Detecting Early Parkinson's Disease from Keystroke Dynamics using the Tensor-Train Decomposition

Hooman Oroojeni M. J.*, James Oldfield*, Mihalis A. Nicolaou†

*Department of Computing, Goldsmiths, University of London, London
†Computation-based Science and Technology Research Center, The Cyprus Institute, Cyprus
Email: h.oroojeni@gold.ac.uk, m.nicolaou@cyi.ac.cy

***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 uni-variate 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 multi-linear form. By performing time-series feature extraction based on scalable hypothesis testing, we show that the proposed 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 on the problem of detecting early Parkinson's disease from keystroke dynamics.**

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

I. Introduction

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 particular, subtle motor impairments become evident shortly after disease onset, but much before actual clinical diagnosis [4].

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 utilisation of the so-called neuroQUERTY 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 utilised in a uni-variate logistic regression setting, achieving an AUC=0.85 compared to 0.81 in [4].

In this paper, we are motivated by the success and wide range of applications of tensor methods and multi-linear analysis in signal processing and machine learning [6], [7], leading to a set of techniques that are both efficient as well as scalable, providing state-of-the-art accuracy in several applications with a significant reduction of parameters in contrast to e.g., deep learning. In more detail, 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 by modelling high-order feature interactions. In more detail, the logistic regression approach utilised in [5] can be considered as a special case of the exponential machines regression presented in [8], where the Tensor Train decomposition is utilised in order to efficiently learn exponentially many interactions in our data, potentially leading to better generalisation 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]).

II. 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 hyper-parameter 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, and then vice-versa. 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

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

difference (MACD)[1]. 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.

III. 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 data set 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 include 18 Parkinson's patients and 13 control.

IV. 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 [9]. 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 data set. We use FRESH on the training data in order to select the most relevant features for this problem, which are subsequently utilised in the compared learning models after normalising for mean and unit variance. We note that we use the `tsfresh` package, implementing the FRESH algorithm [9]. This package combines 63 time series characterisation methods to advance the feature extraction process.

The features with higher significance overall are presented in Table I. 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 [10] and [11].

TABLE I

The list of FRESH functions along with parameters applied to produce the most relevant features including aggregated consecutive differences between elements, estimate of the time series complexity, and Fourier coefficients of the one-dimensional discrete Fourier transform.

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

V. Methodology

Matrix component analysis methods have seen rapid developments over the last decades including Principal Components Analysis (PCA), Non-negative Matrix Factorisation (NMF), Independent Component Analysis (ICA), and Sparse Component Analysis (SCA) [12]–[14]. These approaches evolved into standard tools for classification, feature extraction and blind source separation. The modern heterogeneous sensor modalities provide immense data sets, naturally they can be represented by tensors or multi-way arrays. Reformatting the tensors as a matrix and apply classical two-way analysis instead of 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 decomposition methods are not only matrix factorisation but also they can capture multiple interactions and coupling [15]–[19]. An approach to improve the performance of the machine learning algorithms is to model high-order interactions between features. 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 multi-linear format [20]. In this paper, we propose utilising a Tensor Train-based regression framework, where exponential interactions between our features can be modelled in an efficient and robust manner [8]. Such interactions can be modelled by considering the traditional linear model

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

where the prediction is generated by the dot product of our features $\mathbf{x}$ and parameters $\mathbf{w}$, with an arbitrary loss function ℓ. To consider all interactions, the model above is extended following [8] as,

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

where the weight tensor $\mathcal{W}$ has a dimension d and contains 2^d elements. x_k corresponds to the feature k where $k = 1, ..., d$, while subsets of features are enumerated with a binary vector $(i_1, \ldots, i_d)$, 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

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

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

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

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

[1] MACD is simply the mean of the absolute value of first order differences, applied on hold times.

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 $k = 2, ..., d-1$, 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 [8]. We finally note that the TT-rank of the data tensor $\mathcal{X}$ is always 1, as this tensor can be represented TT-core format as:

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

where $i_k \in \{0, 1\}$. Given features extracted as described in Section IV, we apply the Riemannian gradient descent optimisation scheme proposed in [21] to optimise the parameter tensor $\mathcal{W}$, solving the following optimisation problem,

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

where

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

Where X^f is a d-dimensional feature vector of f-th object, N is the total number of objects or projection, and λ is the regularisation parameter. We further extend this model to incorporate Recurrent Neural Networks (RNNs) for capturing dynamics in the final representation. We utilise an RNN layer for each view of the data tensor $\mathcal{X}$, thus extracting a set of latent features from each. Subsequently, these are factored as a TT-tensor in order to model all 2^d interactions, while a fully connected layer is used for classification. The entire model is trained end-to-end with Stochastic Gradient Descent (SGD) (Section VI-B).

VI. Experiments and Results

In this section, we present the results that compare the proposed tensor-based approach to previous works on the same data set, such as [4] and [5], following the same evaluation protocol and reproducing results presented in each work. Namely, we compare with the nQi method presented in [4], the uni-variate models presented in [5] that include the Stdev and MACD models, as well as the multivariate models that utilise the FRESH feature extraction as described in Sec. IV. 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 II, 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.

TABLE II
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), area under the curve (AUC), and Accuracy.

Model	TP	FN	TN	FP	AUC	Accuracy
nQi [4]	30	12	36	7	0.81	0.77
Stdev [5]	27	15	37	6	0.82	0.75
FRESH (5 Features) [5]	36	6	29	14	0.80	0.76
MACD [5]	34	8	35	8	0.85	0.81
FRESH-GRU	22	20	38	5	0.65	0.70
FRESH-LR (4 Features)	31	11	36	7	0.83	0.79
FRESH - TT	**36**	**6**	**39**	**4**	**0.88**	**0.88**

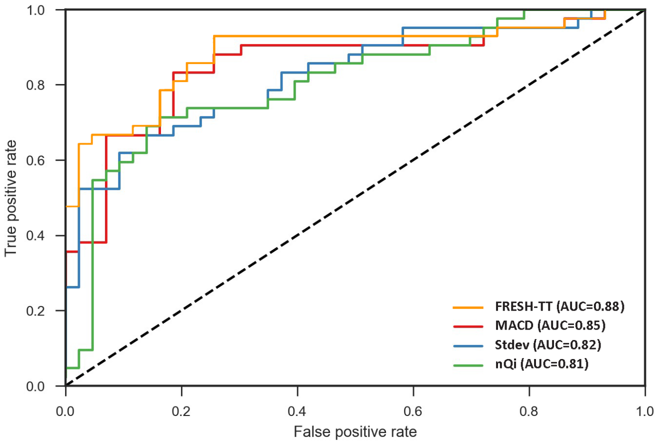

Fig. 1. The ROC curve of the FRESH-TT model and all other models discussed in this paper namely Stdev, FRESH, MACD and nQi is presented. Except nQi, all values are reproduced through the same cross-validation method as described in [4].

A. 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 I. This model is evaluated by using the early PD and the de novo PD data set following [4], that is training on early PD and testing on de-novo and vice versa utilizing cross-validation. This model achieves Area Under curve (AUC)=0.83.

B. 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 [22], acting as universal approximators to non-linear dynamical systems [23], [24]. We specifically utilise Gated Recurrent Unit (GRU) for capturing temporal dependencies, since less parameters are required in comparison to Long Short-Term Memory recurrent neural networks (LSTM) [25] while maintaining the same level of performance such as LSTMs. When directly utilising RNNs on the raw-time series of keystrokes, the results are much worse than compared models. Therefore, by experimenting, we

TABLE III
THE PERFORMANCE OF FRESH TENSOR TRAIN (FRESH-TT), MEAN ABSOLUTE CONSECUTIVE DIFFERENCE (MACD), FRESH-LR, STDEV MODELS OVER DE NOVO AND EARLY PD DATA SETS. TRUE AND FALSE POSITIVE(TP, FP), TRUE AND FALSE NEGATIVE (TN, FN) ARE DEMONSTRATED FOR FURTHER ANALYSIS.

	FRESH-TT				**MACD**				**FRESH-LR**				**Stdev**			
	TP	**FN**	**TN**	**FP**	**TP**	**FN**	**TN**	**FP**	**TP**	**FN**	**TN**	**FP**	**TP**	**FN**	**TN**	**FP**
De novo PD	20	4	28	2	19	5	25	5	18	6	25	5	15	9	26	4
Early PD	16	2	11	2	15	3	10	3	13	5	11	2	12	6	11	2
Total	36	6	39	4	34	8	35	8	31	11	36	7	27	15	37	6

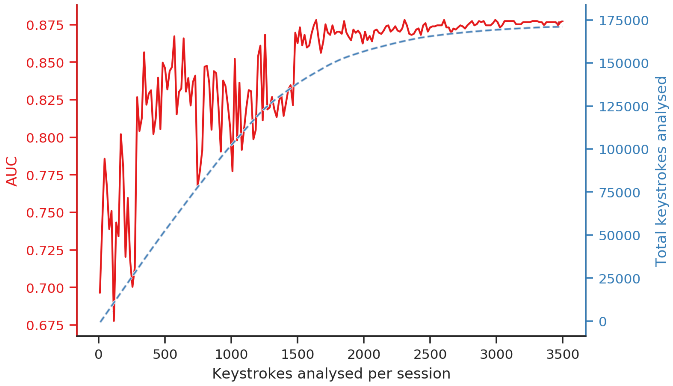

Fig. 2. The relationship between the number of keystrokes and classification performance analysed. The x axis presents the length of truncated time series. In right y axis (blue), shows the total number of keystrokes analysed over all sessions of 85 participants. The left y axis (red) represents the AUC obtained by applying the FRESH-TT model over truncated time series.

concluded that the number of data available is not sufficient for RNNs to discover the appropriate representations. Hence, we resorted in feeding the FRESH features to the GRU layers which increased the accuracy significantly. In this method, first we model each dimension of the data with a GRU, then extract 4 features form each. Then we factor these features as a TT-rank one tensor train to model all 2^d interactions and afterwards we train end-to-end with stochastic gradient descent (SGD). The FRESH-GRU model on this data achieves an AUC=0.65, which is still quite lower than compared models. This is again likely due to the number of data available for the given problem and data set.

C. FRESH - Tensor Train (FRESH-TT)

FRESH with Tensor Train (FRESH-TT) represents the results for the methodology proposed in this paper, as described in Section V. After feature extraction, we utilise the TT decomposition to represent and estimate the model parameters in the TT-format. We utilise the `T3F` library that provides tools for working with the TT decomposition, supporting GPU executing and parallel processing of tensor batches. The challenge of finding the optimal TT-rank is a part of our optimisation process. We experienced various TT-ranks, and experimentally we achieved the best result value of 8 (see table IV to view the relationship between TT-ranks and AUCs). As can be clearly seen in Table II and Figure 1, the proposed method achieves an AUC=0.88, outperforming the second-best method proposed in [5] with AUC=0.85.

By applying the FRESH method to extract features and estimate the model parameters with TT decomposition and classifying with logistic regression, we acquire the evaluation scores shown in Table II. Further exploration regarding performance of FRESH-TT along with other models over de novo PD and early PD data sets, in the same way that discussed in [4], is shown in Table III. This analysis include True positive and negative and also False positive and negative test results. It is clear from observing the results that the proposed FRESH-TT method appears much more robust than all compared methods. Furthermore, in this study we find that the proposed FRESH-TT method achieves an AUC = 0.88, this outperforming significantly all the models previously suggested. Despite prior research that suggest approaches to analyse every element of the hold times series h, FRESH-TT can obtain effective classification without observing the entire time series. Fig. IV exhibits the dependency on classification performance with the number of keystrokes analysed. We curtail time series h after a certain number of elements and perform classification according to the FRESH-TT model. Fig. 2 demonstrates that one may achieve outstanding performance (AUC > 0.87) from analysing approximately 1.5K keystrokes in a typing session.

TABLE IV
THE DEPENDENCE OF THE CLASSIFICATION PERFORMANCE OF THE TOP FIVE TT-RANKS EVALUATED BETWEEN THE RANGE OF 1 TO 100. THE BEST PERFORMANCE (AUC=0.88) ACHIEVED WITH TT-RANK=8.

TT-Rank	**AUC**	**Accuracy**
1	0.5907	0.6648
2	0.8397	0.8013
4	0.8578	0.7921
6	0.8330	0.7691
8	**0.8823**	**0.88**
10	0.8568	0.8013

VII. CONCLUSIONS

In this paper, we proposed a method based on appropriate feature extraction and tensor decomposition applied to the problem of detecting early Parkinson's disease from keystroke dynamics. The proposed method is based on feature extraction with scalable hypothesis testing, as well as utilising the Tensor-Train decomposition for modelling high-order interactions amongst features. We compared against both previous work, as well as extensions of the proposed model with recurrent neural networks. We show that the proposed method improves state-of-the-art results on the problem, reaching an AUC=0.88, while still being efficient in terms of complexity, leading to models that can be easily utilised in embedded systems and other low-power devices for ubiquitous patient monitoring.

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] P. Martínez-Martín, A. Gil-Nagel, L. M. Gracia, J. B. Gómez, J. Martinez-Sarries, F. Bermejo, and C. M. Group, "Unified parkinson's disease rating scale characteristics and structure," Movement disorders, vol. 9, no. 1, pp. 76–83, 1994.

[3] F. L. Pagan, "Improving outcomes through early diagnosis of parkinson's disease," American Journal of Managed Care, vol. 18, no. 7, p. S176, 2012.

[4] L. Giancardo, A. Sanchez-Ferro, T. Arroyo-Gallego, I. Butterworth, C. S. Mendoza, P. Montero, M. Matarazzo, J. A. Obeso, M. L. Gray, and R. S. J. Estépar, "Computer keyboard interaction as an indicator of early parkinsons disease," Scientific reports, vol. 6, p. 34468, 2016.

[5] A. Milne, K. Farrahi, and M. 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] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM review, vol. 51, no. 3, pp. 455–500, 2009.

[7] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 8, no. 2, p. 16, 2017.

[8] A. Novikov, M. Trofimov, and I. Oseledets, "Exponential machines," arXiv preprint arXiv:1605.03795, 2016.

[9] M. Christ, A. W. Kempa-Liehr, and M. Feindt, "Distributed and parallel time series feature extraction for industrial big data applications," arXiv preprint arXiv:1610.07717, 2016.

[10] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," Neurocomputing, 2018.

[11] "tsfresh," https://github.com/blue-yonder/tsfresh, accessed 30 October 2018.

[12] P. Comon and C. Jutten, Handbook of Blind Source Separation: Independent component analysis and applications. Academic press, 2010.

[13] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. John Wiley & Sons, 2009.

[14] A. M. Bruckstein, D. L. Donoho, and M. 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.

[15] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. 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.

[16] P. Comon, "Tensors: A brief survey," IEEE Signal Processing Magazine, vol. 31, no. 3, pp. 44–53, 2014.

[17] J. Landsberg, "Tensors: geometry and applications, ser," Graduate Studies in Mathematics. AMS publ, vol. 128, 2012.

[18] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," Linear algebra and its applications, vol. 18, no. 2, pp. 95–138, 1977.

[19] I. Domanov and L. De Lathauwer, "On the uniqueness of the canonical polyadic decomposition of third-order tensors—part ii: Uniqueness of the overall decomposition," SIAM Journal on Matrix Analysis and Applications, vol. 34, no. 3, pp. 876–903, 2013.

[20] I. V. Oseledets, "Tensor-train decomposition," SIAM Journal on Scientific Computing, vol. 33, no. 5, pp. 2295–2317, 2011.

[21] A. Novikov, P. Izmailov, V. Khrulkov, M. Figurnov, and I. Oseledets, "Tensor train decomposition on tensorflow (t3f)," arXiv preprint arXiv:1801.01928, 2018.

[22] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," arXiv preprint arXiv:1412.3555, 2014.

[23] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[24] K.-i. Funahashi and Y. Nakamura, "Approximation of dynamical systems by continuous time recurrent neural networks," Neural networks, vol. 6, no. 6, pp. 801–806, 1993.

[25] J. Li, Y. Rong, H. Meng, Z. Lu, T. Kwok, and H. Cheng, "Tatc: Predicting alzheimer's disease with actigraphy data," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery &#38; Data Mining, ser. KDD '18. New York, NY, USA: ACM, 2018, pp. 509–518. [Online]. Available: http://doi.acm.org/10.1145/3219819.3219831