Classification of Short Time Series in Early Parkinson's Disease With Deep Learning of Fuzzy Recurrence Plots

Tuan Pham, Karin Wårdell, Anders Eklund, Göran Salerud and Göran Salerud

The self-archived postprint version of this journal article is available at Linköping University Institutional Repository (DiVA):

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-161818>

N.B.: When citing this work, cite the original publication.

Pham, T., Wårdell, K., Eklund, A., Salerud, G., Salerud, G., (2019), Classification of Short Time Series in Early Parkinson's Disease With Deep Learning of Fuzzy Recurrence Plots, *IEEE/CAA Journal of Automatica Sinica*, 6(6), 1306-1317. <https://doi.org/10.1109/JAS.2019.1911774>

Original publication available at:

<https://doi.org/10.1109/JAS.2019.1911774>

Copyright: Institute of Electrical and Electronics Engineers (IEEE)

<http://www.ieee.org/index.html>

©2019 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Classification of Short Time Series in Early Parkinson's Disease with Deep Learning of Fuzzy Recurrence Plots

Tuan D. Pham, *Senior Member, IEEE*, Karin Wårdell, *Member, IEEE*, Anders Eklund, and Göran Salerud

Abstract—There are many techniques using sensors and wearable devices for detecting and monitoring patients with Parkinson's disease (PD). A recent development is the utilization of human interaction with computer keyboards for analyzing and identifying motor signs in the early stages of the disease. Current designs for classification of time series of computer-key hold durations recorded from healthy control and PD subjects require the time series of length to be considerably long. With an attempt to avoid discomfort to participants in performing long physical tasks for data recording, this paper introduces the use of fuzzy recurrence plots of very short time series as input data for the machine training and classification with long short-term memory (LSTM) neural networks. Being an original approach that is able to both significantly increase the feature dimensions and provides the property of deterministic dynamical systems of very short time series for information processing carried out by an LSTM layer architecture, fuzzy recurrence plots provide promising results and outperform the direct input of the time series for the classification of healthy control and early PD subjects.

Index Terms—Early Parkinson's disease, short time series, fuzzy recurrence plots, deep learning, LSTM neural networks, pattern classification.

I. INTRODUCTION

Parkinson's disease (PD) is a neurodegenerative disorder that affects dopaminergic neurons [1]. Statistics on PD have reported it affects approximately 10 million people worldwide, and about 4% of them before the age of 50 [2]. Symptoms of PD slowly develop over years, and the progression of PD can be different among individuals because of the diversity of the disease. People with PD can be observed with tremor, bradykinesia (slowness of movement), limb rigidity, and gait and balance problems. The cause of PD remains unknown [3]. Because a significant amount of the substantia nigra neurons have already been lost or impaired before the onset of motor features, people with PD may first start experiencing

symptoms later in the course of the disease [4], [5]. Treatment options for PD can vary and include medications and sometimes deep brain stimulation [6], [7]. While Parkinson's itself is not fatal, disease complications can be serious [4].

Many scientific efforts have been spent on exploring methods for identifying biomarkers for PD [8] with the hope that these markers can lead to earlier diagnosis and targeted treatments of the disease. However, present therapies used for PD cannot slow or stop the disease in a prodromal stage [9]. There are many techniques using sensors for detecting and monitoring movement patterns on patients with PD. Most techniques using sensor-induced data focus on studying gait dynamics and temporal gait parameters [10]–[14], and others use wrist accelerometers for intraoperative measurements of tremor during surgery [15]. One issue of using sensor data is that gait measurements are to be obtained during relatively long walking periods, causing discomfort to the participants or impracticability of performance in clinical settings. Therefore, research into the minimum strides required for a reliable estimation of temporal gait parameters has recently been carried out with the purpose of avoiding or minimizing discomfort to participants in gait experiments [16].

Apart from gait and balance data, the measurement of computer keystroke time series that contain information of the hold time occurring between pressing and releasing a key collected during the sessions of typing activity using a standard word processor on a personal computer has been proposed for detecting early stages of PD [17]. Being similar to the motivation for determining the minimum number of strides for the analysis of gait dynamics, this study is interested in answering the question if there are methods that can process very short time series and achieve good results for differentiating healthy controls from subjects with early PD. If being successful, the use of computer keystroke time series can be equivalent on a practical basis to the use of mobile sensor data for evaluating upper limbs motor functions by finger tapping [18] that is typically used in clinical trials. The finger tapping test requires a participant to press one or two buttons on a device such as an iPhone as fast as possible for a short period of time.

The method of fuzzy recurrence plots [19] developed for studying nonlinear dynamics in time series data can be useful for creating feature dimensions of short time series. Therefore, the deep learning of fuzzy recurrence plots is proposed in this study for the classification of short time series of computer-key hold time recorded from two cohorts of healthy control

Tuan D. Pham is with the Department of Biomedical Engineering, the Center for Medical Image Science and Visualization, Linköping University, Linköping, Sweden; the Center for Artificial Intelligence, Prince Mohammad Bin Fahd University, Al Khobar, Kingdom of Saudi Arabia. Karin Wårdell is with the Department of Biomedical Engineering, Linköping University, Linköping, Sweden. Anders Eklund is with the Department of Biomedical Engineering, the Department of Computer and Information Science, the Center for Medical Image Science and Visualization, Linköping University, Linköping, Sweden. Göran Salerud is with the Department of Biomedical Engineering, Linköping University, Linköping, Sweden. Corresponding author's e-mail (TDP): tuan.pham@liu.se

Copyright (c) 2019 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

and early PD.

The rest of this paper is organized as follows. Section II includes a survey of literature relating to the present study. Section III outlines the concept and mathematical formulation of fuzzy recurrence plots. Section IV presents the implementation of LSTM neural networks with fuzzy recurrence plots. The public database for testing the proposed classification approach is described in Section V. Section VI shows the experimental results together with discussion. Finally, Section VII consists of the concluding remarks of the research finding and suggestion of issues for future work.

II. RELATED WORK

Having highlighted earlier, the concept of using fuzzy recurrence plots of short time series for classification using LSTM networks is original in that it contributes to the increase of the feature dimensions of short raw time series, which, in turn, can improve the classification. No prior work of similar concepts exists in literature. A survey of recent reports that applied LSTM and convolutional neural networks for time-series or sequential data classification is addressed herein.

The deep-learning method of LSTM neural networks has been adopted for the classification of time-series or sequential data [20], including speech recognition [21] and machine translation [22], [23]. A deep recurrent neural network called TimeNet for extracting features from time series was developed [24]. The TimeNet was designed to generalize time series representation across domains. A trained TimeNet can be used as a feature extractor for time series and was reported to be useful for time series classification by performing better than a domain-specific recurrent neural network and dynamic time warping [24]. Stacked LSTM autoencoder networks were applied to extract features of time-series data, which were then used to train deep feed-forward neural networks for classification of multivariate time series recorded with sensors in the steel industry to detect steel surface defects [25]. In this work, the features extracted with LSTM autoencoders were found to be useful, and therefore the need for domain expert knowledge can be alleviated.

Other time-series classification using convolutional neural networks (CNNs) have recently been introduced. A convolutional LSTM (ConvLSTM) was introduced for a spatiotemporal sequence forecasting problem in which both the input and the prediction target are spatiotemporal sequences [26]. This ConvLSTM model was constructed by extending the fully connected LSTM to have convolutional structures in the input-to-state and state-to-state transitions. The ConvLSTM network was reported to perform better than the fully connected LSTM by being able to capture the spatiotemporal correlations of the sequential data for precipitation nowcasting. A multi-scale convolutional neural network [27], which extracts deep-learning features at different scales and frequencies from three representations of time series, including the original, down-sampled, and smoothed data, was reported to be capable of extracting effective features for time series classification. Baseline full convolutional networks were proposed for time series classification [28]. The proposed baseline models were

reported to be pure end-to-end without demanding heavy pre-processing of the raw data or feature crafting, and achieve competitive performance to other state-of-the-art approaches, including the multilayer perceptron, fully convolutional network, and residual network. This network consists of a branching structure. The first branch is the convolutional part, whereas the second branch is an LSTM block which receives a time series in a transposed form as multivariate time series with a single time step. The outputs of the two branches are concatenated and then fed to a classifier. These models were reported to be able to enhance the performance of fully convolutional networks with a nominal increase in model size as well as require minimal data pre-processing.

In general, time-series classification has been recognized as an important and challenging area of research, particularly with respect to the demand for handling increasing availability of new data of time series. While numerous algorithms for time-series classification have been published in literature and the popularity of deep learning has been pervasive in many disciplines, only a few deep neural networks have been applied to solving time-series classification problems. A recent survey on promising applications of deep neural networks for time series classification in several areas can be found in [29].

Having reviewed recent deep neural networks for time-series classification, the purpose of this study is not to compare the performance of time-series classification between LSTM and CNN models. This study introduces the usefulness of constructing fuzzy recurrence plots of short time series that can be incorporated into LSTM models to improve the classification accuracy.

III. FUZZY RECURRENCE PLOTS

The development of constructing a fuzzy recurrence plot (FRP) of a time series was inspired with the concept of a recurrence plot (RP). An RP [30] is a visualization method for studying patterns of chaos in time series. An RP shows the times at which a phase-space trajectory approximately revisits the same area in the phase space.

Based on the Takens' embedding theorem [31] in the study of dynamical systems, a phase-space reconstruction of a time series can be obtained using an embedding dimension m and time delay τ . Let $X = \{\mathbf{x}\}$ be a set of phase-space vectors, in which $\mathbf{x}_i$ is the i -th state of a dynamical system in m -dimensional space and time delay τ . An RP is constructed as an $N \times N$ matrix in which an element (i, k) , $i = 1, \dots, N$, $k = 1, \dots, N$, is represented with a black dot if $\mathbf{x}_i$ and $\mathbf{x}_k$ are considered to be closed to each other. For a symmetrical RP, a threshold, denoted as ϵ , is used to define the similarity of a state pair $(\mathbf{x}_i, \mathbf{x}_k)$ as follows [32]

$$R(i, k) = H[d(\mathbf{x}_i, \mathbf{x}_k)], \quad (1)$$

where $R(i, k)$ is an element (i, k) of the recurrence matrix $\mathbf{R}$, $d(\mathbf{x}_i, \mathbf{x}_k)$ is a distance function of $\mathbf{x}_i$ and $\mathbf{x}_k$, and $H(\cdot)$ is the Heaviside function expressed as

$$H[d(\mathbf{x}_i, \mathbf{x}_k)] = \begin{cases} 1 & : d(\mathbf{x}_i, \mathbf{x}_k) \leq \epsilon \\ 0 & : d(\mathbf{x}_i, \mathbf{x}_k) > \epsilon \end{cases} \quad (2)$$

An FRP constructs the recurrence of the phase-space vectors as a grayscale image that takes values in $[0, 1]$ without requiring the similarity threshold parameter ϵ needed for the RP analysis. The formulation of an FRP is described as follows [19].

Let $\mathbf{V} = \{\mathbf{v}\}$ be the set of fuzzy clusters of the phase-space vectors. A binary relation $\tilde{\mathbf{R}}$ from $\mathbf{X}$ to $\mathbf{V}$ is a fuzzy subset of $\mathbf{X} \times \mathbf{V}$ characterized by a fuzzy membership function $\mu \in [0, 1]$. This fuzzy membership grade is the degree of relation of each pair $(\mathbf{x}_i, \mathbf{v}_j) \in \tilde{\mathbf{R}}$, $i = 1, \dots, N$, $j = 1, \dots, c$, that has the following properties [33]:

- Reflexivity: $\mu(\mathbf{x}_i, \mathbf{x}_i) = 1$, $\forall \mathbf{x} \in \mathbf{X}$,
- Symmetry: $\mu(\mathbf{x}_i, \mathbf{v}_j) = \mu(\mathbf{v}_j, \mathbf{x}_i)$, $\forall \mathbf{x} \in \mathbf{X}, \forall \mathbf{v} \in \mathbf{V}$, and
- Transitivity: $\mu(\mathbf{x}_i, \mathbf{x}_k) = \bigvee_{\mathbf{v}} [\mu(\mathbf{x}_i, \mathbf{v}_j) \wedge \mu(\mathbf{v}_j, \mathbf{x}_k)]$, $\forall \mathbf{x} \in \mathbf{X}$, which is called the max-min composition, where the symbols $\bigvee$ and $\wedge$ stand for max and min, respectively.

By specifying several clusters c for the data, the fuzzy c -means algorithm [34] is applied to identify the fuzzy clusters of the phase-space vectors and determine the similarity between the states and the fuzzy cluster centers. Based on this direct similarity measure, the inference of the similarity between the pairs of the states can be carried out using the max-min composition of a fuzzy relation.

The fuzzy membership of $\mathbf{x}_i$ assigned to a cluster center $\mathbf{v}_j$ of $\mathbf{X}$, $\mu(\mathbf{x}_i, \mathbf{v}_j)$ or denoted as μ_{ij} , is computed using the fuzzy c -means (FCM) algorithm [34] that attempts to partition N elements of $\mathbf{X}$ into a set of c fuzzy clusters, $1 < c < N$, by minimizing the following objective function F :

$$F = \sum_{i=1}^N \sum_{j=1}^c (\mu_{ij})^w \|\mathbf{x}_i - \mathbf{v}_j\|^2, \quad (3)$$

where $w \in [1, \infty)$ is the fuzzy weighting exponent, and F is subject to

$$\sum_{j=1}^c \mu_{ij} = 1, i = 1, \dots, N. \quad (4)$$

The minimization of the objective function of the FCM is numerically carried out by an iterative process of updating the fuzzy membership grades and cluster centers until the convergence or maximum number of iterations is reached. The fuzzy membership grades and cluster centers are updated as

$$\mu_{ij} = \frac{1}{\sum_{q=1}^c \left(\frac{\|\mathbf{x}_i - \mathbf{v}_j\|}{\|\mathbf{x}_i - \mathbf{v}_q\|} \right)^{2/(w-1)}}; \quad (5)$$

$$\mathbf{v}_j = \frac{\sum_{i=1}^N (\mu_{ij})^w \mathbf{x}_i}{\sum_{i=1}^N (\mu_{ij})^w}, j = 1, \dots, c. \quad (6)$$

Using the values of the fuzzy membership derived from the FCM and the fuzzy relation to represent the degree of recurrence among the phase-space vectors of the time series, an FRP can be visualized as a grayscale image by taking the complement of the FRP matrix that displays a black pixel if $\mathbf{x}_i = \mathbf{x}_k$, $i, j = 1, \dots, N$, otherwise a pixel with a shade of gray.

IV. LSTM NEURAL NETWORKS WITH FRPS

An LSTM neural network [35] is an artificial recurrent neural network (RNN) used in deep learning. Unlike conventional feedforward neural networks, an LSTM model has feedback loops that allow information of previous events to be carried on in the sequential learning process. Therefore, LSTM networks are effective in learning and classifying sequential data such as speech and video analysis [36], [37], [38], [39].

The internal state of an RNN is used as a memory cell to map real values of input sequences to those of output sequences that reflect the dynamic pattern of time series, and therefore is considered an effective algorithm for learning and modeling temporal data [40]. However, because an RNN uses sequential processing over time steps that can easily degrade the parameters capturing short-term dependencies of information sequentially passing through all cells before arriving at the current processing cell. This effect causes the gradient of the output error with respect to previous inputs to vanish by the multiplication of many small numbers being less than zero. This problem is known as vanishing gradients [41]. LSTM networks attempt to overcome the problem of the vanishing gradients encountered by conventional RNNs by using gates to keep relevant information and forget irrelevant information.

The difference between an LSTM neural network and a conventional RNN is the use of memory blocks for the former network instead of hidden units for the latter [42]. The input gate of an LSTM network guides the input activations into the memory cell and the output gate carry out the output flow of cell activations into the rest of the network. The forget gate allows the flow of information from the memory block to the cell as an additive input, therefore adaptively forgetting or resetting the cell memory. Thus, being less sensitive to the time steps makes LSTM networks better for analysis of sequential data than conventional RNNs. A common LSTM model is composed of a memory cell, an input gate, an output gate, and a forget gate. The cell memorizes values over time steps and the three gates control the flow of information into and out of the cell. The weights and biases to the input gate regulate the amount of new value flowing into the cell, while the weights and biases to the forget gate and output gate control the amount of information to remain in the cell and the extent to which the value in the cell is used to compute the output activation of the LSTM block, respectively.

The architecture for an LSTM block, in which the fuzzy membership grades of the FRP are the input values, is described in Figure 1. Furthermore, it can be visualized from Figure 1 that the use of FRPs can increase the feature of a time series from one dimension to N dimensions, where N is the number of the phase-space vectors of the time series, enhancing the training of the LSTM network. Figure 2 shows the forget, update, and output gates of the cell and hidden states. The mathematical expressions for the four gates of an LSTM block at time step t with the input of a FRP represented with its discrete fuzzy membership vector of N dimensions at time t , denoted as $\mathbf{u}_t = (\mu_t^1, \mu_t^2, \dots, \mu_t^N)^T$, are given as follows [35].

Fig. 1: For an LSTM layer, the first LSTM block takes the initial state of the network and the first time step of the FRP, and then computes the first output h_1 and the updated cell state c_1 , then at time step t , the LSTM block takes the current state of the network c_{t-1} , h_{t-1} and the next time step of the FRP at t , and then computes the output h_t and the updated cell state c_t . Note: Fuzzy membership and time steps of FRP is not drawn to scale, and the image gradient is a virtual representation of the fuzzy membership of recurrences.

Fig. 2: Cell and hidden states at time step t are processed by forget, update, and output gates.

Fig. 3: Architecture of an LSTM network for time-series classification.

$$\mathbf{f}_t = \sigma_g(\mathbf{W}_f \mathbf{u}_t + \mathbf{R}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \quad (7)$$

where $\mathbf{f}_t \in \mathbb{R}^N$ is the activation vector of the forget gate at time t , σ_g denotes the sigmoid function, $\mathbf{W}_f \in \mathbb{R}^{M \times N}$ is the input weight matrix, M refers to the number of hidden layers, $\mathbf{R}_f \in \mathbb{R}^{M \times M}$ is the recurrent weight matrix, $\mathbf{h}_{t-1} \in \mathbb{R}^M$ is the hidden state vector at time $(t-1)$, which is also known as the output vector of the LSTM unit and the initial $\mathbf{h}_0 = 0$, and $\mathbf{b}_f \in \mathbb{R}^M$ is the bias vector of the forget gate.

The input gate at time t , denoted as $\mathbf{i}_t \in \mathbb{R}^M$, is expressed as

$$\mathbf{i}_t = \sigma_g(\mathbf{W}_i \mathbf{u}_t + \mathbf{R}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \quad (8)$$

where $\mathbf{W}_i$, $\mathbf{R}_i$, and $\mathbf{b}_i$, are similarly defined as in Eq. (7).

The cell candidate vector that adds information to the cell state at time step t , denoted as $\mathbf{g}_t \in \mathbb{R}^M$, is defined as

$$\mathbf{g}_t = \sigma_c(\mathbf{W}_g \mathbf{u}_t + \mathbf{R}_g \mathbf{h}_{t-1} + \mathbf{b}_g), \quad (9)$$

where σ_c is hyperbolic tangent function ($\tanh$), $\mathbf{W}_g$, $\mathbf{R}_g$, and $\mathbf{b}_g$ are similarly defined as in Eq. (7).

The output gate, denoted as $\mathbf{o}_t \in \mathbb{R}^M$, which controls the level of the cell state added to hidden state is expressed as

$$\mathbf{o}_t = \sigma_g(\mathbf{W}_o \mathbf{u}_t + \mathbf{R}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \quad (10)$$

where $\mathbf{W}_o$, $\mathbf{R}_o$, and $\mathbf{b}_o$ are similarly defined as in Eq. (7).

The cell state vector at time step t , denoted as $\mathbf{c}_t \in \mathbb{R}^M$, is given by

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t, \quad (11)$$

where the initial values for $\mathbf{c}_0 = 0$, and the operator $\circ$ denotes the Hadamard product.

Finally, the hidden state vector at time step t is given by

$$\mathbf{h}_t = \mathbf{o}_t \circ \sigma_c(\mathbf{c}_t). \quad (12)$$

Figure 3 illustrates the architecture of an LSTM network for classification of time series. The network starts with an input layer of time series followed by an LSTM layer. To predict class labels, the LSTM network ends with a fully connected layer, a softmax layer, and a classification output layer.

V. DATABASE

The neuroQWERTY MIT-CSXPD database [43], which is publicly available from the PhysioNet (research resource for complex physiologic complex signals), was used in this study. The data contains keystroke logs collected from 85 subjects with and without PD. This dataset was collected and analyzed for investigating if the routine interaction with computer keyboards can be used to detect motor signs in the early stages of the PD subjects whose average time since diagnosis was 3.9 years, who were on PD medication but had no medication for the 18 hours before the typing test [17]. The subjects were recruited from two movement disorder units in Madrid, Spain, following the institutional protocols approved

by the Massachusetts Institute of Technology, USA, Hospital 12 de Octubre, Spain, and Hospital Clinico San Carlos, Spain.

Each data file collected includes the timing information collected during the sessions of typing activity using a standard word processor on a Lenovo G50-70 i3-4005U with 4MB of memory and a 15 inches screen running Manjaro Linux. The lengths of computer-key hold time series are highly variable, some have around 500, and others around 2500 time points.

Subjects were instructed to type as they normally would do at home and they were left free to correct typing mistakes only if they wanted to. The key acquisition software presented a temporal resolution of 3/0.28 (mean/standard deviation) milliseconds. Along with the raw typing collections, clinical evaluations were also performed on each subject, including UPDRS-III (Unified Parkinson's Disease Rating Scale: Part III) [44] and finger tapping tests.

VI. RESULTS AND DISCUSSION

TABLE I: Average accuracy (%), sensitivity (%), and specificity (%) rates obtained from classification of control and early PD using short time series of length = 50 and different methods.

Method	Accuracy	Sensitivity	Specificity
ID-CNN	64.71 $\pm$ 21.81	70.83 $\pm$ 24.92	58.33 $\pm$ 30.68
LSTM-Time series	63.43 $\pm$ 4.55	100 $\pm$ 0.00	0.00 $\pm$ 0.00
CNN-GoogLeNet	54.29 $\pm$ 18.63	65.00 $\pm$ 28.50	40.00 $\pm$ 27.89
CNN-AlexNet	37.14 $\pm$ 7.82	35.00 $\pm$ 28.50	40.00 $\pm$ 43.46
LSTM-FRP ($m=1$)	72.00 $\pm$ 15.92	90.00 $\pm$ 22.36	46.67 $\pm$ 50.55
LSTM-FRP ($m=3$)	65.14 $\pm$ 11.50	66.67 $\pm$ 33.33	63.33 $\pm$ 41.50
LSTM-FRP ($m=5$)	63.43 $\pm$ 4.55	100 $\pm$ 0.00	0.00 $\pm$ 0.00

TABLE II: Average accuracy (%), sensitivity (%), and specificity (%) rates obtained from classification of control and early PD using short time series of length = 30 and different methods.

Method	Accuracy	Sensitivity	Specificity
ID-CNN	61.71 $\pm$ 15.56	59.17 $\pm$ 32.26	66.67 $\pm$ 20.79
LSTM-Time series	62.10 $\pm$ 4.33	93.33 $\pm$ 14.91	10.00 $\pm$ 22.36
CNN-GoogLeNet	65.71 $\pm$ 21.67	70.00 $\pm$ 20.92	60.00 $\pm$ 27.89
CNN-AlexNet	68.57 $\pm$ 6.39	55.00 $\pm$ 11.18	86.67 $\pm$ 29.81
LSTM-FRP ($m=1$)	72.38 $\pm$ 11.24	78.33 $\pm$ 21.73	66.67 $\pm$ 23.57
LSTM-FRP ($m=3$)	81.90 $\pm$ 11.74	95.00 $\pm$ 11.18	66.67 $\pm$ 23.57
LSTM-FRP ($m=5$)	70.10 $\pm$ 9.63	95.00 $\pm$ 11.18	36.67 $\pm$ 24.72

Based on a previous study [46], outliers existing in the raw time series of the HC and PD individuals, where several data points are in the magnitude of 10^9 , were removed from the time series. Because the purpose of this study is to classify the signals of short lengths, short segments from the start of the original time series were selected for testing the use of the LSTM neural-network model with FRPs.

Figure 4 shows two short time series of 30 time steps of the computer-key hold durations, extracted from the start of the original time series, recorded from a healthy control (HC) and early PD subjects, and their associated RPs and FRPs with an embedding dimension = 1, and time delay =

Fig. 4: Time series, and corresponding RPs, and FRPs of a control subject and an early PD subject. Note: RPs are displayed as the sparsity patterns of the RP matrices to make the plots visualizable, while FRPs are shown as grayscale images of the FRP matrices.

1, similarity tolerance = 0.1 for RPs, and number of clusters = 6 for FRPs. It can be observed from Figure 4 that the

binary information obtained from the RPs are very sparse for both HC and early PD subjects. The RPs become sparser with

Fig. 5: Training of LSTM neural network with: (a) short computer-key hold time series, and (b) FRPs of short computer-key hold time series.

increased embedding dimensions of 3 and 5. The FRPs display rich information as texture images as the values of the fuzzy membership grades about the recurrences of the underlying dynamics of the time series of the two subjects. Therefore, the fuzzy membership grades of the phase-space vectors of the time series were used as the feature with dimensions being equal to the number of the phase-space vectors for training and

classification using the LSTM network.

The LSTM neural network of the Matlab Deep Learning Toolbox (R2018b) was used in this study. The number of hidden layers = 100, maximum number of epochs = 200, and learning rate = 0.001. L_2 regularization was used for the biases, input weights, and recurrent weights to reduce model overfitting. To construct FRPs, FCM parameters that are the

Fig. 6: Training of computer-key hold time series with 1D-CNN: (a) time-series length = 30, and (b) time-series length = 50.

fuzzy weighting exponent w , the number of clusters c , and the maximum number of iterations were chosen to be 2, 6, and 100, respectively. Given an embedding dimension m , time delay τ , the number of the phase-space vectors of a time series of length L , which is also the number of feature dimensions used in the LSTM network, is calculated as $N = L - (m - 1)\tau$. Thus, keeping $\tau = 1$, the feature dimensions for $m = 1, 3$, and 5 for $L = 50$ are 50, 48, and 46, respectively; and for

$L=30$, the feature dimensions for $m = 1, 3$, and 5 are 30, 28, and 26, respectively. The time delay was set to be 1 for both lengths of the time series. There are methods for estimating the time delay and embedding dimension for the phase-space reconstruction such as the false nearest neighbor (FNN) and average mutual information (AMI), respectively, where the first local minima of the FNN and AMI functions are indicative of the embedding dimension and time delay, respectively

(a) Control

(b) PD

Fig. 7: Scalograms of time series of the control and early PD subjects shown in Figure 4.

[45]. However, estimate for the embedding dimension and time delay for the phase-space reconstruction of each time series of the computer-key hold duration is not convenient for implementing the LSTM network because of the variation in the feature dimensions. It has been reported that the use of time delay = 1 is well adopted for studying nonlinear time series [46], and several embedding dimensions were adopted in this study.

Tables I and II show the results of classifying HC and early PD subjects using the short time series of lengths 50 and 30, respectively. Values of the accuracy, sensitivity, and specificity are based on the average of five repetitions of the 10-fold cross validation results. The sensitivity, which is also called the true positive rate, measures the proportion of actual positives (early PD subjects) that are correctly identified as such; whereas specificity, which is also known as the true negative rate, measures the proportion of actual negatives (control subjects) that are correctly identified as such. The sensitivity (SEN) and specificity (SPE) are computed as follows.

$$SEN = \frac{TP}{TP + FN}, \quad (13)$$

where TP and FN are the numbers of true positives and false negatives, respectively, which are obtained from the (2×2) confusion matrix for each classification method.

$$SPE = \frac{TN}{TN + FP}, \quad (14)$$

where TN and FP are the numbers of true negatives and false positives, respectively, which are obtained from the (2×2) confusion matrix for each classification method.

For the direct use of the time series (LSTM-Time series) of length = 50, accuracy = 63%, sensitivity = 100%, and specificity = 0%. For the use of FRPs of the time series (LSTM-FRP) of length = 50, with embedding dimension $m=1$, accuracy = 72%, sensitivity = 90%, and specificity = 47%; with $m=3$, accuracy = 65%, sensitivity = 67%, and specificity = 63%; and for $m=5$, accuracy = 63%, sensitivity = 100%, and specificity = 0%. The accuracy results obtained from the use FRPs are equal to or higher than the accuracy obtained from the direct use of the time series. With the use of FRPs as features, the accuracy decreases with increasing value for m , where the best accuracy is obtained with $m=1$. The direct use of the time series gives 100% for sensitivity but 0% for specificity, such results are not practically helpful because all are identified as early PD, which may lead to a large false positive rate.

For the time series of shorter length = 30, the direct use of the time series for LSTM network training and validation results in accuracy = 62%, sensitivity = 93%, and specificity = 10%. Once again, while the sensitivity is very high, the specificity is very low. The results obtained from the direct use of the time series for both time-series of lengths = 50 and 30 are similar in accuracy, sensitivity, and specificity. For the use of FRPs of the time series of length = 30, with $m=1$, accuracy = 72%, sensitivity = 78%, and specificity = 67%; with $m=3$, accuracy = 82%, sensitivity = 95%, and specificity = 67%; and for $m=5$, accuracy = 70%, sensitivity = 95%, and specificity = 37%. The FRPs with $m=3$ give the highest accuracy (82%) among the others. All accuracy results obtained from the FRPs are higher than those obtained from the direct use of the time series.

The standard deviations of the results obtained from the use of FRPs for signal length = 50 with $m=1$ and 3 are higher than those of the raw time series, because some rates of accuracy obtained from the FRPs reached 100% and 85%, respectively. However, even the average accuracy obtained from the use of the raw signals of the same length is lower than those obtained from the FRPs, the average specificity (true negative rate that is the ability to correctly identify those without PD) obtained from using the raw signals is zero, which is obviously not useful at all (Table I). Similarly, the standard deviations of the accuracy results obtained from the use of FRPs for signal length = 50 with $m=1, 3$, and 5 are higher than those of the raw time series, because some rates of accuracy obtained from the FRPs reached 83%, 100%, and 86%, respectively. Once again, even the average accuracy

obtained from the use of the raw signals of the same length is lower than those obtained from the FRPs, the average specificity obtained from using the raw signals is very low (10%), which is not useful for the classification (Table II).

As an illustration for the performance of the FRPs preferred to that of the direct use of the short time series, Figure 5 shows the training progresses and metrics of the LSTM network with the time series of length = 30 and the associated FRPs. Each iteration is an estimation of the gradient and an update of the network parameters. The accuracy of the direct use of the short time series converged to around 60% and the loss around 0.7, while the accuracy and loss for the use of the FRPs converged to 100% and 0, respectively. Furthermore, it can be observed that for the direct use of the time series in the LSTM network, the longer time series ($L = 50$) yields higher accuracy than the shorter ones ($L=30$), but for the use of the FRPs, the accuracy depends on the selection of the embedding dimension (m) parameter, suggesting the influence of the embedding dimension over the time-series length, and potential exploration of the use of FRPs for classifying short sequences. The accuracy values obtained from the direct input of the time series of two different lengths are similar, while these are highly variable for the FRPs. The augmentation of more training data for the two classes would be expected to reduce the accuracy variation obtained from the input of the FRPs. Another potential factor for the higher accuracy obtained from the FRPs of the shorter time series is the redundancy with respect to the higher feature dimensions provided by the FRPs of the longer time series. This factor also suggests the ability of FRPs to extract effective dynamical features from short time series with an appropriate selection of collective parameters for the phase-space reconstruction of different time series.

Tables I and II also show the average cross-validation results of classifying HC and early PD subjects obtained from two popular pre-trained deep CNNs known as GoogLeNet [47] and AlexNet [48], using the short time series of lengths 50 and 30, respectively. The implementations of these two pre-trained CNN models for time-series classification were based on the work proposed in [51]. It is known that training a deep CNN from scratch is computationally expensive and requires a large amount of training data. In this study, a large amount of training data is not available. Thus, taking advantage of existing deep CNNs that have been trained on large data sets for conceptually similar tasks is desirable. This leveraging of existing neural networks is called transfer learning, which has recently been applied to time-series classification [49], [50]. GoogLeNet and AlexNet, which were pretrained for image recognition, were adopted to classify transformed images of the short time series based on a time-frequency representation [51]. Scalograms were used to obtain the RGB images of time-frequency representations of the time series. A scalogram is the absolute value of the continuous wavelet transform (CWT) coefficients of a signal. Parameters used for obtaining the scalograms of the time series and modifying GoogLeNet and AlexNet for the time-series classification are the same as described in [51]. Figure 7 shows two scalograms of two time series recorded from a control subject and an early

PD subject, respectively. For the time series of length = 50, classification results obtained from both deep convolutional networks GoogLeNet (CNN-GoogLeNet) and AlexNet (CNN-AlexNet) are lower than LSTM-Time series and LSTM-FRP. CNN-AlexNet has the lowest average accuracy (37%). For the time series of shorter length = 30, classification results obtained from CNN-AlexNet (69%) is higher than those obtained from the CNN-GoogLeNet and LSTM-Time series, but lower than the LSTM-FRP.

One-dimensional CNN (1D-CNN) was also applied as a baseline model for directly training and classifying the short time series. The CNN architecture of the Matlab Deep Learning Toolbox (2019a) was created as follows. The input size of the time series to the CNN model was specified as $L \times 1 \times 1$, where L is the length of the time series. A convolutional layer was constructed with 16 filters that have the height and width of 3. Padding was applied to the input along the edges. Padding was set so that the output size was the same as the input size where the stride is 1. A fully connected layer with an output size of 384 in the hidden layer and 2 as the output classes were specified. The maximum number of epochs was 400 for training the 1D-CNN. The average cross-validation results of classifying HC and early PD subjects using the time series of lengths = 50, and 30 obtained from the 1D-CNN are shown in Tables I and II, respectively. For both time series of lengths = 50 and 30, the 1D-CNN model provides similar classification accuracy rates in comparison with the LSTM directly using the time series (LSTM-Time series). However, the sensitivity and specificity obtained from the 1D-CNN are more balanced than those obtained from the LSTM-Time series. For time series length = 50, LSTM-FRP models with $m = 1$ and 3 outperform the 1D-CNN, while the accuracy obtained from LSTM-FRP with $m = 5$ is slightly slower than the 1D-CNN. For time series length = 30, all LSTM-FRP models ($m = 1, 3$, and 5) provide much higher classification accuracy rates than the 1D-CNN. Figure 6 shows the training processes of the 1D-CNN with the time series of lengths = 30 and 50. The converged accuracy and loss rates obtained from the 1D-CNN are less desirable than those from the LSTM-FRP (Figure 5(b)).

Once again, the idea of using FRPs of short raw time series for classification with an LSTM network is original, and there are no previous reports on this kind of research. Furthermore, on the comparison between the LSTM-based classification using raw time series and FRPs of raw time series, an appropriate construction of an FRP in order to correctly reflect the dynamics underlying the signal is mainly subject to the selection of a good embedding dimension m , whereas m being not applicable for the case of LSTM-based classification of raw time series. Hence, three different values of m were chosen for the construction of FRPs of the raw signals to gain insight into the influence of the embedding dimension over the classification. The LSTM-based classification using any of the three values for m (except with $m = 5$ for length = 50, the accuracy rates of LSTM-FRP and LSTM-Time series are the same) specified for constructing the FRPs outperformed those using LSTM with raw time series and the two pre-trained CNN models.

The aim of this study is to show that the incorporation of

FRPs of short time series, which creates several dimensions or channels for each time step of the time series, as input into the LSTM model can improve the LSTM-based classification and outperform the direct classification of time series using a 1D-CNN model as the baseline. The number of dimensions associated with the time steps of a sequence is considered as the number of features flown through an LSTM layer, which constitutes to the LSTM layer architecture as described by LSTM networks in the Matlab Deep Learning Toolbox (R2018b and R2019a). The core components of an LSTM network are a time-series input layer and an LSTM layer. The input layer inputs time series into the network. An LSTM layer learns long-term dependencies between time steps of the data.

VII. CONCLUSION

An approach for classification of short time series for differentiating healthy control from early PD subjects using a deep learning LSTM network of FRPs has been presented and discussed. The purpose of classifying short time series is to reduce potential discomfort caused to individuals participating in the test. The use of the fuzzy membership grades of the recurrences of the phase-space vectors of the time series contributes to the increase of the feature dimensions for the learning of the LSTM network and therefore can improve the classification over the direct input of the time series. The results obtained from the FRPs are encouraging for the collection of practical data recorded from participants and their usage for the classification task.

The selection of a value for the embedding dimension of the phase-space reconstruction of the time series can influence the classification. Designing an optimal procedure for estimating collective embedding dimensions as well as time delays for the sets of time series of different classes would be worth investigating to improve the effective use of FRPs in LSTM networks for short time-series classification. Furthermore, application of bidirectional LSTM (BiLSTM) networks [36] that is based on the concept of bidirectional RNNs [52] to simultaneously learn from past (backward) and future (forward) dependencies between time steps of time series or sequence data is worth exploring for improving the classification.

In this study, the FRPs of short time series were used as sequential data for classification. The inherent texture of FRPs can be extracted with several methods of texture analysis in image processing [14] or pre-trained image-based deep-learning models for classification as a transformation of one-dimensional signals into two-dimensional images. Furthermore, the approach proposed in this study is not limited to differentiating healthy control from early PD subjects with short time series but can also be applied to other problems concerning with machine learning and classification with short time-series data.

REFERENCES

- [1] S.J. Chinta, J.K. Andersen, "Dopaminergic neurons", *Int J Biochem Cell Biol.*, vol. 37, 942-946, 2005.
- [2] Statistics on Parkinson's, 2017 *Parkinson's Disease Foundation*. Available: http://www.pdf.org/parkinson_statistics.
- [3] P. Rizek, N. Kumar, M.S. Jog, "An update on the diagnosis and treatment of Parkinson disease", *CMAJ*, vol. 188, pp.1157-1165, 2016.
- [4] A. Elkouzi, "What is Parkinson's?", *Parkinson's Foundation*. Available: <https://parkinson.org/understanding-parkinsons/what-is-parkinsons>.
- [5] G. DeMaagd, A. Philip, "Parkinson's disease and its management: Part 1: disease entity, risk factors, pathophysiology, clinical presentation, and diagnosis, *P&T*, vol. 40, pp. 504-532, 2015.
- [6] M. Hariz, P. Blomstedt, L. Zrinzo, "Future of brain stimulation: new targets, new indications, new technology", *Mov Disord.*, vol. 28, 1784-1792, 2013.
- [7] M. Hariz, "Deep brain stimulation: new techniques", *Parkinsonism Relat Disord.*, vol. 20 Suppl 1:S192-S196, 2014.
- [8] F.N. Emamzadeh, A. Surguchov, "Parkinson's disease: Biomarkers, treatment, and risk factors", *Frontiers in Neuroscience*, vol. 12, 612, 2018.
- [9] W.H. Oertel, "Recent advances in treating Parkinson's disease", *F1000Research*, vol. 6, 260, 2017.
- [10] J.M. Hausdorff, "Gait dynamics in Parkinson's disease: common and distinct behavior among stride length, gait variability, and fractal-like scaling", *Chaos*, vol. 19, 026113, 2009.
- [11] P.H. Chen, R.L. Wang, D.J. Liou, J.S. Shaw, "Gait Disorders in Parkinson's Disease: Assessment and Management", *International Journal of Gerontology*, vol. 7, pp. 189-193, 2013.
- [12] W. Zeng, C. Wang, "Parkinson's disease classification using gait analysis via deterministic learning", *Neuroscience Letters*, vol. 633, pp. 268-278, 2016.
- [13] P. Ren, S. Tang, F. Fang, L. Luo, L. Xu, M.L. Bringas-Vega, D. Yao, K.M. Kendrick, P.A. Valdes-Sosa, "Gait rhythm fluctuation analysis for neurodegenerative diseases by empirical mode decomposition", *IEEE Trans Biomedical Engineering*, vol. 64, pp. 52-60, 2017.
- [14] T.D. Pham, "Texture classification and visualization of time series of gait dynamics in patients with neuro-degenerative diseases", *IEEE Trans Neural Systems and Rehabilitation Engineering*, vol. 26, pp. 188-196, 2018.
- [15] S. Hemm, D. Pison, F. Alonso, A. Shah, J. Coste, J.J. Lemaire, K. Wardell, "Patient-specific electric field simulations and acceleration measurements for objective analysis of intraoperative stimulation tests in the thalamus", *Frontiers in Human Neuroscience*, vol. 10, article 577, 2016.
- [16] L. Kribus-Shmiel, G. Zeilig, B. Sokolovski, M. Plotnik M, "How many strides are required for a reliable estimation of temporal gait parameters? Implementation of a new algorithm on the phase coordination index", *PLoS ONE*, vol. 13, e0192049, 2018.
- [17] L. Giancardo, A. Sanchez-Ferro, T. Arroyo-Gallego, I. Butterworth, C. S. Mendoza, P. Montero, M. Matarazzo, J. A. Obeso, M. L. Gray, R. San Jose Estepar, "Computer keyboard interaction as an indicator of early Parkinson's disease", *Scientific Reports*, vol. 6, 34468, 2016.
- [18] T.A.L. Tavares, G.S. Jefferis, M. Koop, B.C. Hill, T. Hastie, G. Heit, H.M. Bronte-Stewart, "Quantitative measurements of alternating finger tapping in Parkinson's disease correlate with UPDRS motor disability and reveal the improvement in fine motor control from medication and deep brain stimulation", *Mov. Disord.*, vol. 20, pp. 1286-1298, 2005.
- [19] T.D. Pham, "Fuzzy recurrence plots", *EPL (Europhysics Letters)*, vol. 116, 50008, 2016.
- [20] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, "Recurrent neural networks for multivariate time series with missing values", *Scientific Reports*, vol. 8, 6085, 2018.
- [21] G. Hinton, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups", *IEEE Signal Process. Mag.*, vol.29, pp. 82-97, 2012.
- [22] I. Sutskever, O. Vinyals, Q.V. Le, "Sequence to sequence learning with neural networks", in *Advances in Neural Information Processing Systems*, pp. 3104-3112, 2014.
- [23] D. Bahdanau, K. Cho, Y. Bengio, "Neural machine translation by jointly learning to align and translate", in: *Proceedings of 3rd International Conference on Learning Representations*, 2015.
- [24] P. Malhotra, T.V. Vishnu, L. Vig, P. Agarwal, G. Shroff, "TimeNet: Pre-trained deep recurrent neural network for time series classification", in: *25th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 607-612, 2017.
- [25] N. Mehdiyeva, J. Lahanna, A. Emrich, D. Enke, P. Fetteke, P. Loos, "Time series classification using deep learning for process planning: A case from the process industry", *Procedia Computer Science*, vol. 114, 242-249, 2017.
- [26] X. Shi, Z. Chen, H. Wang, D.Y. Yeung, W.K. Wong, W.C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting", in: *Advances in Neural Information Processing Systems*, vol. 28, pp. 802-810, 2015.

- [27] Z. Cui, W. Chen, Y. Chen, "Multi-scale convolutional neural networks for time series classification", in: *CoRR* abs/1603.06995, arXiv: 1603.06995, 2016.
- [28] Z. Wang, W. Yan, T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline", in: *2017 International Joint Conference on Neural Networks*, pp. 1578-1585, 2017.
- [29] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.A. Muller, "Deep learning for time series classification: a review", *Data Mining and Knowledge Discovery*, pp. 1-47, 2019. <https://doi.org/10.1007/s10618-019-00619-1>
- [30] J.P. Eckmann, S.O. Kamphorst, D. Ruelle, "Recurrence plots of dynamical systems", *Europhysics Letters*, vol. 5, pp. 973-977, 1987.
- [31] F. Takens, "Detecting strange attractors in turbulence", *Lecture Notes in Mathematics*, vol. 898, pp. 366-381, 1981
- [32] N. Marwan, M.C. Romano, M. Thiel, J. Kurths, "Recurrence plots for the analysis of complex systems", *Physics Reports*, vol. 438, 237, 2007.
- [33] L.A. Zadeh, "Similarity relations and fuzzy orderings", *Information Sciences*, vol. 3, pp. 177-200, 1971.
- [34] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [35] S. Hochreiter, J. Schmidhuber, "Long short-term memory", *Neural Computation*, vol. 9, pp. 1735-1780, 1997.
- [36] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks*, vol. 18, pp. 602-610, 2005.
- [37] A. Graves, N. Jaitly and A. Mohamed, "Hybrid speech recognition with Deep Bidirectional LSTM", in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, Olomouc, Czech Republic, 2013, pp. 273-278.
- [38] R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D.T. Toledano, J. Gonzalez-Rodriguez "Language Identification in Short Utterances Using Long Short-Term Memory (LSTM) Recurrent Neural Networks", *PLoS ONE*, vol. 11, e0146917, 2016.
- [39] L. Zhenyang, K. Gavriluk, E. Gavves, M. Jain, C.G.M. Snoek, "VideoLSTM convolves, attends and flows for action recognition", *Computer Vision and Image Understanding*, vol. 166, pp. 41-50, 2018.
- [40] T. Mikolov, S. Kombrink, L. Burget, J.H. Cernocky, S. Khudanpur, "Extensions of recurrent neural network language model", in 2011 IEEE International Conference on Acoustics, Speech and Signal Processing, Prague, Czech Republic, 2011, pp. 5528-5531.
- [41] R. Pascanu, T. Mikolov, Y. Bengio, "On the difficulty of training recurrent neural networks", in *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA, 2013, pp. III-1310-III-1318.
- [42] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, "LSTM: A search space odyssey", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, pp. 2222-2232, 2017.
- [43] neuroQWERTY MIT-CSXPD Dataset, *PhysioNet*. Available: <https://www.physionet.org/physiobank/database/nqmitchxpd/>.
- [44] P. Martinez-Martin, A. Gil-Nagel, L.M. Gracia, J.B. Gomez, J. Martinez-Sarries, F. Bermejo, "Unified Parkinson's disease rating scale characteristics and structure. The Cooperative Multicentric Group", *Mov. Disord.*, vol. 9, pp. 76-83, 1994.
- [45] H. Kantz, T. Schreiber. *Nonlinear Time Series Analysis*. Cambridge: Cambridge University Press, 2004.
- [46] T.D. Pham, "Pattern analysis of computer keystroke time series in healthy control and early-stage Parkinson's disease subjects using fuzzy recurrence and scalable network features", *Journal of Neuroscience Methods*, vol. 307, pp. 194-202, 2018.
- [47] C. Szegedy, et al., "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 1-9.
- [48] A. Krizhevsky, I. Sutskever, G.E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84-90, 2017
- [49] S. Karimi-Bidhendi, F. Munshi, A. Munshi, "Scalable classification of univariate and multivariate time series", in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, posted online 24 Jan 2019. <https://ieeexplore.ieee.org/document/8621889>.
- [50] H.I. Fawaz, et al., "Transfer learning for time series classification", in *2018 IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, posted online 24 Jan 2019. <https://ieeexplore.ieee.org/document/8621990>.
- [51] MathWork, "Classify time series using wavelet analysis and deep learning". <https://mathworks.com/help/deeplearning/examples/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html#d117e22181>. Accessed 25 April 2019.
- [52] M. Schuster, K.K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Trans Signal Processing*, vol. 45, pp. 2673-2681, 1997.

Tuan D. Pham (SM'01) received his Ph.D. degree in Civil Engineering from the University of New South Wales, Sydney, Australia, in 1995. He has been a Professor of Biomedical Engineering, leading the Biomedical Pattern Recognition Group at Linköping University, Sweden. He held previous positions as Professor and the Leader of the Aizu Research Cluster for Medical Engineering the Informatics and Head of the Medical Image Processing Laboratory at the University of Aizu, Japan; and the Bioinformatics Research Group Leader at the School of Engineering and Information Technology, the University of New South Wales, Australia. His teaching and research span across several disciplines of computer science and engineering. His current research areas include artificial intelligence, image processing, time series, nonlinear dynamics, novel methods in pattern recognition and machine learning applied to medicine, physiology, biology, and health. He has been serving as Section Editor and Associate Editor of several international journals, conference chair, and keynote speaker. Dr. Pham has been internationally recognized by his peers as a highly prolific scientist, publishing extensively as the lead author on various topics in books by well-known publishers, well-respected journals, and refereed conferences.

Karin Wårdell received her PhD in Biomedical Instrumentation in 1994. Since 2002 she has been a full professor of Biomedical Engineering at Linköping University and the Head of the Neuro-engineering Group. The research is focused on deep brain stimulation, optical techniques in neurosurgery, neuronavigation and brain microcirculation. She is an IAMBES and EAMBES Fellow.

Anders Eklund received the M.Sc. degree in applied physics and electrical engineering in 2007, and the Ph.D. degree in medical informatics in 2012, both from Linköping University, Sweden. During 2012 - 2014 he was a postdoc at the Virginia Tech Carilion Research Institute, Roanoke, USA. He is currently an associate professor at Linköping University, working with image processing, machine learning and statistics for neuroimaging.

Göran Salerud was born in Vimmerby, Sweden, on January 7th, 1954. He received the MSc degree from Linköpings universitet in 1979 and the Ph.D. degree in Biomedical Engineering in 1986 also from Linköping University, Sweden. Dr. Salerud was awarded the Young investigator's award in Microcirculation in 1986. He has held the position as director of studies in Biomedical Engineering from 1986 to 1990 and again from 2000 and forward. Since 1986 he also has been responsible for the curriculum in Biomedical Engineering. His passion and interest in learning science and microcirculation research has resulted in as invited speaker at international meetings, workshops and conferences. He has been appointed as professor in Biomedical Instrumentation since 2003. Professor Salerud's main research fields are biomedical optics, biomedical signal and image processing. Developing research fields are spectroscopy, pattern recognition and the use of information technology in Biomedical Engineering research especially in microcirculation.