

# DETECTING EARLY PARKINSON'S DISEASE FROM KEYSTROKE DYNAMICS USING TENSOR DECOMPOSITIONS

Hooman Oroojeni M J<sup>1</sup>, James Oldfield<sup>1</sup>, Mihalis A. Nicolaou<sup>2,1</sup>

<sup>1</sup>Department of Computing, Goldsmiths, University of London, UK

<sup>2</sup>Computation-based Science and Technology Research Center, The Cyprus Institute, Cyprus

## ABSTRACT

We present a method for detecting early signs of Parkinson's disease from keystroke hold times that is based on the Tensor Train (TT) decomposition. While simple univariate methods such as logistic regression have shown good performance on the given problem by using appropriate features, the TT format facilitates modelling high-order interactions by representing the exponentially large parameter tensor in a compact multilinear form. By performing time-series feature selection, we show that this approach can significantly improve upon state-of-the-art for the given problem, reaching a performance of AUC=0.88, outperforming compared methods such as deep neural networks and other linear models.

**Index Terms**— Tensor Decomposition, Tensor Train, Feature Extraction, Parkinson's Disease.

## 1. INTRODUCTION

take care of whitespace in paper - try to get it to 5 pages exactly (that is the limit right?) - also make sure format is OK Parkinson's disease (PD) is one of the world's most prevalent neurodegenerative diseases, second only to Alzheimer's. Despite that, PD is diagnosed through a set of neurological tests at a clinic [1, 2], and is largely based on a specialist interpretation of symptoms. These tests are subjective, costly, protracted and imprecise, in particular for those who suffer from Parkinson's disease at the early stages [3].

In order to provide tools for the early detection and diagnosis of Parkinson's disease that are unobtrusive, ubiquitous, and cost-effective, the authors of [4, 5] evaluate the accuracy of predicting early detection of PD through the analysis of typing logs by several subjects that have PD or belong to the control group. In these works, keystroke dynamics are analysed with a focus on hold times (i.e. the length of time between pressing and releasing a key), as this measure is considered independent of typing skills. In [4], the utilization of the so-called neuroQWERTY index (nQi) method is used, in order to detect PD patients during a testing session. In [5], a simpler and easier to reproduce method is proposed that is based on logistic regression and features designed specifically

for this problem. In more detail, the mean absolute consecutive difference (MACD) feature is utilized in a univariate logistic regression setting, achieving an AUC=0.85 compared to 0.81 in [4].

In this paper, we propose a method based on the Tensor Train decomposition in order to provide even more accurate models for the detection of early Parkinson's Disease from keystroke dynamics. In more detail, the logistic regression approach utilized in [5] is a special case of the exponential machines regression presented in [6], where the Tensor Train decomposition is utilized in order to efficiently learn exponentially many interactions in our data, potentially leading to better generalization models. As we show in what follows, the proposed method can achieve an AUC=0.88, in comparison to previous work that achieve AUC=0.81 ([4]) and AUC=0.85 ([5]).

## 2. RELATED WORK

In this section, we briefly review some of the related work to this paper. In particular, the neuroQWERTY index(nQi) method was proposed in [4] to classify the typing sessions of participants to Parkinson's sufferer or control group. This paper partitions each typing session into a set of 90 seconds-long window. These partitions do not overlap and a partition is removed if it contains less than 30 elements. A 7-dimensional feature vector is created for each window, where each vector includes the partition's outliers proportion, skewness, flight time between consecutive keystrokes, and the proportion of elements in four equal bins. An ensemble of 200 linear support vector regression models with grid search hyperparameter optimisation is used to be trained with an external data set. The median of the 200 regression model of each partition  $i$  is the  $nQi^i$  value. The  $nQi$  score for a typing session is defined as the average of medians over  $I$  partitions. Ref. [4] achieved Area Under Receiving Operating Character curve (AUC)= 0.81 by applying cross-validation training on early PD data set and test on de novo data set. However, Ref. [5] achieved a similar AUC=0.82 by utilising a simpler approach that is based on a single feature from each session, the standard deviation, with a simple logistic regression model. Furthermore, in [5] a more sophisticated time

series feature has been proposed, namely the mean absolute consecutive difference (MACD). By using this single feature from a typing session in the same logistic regression setting, the authors are able to achieve a performance of AUC=0.85, while a performance of more than AUC=0.80 is achieved by just a few hundred keystrokes.

### 3. DATA SET

The data set used in this paper is drawn from the original study of [4]. 85 participants are included, with each participating in a typing session of around 15 minutes. The dataset includes 42 Parkinson’s Disease patients and 43 control subjects, that are further separated into two sets. Namely, the first set includes patients that are newly diagnosed and untreated (de novo PD), and the second set contains recordings of patients that have had a confirmed diagnosis in less than five years (early PD). The de novo PD contains 24 subjects with Parkinson’s and 30 control, while the early PD 18 Parkinson’s patients and 13 control.

### 4. FEATURE EXTRACTION

While many features can be extracted from data, and in particular from time-series, not every feature is informative and relevant to the target problem. In order to facilitate feature extraction in this paper, we utilise the Scalable Hypothesis (FRESH) algorithm [7]. FRESH encapsulates a collection of both static and dynamic features, while by performing significance testing is able to select the relevant features that are highly significant with respect to the true labels of the dataset. We use FRESH on the training data in order to select the most relevant features for this problem, which are subsequently utilized in the compared learning models after normalising for mean and unit variance. We note that we use the `tsfresh` package, implementing the FRESH algorithm [7]. This package combines 63 time series characterisation methods to advance the feature extraction process.

The features with higher significance overall are presented in Table 4. Briefly, `Change_quantiles`, aggregates consecutive differences between elements of a data record. `Cid_ce` is an estimate of the time series complexity, and `Fft_coefficient` calculates the Fourier coefficients of the one-dimensional discrete Fourier Transform. More details regarding these features can be found in [8].

### 5. METHODOLOGY

Matrix component analysis methods have seen rapid developments over the last decades including Principal Components Analysis (PCA), Nonnegative Matrix Factorisation (NMF), Independent Component Analysis (ICA), and Sparse Component Analysis (SCA) [10, 11, 12]. These approaches evolved into standard tools for classification, feature extraction and

**Table 1.** The list of FRESH functions along with parameters to produce features. These functions are part of the `tsfresh` package [9, 8]

Feature names and related parameters
<code>cid_ce(normalize=False)</code>
<code>fft_coefficient(coeff=53, attr=abs)</code>
<code>change_quantiles(ql=0.6, qh=1.0, isabs=True, f_agg=mean)</code>
<code>change_quantiles(ql=0.6, qh=0.8, isabs=True, f_agg=mean)</code>

blind source separation. Since **recent heterogeneous sensor data has a multiway character**<sup>1</sup>, reformatting them as a matrix and apply classical two-way analysis instead of multiway array (tensor) operations are not always a good practice. Instead of pair-wise analysis, the higher order tensor decomposition offers an opportunity to capture multiple interactions and coupling through developing complex models. Tensor decompositions are not only matrix factorisation but also they can capture multiple interactions and coupling [13]. An approach to improve the performance of the machine learning algorithms is to model interactions between features in every order. This is in contrast to traditional linear models, as modelling such interactions results in a gigantic parameter tensor, which is challenging to both train and fit into memory. This problem can be alleviated by adopting the Tensor Train (TT) representation, where an exponentially large tensor can be represented in a compact multilinear format [14]. In this paper, we propose utilizing a Tensor Train-based regression framework, where exponential interactions between our features can be modelled in an efficient and robust manner. Such interactions can be modelled by considering the traditional linear model

$$\hat{y}(\mathbf{x}) = \langle \mathbf{x}, \mathbf{w} \rangle + b,$$

where the prediction is generated by the dot product of our features  $\mathbf{x}$  and parameters  $\mathbf{w}$ , with an arbitrary loss function  $\ell$ . To consider all interactions, the model above is extended following [6] as,

$$\hat{y}(x) = \sum_{i_1=0}^1 \dots \sum_{i_d=0}^1 \mathcal{W}_{i_1 \dots i_d} \prod_{k=1}^d x_k^{i_k} \quad (1)$$

where the weight tensor  $\mathcal{W}$  has a dimension  $d$  and contains  $2^d$  elements.  $x_k$  corresponds to the feature  $k$  where  $k = 1, \dots, d$ , while subsets of features are enumerated with a binary vector  $(i_1, \dots, i_d)$ , with  $i_k = 1$  if the  $k$ -th feature belongs to the subset. Given that Eq. 1 can be written as a tensor dot product,  $\hat{y}(\mathbf{x}) = \langle \mathcal{X}, \mathcal{W} \rangle$ , where

$$\mathcal{X}_{i_1, \dots, i_d} = \prod_{k=1}^d x_k^{i_k}. \quad (2)$$

In this way, the Tensor Train format can be utilized to compactly represent the parameter tensor  $\mathcal{W}$ .

<sup>1</sup>not sure what we mean recent here

In more detail, the  $d$ -dimensional tensor  $\mathcal{W}$  is computed as a product of  $d - 2$  matrices and 2 vectors,

$$W_{i_1 \dots i_d} = G_1[i_1] \dots G_d[i_d], \quad (3)$$

where  $G_1[i_1]$  and  $G_d[i_d]$  are vectors with dimensions of  $1 \times r$  and  $r \times 1$ . For any  $i_k$ ,  $G_k[i_k]$  where  $k = 2, \dots, d-1$ , is a  $r \times r$  matrix.  $G_k$  matrix matching, the same dimension  $k$ , is called as the  $k$ -th TT-core. The size  $r$  is called as TT-rank of the tensor  $\mathcal{W}$  which is the slice-size of  $G_k[i_k]$ . We note that the TT-rank adjusts the balance between computation efficiency of the tensor operations and the representational power of the TT-format itself [6]. We finally note that the TT-rank of the Data tensor  $\mathcal{X}$  is always 1 and this tensor can be represented TT-core format as:

$$G_k[i_k] = x_k^{i_k} \in \mathbb{R}^{1 \times 1}, k = 1, \dots, d. \quad (4)$$

Given features extracted as described in Section 4, we apply the Riemannian optimization<sup>2</sup> procedure described in [6] to which exploits tensor geometry to optimise the parameter tensor  $\mathcal{W}$  in the following optimisation problem,

$$\begin{aligned} \min_{\mathcal{W}} \quad & L(\mathcal{W}) \\ \text{subject to} \quad & \text{TT-rank}(\mathcal{W}) = r_0 \end{aligned} \quad (5)$$

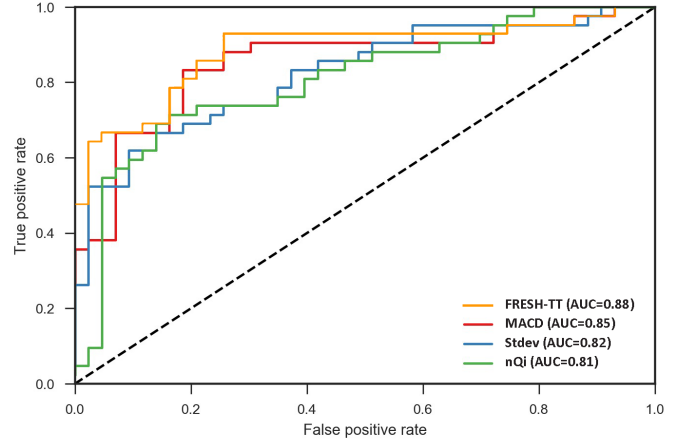
where

$$L(\mathcal{W}) = \sum_{f=1}^N \ell(\langle \mathcal{X}^f, \mathcal{W} \rangle, y^{(f)}) + \frac{\lambda}{2} \|\mathcal{W}\|_F^2. \quad (6)$$

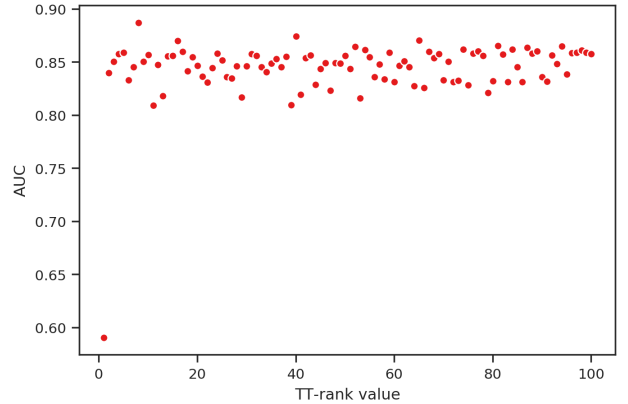
## 6. EXPERIMENTS AND RESULTS

In this section, we present results that compare the proposed Tensor Decomposition based approach to previous works on the same dataset, such as [4] and [5], following the same evaluation protocol as in previous works. Namely, we compare with the nQi method presented in [4], the univariate models presented in [5] that include the Stdev and MACD models, as well as the multivariate models that utilize the FRESH feature extraction as described in Sec. 4. Furthermore, we compare with a model based on Recurrent Neural Networks, and in particular the so-called Gated Recurrent Units (GRU).

Detailed results are presented in Table 2, where we show both accuracy and area under the curve (AUC) for each of the compared methods. Furthermore, in Figure 1, the ROC curve of the proposed method in comparison to related work is shown, where FRESH-TT clearly outperforms all compared methods. In the following, we discuss the different approaches employed along with the resulting scores.



**Fig. 1.** The ROC curve of the FRESH-TT model and all other models discussed in this paper namely Stdev, FRESH, MACD and nQi is presented. Except nQi, all values are reproduced through the same cross-validation method as described in [4].

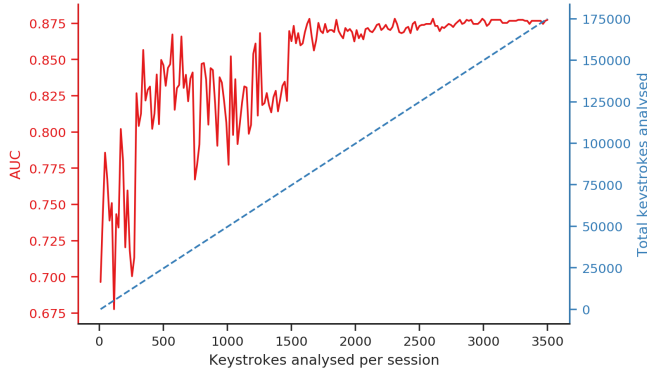


**Fig. 2.** The dependence of classification performance on the values of TT-ranks between 1 to 100. The best performance (AUC=0.88) achieved with TT-rank=8. perhaps this could look better in a table or in-text

<sup>2</sup>do we? or is it just gradient descent?

**Table 2.** The performance of all the models evaluated in the recent papers along with this paper, including True and False Positives (TP, FP), True and False Negatives (TN, FN), and area under the curve (AUC).

Model	TP	FN	TN	FP	AUC
nQi [4]	30	12	36	7	0.81
Stdev [5]	27	15	37	6	0.82
FRESH (5 Features) [5]	36	6	26	17	0.80
MACD [5]	34	8	35	8	0.85
FRESH-GRU	22	20	38	5	0.65
FRESH-LR (4 Features)	29	13	36	7	0.83
FRESH - TT	22	20	39	4	0.88



**Fig. 3.** The relationship between the number of keystrokes and classification performance analysed. The  $x$  axis presents the length of truncated time series. In right  $y$  axis (blue), shows the total number of keystrokes analysed over all sessions of 85 participants. The left  $y$  axis (red) represents the AUC obtained by applying the FRESH-TT model over truncated time series. **seems to me that the blue line should not be diagonal**

### 6.1. FRESH - Logistic Regression (FRESH-LR)

To offer a baseline, we use a feature extraction method based on Scalable Hypothesis algorithm (FRESH) and perform binary classification by using logistic regression. The selected features that showed higher significance for the data set are listed in Table 4. This model evaluated by using the early PD and the de novo PD data set in the same way that discussed in [4] to calculate cross-validation. After applying cross-validation while training on the de novo PD data set and test on the early PD data set and vice-versa, this model achieved Area Under curve (AUC)=0.83.

### 6.2. FRESH - Gated Recurrent Units (FRESH-GRU)

Recurrent Neural Networks (RNN) are well-known for being able to model arbitrary temporal dependencies when analysing time series data [15]. We specifically utilize Gated Recurrent Unit (GRU) since less parameters are required in comparison to traditional Long Short-Term Memory re-

current neural network (LSTM) [16]. To capture temporal dependencies, we use GRU as part of this experiment to benefit the same level of LSTM performance while using less parameters. By experimenting, we concluded that the data is not sufficient for RNNs to discover the appropriate features, and as results were low, we resorted in feeding the FRESH features to the GRU layers which increased the accuracy. The GRU model on this data achieves an AUC=0.65, which is quite lower than compared models. This is likely due to the number of data available for the given problem and dataset.

### 6.3. FRESH - Tensor Train (FRESH-TT)

Fresh with Tensor Train (TT-FRESH) represents the results for the methodology proposed in this paper, as described in Section 5. After feature extraction, we utilize the TT decomposition to represent and estimate the model parameters in the TT-format. We utilize the T3F library that provides tools for working with the TT decomposition, supporting GPU executing and parallel processing of tensor batches. We utilize this library for implementing the TT model. As can be clearly seen in Table 2 and Figure 1, the proposed method achieves an AUC=0.88, outperforming the second-best method proposed in [5] with AUC=0.85.

## 7. CONCLUSION

In this paper, we proposed a method based on appropriate feature extraction and tensor decompositions in order to model high-order interactions in the problem of detecting early Parkinson's disease from keystroke dynamics. We compare against related methods in literature, including recurrent neural networks and linear models. We show that our method improves results, providing an AUC=0.88, while still being efficient in terms of complexity.

## 8. REFERENCES

- [1] A Elbaz, L Carcaillon, S Kab, and F Moisan, "Epidemiology of parkinson's disease," *Revue neurologique*, vol. 172, no. 1, pp. 14–26, 2016.
- [2] P1 Martínez-Martín, A Gil-Nagel, L Morlán Gracia, J Balseiro Gómez, J Martinez-Sarries, F Bermejo, and Cooperative Multicentric Group, "Unified parkinson's disease rating scale characteristics and structure," *Movement disorders*, vol. 9, no. 1, pp. 76–83, 1994.
- [3] Fernando L Pagan, "Improving outcomes through early diagnosis of parkinson's disease," *American Journal of Managed Care*, vol. 18, no. 7, pp. S176, 2012.
- [4] Luca Giancardo, Alvaro Sanchez-Ferro, Teresa Arroyo-Gallego, Ian Butterworth, Carlos S Mendoza, Paloma Montero, Michele Matarazzo, José A Obeso, Martha L

- Gray, and R San José Estépar, “Computer keyboard interaction as an indicator of early parkinsons disease,” *Scientific reports*, vol. 6, pp. 34468, 2016.
- [5] Antony Milne, Katayoun Farrahi, and Mihalis A Nicolaou, “Less is more: Univariate modelling to detect early parkinsons disease from keystroke dynamics,” in *International Conference on Discovery Science*. Springer, 2018, pp. 435–446.
- [6] Alexander Novikov, Mikhail Trofimov, and Ivan Oseledets, “Exponential machines,” *arXiv preprint arXiv:1605.03795*, 2016.
- [7] Maximilian Christ, Andreas W Kempa-Liehr, and Michael Feindt, “Distributed and parallel time series feature extraction for industrial big data applications,” *arXiv preprint arXiv:1610.07717*, 2016.
- [8] “tsfresh,” <https://github.com/blue-yonder/tsfresh>, accessed 30 October 2018.
- [9] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W Kempa-Liehr, “Time series feature extraction on basis of scalable hypothesis tests (tsfresh—a python package),” *Neurocomputing*, 2018.
- [10] Pierre Comon and Christian Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*, Academic press, 2010.
- [11] Andrzej Cichocki, Rafal Zdunek, Anh Huy Phan, and Shun-ichi Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*, John Wiley & Sons, 2009.
- [12] Alfred M Bruckstein, David L Donoho, and Michael Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.
- [13] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis,” *IEEE Signal Processing Magazine*, vol. 32, no. 2, pp. 145–163, 2015.
- [14] Ivan V Oseledets, “Tensor-train decomposition,” *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [15] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [16] Jia Li, Yu Rong, Helen Meng, Zhihui Lu, Timothy Kwok, and Hong Cheng, “Tatc: Predicting alzheimer’s disease with actigraphy data,” in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, 2018, KDD ’18, pp. 509–518, ACM.