TU Delft

Delft University of Technology

Document Version
Final published version

Citation (APA)
de Rooij, S. J. S., Wesel, F., & Hunyadi, B. (2024). Efficient Patient Fine-Tuned Seizure Detection with a Tensor Kernel Machine. In *Proceedings of the 32nd European Signal Processing Conference, EUSIPCO 2024* (pp. 1372-1376). (European Signal Processing Conference). European Signal Processing Conference, EUSIPCO. https://doi.org/10.23919/EUSIPCO63174.2024.10715227

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Efficient Patient Fine-Tuned Seizure Detection with a Tensor Kernel Machine

Seline J.S. de Rooij
Signal Processing Systems
Delft University of Technology
Delft, Netherlands
s.j.s.derooij@tudelft.nl

Frederiek Wesel
Delft Center for Systems and Control
Delft University of Technology
Delft, Netherlands
f.wesel@tudelft.nl

Borbála Hunyadi
Signal Processing Systems
Delft University of Technology
Delft, Netherlands
b.hunyadi@tudelft.nl

***Abstract*—Recent developments in wearable devices have made accurate and efficient seizure detection more important than ever. A challenge in seizure detection is that patient-specific models typically outperform patient-independent models. However, in a wearable device one typically starts with a patient-independent model, until such patient-specific data is available. To avoid having to construct a new classifier with this data, as required in conventional kernel machines, we propose a transfer learning approach with a tensor kernel machine. This method learns the primal weights in a compressed form using the canonical polyadic decomposition, making it possible to efficiently update the weights of the patient-independent model with patient-specific data. The results show that this patient fine-tuned model reaches as high a performance as a patient-specific SVM model with a model size that is twice as small as the patient-specific model and ten times as small as the patient-independent model.**

***Index Terms*—seizure detection, tensors, kernel machine, transfer learning**

I. Introduction

Epilepsy is a common neurological disorder, it is characterized by abnormal electrical activity in the brain which causes seizures. Anti-epileptic drugs are only effective for about 70% of the patients. Thus, the remaining 30% continue to suffer from seizures, which impacts their quality of life [1].

To improve this quality of life, wearable seizure detection devices are being developed, such as subcutaneous EEG [2], wrist-worn [3] or behind-the-ear EEG devices [4], [5]. These devices could lead to more accurate diary keeping [4], and they may serve as a warning system when implemented in real-time [6].

This all requires an accurate and efficient seizure detection algorithm. Due to the large variability between epilepsy patients, patient-specific (PS) seizure detection algorithms outperform their patient-independent (PI) counterparts [7]. However, there is often a lack of labelled patient-specific data, which makes it hard to implement a PS model. Furthermore, recent research has shown that there is also an increasing intra-patient variability over time [8]. Suggesting, that even a patient-specific detector would need adaptation after some time.

This is why *transfer learning* has become increasingly popular for seizure detection [9], [10]. In transfer learning information from a source task is used to improve the performance of a target task. While there are many different types of transfer learning, in this paper we consider the case of *inductive* transfer learning, where both the source and the target task contain labels [9], [11]. Specifically, we consider the case where a patient-independent model is learned (*source*), which is later updated with a limited amount of patient-specific data (the *target* or *patient fine-tuned* model).

This type of transfer learning has been considered previously for seizure detection on wearable devices, using neural networks (CNN [12]) or kernel machines (SVM [13]). However, both of these methods have their limitations. Deep neural networks, for instance, typically contain a large number of model parameters and require a correspondingly large amount of data for training. This makes training computationally expensive. Although this could be remedied by fine-tuning only a few layers of the network [14], such an approach may lead to suboptimal results and the resulting model remains large in size.

Kernel machines, on the other hand, are well suited to small and medium size problems [15]. However, because they are typically trained in their dual form, they can become intractable for large-scale problems due to the construction of the kernel matrix and because the number of model parameters scales with the training data (the *support vectors*) [15]. Transfer learning with a kernel machine can be done by learning a *delta* model, to learn the difference between source and target [16]. This delta model is then added to the source model. The downside of this approach is that the update can only be done once and both the source and delta model need to be kept in memory. This makes it less suitable for a wearable device.

In this paper, we propose an efficient transfer learning method based on tensor kernel ridge regression (T-KRR) [17]. The advantage of the T-KRR is that it learns the kernel ridge regression problem in the primal form and uses the canonical polyadic decomposition (CPD) [18] to learn the weights in compressed form. The use of the primal form and the CPD makes it suitable for large-scale and high-dimensional data. Furthermore, it becomes easier to adapt the model to new data by simply updating (part of) the weights. To our knowledge, this is the first time the T-KRR classifier has been used for seizure detection and transfer learning.

Authorized licensed use limited to: TU Delft Library. Downloaded on November 18,2024 at 14:20:07 UTC from IEEE Xplore. Restrictions apply.

II. Classification Method

A. Notation and Preliminaries

Throughout this paper, the following notation conventions are used. Vectors and matrices are denoted by boldface lowercase and uppercase letters, respectively. Tensors, which are multi-dimensional arrays [19], are denoted by calligraphic letters, e.g. $\mathcal{A}$.

The symbols $\otimes$, $\odot$ and $\circledast$ are used to denote the tensor outer product, the Khatri-Rao product and the Hadamard product, respectively. The Frobenius inner product between two tensors is defined as, $\langle \mathcal{A}, \mathcal{B} \rangle_{\mathrm{F}} := \mathrm{vec}(\mathcal{A})^T \mathrm{vec}(\mathcal{B})$, where $\mathrm{vec}(\mathcal{A})_i = a_{i_1 i_2 \ldots i_D}$ is the vectorization of $\mathcal{A} \in \mathbb{R}^{I_1 \times \cdots \times I_D}$ to $\boldsymbol{a} \in \mathbb{R}^I$ where $I = \prod_{d=1}^{D} I_d$. A tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$ is considered to be *rank-one* if it can be written as the outer product of D vectors, $\mathcal{A} = \boldsymbol{a}^{(1)} \otimes \boldsymbol{a}^{(2)} \otimes \cdots \otimes \boldsymbol{a}^{(D)}$ [19].

As the number of elements in a tensor grows exponentially in the number of dimensions, low-rank tensor decompositions are used to represent the tensor with a reduced number of parameters [20]. In this paper, we make use of the Canonical Polyadic decomposition (CPD) [18], [21]. A rank-R CPD decomposes a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D}$ into a sum of R rank-one tensors,

$$\mathcal{A} := \sum_{r=1}^{R} \mu_r \; \boldsymbol{a}_r^{(1)} \otimes \boldsymbol{a}_r^{(2)} \otimes \cdots \otimes \boldsymbol{a}_r^{(D)}. \tag{1}$$

Here, the vectors $\boldsymbol{a}_r^{(d)}$ are normalized to length one and the scaling is absorbed by $\boldsymbol{\mu} \in \mathbb{R}^R$. The CPD can also be expressed as the Khatri-Rao product of *factor matrices*,

$$\mathrm{vec}(\mathcal{A}) := \left(\boldsymbol{A}^{(D)} \odot \boldsymbol{A}^{(D-1)} \odot \cdots \odot \boldsymbol{A}^{(1)} \right) \boldsymbol{\mu} \; .$$

The columns of these factor matrices contain the vectors of the rank-one components, i.e. $\boldsymbol{A}^{(d)} := [\boldsymbol{a}_1^{(d)} \; \boldsymbol{a}_2^{(d)} \; \cdots \; \boldsymbol{a}_R^{(d)}] \in \mathbb{R}^{I_d \times R}$.

For a sufficiently low rank this CPD results in a significant reduction in storage complexity compared to the full tensor, $\mathcal{O}\left(R \sum_d I_d\right)$ instead of $\mathcal{O}\left(\prod_d I_d\right)$.

B. Kernel Machines

In binary classification one aims to find a function $f(\cdot)$ that maps the input data $\boldsymbol{x} \in \mathbb{R}^D$ to the correct label $y \in \{-1, +1\}$.

Kernel machines typically assume the following model form,

$$f(\boldsymbol{x}) = \boldsymbol{\phi}(\boldsymbol{x})^T \boldsymbol{w} = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{w} \rangle_{\mathrm{F}}, \tag{2}$$

where $\boldsymbol{w} \in \mathbb{R}^M$ are the model weights. And $\boldsymbol{\phi}(\cdot) : \mathbb{R}^D \to \mathbb{R}^M$ is a feature map, which maps the input data to a higher dimensional feature space, a kernel Hilbert space, to enable nonlinear classification. The decision boundary for classification can be obtained by applying the sign function to the model response [15].

The model weights $\boldsymbol{w}$ can be learned by minimizing a convex and symmetric regularized loss function. An SVM [22], for instance, uses a hinge loss function, whereas kernel ridge regression (KRR) (or the LS-SVM) [15] uses the squared loss with Frobenius norm regularization. This leads to the following minimization problem,

$$\min_{\boldsymbol{w}} \sum_{n=1}^{N} \left(\langle \boldsymbol{\phi}\left(\boldsymbol{x}_n\right), \boldsymbol{w} \rangle_{\mathrm{F}} - y_n \right)^2 + \lambda \langle \boldsymbol{w}, \boldsymbol{w} \rangle_{\mathrm{F}}, \tag{3}$$

for N training samples.

Due to the high dimensionality of the feature map, $\phi(\cdot)$, often the *kernel trick* is applied, which results in the non-parametric dual formulation. Therein, explicit feature maps only manifest as inner products, thus they can be replaced by a kernel function $\kappa(\boldsymbol{x}, \boldsymbol{x}') := \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{x}') \rangle_{\mathrm{F}}$. The downside of this dual formulation is that it requires the calculation (and the formal inversion [15]) of the kernel matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$, $K_{i,j} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$, which becomes intractable for large N.

C. Tensor Kernel Ridge Regression

In tensor kernel ridge regression (T-KRR) [17] the following feature mappings are considered,

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{\phi}^{(1)}(x_1) \otimes \boldsymbol{\phi}^{(2)}(x_2) \otimes \cdots \otimes \boldsymbol{\phi}^{(D)}(x_D), \tag{4}$$

where $\boldsymbol{\Phi}(\cdot) : \mathbb{R}^D \to \mathbb{R}^{M_1} \otimes \mathbb{R}^{M_2} \otimes \cdots \otimes \mathbb{R}^{M_D}$ is a tensor product of D vectors and creates a rank one tensor. Here, $\boldsymbol{\phi}^{(d)}(\cdot) : \mathbb{R} \to \mathbb{R}^{M_d}$ is the (local) feature map applied to the d-th dimension of the input.

Using (4) as the feature map we can reformulate the primal form of kernel ridge regression (3) as follows,

$$\min_{\mathcal{W}} \sum_{n=1}^{N} \left(\langle \boldsymbol{\Phi}(\boldsymbol{x}_n), \mathcal{W} \rangle_{\mathrm{F}} - y_n \right)^2 + \lambda \langle \mathcal{W}, \mathcal{W} \rangle_{\mathrm{F}}, \tag{5}$$

where the weight vector $\boldsymbol{w}$ has been reshaped into a D-dimensional weight tensor $\mathcal{W} \in \mathbb{R}^{M_1 \times M_2 \times \cdots \times M_D}$ by means of the vectorization identity (Section II-A).

As the size of the weight tensor scales exponentially with the dimension of the input data D, (5) becomes intractable for high-dimensional input data. This is why in [17] the weight tensor is constrained to have a rank-R CPD structure. The CPD weight tensor can be determined using an alternating linear scheme (ALS)[1]. Starting from some initial value, $\{\boldsymbol{W}_0^{(d)}\}_{d=1}^{D}$, the ALS algorithm updates the factor matrices of the CPD one at a time by solving,

$$\begin{aligned} \min_{\boldsymbol{W}^{(d)}} \sum_{n=1}^{N} \left(\left\langle \boldsymbol{g}^{(d)}(\boldsymbol{x}_n), \mathrm{vec}(\boldsymbol{W}^{(d)}) \right\rangle_{\mathrm{F}} - y_n \right)^2 \\ + \lambda \left\langle \mathrm{vec}\left(\boldsymbol{W}^{(d)^T} \boldsymbol{W}^{(d)} \right), \mathrm{vec}\left(\boldsymbol{H}^{(d)} \right) \right\rangle_{\mathrm{F}}, \end{aligned} \tag{6}$$

where, $\boldsymbol{g}^{(d)}(\boldsymbol{x}) := \boldsymbol{\phi}^{(d)} \otimes \left(\boldsymbol{\phi}^{(1)^{\mathrm{T}}} \boldsymbol{W}^{(1)^{\mathrm{T}}} \circledast \cdots \circledast \boldsymbol{\phi}^{(D)^{\mathrm{T}}} \boldsymbol{W}^{(D)^{\mathrm{T}}} \right)$ and $\boldsymbol{H}^{(d)} := \boldsymbol{W}^{(1)^{\mathrm{T}}} \boldsymbol{W}^{(1)} \circledast \cdots \circledast \boldsymbol{W}^{(D)^{\mathrm{T}}} \boldsymbol{W}^{(D)}$.

As in [17] we approximate the RBF kernel with Laplace basis functions [23]. Thus, the feature map is defined by,

$$\left(\boldsymbol{\phi}^{(d)}\left(x_d\right) \right)_{i_d} = \frac{1}{\sqrt{U_d}} p\left(\frac{\pi i_d}{2U_d} \right) \sin\left(\frac{\pi i_d \left(x_d + U_d\right)}{2U_d} \right), \tag{7}$$

[1]For a detailed derivation please refer to [17] and its appendix.

Authorized licensed use limited to: TU Delft Library. Downloaded on November 18,2024 at 14:20:07 UTC from IEEE Xplore. Restrictions apply.

for input data centered in a hyperbox, $x_d \in [-U_d, U_d]$, and where $i_d = 1, \ldots, M_d$ and $p(\cdot)$ is the spectral density of the RBF kernel [24].

D. Transfer Learning with T-KRR

As stated in the introduction, in transfer learning knowledge from a *source* task is transferred to a related *target* task in order to improve its performance [9], [11]. A common technique to perform inductive transfer learning is *fine-tuning*. In fine-tuning pre-trained weights from the source task are (partially) updated using data from the target task [14].

For the T-KRR, fine-tuning can easily be accomplished by using the weights of the source task as the initial value for the ALS optimization of the target task, and then updating (some of the) factor matrices of the CPD weight tensor using the target data, i.e. $\forall d : \boldsymbol{W}^{(d)}_{0,\ \text{target}} = \boldsymbol{W}^{(d)}_{\text{source}}$.

In this paper, we consider the specific case where the weights are first trained patient-independently and then *fine-tuned* with a small amount of patient-specific data.

III. Data Processing

A. Data

In this paper, EEG data from the Temple University Seizure Corpus (TUSZ, v1.5.2) [25] was used to train and test the models. This is the largest publicly available EEG seizure detection dataset. It contains a total of 3.9×10^6 seconds of annotated EEG data. Each recording session contains a `.txt` file with information about the patient and the seizure morphology. The dataset contains both EEG recordings that use the average reference montage and ones that use the linked ears reference montage. In our case, only the latter was used.

Furthermore, we wanted to create a scenario that most closely reflects that of a wearable behind-the-ear EEG device, such as the ones presented in [4], [5]. Therefore, we used the T1 and T2 channels [26], located at the temporal lobe close to the ear, as a surrogate for the behind-the-ear channels. Additionally, we selected only patients who had temporal lobe seizures to ensure that the seizures were detectable with these channels. The following procedure was used to select the relevant recordings:

1) Perform a keyword search on the `.txt` files for the terms "left temporal" and "right temporal".
2) Check if the recordings contain the T1 and T2 channels.
3) Count the number of seizures per patient with a minimal seizure duration of 10 s.
4) Select patients with ≥ 5 seizures and perform a manual check of the `.txt` files to confirm the patient has temporal lobe seizures.

This leads us to 6 patients with a median seizure count of 9 (± 4) seizures per patient. In total, these recordings contain 7.3×10^4 seconds of background data and 2.7×10^4 seconds of seizure data.

TABLE I
Extracted features [4], [27].

Time domain	1-3. Number of zero crossings, maxima and minima 4. Skewness 5. Kurtosis 6. RMS amplitude 7. Linelength
Frequency domain	8. Total power 9. Peak frequency 10-14. Mean power in frequency bands: δ (1-3 Hz), θ (4-8 Hz), α (9-13 Hz), β (14-20 Hz), HF (40-80 Hz).
Entropy	15. Spectral entropy 16. Sample entropy

B. Preprocessing and Feature Extraction

All recordings were resampled to a sampling frequency of 250 Hz. Then, a bandpass filter (0.1-50 Hz, 4th order Butterworth) and a notch filter (50 Hz) was applied to each channel to reduce the presence of artifacts.

After filtering the EEG data was divided into 2s windows. For the seizure data 50% overlap between windows was used, while no overlap was used for the non-seizure windows. This was done to balance the data for training. For each window features were extracted. The extracted features per channel are presented in Table I. It should be noted that the high-frequency band features (HF) were extracted prior to applying the low-pass filter. The presented features are a selection of features used in [4] with the addition of the line length feature, which was shown to work well for seizure detection [27]. Since we only use two EEG channels (T1 and T2), this amounts to a total of 32 features.

Given our dataset contains recordings of seizures originating in both the left and right temporal lobes, we sort the features to remove this spatial structure [4]. Furthermore, all features (Table I) were scaled to lie in a unit hyperbox (i.e. *min-max* scaling was used).

IV. Results and Discussion

All models described in this section were implemented in Python [28]. The code and the corresponding models (with the used hyperparameters) can be found on Github.[2]

A. Validation and Hyperparameter Tuning

The patient-independent (PI) models were evaluated using leave-one-patient-out cross-validation. The hyperparameters were tuned by gridsearch on the training set, using 5-fold cross-validation and maximizing the area under the receiver operating characteristic curve (AUROC).

For the patient fine-tuned (PF) models, data from only one seizure (with some background data) was used to fine-tune the corresponding PI model using the same hyperparameters. Thus, in this case, a leave-one-seizure-*in* cross-validation scheme was used. This validation scheme was also used for the patient-specific models.

[2]https://github.com/sderooij/PF-TKRR-public/

Authorized licensed use limited to: TU Delft Library. Downloaded on November 18,2024 at 14:20:07 UTC from IEEE Xplore. Restrictions apply.

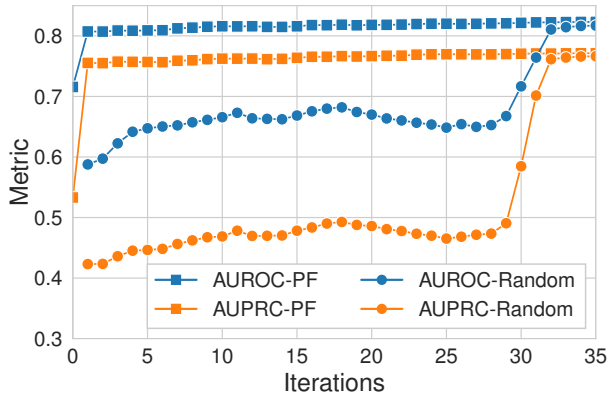


Fig. 1. AUC against the number of iterations of the ALS algorithm for T-KRR. Blue shows the AUROC and orange the AUPRC. Square dots represent the T-KRR$_{PF}$ model and round dots the T-KRR$_{PS}$ model with random initialization.

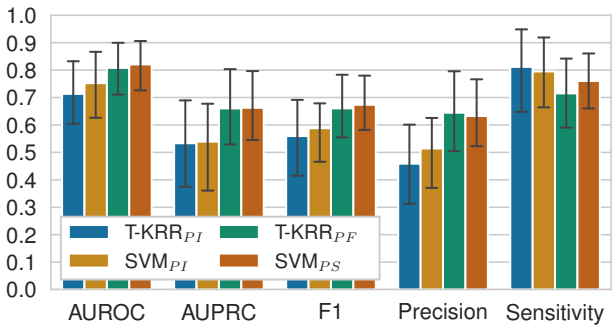


Fig. 2. Barplot of the performance for the different classifiers. The error bars show the 95% confidence interval.

To evaluate the performance of the classifier we used the following metrics: AUROC, area under the precision-recall curve (AUPRC), F1-score, sensitivity and precision. The decision boundaries were chosen to maximize the F1-score. The overlapped windows were removed from the test set before calculating the performance metrics.

B. Classification Performance

To investigate the influence of the number of iterations for fine-tuning the T-KRR, we plot the AUCPR and the AUROC (patient-mean) against the number of iterations (Figure 1). An iteration equals solving (6), i.e. updating one factor matrix of the weight tensor in CPD format. In the plot, we show two cases. One is the aforementioned fine-tuning approach, where the PI weights are used as the initial value for training with patient-specific data: the T-KRR$_{PF}$ model. In the other case, the elements of the CPD weight tensor are initialized from a standard normal distribution and trained on the same PS data: the T-KRR$_{PS}$ model.

From Figure 1 one can see that the T-KRR$_{PF}$ model has a significant increase in performance already after the first iteration compared to the patient-independent T-KRR$_{PI}$ model (at zero iterations). As the number of iterations grows, the performance increase is more gradual. The T-KRR$_{PS}$ model needs at least 32 iterations, i.e. updating all the factor matrices, to reach the performance level of the T-KRR$_{PF}$ model. Thus, starting from the PI weights yields a much faster convergence.

Next, we compare the performance of the T-KRR classifier to that of a conventional SVM classifier with an RBF kernel. Both a patient-independent and a patient-specific SVM model were constructed. For the T-KRR$_{PF}$ a full sweep of 32 iterations was used. Figure 2 provides barplot of the considered performance metrics. This barplot shows the mean performance across the patients and the corresponding 95% confidence interval.

In these results, we see that the performance of the T-KRR$_{PI}$ model seems slightly worse than the performance of the SVM$_{PI}$ model. This result may be due to the approximation of the RBF kernel (7) or the low-rank CPD approximation of the weight tensor. Furthermore, it is clear that being able to train on even a little bit of patient-specific data improves the performance significantly. The T-KRR$_{PF}$ model shows a comparable performance to the SVM$_{PS}$ model.

C. Model complexity

Aside from the performance of the different classifiers, we also compare the complexity of the different models. When it comes to training an SVM, the cost is at least $\mathcal{O}\left(N^2\right)$ and at most $\mathcal{O}\left(N^3\right)$ [29], whereas for the T-KRR every ALS iteration costs $\mathcal{O}\left(NM^2R^2\right)$, for $M = \max(M_d)$.

When it comes to inference, the complexity is dependent on the number of model parameters. For the SVM this is $\mathcal{O}\left(N_{SV}D\right)$, where N_{SV} is the number of support vectors ($N_{SV} \leq N$). And for T-KRR this is $\mathcal{O}\left(DMR\right)$. This means that for the models shown in Figure 2, the T-KRR$_{PI}$ and T-KRR$_{PF}$ have the same number of parameters ($\sim 2 \times 10^4$), while the SVM$_{PI}$ models have a magnitude larger size than the SVM$_{PS}$ models ($\sim 9 \times 10^5$ vs. $\sim 4 \times 10^4$ parameters).

V. Conclusion

This paper investigated using the T-KRR classifier for seizure detection in a transfer learning setting. It was shown that this model is easily adaptable by using the previously trained weights as an initial value for training with new data. This concept was tested by updating the patient-independent model with a small amount of patient-specific data. The results showed that updating just one of the 32 factor matrices of the CPD weight tensor already improved the performance significantly. Ultimately, the patient fine-tuned model performed as well as a patient-specific SVM classifier trained on the same data, while using twice as few parameters.

Furthermore, for a patient-independent model, the performance of the T-KRR classifier is only slightly worse than that of the standardly used SVM classifier, while the T-KRR has ten times fewer parameters. This makes T-KRR classifier more suitable for a wearable device in case of strict memory requirements.

Authorized licensed use limited to: TU Delft Library. Downloaded on November 18,2024 at 14:20:07 UTC from IEEE Xplore. Restrictions apply.

In future work, it would be of interest to investigate the effectiveness of the T-KRR classifier for *online* transfer learning.

VI. Acknowledgements

S.J.S. de Rooij and F. Wesel, and thereby this work, are supported by the TU Delft AI labs program.

References

[1] World Health Organization, "Epilepsy: a public health imperative: summary," Tech. Rep. WHO/MSD/MER/19.2, 2019.

[2] S. Weisdorf, J. Duun-Henriksen, M. J. Kjeldsen, F. R. Poulsen, S. W. Gangstad, and T. W. Kjær, "Ultra-long-term subcutaneous home monitoring of epilepsy—490 days of EEG from nine patients," *Epilepsia*, vol. 60, no. 11, pp. 2204–2214, 2019.

[3] S. Beniczky, T. Polster, T. W. Kjaer, and H. Hjalgrim, "Detection of generalized tonic–clonic seizures by a wireless wrist accelerometer: A prospective, multicenter study," *Epilepsia*, vol. 54, no. 4, pp. e58–e61, 2013.

[4] K. Vandecasteele, T. De Cooman, J. Dan, E. Cleeren, S. Van Huffel, B. Hunyadi, and W. Van Paesschen, "Visual seizure annotation and automated seizure detection using behind-the-ear electroencephalographic channels," *Epilepsia*, vol. 61, no. 4, pp. 766–775, Apr. 2020.

[5] S. You, B. Hwan Cho, Y.-M. Shon, D.-W. Seo, and I. Y. Kim, "Semi-supervised automatic seizure detection using personalized anomaly detecting variational autoencoder with behind-the-ear EEG," *Computer Methods and Programs in Biomedicine*, vol. 213, p. 106542, Jan. 2022.

[6] V. Naganur, S. Sivathamboo, Z. Chen, S. Kusmakar, A. Antonic-Baker, T. J. O'Brien, and P. Kwan, "Automated seizure detection with noninvasive wearable devices: A systematic review and meta-analysis," *Epilepsia*, vol. 63, no. 8, pp. 1930–1941, 2022.

[7] M. K. Siddiqui, R. Morales-Menendez, X. Huang, and N. Hussain, "A review of epileptic seizure detection using machine learning classifiers," *Brain Informatics*, vol. 7, no. 1, p. 5, Dec. 2020.

[8] G. M. Schroeder, B. Diehl, F. A. Chowdhury, J. S. Duncan, J. de Tisi, A. J. Trevelyan, R. Forsyth, A. Jackson, P. N. Taylor, and Y. Wang, "Seizure pathways change on circadian and slower timescales in individual patients with focal epilepsy," *Proceedings of the National Academy of Sciences*, vol. 117, no. 20, pp. 11 048–11 058, May 2020.

[9] X. Cui, J. Cao, T. Jiang, and F. Gao, "Transfer Learning Based Seizure Detection: A Review," in *Cognitive Computation and Systems*, ser. Communications in Computer and Information Science, F. Sun, J. Li, H. Liu, and Z. Chu, Eds. Singapore: Springer Nature, 2023, pp. 160–175.

[10] Z. Wan, R. Yang, M. Huang, N. Zeng, and X. Liu, "A review on transfer learning in EEG signal analysis," *Neurocomputing*, vol. 421, pp. 1–14, Jan. 2021.

[11] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[12] F. Pisano, G. Sias, A. Fanni, B. Cannas, A. Dourado, B. Pisano, and C. A. Teixeira, "Convolutional Neural Network for Seizure Detection of Nocturnal Frontal Lobe Epilepsy," *Complexity*, vol. 2020, p. e4825767, Mar. 2020.

[13] T. De Cooman, K. Vandecasteele, C. Varon, B. Hunyadi, E. Cleeren, W. Van Paesschen, and S. Van Huffel, "Personalizing Heart Rate-Based Seizure Detection Using Supervised SVM Transfer Learning," *Frontiers in Neurology*, vol. 11, 2020.

[14] N. Tajbakhsh, J. Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?" *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, May 2016.

[15] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vandewalle, *Least Squares Support Vector Machines*. World Scientific, Nov. 2002.

[16] J. Yang, R. Yan, and A. G. Hauptmann, "Adapting SVM Classifiers to Data with Shifted Distributions," in *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, Oct. 2007, pp. 69–76.

[17] F. Wesel and K. Batselier, "Large-Scale Learning with Fourier Features and Tensor Decompositions," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 17 543–17 554.

[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, Jan. 1977.

[19] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[20] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, and D. P. Mandic, "Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions," *Foundations and Trends® in Machine Learning*, vol. 9, no. 4-5, pp. 249–429, 2016.

[21] F. L. Hitchcock, "The Expression of a Tensor or a Polyadic as a Sum of Products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[23] A. Solin and S. Särkkä, "Hilbert space methods for reduced-rank Gaussian process regression," *Statistics and Computing*, vol. 30, no. 2, pp. 419–446, Mar. 2020.

[24] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, Nov. 2005.

[25] I. Obeid and J. Picone, "The Temple University Hospital EEG Data Corpus," *Frontiers in Neuroscience*, vol. 10, May 2016.

[26] S. Obretenova, M. F. Villamar, and S. Tobochnik, "Addition of Anterior Temporal EEG Electrodes to Improve Seizure Detection," *The Neurohospitalist*, vol. 11, no. 1, pp. 89–90, Jan. 2021.

[27] B. R. Greene, S. Faul, W. P. Marnane, G. Lightbody, I. Korotchikova, and G. B. Boylan, "A comparison of quantitative EEG features for neonatal seizure detection," *Clinical Neurophysiology*, vol. 119, no. 6, pp. 1248–1261, Jun. 2008.

[28] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[29] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

Authorized licensed use limited to: TU Delft Library. Downloaded on November 18,2024 at 14:20:07 UTC from IEEE Xplore. Restrictions apply.