A data-driven approach for detecting gait events during turning in people with Parkinson's disease and freezing of gait

DOI: 10.1016/j.gaitpost.2020.05.026
Handle: http://hdl.handle.net/1942/32615

1 INTRODUCTION

Freezing of gait (FOG) is a devastating gait disorder manifesting itself frequently in Parkinson's disease (PD). FOG is defined by Nieuwboer and Giladi as "an episodic inability to generate effective stepping in the absence of any known cause other than Parkinsonism or high-level gait disorder" [1]. Patients describe a FOG episode as "the feeling that their feet are glued to the ground" [2]. FOG occurs most reliably during complex gait tasks, such as turning with fast speed or walking while performing a dual task [3]. To study FOG and the highly abnormal steps leading up to it, gait analysis has been adopted, using instrumented gait analysis systems based on 3D motion capturing techniques [4, 5]. The gait data generated from these systems are typically normalized to a gait cycle. This normalization requires accurate timing of initial contact (IC) and end contact (EC) of the foot. The detection of these gait events is based on visual inspection by a clinical expert [4, 5]. Due to the small and shuffling steps, reduced heel strike and inadequate swing phase prior to FOG [6], and altered steps between FOG episodes [7], this process is imprecise. In addition, visual detection of gait events are more time consuming, during more complex gait tasks such as 360 degree turning [8].

To find a solution for this problem, this paper aimed to investigate the validity of an automated approach for gait cycle detection. Heuristic based methods are most commonly used to automatically detect the defined gait events. These methods utilize domain knowledge to extract kinematic features that correlate with the timing of gait events. However, owing to the variable gait patterns apparent in PD patients with FOG, these features do not necessarily generalize to this pathology. Furthermore, heuristic methods typically lack validation in challenging movement sequences, such as turning and dual tasking, commonly used to trigger FOG [3].

Powered by large datasets, data-driven approaches, such as recurrent neural networks (RNN), have shown great success in many problems that contain temporal information. These approaches can infer relevant features directly from the raw input data, a technique called end-to-end learning [9]. The success of these approaches for gait event detection was recently demonstrated [10], utilizing a long short-term memory (LSTM) network to classify gait events in children. The focus of this paper was to provide a robust tool to automatically annotate gait events for PD patients with FOG during straight-line gait and turning, which can be trained end-to-end with minimal data pre-processing.

2 MATERIAL AND METHODS

2.1 Sequence to Sequence Learning

In this study, gait event detection is cast as a sequence to sequence classification problem [11]. Each input sample x is associated with a ground-truth label y_{obs}. A model is trained to learn a function $f : x \rightarrow y$ that transforms a given input sequence $X = x_0, \ldots, x_t$ into an output sequence $Y = y_0, \ldots, y_t$ that closely resembles the manual annotations Y_{obs}. Separate datasets are generated for each gait event by encoding each sample as a binary vector $y_{obs} \in \{0, 1\}$. The input sequence $X_{in} \in \mathbb{R}^{s \times t}$ is comprised of a spatial dimension s and time dimension t.

2.2 Dataset

An existing dataset [12] including fifteen PD patients with freezing of gait (FOG) was used. Patients were diagnosed by a Movement disorders specialist as having PD and were classified as freezers based on the first question of the New Freezing of Gait Questionnaire (NFOG-Q): “Did you experience “freezing episodes” over the past month?" [13]. The study was approved by the local ethics committee of the University Hospital Leuven and all subjects gave written informed consent.

2.3 Procedure

Gait analysis was performed using an eight camera Vicon 3D motion analysis system recording at a sample frequency of 100hz. Thirty-four retro-reflective markers were placed on anatomical landmarks according to the full body plug-in-gait model. All experiments were done during the off-state of the subjects medication cycle, except for clinical testing which was conducted ON-medication. The subjects were instructed to complete three straight-line and six 360 degree turning trials, according to the standardized protocol described in a previous paper [12]. Two researchers, blinded for NFOG-Q score, visually detected all FOG episodes. The onset of FOG, defined as the start of delayed knee flexion, was detected by visual inspection of the knee-angle data (flexion-extension) in combination with the 3D images. Termination of FOG was determined at the time point when at least two consecutive movement cycles were regained. These two gait cycles were not included in the FOG episode [14]. The dataset was partitioned into two groups. Trials that contained a freezing episode were indicated as freezing trials (FOG) and trials without a freezing episode were termed as functional gait trials (FG). For both groups, the left-sided gait events were manually annotated based on visual inspection of the 3D marker coordinates. Furthermore, the highly varied gait data between onset and termination of a FOG episode was excluded during evaluation.

2.4 Deep Learning Models

2.4.1 Recurrent Neural Network

Recurrent neural networks (RNN) are commonly associated with the modelling of sequential data. Recurrent architectures solve the sequence to sequence learning by iterating over the following equation [11]:

$$h_t = \sigma(x_t W^{xh} + h_{t-1} W^{hh}),$$

$$y_t = h_t W^{hy}.$$

The weight matrices are represented by W, with superscripts representing from-to relationships. The terms x_t and y_t are the input and output at time t, respectively. However, computing the complete gradient by unrolling over long temporal sequences can lead to vanishing or exploding of the gradient [15]. Long Short Term Memory (LSTM) networks [16] extend RNNs with memory cells, instead of recurrent units, to store and output information. An LSTM cell is comprised out of four gates, formally defined as:

$$i_t = \sigma(x_t W^{xi} + h_{t-1} W^{hi} + C_{t-1} W^{ci}),$$

$$f_t = \sigma(x_t W^{xf} + h_{t-1} W^{hf} + C_{t-1} W^{cf}),$$

$$o_t = \sigma(x_t W^{xo} + h_{t-1} W^{ho} + C_{t-1} W^{co}),$$

$$\tilde{c}_t = tanh(x_t W^{xc} + h_{t-1} W^{hc}),$$

$$c_t = \sigma(f_t * c_{t-1} + i_t * \tilde{c}_t),$$

$$h_t = tanh(c_t) * o_t.$$

The weight matrices are represented by W, with superscripts representing from-to relationships. The term x_t is the input to the memory cell at time t. The terms σ and $tanh$ are the sigmoid and hyperbolic tangent activation functions. The terms i, f, o, and c are the input gate, forget gate, output gate, and cell activation vectors, respectively. The multiplicative gates allow the LSTM cells to store and access information over long periods of time, thereby avoiding the aforementioned vanishing and exploding gradient problem. Our recurrent model consists out of one to three LSTM layers mapping the input x_t to a p-dimensional time series, where $p \in \{2, 4, 8, 16, 32\}$. Our model is based on the architecture of [10], who successfully exploited LSTMs for gait event detection in children.

2.4.2 Convolutional Neural Network

Results from a systematic evaluation of convolutional neural networks (CNN) and recurrent neural networks (RNN) suggests that the common association between RNNs and sequence modelling should be reconsidered, and that CNNs should be regarded as the natural starting point for sequence modelling [17]. The authors show that a simple temporal convolutional neural network (TCN) outperforms RNNs, such as LSTMs. The nature of our sequence to sequence learning framework is based upon two constraints: (1) given an input sequence $x_0, \ldots, x_t \in X$ the network produces an output $y_0, \ldots, y_t \in Y$ of the same length, and (2) that the mapping satisfies the causal constraint, such that y_t only depends on the observations $x_0, \ldots, x_t$ and not on x_{t+1}, i.e. there is no leakage of information from future observations. To satisfy the first constraint, the TCN network utilises 1D fully convolutional layers (FCN) [18]. FCN layers preserve the time dimension throughout the network by omitting local pooling layers, thereby ensuring that each hidden layer is the same length as the input sequence. To satisfy the second constraint, the TCN network utilises causal convolutions, i.e. convolutions that ensure that an output at time t is only convolved with elements from time t and earlier. Our model consists of one to three repeating blocks of causal convolutions mapping the input to a p-dimensional time series with a kernel size of five, where $p \in \{2, 4, 8, 16, 32\}$. The convolutions are followed by batch normalization [19], ReLU activation, 1x convolution (bottleneck) [17], and dropout [20]. The repeating blocks are concatenated to form a residual temporal convolutional neural network, based on the architecture of [17].

2.4.3 Hyperparameter Optimization and Model Training

The gait trials were partitioned into equal length time windows of 128 samples. Each input sample x_t is comprised out of the sagittal plane kinematics of the hip, knee, and ankle of both legs. Additionally, angular velocities were extracted by using first order finite difference equations. The input sequence is thus a matrix of $X_{in} \in \mathbb{R}^{12\times128}$. All signals were low-pass filtered with a cut-off frequency of 7 Hz [21] using a zero phase fourth order butter-worth filter. Separate models were trained for EC and IC by encoding the manual annotations as a binary vector $y_{obs} \in \{0, 1\}$.

The convolutional and LSTM layers are followed by a fully connected layer which learns the non-linear function $f : x \rightarrow y$ from the proposed feature space that best separates the two classes $y_{obs} \in \{0, 1\}$ by minimizing a certain loss function. Since gait events occur sparsely compared to non-events, class imbalance is accounted for by using a weighted binary cross entropy loss function [10]. The number of residual blocks and filters (CNN) or layers and units (LSTM) were optimized using the tree-structured Parzen estimator (TPE) [22], a Bayesian optimization approach which was proven to have an overall better test performance than grid and random search [23]. The models were trained for 150 epochs and are visualized in Figure 1. To ensure generalization to new subjects, a leave one subject out cross validation approach was utilized, as visualized in Figure 2. The optimization algorithm was run for 10 iterations and the cross validated loss was the objective function to be minimized.

2.5 Heuristic Method

The deep learning models were quantitatively compared to a commonly used heuristic method [21]. This method was chosen due to excellent performance, when compared to other heuristic methods, for different gait pathologies [24] and for 360-degree turning [25]. This method uses the maximum anterior position of the posterior calcaneus marker relative to the sacrum marker to detect IC. EC is detected by the maximum posterior position of the metatarsal head marker relative to the sacrum marker. During straight-line gait, the anterior posterior axis is collinear to the walking axis of the gait laboratory. However, during a 360 degree turn, the anterior posterior axis continuously varies over time. Inspired by [25], this method was generalized to 360 degree turning by defining a rotation matrix R_z around the coronal plane to map the position of the calcaneus, metatarsal, and sacrum marker back to the transverse plane.

$$R_z = \begin{bmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The angle θ was defined as the pelvis angle, corresponding to the turning radius.

2.6 Peak Detection

The predicted output sequence Y returns the likelihood of a gait event for each sample. The peaks within an output sequence thus corresponded to a gait event. A peak detection algorithm [26] was employed to detect the local maxima in the likelihood vector and in the characteristic kinematic shapes of the heuristic methods. A constraint was imposed on the minimum distance between two consecutive gait events. The threshold for this constraint was empirically defined at 15 frames or 150ms.

2.7 Statistical Analysis

The model predictions were validated in terms of accuracy and timing agreement with respect to the manual annotations which were considered as the golden standard [27]. The accuracy was assessed using the true positive (TP), false positive (FP), false negative (FN), and summarized with the F1-score. Bland-Altman plots were created to assess the timing agreement between the methods. The agreement was quantified in terms of mean values, 95% confidence intervals, and limits of agreement (mean ± 1.96 standard deviation).

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{1}$$

3 RESULTS

For the freezing trials (FOG), a total of 506 IC and 491 EC events were acquired. The TCN model shows F1-scores of 0.995 and 0.992 for IC and EC, respectively. The LSTM model shows F1-scores of 0.989 and 0.976 for IC and EC, respectively. The heuristic method shows F1-scores of 0.976 and 0.956 for IC and EC, respectively. For the functional gait trials (FG), a total of 741 IC and 669 EC events were acquired. The TCN model shows F1-scores of 0.997 and 0.999 for IC and EC, respectively. The LSTM model shows F1-scores of 0.997 and 0.990 for IC and EC, respectively. The heuristic method shows F1-scores of 0.997 and 1 for IC and EC, respectively. The results are summarized in Table 1, reporting the total number of steps and the accuracy of the algorithms in terms of TP, FP, FN, and F1-score.

Error analysis showed that a large amount of the missed detections by the heuristic method were caused by a festination pattern of walking, which is the tendency to move forward with increasingly rapid, but ever smaller steps, associated with the centre of gravity falling forward over the stepping feet [28]. This phenomenon was especially evident in one patient, who accounted for 83% of all false detections for the heuristic method. Exclusion of this patient results in comparable levels of accuracy between the heuristic method and the TCN model, which shows the highest overall accuracy.

Bland-Altman plots were obtained, assessing the timing agreement of the deep learning models and the heuristic method, to the manual annotations. The differences between the proposed annotations and the manual annotations (vertical axis) are plotted against their average (horizontal axis). A positive time difference represents a delay in the annotations with respect to the manual annotations, while the limits of agreement (LoA) estimate the interval within which a proportion of the differences between the methods lie. All results are given in terms of frames.

Firstly, A Bland-Altman plot was obtained for both the FOG and FG trials, assessing the timing agreement of the TCN model versus the manual annotations, visualized in Figure 3 (a). For FOG-trials, the mean time differences [lower LoA, upper LoA] were 0.55 [-5.0, 6.1] for IC and -1.7 [-7.4, 4.1] for EC. For FG-trials, the mean time differences [lower LoA, upper LoA] were -0.93 [-6.5, 4.7] for IC and -0.01 [-4.7, 4.5] for EC. Secondly, A Bland-Altman plot was obtained for both the FOG and FG trials, assessing the timing agreement of the LSTM model versus the manual annotations, visualized in Figure 3 (b). For FOG-trials, the mean time differences [lower LoA, upper LoA] were 1.0 [-4.3, 6.3] for IC and -2.1 [-8.0, 3.8] for EC. For FG-trials, the mean time differences [lower LoA, upper LoA] were -0.4 [-5.5, 4.7] for IC and 1.2 [-5.0, 7.5] for EC. Lastly, A Bland-Altman plot was obtained for both the FOG and FG trials, assessing the timing agreement of the heuristic method versus the manual annotations, visualized in Figure 3 (c). For FOG-trials, the mean time differences [lower LoA, upper LoA] were -4.4 [-13, 4.2] for IC and -3.3 [-13, 6.2] for EC. For FG-trials, the mean time differences [lower LoA, upper LoA] were -3.5 [-8.1, 1.1] for IC and 1.7 [-3.7, 7.1] for EC.

For the FG trials, all three algorithms performed excellently with low variability. The deep learning algorithms additionally show minimal mean time differences with the manual annotations. For the FOG trials, several early detections that were still within the fifteen frame limit resulted in large mean time differences and variability for the heuristic method. For the deep learning models, a few hastened EC detections can be observed. These hastened detections were the result of delayed swing-phase during gait re-initiation after a FOG episode. Overall, the TCN model shows the most consistent results for both gait events.

4 DISCUSSION

We evaluated two data-driven approaches for the detection of gait events that were trained end-to-end on a small dataset of straight-line gait and 360 degree turning of PD patients with FOG. A total of 2407 events have been manually annotated and these events were used to quantitatively validate the algorithms in terms of accuracy and timing agreement. A commonly used heuristic method proposed in [21] was reproduced to allow a quantitative comparison with the deep learning models on the same dataset. The heuristic method showed a large mean time difference with the manual annotations. For the functional gait trials, the mean time difference could be associated with a systematic error on the manual annotations. For the freezing trials, the line between false and hastened detections blurred, resulting in large variability and an indication that this method is ill-suited for detecting gait events in PD patients with FOG when OFF-medication. In contrast, the Bland-Altman plots indicate that both deep learning models share a similar small mean time difference with the manual annotations. While these results suggest that both models focus on similar patterns in the data, the TCN model detects gait events with fewer false detections. Overall, the TCN model showed excellent levels of accuracy and timing agreement, with on average 39% and 47% of the detections occurring within 10ms from the manual annotations for FOG and FG, respectively. However, delayed swing-phase during gait re-initiation after a freezing episode, resulted in a few hastened EC detections. Additionally, research shows that strides directly preceding FOG were reduced by 35% in comparison with normal (functional) strides [5], which impacts the acceptable limits of agreement. Therefore, we suggest to visually verify the timing of the gait events that directly precede and proceed a FOG episode.

When repeating the analysis on a different cohort of non-freezing patients with PD through random selection of five gait trials we found very similar results confirming the robustness of the present findings (see supplm 1).

In conclusion, we were able to establish that the TCN model was able to accurately demarcate gait cycles based on kinematic data obtained with a 3D motion capturing system. The most remarkable finding was that this methodology proved robust for people experiencing severe gait disorders such as FOG when OFF-medication. Hence, our results suggest that the TCN model allows analyzing stepping behavior even during 360 degree turning tasks, when FOG episodes are provoked most consistently. Furthermore, future work is now possible in which automated step annotations based on kinematic data acquired from wearable devices, could be compared with automated step annotations based on kinematic data from 3D gait analysis systems. Such work is important to increase the understanding of FOG and to assess the effects of interventions during everyday life to alleviate this debilitating symptom.

5 Data Availability

The input set was imported and labelled using Python version 2.7.12 with Biomechanical Toolkit (btk) version 0.3 [29]. Peak detection was done with Scipy [26]. Deep learning models were trained on an NVIDIA Tesla K80 GPU using Python version 3.6.8 and Tensorflow version 1.14 [30]. Hyperparameters were optimized using the Hyperopt python library [31], with cross validation splits created with scikit-learn version 0.21.3 [32]. Utility functions for processing c3d files were adopted from [10]. All code, including a deployable model, is made available at `https://github.com/BenjaminFiltjens/gait_event`. Statistical analysis was conducted using R statistical software version 3.5.3 [33].

6 Conflict of Interest Statement

The authors declare that there is no conflict of interest regarding the publication of this article. This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] N. Giladi, A. Nieuwboer, Understanding and treating freezing of gait in parkinsonism, proposed working definition, and setting the stage, Mov. Disord. 23 Suppl 2 (2008) S423–5.

[2] A. H. Snijders, M. J. Nijkrake, M. Bakker, M. Munneke, C. Wind, B. R. Bloem, Clinimetrics of freezing of gait, Mov. Disord. 23 Suppl 2 (2008) S468–74.

[3] H. Zach, A. M. Janssen, A. H. Snijders, A. Delval, M. U. Ferraye, E. Auff, V. Weerdesteyn, B. R. Bloem, J. Nonnekes, Identifying freezing of gait in parkinson's disease during freezing provoking tasks using waist-mounted accelerometry, Parkinsonism Relat. Disord. 21 (11) (2015) 1362–1366.

[4] N. Alice, C. Fabienne, W. Anne-Marie, D. Kaat, Does freezing in parkinson's disease change limb coordination? a kinematic analysis.

[5] A. Nieuwboer, R. Dom, W. De Weerdt, K. Desloovere, S. Fieuws, E. Broens-Kaucsik, Abnormalities of the spatiotemporal characteristics of gait at the onset of freezing in parkinson's disease, Mov. Disord. 16 (6) (2001) 1066–1075.

[6] J. Nonnekes, A. M. Janssen, S. H. G. Mensink, L. B. Oude Nijhuis, B. R. Bloem, A. H. Snijders, Short rapid steps to provoke freezing of gait in parkinson's disease, J. Neurol. 261 (9) (2014) 1763–1767.

[7] M. Plotnik, J. M. Hausdorff, The role of gait rhythmicity and bilateral coordination of stepping in the pathophysiology of freezing of gait in parkinson's disease, Mov. Disord. 23 Suppl 2 (2008) S444–50.

[8] M. Mancini, B. R. Bloem, F. B. Horak, S. J. G. Lewis, A. Nieuwboer, J. Nonnekes, Clinical and methodological challenges for assessing freezing of gait: Future perspectives, Mov. Disord. 34 (6) (2019) 783–790.

[9] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baselinearXiv:1611.06455.

[10] Ł. Kidziński, S. Delp, M. Schwartz, Automatic real-time gait event detection in children using deep neural networks, PLoS One 14 (1) (2019) e0211466.

[11] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 3104–3112.

[12] J. Spildooren, S. Vercruysse, K. Desloovere, W. Vandenberghe, E. Kerckhofs, A. Nieuwboer, Freezing of gait in parkinson's disease: the impact of dual-tasking and turning, Mov. Disord. 25 (15) (2010) 2563–2570.

[13] A. Nieuwboer, L. Rochester, T. Herman, W. Vandenberghe, G. E. Emil, T. Thomaes, N. Giladi, Reliability of the new freezing of gait questionnaire: agreement between patients with parkinson's disease and their carers, Gait Posture 30 (4) (2009) 459–463.

[14] J. Spildooren, S. Vercruysse, P. Meyns, J. Vandenbossche, E. Heremans, K. Desloovere, W. Vandenberghe, A. Nieuwboer, Turning and unilateral cueing in parkinson's disease patients with and without freezing of gait, Neuroscience 207 (2012) 298–306.

[15] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, in: S. C.

Kremer, J. F. Kolen (Eds.), A Field Guide to Dynamical Recurrent Neural Networks, IEEE Press, 2001.

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

[17] S. Bai, J. Zico Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modelingarXiv:1803.01271.

[18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentationarXiv:1411.4038.

[19] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shiftarXiv:1502.03167.

[20] 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 (2014) 1929–1958.

[21] J. A. Zeni, Jr, J. G. Richards, J. S. Higginson, Two simple methods for determining gait events during treadmill and overground walking using kinematic data, Gait Posture 27 (4) (2008) 710–714.

[22] J. S. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, Algorithms for Hyper-Parameter optimization, in: J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 24, Curran Associates, Inc., 2011, pp. 2546–2554.

[23] J. Bergstra, D. Yamins, D. Cox, Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures, in: International Conference on Machine Learning, 2013, pp. 115–123.

[24] D. A. Bruening, S. T. Ridge, Automated event detection algorithms in pathological gait, Gait Posture 39 (1) (2014) 472–477.

[25] B. Ulrich, A. N. Santos, B. M. Jolles, D. H. Benninger, J. Favre, Gait events during turning can be detected using kinematic features originally proposed for the analysis of straight-line walking, J. Biomech. 91 (2019) 69–78.

[26] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. Jarrod Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. Carey, İ. Polat, Y. Feng, E. W. Moore, J. Vand erPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, S. . . Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:https://doi.org/10.1038/s41592-019-0686-2.

[27] N. Chia Bejarano, E. Ambrosini, A. Pedrocchi, G. Ferrigno, M. Monticone, S. Ferrante, A novel adaptive, real-time algorithm to detect gait events from wearable sensors, IEEE Trans. Neural Syst. Rehabil. Eng. 23 (3) (2015) 413–422.

[28] J. Nonnekes, N. Giladi, A. Guha, U. M. Fietzek, B. R. Bloem, E. Růžička, Gait festination in parkinsonism: introduction of two phenotypes, J. Neurol. 266 (2) (2019) 426–430.

[29] A. Barre, S. Armand, Biomechanical ToolKit: Open-source framework to visualize and process biomechanical data, Comput. Methods Programs Biomed. 114 (1) (2014) 80–87.

[30] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden,

M. Wicke, Y. Yu, X. Zheng, TensorFlow: a system for large-scale machine learning, in: Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation, OSDI'16, USENIX Association, USA, 2016, pp. 265–283.

[31] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, D. D. Cox, Hyperopt: a python library for model selection and hyperparameter optimization, Comput. Sci. Discov. 8 (1) (2015) 014008.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (85) (2011) 2825–2830.

[33] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria (2013). URL http://www.R-project.org/