Tzallas, M.G. Tsipouras, P. Bougia, E.E. Tripoliti, D. Baga, D. I. Fotiadis, S.G. Tsouli, S. Konitsiotis, Assessment of tremor activity in the

Parkinson's disease using a set of wearable sensors, IEEE Trans. Inf. Technol. Biomed. 16 (3) (2012) 478–487.

[25] A. Samà, C. Pérez-López, D. Rodríguez-Martín, A. Català, J. Moreno-Aróstegui, J. Cabestany, E. de Mingo, A. Rodríguez-Molinero, Estimating bradykinesia severity in Parkinson's disease by analysing gait through a waist-worn sensor, Comput. Biol. Med. 84 (2017) 114–123.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.

[27] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.

[28] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.

[29] X. Zhang, J. Chou, J. Liang, C. Xiao, Y. Zhao, H. Sarva, C. Henchcliffe, F. Wang, Data-driven subtyping of Parkinson's disease using longitudinal clinical records: a cohort study, Sci. Rep. 9 (1) (2019) 797.