Epileptic seizure classification using scalp EEG data

A support tensor machine approach

M.P. van Dijk

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Epileptic seizure classification using scalp EEG data

A support tensor machine approach

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

M.P. van Dijk

April 9, 2024

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

Abstract

Algorithms which can effectively detect epileptic seizures have the potential to improve current treatment methods for people who suffer from epilepsy. The current state-of-the-art methods use neural networks, which are able to learn directly from the electroencephalogram (EEG) data without feature extraction. However, neural networks have drawbacks—they are time and data intensive to train and require a high number of parameters for efficient classification. This thesis proposes a novel approach to the epileptic seizure detection problem using Support Tensor Machines (STMs). The final models are able to learn directly from EEG data and use far less model parameters than a state-of-the-art model using a convolutional neural network. Three types of experiments have been conducted using different representations for the EEG data. The STMs that have been used in the experiments are the Support Higher-Order Tensor Machine, Dual Structure-preserving Kernel (DuSK) and Tensor Train Multi-way Multi-level Kernel (TT-MMK). The results show that by using TT-MMK (with a tensorized data representation) for leave-one-seizure-out validation and DuSK (with the original data representation) for leave-one-patient-out validation, the models are able to rival a state-of-the-art epileptic seizure detector.

Table of Contents

List of Figures

List of Tables

Preface & Acknowledgements

This document is a part of my Master of Science graduation thesis within the MSc Systems & Control program at Delft University of Technology. The subject of the thesis was proposed by my daily supervisor: ir. S.J.S. de Rooij. All code written for this research is publicly available on Github: **https://github.com/MeesvanDijk/MSc-Thesis**.

I would like to thank my supervisors dr.ir. K.Batselier and ir. S.J.S. de Rooij for their assistance and guidance during the writing of this thesis.

Delft, University of Technology
April 9, 2024

M.P. van Dijk

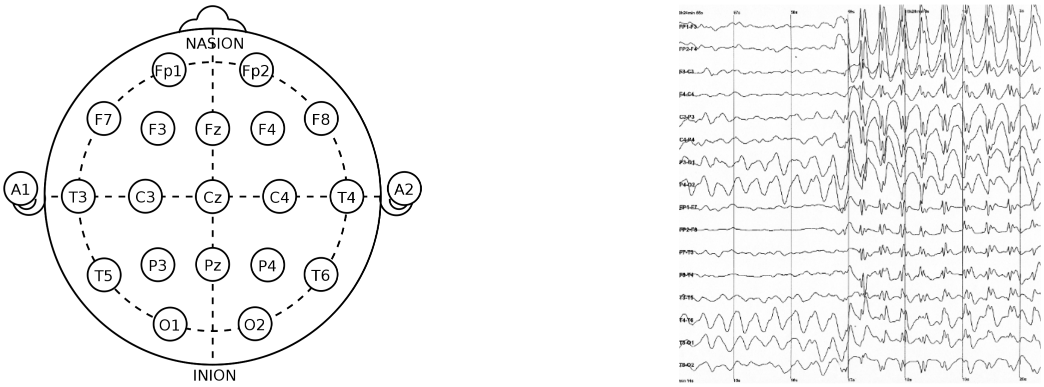
Chapter 1

Introduction

1-1 Epileptic seizure detection

Epilepsy is a group of chronic neurological disorders distinguished by spontaneous recurrent seizures caused by abnormal electrical activity in the brain. Epilepsy is a life shortening disease affecting around 70 million people worldwide [4]. Despite recent advances in the development of antiepileptic medication there is still a group of around 30% of patients that are resistant to the current medicines, this is called intractable (drug-resistant) epilepsy [5].

The most widely used method used to diagnose epilepsy is the electroencephalogram (EEG). EEG is a method to record an electrogram of the electrical activity of the brain using electrodes on the scalp. The internationally recognized 10-20 system is used to standardize the location of the applied scalp electrodes. This setup can be seen in figure 1-1a. The kind of data collected by the EEG monitoring device can be seen in figure 1-1b, where every track on the y-axis corresponds to data from a single electrode with respect to a reference. This reference could be a fixed electrode or a different electrode depending on the location on the scalp.



(a) The 10-20 electrode setup [6].

(b) Example of EEG data [7].

Figure 1-1: Figures illustrating the measurement method of scalp EEG data and example EEG data.

An EEG recording can be divided into different phases as can been seen in figure 1-2 , which has labeled phases for an example seizure. The interictal phase is the phase in between seizures where no seizure activity is present. The preictal phase is the phase right before a seizure where abnormal activity is present. The ictal phase is the phase where the most noticeable part of the seizure occurs. The postictal phase is the phase right after the ictal phase where there is still some abnormal behaviour as a result of the seizure. When the postical phase returns to baseline, it is called interictal again. The exact moment phase-transitions happen can however be ambiguous, therefore it might be better not to label arbitrary stages as being ictal or postical except for when the boundaries are very clear [8]. The frequency range of EEG signals is commonly divided into different frequency bands associated with specific brain states and activities. The main frequency bands observed in EEG are delta waves (0.5 -4 Hz), theta waves (4 - 7 Hz), alpha waves (8 - 12 Hz), beta waves (13 - 30 Hz) and Gamma waves (30 - 50 Hz) [9].

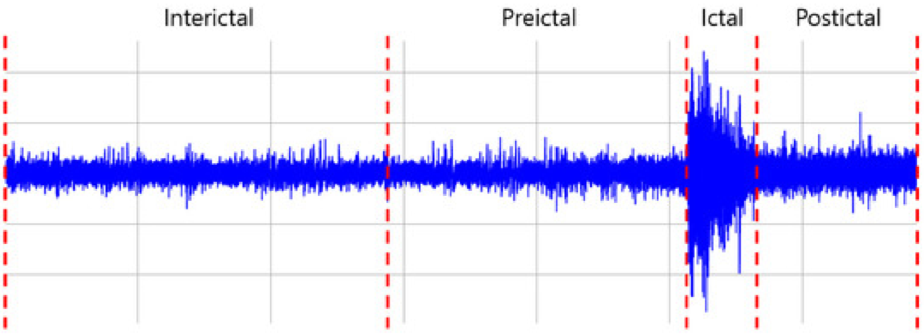


Figure 1-2: Example EEG data illustrating the various phases [1].

Visual inspection of EEG data for epilepsy diagnosis can be time-consuming and costly since recordings can be up to days long and have to be reviewed manually by a specialist. In order to overcome this problem, Gotman [10] created the first algorithm in 1982 to automatically detect epileptic seizures using EEG data. The algorithm presented by Gotman combined a decomposition with set detection criteria, which is called a thresholding algorithm. Although the algorithm by Gotman is currently outdated, it paved the way for the epileptic seizure detection field of research. Today, far more advanced algorithms exist which are improved every year [11]. These algorithms have the potential of being used by neurologists for more time-efficient diagnosis by letting the algorithm pre-select time segments with a high probability of a seizure.

The development of these types of algorithms, together with the tremendous growth of computational power of computer chips over the last decades, has opened up the path for new monitoring and treatment methods for epilepsy. This is especially beneficial for patients suffering from intractable epilepsy. Some of these applications will be discussed in the next section.

1-2 Applications of epileptic seizure detection

Wearable devices for monitoring EEG data of patients have the potential to make the diagnosis of epilepsy easier and less time-consuming. This is because the devices can be worn outside the hospital in the daily lives of the patients, eliminating the need for continuous monitoring by a specialist in the hospital. This has both economical benefits, as it requires less time from a specialist, and social benefits, as it is more convenient for the patient. Currently, there are two wearable devices on the market that collect data for diagnosis. These wearables can monitor 2-channel behind-the-ear EEG data and are known as Sensordot and Epilog [12]. The algorithms embedded in these devices automatically detect seizure segments from the behind-the-ear EEG and transmit them to specialists. While these devices already demonstrate effectiveness in practice, the development of better and more efficient algorithms is necessary to enhance the viability of such wearables for all patients.

A relatively new treatment method for patients with intractable epilepsy is responsive neurostimulation (RNS). This treatment method uses an implanted neurostimulator that continuously monitors brain activity. When an upcoming seizure is detected the neurostimulator tries to terminate it in its early stages by sending electrical signals to the part of the brain where the seizure occurs. The main difference with other treatment methods that use an implant, like deep brain stimulation (DBS) and vagus nerve stimulation (VNS), is the closed-loop configuration of RNS. DBS and VNS have a preset cycle on which they send their signals to the brain. RNS only sends control inputs to the brain if a seizure is detected. This closed-loop configuration of RNS can be seen in figure 1-3.

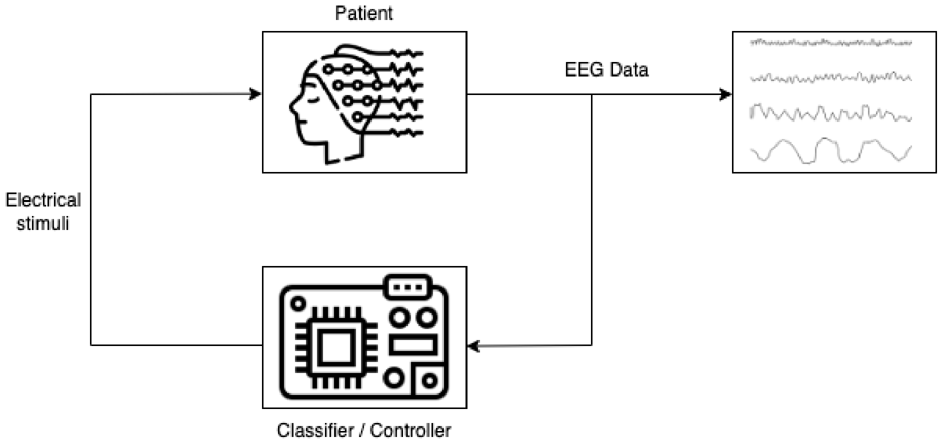


Figure 1-3: Closed-loop control of a RNS system.

There already is a RNS system on the market which claims that 44% of patients that participated in their study had at least 50% fewer seizures [13]. These type of closed-loop applications of electrical stimulation via chronically implanted electrodes should be designed in a way to keep the size of the implant as small as possible. In order to keep the device as small as possible the battery size needs to be minimized. As a consequence of a smaller battery, the seizure detection algorithm has to be computationally more efficient while still scoring high on performance metrics. As a consequence, more energy efficient and better algorithms have to be designed in order to improve wearable EEG detectors and EEG implants.

1-3 Research focus

In this research, a tensor-network approach to the epileptic seizure detection problem is carried out. This is done by making use of Support Tensor Machines (STMs), which are the multidimensional extension to the support vector machine (SVM). These STMs make use of tensor networks, which will be discussed in more detail in chapter 4. In this research three different types of STMs are considered:

- *Support Higher Order Tensor Machine*, which is a linear Support Tensor Machine (STM) using the Canonical Polyadic (CP) decomposition.
- *Dual Structure Preserving Kernel*, which is a kernelized (non-linear) STM using the CP decomposition.
- *Tensor Train Multi-Level Multi-way Kernel*, which is a kernelized (non-linear) STM using the Tensor-Train (TT) and the CP decompositions.

These methods have been selected based on their interdependent nature, as will become clear from chapter 3.

In order to obtain third order tensors form the original (matrix) EEG data, two methods are used. Firstly, the Continuous Wavelet Transform (CWT) is used in order to transform the data from the time domain into the time-frequency domain, yielding a third order tensor. The CWT will be discussed in more detail in chapter 4. The second method used to obtain a tensor is by reshaping the dimensions, this is called tensorization.

The main goal of the research is to design a classifier capable of detecting epileptic seizures using a minimal amount of pre-processing steps and without manually extracting features. The methods currently capable of learning directly from the data using minimal pre-processing are neural networks and convolutional neural networks in particular. Unfortunately, these models take a long time to train since the amount of model parameters are usually in the 100,000's. This research is an attempt at achieving the aforementioned mentioned design goals using less model parameters with a STM approach.

1-4 Chapter outline and basic notation

In the first part of chapter 2 the basics of tensors and some relevant multi-linear operations are introduced. Later on in chapter 2 the relevant tensor decompositions used in the research are presented. Subsequently, in chapter 3 a brief summary to the SVM is given and the relevant STMs are explained in more detail. Hereafter, in chapter 4, an overview and explanation of all the experiments is given. In chapter 5 all the results will be presented and discussed. Lastly, in chapter 6 the conclusions will be presented and recommendations for future research will be given.

In table 1-1, the research questions that will be addressed in this thesis are presented. The table also indicates in which section the research question will be addressed.

Section	Research question
5-1	1. *"Do the proposed STMs possess classification capabilities when first applying the CWT?"*
5-2	2. *"Do the proposed STMs possess classification capabilities when working with the original EEG data?"*
5-3	3. *"Does tensorizing the data by reshaping, increase the performance when comparing it to the performance on the original data?*
5-4-2	4. *"Are the newly proposed methods able to rival the state-of-the-art methods in terms of performance (Accuracy & F1-score) and model complexity?"* .

Table 1-1: Addressed research question per section.

In table 1-2, the basic notation is listed that will be used throughout this thesis. Some of the notation will be explained in more detail in chapter 2.

Notation	Definition
x	Scalar
$\mathbf{x}$	Vector
$\mathbf{X}$	Matrix
$\underline{\mathbf{X}}$	Tensor
$\mathbf{x}(i)$	i-th entry of vector $\mathbf{x}$
$\mathbf{X}(i,j)$	Element (i,j) of matrix $\mathbf{X}$
$\underline{\mathbf{X}}(i_1, i_2, \ldots, i_d)$	Element $i_1, i_2, \ldots, i_d$ of d-dimensional tensor $\underline{\mathbf{X}}$
$A^T, A^{-1}, A^\dagger$	transpose, inverse and Moore-Penrose pseudo-inverse of matrix $\mathbf{A}$
$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \ldots I_{n-1} I_{n+1} \ldots I_N}$	mode-n matricization of $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$
$\circ, \odot, \otimes$	outer, Khatri-Rao, Kronecker products
$\mathbf{x} = \mathbf{vec}(\underline{\mathbf{X}})$	vectorization of $\underline{\mathbf{X}}$
$\lVert \ldots \rVert_F$	Frobenius norm
$\lVert \ldots \rVert_\star$	Nuclear norm

Table 1-2: Basic notation.

Chapter 2

Tensor Networks

2-1 Tensor basics

Tensors are mathematical objects that generalize scalars, vectors, and matrices to higher dimensions. A scalar is a 0-order tensor, a vector a 1st order tensor, a matrix a 2nd order tensor and a cube a 3rd order tensor. Above 3-dimensions tensors become difficult to visualize for humans, since we live in a 3-dimensional world. In the next section a method is introduced to visualize tensors of higher order. The mathematical notation for a N^{th} order tensor is $\underline{\mathbf{X}}$ $\in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, where N can be arbitrarily large. In figure 2-1 a 3rd order tensor is displayed where two dimensions are fixed, resulting in fibers. In figure 2-2 only one dimension of the same 3rd order tensor is fixed resulting in slices.

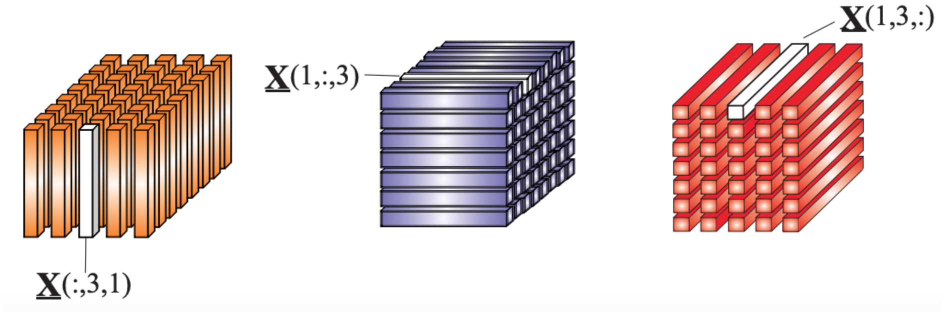

$\underline{\mathbf{X}}(:,3,1)$
$\underline{\mathbf{X}}(1,:,3)$
$\underline{\mathbf{X}}(1,3,:)$

Figure 2-1: Fibers in a 3rd order tensor [2].

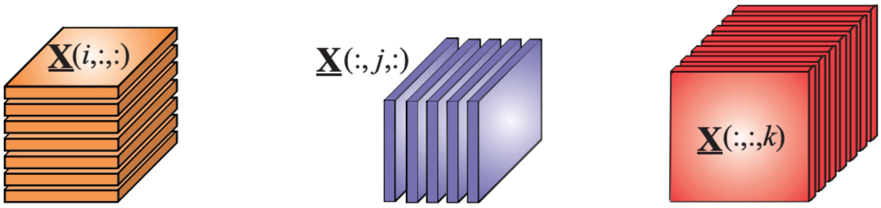

$\underline{\mathbf{X}}(i,:,:)$
$\underline{\mathbf{X}}(:,j,:)$
$\underline{\mathbf{X}}(:,:,k)$

Figure 2-2: Slices in a 3rd order tensor [2].

In order to work with tensor structures, some basic notation and (multilinear) operations need to be introduced.

Definition 2-1.1 (Multi-indices)**.** A multi-index is written as

$$i = \overline{i_1 i_2 \cdots i_N}$$

and refers to an index which takes all possible combinations of values of indices, $i_1, i_2, \ldots, i_N$ in a specific order. They can be defined using the Little-endian convention (reverse lexicographic ordering) or the big-endian (colexicographic ordering). In this thesis the little-endian convention will be used, it is defined as:

$$\overline{i_1 i_2 \ldots i_N} = i_1 + (i_2 - 1)\, I_1 + (i_3 - 1)\, I_1 I_2 + \cdots + (i_N - 1)\, I_1 \cdots I_{N-1}.$$

where $i_1, i_2, \ldots, i_N$ represents the index along each mode and $I_1, I_2, \ldots, I_N$.

Definition 2-1.2 (mode-n matricization)**.** A tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ can be unfolded along each mode in order to create a matrix. The result is the mode-n matricization, which is a matrix of the form

$$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 I_2 \cdots I_{n-1} I_{n+1} \cdots I_N}.$$

Definition 2-1.3 (mode-n product)**.** The mode-n product of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and vector $\mathbf{b}$ is denoted by

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \times_n \mathbf{b}$$

yields a tensor

$$\underline{\mathbf{C}} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}.$$

Similarly, the mode-n product of tensor $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and matrix $\mathbf{B}$ is denoted by

$$\underline{\mathbf{C}} = \underline{\mathbf{A}} \times_n \mathbf{B}$$

yields the tensor

$$\underline{\mathbf{C}} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$$

where J is the remaining dimension of matrix $\mathbf{B}$.

Definition 2-1.4 (Outer product of vectors)**.** The outer product of vectors is denoted by

$$\underline{\mathbf{X}} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \in \mathbb{R}^{I \times J \times K}.$$

It is the outer product of vectors $\mathbf{a}, \mathbf{b}$ and $\mathbf{c}$ which form the rank-1 tensor $\underline{\mathbf{X}}$, with entries $x_{ijk} = a_i b_j c_k$.

Definition 2-1.5 (Tensor contractions)**.** A tensor contraction can be considered a higher dimensional analogue of matrix multiplication, inner product and outer product [2]. In a similar way as the mode-n product, a tensor contraction of two tensors, $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$, with common modes, $I_n = J_m$, yields an (N+M-2)-order tensor $\underline{\mathbf{C}} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N \times J_1 \times \cdots \times J_{m-1} \times J_{m+1} \times \cdots \times J_M}$.

Definition 2-1.6 (Kronecker product [2])**.** The (left) Kronecker product of tensors $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\underline{\mathbf{B}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ yields a tensor $\underline{\mathbf{C}} \in \mathbb{R}^{I_1 J_1 \times \cdots \times I_N J_N}$, with entries $c_{\overline{i_1 j_1}, \ldots, \overline{i_N j_N}} = a_{i_1, \ldots, i_N} b_{j_1, \ldots, j_N}$.

Definition 2-1.7 (Khatri-Rao product [2])**.** The (left) Khatri-Rao product of matrices $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_J] \in \mathbb{R}^{I \times J}$ and $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_J] \in \mathbb{R}^{K \times J}$ yields a matrix $\mathbf{C} \in \mathbb{R}^{IK \times J}$, with columns $\mathbf{c}_j = \mathbf{a}_j \otimes \mathbf{b}_j \in \mathbb{R}^{IK}$. The Khatri-Rao product is also called the column-wise Kronecker product.

2-2 Tensor diagrams

In order to be able to visualize tensors of higher dimensions, tensor diagrams were introduced by R. Penrose [14]. They were first used in the field of quantum mechanics but found their way into the field of multi-linear algebra. The diagrams consist out of nodes and edges. The nodes correspond to the core of the tensor and each edge coming from the node corresponds to a single dimension or mode. In figure 2-3 examples are given for up to the 4th order case.

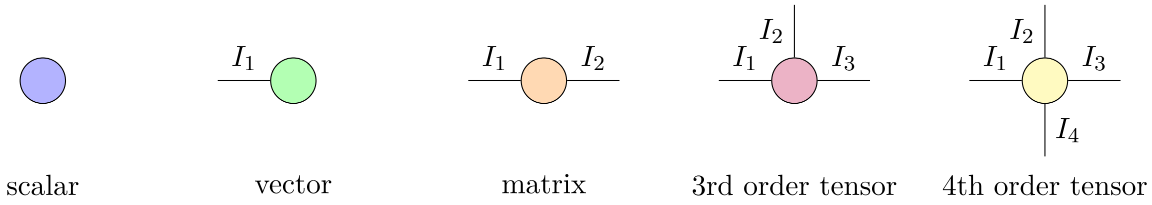
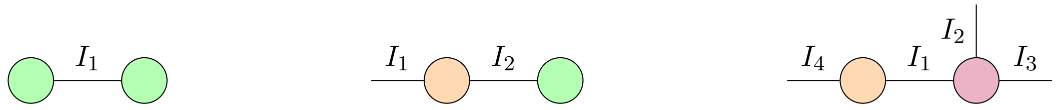


Figure 2-3: Tensor diagrams of a scalar, vector, matrix, 3rd order tensor and 4th order tensor.

It is now possible to visualize a 4th order tensor using a two dimensional diagram. Tensor diagrams are also capable of visualizing tensor products by connecting two (or more) cores with their edges. The interconnections between different cores represent contractions. In figure 2-4 some basic contractions are shown. The first contraction which can be seen in the figure is a simple vector dot product which can be denoted by $< \mathbf{a}, \mathbf{b} > = \sum_{i=1}^{I_1} \mathbf{a}_i \mathbf{b}_i$. The dot product is a contraction over the specified dimension I_1, yielding a scalar as outcome. The matrix-vector product can be denoted by: $\mathbf{A} \cdot \mathbf{b} = \sum_{j=1}^{I_2} \mathbf{A}_{(:,j)} \mathbf{b}_j$. The second subfigure is the corresponding graphical representation of the vector-matrix product, which results in a vector of dimension I_1. The last subfigure is the mode-1 product of a third order tensor and a matrix. It can be denoted by: $\underline{\mathbf{A}} \times_1 \mathbf{B} = \sum_{k=1}^{I_1} \underline{\mathbf{A}}_{(k,:,:)} \mathbf{B}_{(j,k)}$ and it yields a new third order tensor $\underline{\mathbf{C}}$, with $\underline{\mathbf{C}} \in \mathbb{R}^{I_4 \times I_2 \times I_3}$.



Vector dot product Vector-Matrix product mode-1 product with matrix

Figure 2-4: Various basic tensor contractions.

2-3 Tensor decompositions

The term *curse of dimensionality*, in tensor context, refers to the phenomenon where the number of elements of an Nth-order tensor grows exponentially with the tensor order, N. Processing such tensors can therefore require an enormous amount of computational and memory resources. The *curse of dimensionality* can be alleviated or even fully dealt with by using tensor network representations. The tensor network representations do come with a loss in accuracy due to the the necessity to involve various approximations [2]. In this section the Canonical Polyadic Decomposition (CPD) and the Tensor-Train Decomposition (TTD) are introduced and briefly discussed.

2-3-1 Cannonical polyadic decomposition

The CPD decomposes an Nth-order tensor, $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \dots I_N}$, into a linear combination of terms, ${\mathbf{b_r}}^{(1)} \circ {\mathbf{b_r}}^{(2)} \circ \cdots \circ {\mathbf{b_r}}^{(N)}$, which are rank-1 tensors and is denoted by:

$$\begin{aligned} \underline{\mathbf{X}} &\cong \sum_{r=1}^{R} \lambda_r \mathbf{b}_r^{(1)} \circ \mathbf{b}_r^{(2)} \circ \cdots \circ \mathbf{b}_r^{(N)} \\ &= \underline{\mathbf{\Lambda}} \times {}_1\mathbf{B}^{(1)} \times_2 \mathbf{B}^{(2)} \cdots \times {}_N\mathbf{B}^{(N)} \\ &= [\![\underline{\mathbf{\Lambda}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \dots, \mathbf{B}^{(N)}]\!] \end{aligned} \tag{2-1}$$

where $\mathbf{B}^{(n)} = [\mathbf{b}_1^{(n)}, \mathbf{b}_2^{(n)}, \dots, \mathbf{b}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$ and λ_r are non-zero entries of the diagonal core tensor $\underline{\mathbf{\Lambda}} \in \mathbb{R}^{R \times R \times \cdots \times R}$. In figure 2-5 the CPD of a 4th-order tensor displayed as a tensor diagram.

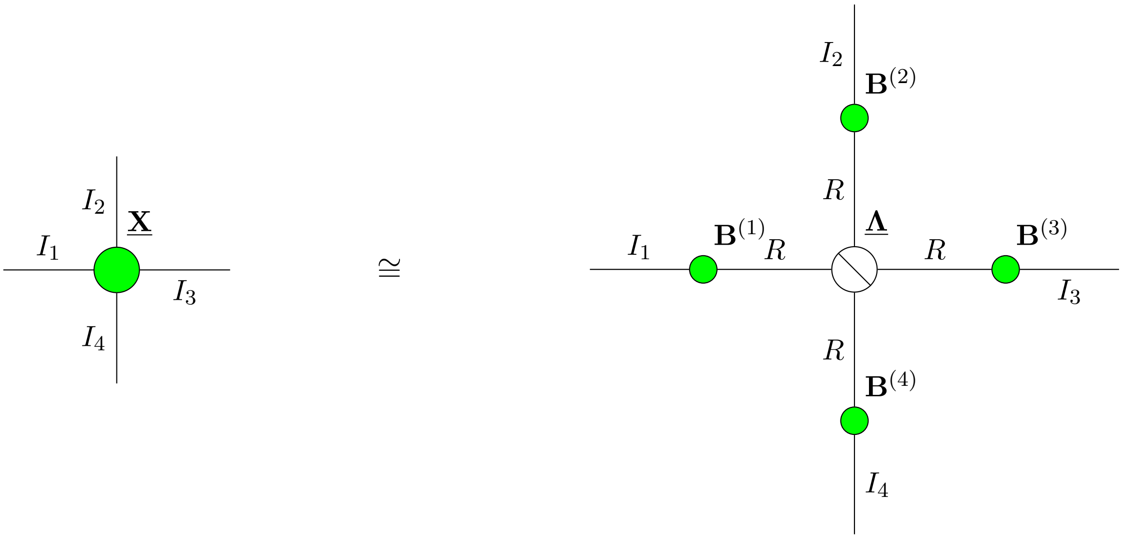


Figure 2-5: CPD decomposition of a 4th order tensor.

The rank of a tensor $\underline{\mathbf{X}}$ is defined as the smallest number of rank-one tensors from which their sum generates $\underline{\mathbf{X}}$ [15]. This is equal to the smallest number of components in an exact CPD. The tensors found in real world applications are often not exact since most real world tensors are corrupted by noise. Therefore the CPD has to be estimated using a cost-function of some form and the rank becomes more difficult to determine, since it is an NP-hard problem [15]. These cost-functions are mostly of the least-squares type, using the Frobenius norm:

$$J\left(\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}\right) = \left\| \underline{\mathbf{X}} - [\![\underline{\mathbf{\Lambda}}; \mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \ldots, \mathbf{B}^{(N)}]\!] \right\|_F^2. \tag{2-2}$$

An algorithm that uses these types of cost-functions is the alternating least-squares (ALS) algorithm, which is one of the most commonly used algorithms for computing the CPD. The ALS algorithm optimizes each factor matrix $\mathbf{B}^{(n)}$ individually by minimizing the cost function iteratively. The pseudo code for the ALS algorithm for a 3rd order tensor is given in algorithm 1.

Algorithm 1 Basic ALS for a 3rd-order tensor

```
Input: Tensor $\underline{\mathbf{X}}^{I \times J \times K}$ and rank R
Output: Factor matrices $\mathbf{B}^{(1)} \in \mathbb{R}^{I \times R}, \mathbf{B}^{(2)} \in \mathbb{R}^{J \times R}, \mathbf{B}^{(3)} \in \mathbb{R}^{K \times R}$

Initialization of $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \mathbf{B}^{(3)}$
while not converged or iteration limit not reached do
    $\mathbf{B}^{(1)} \leftarrow \underline{\mathbf{X}}_{(1)}(\mathbf{B}^{(3)} \odot \mathbf{B}^{(2)})(\mathbf{B}^{(3)\mathrm{T}}\mathbf{B}^{(3)} \circledast \mathbf{B}^{(2)\mathrm{T}}\mathbf{B}^{(2)})^{\dagger}$
    Normalize column vectors of $\mathbf{B}^{(1)}$
    $\mathbf{B}^{(2)} \leftarrow \underline{\mathbf{X}}_{(2)}(\mathbf{B}^{(3)} \odot \mathbf{B}^{(1)})(\mathbf{B}^{(3)\mathrm{T}}\mathbf{B}^{(3)} \circledast \mathbf{B}^{(1)\mathrm{T}}\mathbf{B}^{(1)})^{\dagger}$
    Normalize column vectors of $\mathbf{B}^{(2)}$
    $\mathbf{B}^{(3)} \leftarrow \underline{\mathbf{X}}_{(3)}(\mathbf{B}^{(2)} \odot \mathbf{B}^{(1)})(\mathbf{B}^{(2)\mathrm{T}}\mathbf{B}^{(2)} \circledast \mathbf{B}^{(3)\mathrm{T}}\mathbf{B}^{(3)})^{\dagger}$
    Normalize column vectors of $\mathbf{B}^{(3)}$, store the norms in vector $\lambda$
end while
return $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \mathbf{B}^{(3)}$ and $\lambda$
```

The computationally most intensive steps of the algorithm are multiplying a matricized tensor with a Khatri-Rao product of factor matrices and the computation of the pseudo-inverse of the R × R matrices. The algorithm is given for a 3-dimensional tensor case, but it can be easily extended for a higher dimensional tensor.

An interesting property of the CPD is the fact that it is generally unique under mild conditions. The individual components should be sufficiently different and their number must not be unreasonably large. This is unlike matrix decompositions, since they are often not unique [15]. The storage complexity of the original tensor is $\mathcal{O}(I^N)$ and the storage complexity of of the CPD is $\mathcal{O}(NIR)$. Hence for higher order tensors, the CPD has benefits regarding storage complexity. The CPD can be used for a wide variety of applications, one of them being electroencephalogram (EEG) data analysis, which will be discussed in more detail in chapter 4.

2-3-2 Tensor-Train decomposition

One of the most promising tensor decompositions is the TTD, which decomposes a tensor into a Tensor-Train (TT). The TT format has it's roots in quantum physics [16], but has gained more attention since a paper by I. V. Oseledets [17] introducing the current name of the decomposition. The TT-decomposition is a low-rank representation of a tensor which reduces the storage complexity of a tensor and computational complexity when doing operations. The mathematical form of the TT is given in definition 2-3.1.

Definition 2-3.1 (Index form of a TT [17])**.** The index form of the Tensor-Train Decomposition is written as

$$\underline{\mathbf{X}}\left(i_1, \ldots, i_d\right) = \sum_{\alpha_0, \ldots, \alpha_{d-1}, \alpha_d} \underline{\mathbf{X}}^{(1)}\left(\alpha_0, i_1, \alpha_1\right) \underline{\mathbf{X}}^{(2)}\left(\alpha_1, i_2, \alpha_2\right) \ldots \underline{\mathbf{X}}^{(d)}\left(\alpha_{d-1}, i_d, \alpha_d\right)$$

where $\underline{\mathbf{X}}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$ for $n = 1, 2, \ldots, d$ are the *TT-cores* which represent a single dimension of the original tensor. For simplicity the TT is often denoted by: $\underline{\mathbf{X}} = \langle\langle \underline{\mathbf{X}}^{(1)}, \underline{\mathbf{X}}^{(2)}, \ldots, \underline{\mathbf{X}}^{(d)} \rangle\rangle$.

The graphical representation of the TTD is displayed in figure 2-6. It can be seen that the graphical representation of the decomposition looks a bit like a train, which is the origin of the name for the TT.

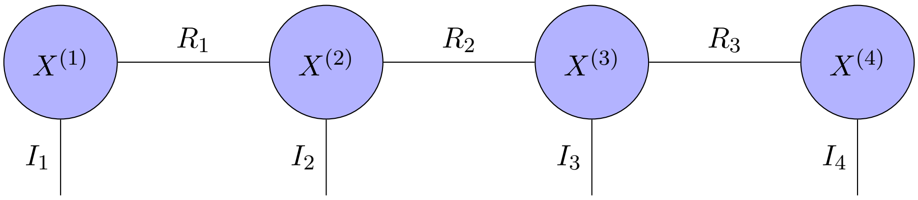


Figure 2-6: Graphical representation of the tensor-train decomposition of a 4th order tensor.

In order to convert the original tensor into the TT format the TT-SVD algorithm can be used. The TT-SVD algorithm takes as input a tensor $\underline{\mathbf{A}}$ and a prescribed accuracy ϵ and outputs the TT $\underline{\mathbf{B}}$. The algorithm will make sure that the following holds:

$$\frac{||\,\underline{\mathbf{A}} - \underline{\mathbf{B}}\,||_F}{||\,\underline{\mathbf{A}}\,||_F} \leq \varepsilon. \tag{2-3}$$

The TT-SVD algorithm can also be initiated with prescribed TT-ranks for each of the edges when the desired ranks are known. The complete TT-SVD is given in algorithm 2.

Algorithm 2 TT-SVD [17]

```
Input: Tensor $\underline{\mathbf{A}} \in \mathbb{R}^{I_1 \times I_2 \dots I_N}$
Output: Tensor-Train $\underline{\mathbf{B}}$ with cores $\mathbf{G_1}, \dots, \mathbf{G_d}$

Compute truncation parameter $\delta = \frac{\varepsilon}{\sqrt{d-1}} ||\,\underline{\mathbf{A}}\,||_F$
Temporary tensor $\underline{\mathbf{C}} \leftarrow \underline{\mathbf{A}}$, $r_0 \leftarrow 1$
for $k = 1$ to $N-1$ do
    C $\leftarrow$ reshape($C$, $[r_{k-1}n_k, \frac{\text{NUMEL(C)}}{r_{k-1}n_k}]$)
    Compute $\delta$-truncated SVD: $C = USV + E, ||E||_F \leq \delta, r_k = rank_\delta(C)$
    New core: $G_k \leftarrow$ reshape($U$, $[r_{k-1}, n_k, r_k]$)
    C $\leftarrow SV^\top$
    $G_d = C$
end for
return $\underline{\mathbf{B}}$ in TT-format with cores $\mathbf{G_1}, \dots, \mathbf{G_d}$
```

The computationally most expensive step of the algorithm is the first SVD which has a computational complexity of $(\mathcal{O}(I^{N+1}))$ [17]. The TT decomposition is not unique, since it is possible to multiply a node with a transformation matrix and the node right next to it with the inverse of the transformation matrix. If these transformation matrices are now absorbed into the node itself, a new TT decomposition is created which represents the same initial tensor. This property can be exploited in order to orthogonalize the TT with respect to a node. This can be done by using a recursive thin QR decomposition from the left of the target node and from the right of the target node. If the index of the node is k, the TT is now in site-k-mixed canonical form.

Definition 2-3.2 (site-k-mixed canonical form [2])**.** A tensor $\underline{\mathbf{X}}$ in TT format, denoted by $\langle\langle\underline{\mathbf{X}}^{(1)}, \underline{\mathbf{X}}^{(2)}, \ldots, \underline{\mathbf{X}}^{(N)}\rangle\rangle$ is in site-k-mixed canonical form, if

$$\begin{aligned}\left(\underline{\mathbf{X}}_L^{(m)}\right)^{\mathrm{T}} \underline{\mathbf{X}}_L^{(m)} &= \mathbf{I}_{R_m}, \quad m = 1, \ldots, k-1 \\ \underline{\mathbf{X}}_R^{(m)} \left(\underline{\mathbf{X}}_R^{(m)}\right)^{\mathrm{T}} &= \mathbf{I}_{R_{m-1}}, \quad m = k+1, \ldots, N.\end{aligned}$$

with $1 \leq k \leq N$.

A useful property of the TT is the fact that the ranks of the nodes can be upper bounded by using:

$$R_n \leq \min(I_1 \cdot_{\ldots} \cdot I_n, I_{n+1} \cdot_{\ldots} \cdot I_N). \tag{2-4}$$

Since the nodes in a TT can be permuted in any order, the upperbound on the rank of the nodes can change depending on the order of the nodes. This is something to take into account when constructing the TT. The original storage complexity of an Nth order tensor is reduced from $\mathcal{O}(I^N)$ to approximately $\mathcal{O}(NIR^2)$. The storage complexity is thus larger for the TT than for the CPD; however, it presents some advantages over the CPD. One of these advantages is the reliable construction of the TT since it depends only on regular singular value decompositions (SVDs). Another advantage is the distinct ranks between TT cores, making it a more flexible tool since multiple ranks for a tensor can be chosen. The downside of this flexibility is the extra amount of parameters that have to be chosen or has to be optimized for. This is especially true for higher order tensors, where there is an exponentially increasing number of rank combinations for a linearly increasing rank. Since, real world data is commonly low-rank [18], the number of combinations of ranks is relatively small.

Chapter 3

Support Tensor Machines

In this chapter a brief summary of the classic support vector machine (SVM) is given. Hereafter the first Support Tensor Machine (STM) and variations of the STM relevant for the research are introduced.

3-1 Support vector machines

The support vector machine is a supervised machine learning model that can be used for classification and regression. It was developed at AT&T Bell Laboratories by Vladimir Vapnik and his colleagues [19]. In this subsection the application of the SVM to binary classification problems will be discussed.

Given a dataset $\{x_i, y_i\}_{i=1}^n$ where $x_i \in \mathcal{X}$ represents the i-th input vector and where $y_i \in \{-1, +1\}$ represents the corresponding binary class labels, the classical SVM aims to find a classification hyperplane which maximizes the margin between the two classes. That is finding a projection vector $\mathbf{w}$ and a bias b. The decision boundary is usually specified by a small subsection of training samples, called the support vectors. Hence, the support vectors are the points which are the closest to the decision boundary.

3-1-1 Hard margin support vector machine

If the data is linearly separable it is possible to construct two parallel hyperplanes to separate the data. These hyperplanes can be described by the following equations:

$$\begin{cases} \mathbf{w}^\top \mathbf{x}_i + b \geq +1, & \text{if} \quad y_i = +1 \\ \mathbf{w}^\top \mathbf{x}_i + b \leq -1, & \text{if} \quad y_i = -1. \end{cases} \tag{3-1}$$

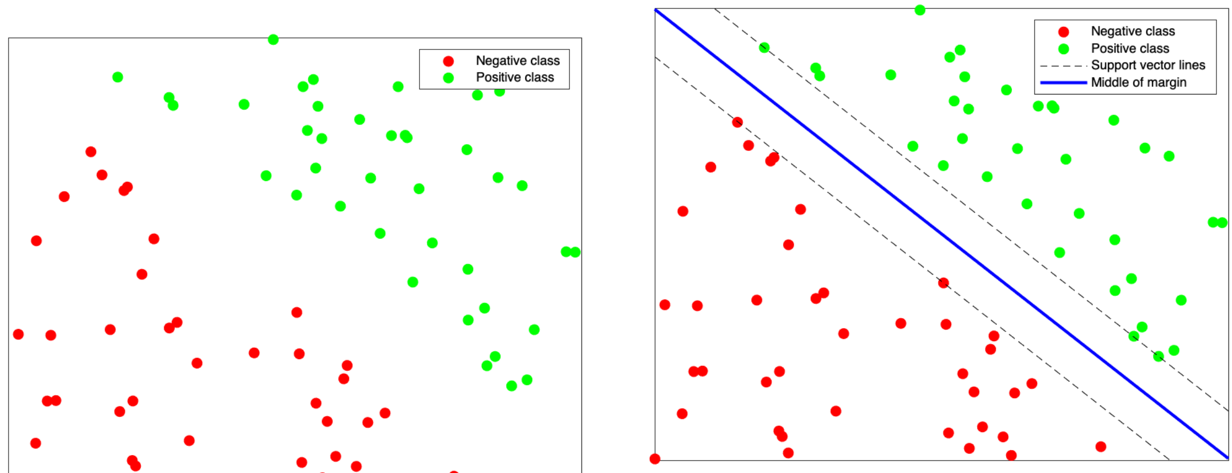
Which can be rewritten into a single equation:

$$y_i \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i + b \right) \geqslant 1, \text{for } \ i = 1, \ldots, N. \tag{3-2}$$

The distance between these two hyper planes is called the margin and is equal to $\frac{2}{\|w\|}$. In order to maximize the margin, $\|w\|$ has to be minimized. This can be formulated as an optimization problem:

$$\begin{aligned} &\min_{\mathbf{w},b} J(\mathbf{w}, b, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2, \\ &\text{s.t.} \quad y_i \left(\mathbf{w}^{\mathrm{T}} \mathbf{x}_i + b \right) \geqslant 1, \quad i = 1, \ldots, N. \end{aligned} \tag{3-3}$$

This optimization problem is called the primal problem and can be solved using quadratic programming algorithms. In most applications the dual problem is constructed using Lagrange multipliers, which will be discussed later in this chapter. When solving the optimization problem for randomly initiated separable samples figure 3-1b will be obtained.



(a) Linearly separable data.

(b) Separated data with maximum margin.

Figure 3-1: Example of a SVM applied to linearly separable data.

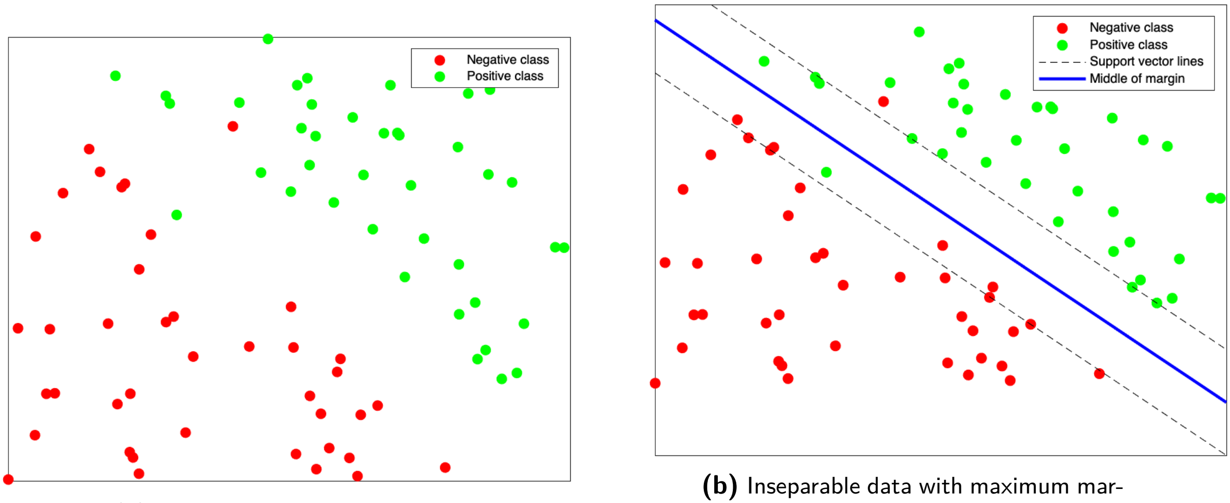
Where the red dots are the negative class samples, the green dots the positive class samples, the black striped lines indicate the boundaries of the optimized margin (support vector lines) and the blue line indicates the middle of the margin.

3-1-2 Soft margin support vector machine

Since real datasets are rarely linearly separable the hard margin optimization can not always be used. Furthermore, the hard margin optimization is sensitive to outliers rendering it unreliable for real world applications. The Soft margin optimization deals with these problems by introducing a slack variable ξ_i. The optimization problem can now be written as follows:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\xi}} J(\mathbf{w},b,\boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{M}\xi_i, \\ &\text{s.t.} \quad y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) \geqslant 1, \quad \xi_i \geqslant 0, \quad i = 1,\dots,N. \end{aligned} \tag{3-4}$$

Where $\boldsymbol{\xi} = [\xi_1, \xi_2, \dots, \xi_N]^T$ is the vector of all slack variables needed to deal with the linearly inseparable problem. Whenever the classification problem is linearly separable, $\boldsymbol{\xi}$ can be set to 0, the hard margin optimization problem is then obtained. In order to classify new incoming measurements the decision function $y(x_i) = sign(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)$ decides to which class, {+1, -1}, it belongs. This problem is however not well defined without the addition of regularization term C. Without it, it would be possible to make ξ_i arbitrarily large, which would allow any $\mathbf{w}$ to satisfy the constraints. C penalizes large values of ξ_i which reduces overfitting. The choice for the value of C is a design decision and depends on the the amount of error which is allowable in the classifiers design. It is now possible to deal with linearly inseparable data. An example is shown in figure 3-2b.



(a) Linearly inseparable data.

(b) Inseparable data with maximum margin.

Figure 3-2: Example of a SVM applied to linearly inseparable data.

Even though the data is not linearly separable, the optimization problem is still able to find the best support vectors when giving some space for errors. The figure will look different depending on the given value for the regularization parameter C.

3-1-3 Least-squares support vector machine

The soft-margin SVM can be rewritten into the least-squares support vector machine (LS-SVM) [20] as follows:

$$\begin{aligned} &\min_{\mathbf{w},b,\boldsymbol{\varepsilon}} J(\mathbf{w},b,\boldsymbol{\varepsilon}) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\gamma}{2}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon}, \\ &\text{s.t.} \quad y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b\right) = 1 - \varepsilon_i, \quad i = 1,\dots,N. \end{aligned} \tag{3-5}$$

This new formulation allows to find the solution by solving a set of linear equations instead of a convex quadratic programming (QP) problem used for solving the classic SVMs. This is now possible because the inequality constraints are now equality constraints and the loss is replaced by a square loss.

3-1-4 Dual problem & kernel trick

It is common practice to solve the dual problem of the optimization problem, especially when the feature size n is larger than the sample size M. The dual problem is constructed by using Lagrange multipliers $\alpha_i \geq 0$ and a Lagrangian:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{M} \alpha_i \left(y_i \left(\langle \mathbf{x}_i, \mathbf{w} \rangle + b\right) - 1\right). \tag{3-6}$$

The Lagrangian L has to be minimized with respect to the primal variables $\mathbf{w}$ and b and maximized with respect to the dual variables α_i. This means that a saddle point has to be found. In order to find this saddle point the derivatives of L with respect to the primal variables must be zero:

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0 \text{ and } \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0. \tag{3-7}$$

These equations will yield the following expressions:

$$\sum_{i=1}^{M} \alpha_i y_i = 0 \text{ and } \mathbf{w} = \sum_{i=1}^{M} \alpha_i y_i \mathbf{x}_i. \tag{3-8}$$

If the expressions in equation 3-8 are substituted in the Lagrangian of equation 3-6, the dual optimization problem is obtained:

$$\begin{aligned} &\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} W(\boldsymbol{\alpha}) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ &\text{s.t. } 0 \leq \alpha_i \leq C \text{ for all } i = 1, \ldots, M \text{ and } \sum_{i=1}^{M} \alpha_i y_i = 0. \end{aligned} \tag{3-9}$$

The dual problem can be solved using quadratic programming algorithms, sub-gradient descent and coordinate descent techniques. The most widely used technique for solving the QP problem in the dual is Sequential Minimal Optimization (SMO). SMO was invented by J. Platt in 1998 [21] and quickly replaced more complex QP solvers. SMO is an iterative algorithm that breaks the original optimization problem down into a series of sub-problems. The smallest possible problem involves two Lagrange multipliers and can be denoted by:

$$0 \leq \alpha_1, \alpha_2 \leq C, \tag{3-10}$$

$$y_1\alpha_1 + y_2\alpha_2 = k, \tag{3-11}$$

and this problem can be solved analytically. When all the Lagrange multipliers satisfy the Karush-Kuhn-Tucker (KKT) conditions, the complete problem is solved. The complete algorithm can be found in the original paper of J. Platt [21]. SMO is used extensively in one of the most popular SVM libraries called LIBSVM, which is a library used in this thesis.

When the expression for $\mathbf{w}$ of equation 3-8 is substituted into the hyperplane decision function, it can now be written as:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{M} y_i\alpha_i \langle \mathbf{x}, \mathbf{x}_i\rangle + b\right). \tag{3-12}$$

Using the kernel trick the linear support vector machine can be extended to a kernelized support vector machine which has better capabilities with respect to non-linear classification. The kernel trick replaces the dot product with a non-linear kernel function, which allows the margin hyperplane to be constructed in a transformed feature space. The main advantage of using kernel methods is the fact that the dot product can be computed in transformed feature space without knowing the exact mapping function. The kernel can be denoted by:

$$k\left(x_i, x_j\right) = \langle \boldsymbol{\phi}(x_i), \boldsymbol{\phi}\left(x_j\right)\rangle. \tag{3-13}$$

Replacing the dot product in the dual optimization problem will yield the following optimization problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}\in\mathbb{R}^m} \; W(\boldsymbol{\alpha}) &= \sum_{i=1}^{M} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{M} \alpha_i\alpha_j y_i y_j \left(k(\mathbf{x}_i, \mathbf{x}_j)\right) \\ \text{s.t. } 0 \leq \alpha_i \leq C &\text{ for all } i = 1, \ldots, M \text{ and } \sum_{i=1}^{M} \alpha_i y_i = 0. \end{aligned} \tag{3-14}$$

The hyperplane decision function is now equal to:

$$f(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{M} y_i\alpha_i \left(k(\mathbf{x}_i, \mathbf{x}_j) + b\right)\right). \tag{3-15}$$

Some commonly used kernels include:

- Homogeneous polynomial kernel: $k(\mathbf{x}_i \cdot \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d$. For $d = 1$ the kernel becomes the linear kernel.
- Inhomogeneous polynomial kernel: $k(\mathbf{x}_i \cdot \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + r)^d$.
- Gaussian radial basis function kernel: $k(\mathbf{x}_i \cdot \mathbf{x}_j) = exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||^2)$ for $\gamma > 0$.
- Sigmoid function kernel: $k(\mathbf{x}_i \cdot \mathbf{x}_j) = tanh(\kappa\mathbf{x}_i \cdot \mathbf{x}_j + c)$ for $\kappa > 0$ and $c < 0$.

3-2 Support Tensor Machines

The conventional SVM model operates within a vector space framework and lacks inherent capability to process non-vector structures. Conversely, real-world image and video data find more intuitive representation as matrices (second-order tensors) or tensors of higher order. As a result, there's an increasing focus on creating better learning machines for classifying tensors. In this section first the original STM is introduced and subsequently the more advanced STM techniques that have been used in this thesis will be presented.

3-2-1 The first support tensor machine

In order to avoid vectorizing multi-dimensional data and thus retaining the information contained in the structure of a tensor, Tao et al. extended the classical SVM to the the first STM for classification of higher order tensors [22]. For classification purposes the STM can be formulated using the following minimization problem:

$$\begin{aligned} &\min_{\mathbf{w}_n, b, \boldsymbol{\xi}} J\left(\mathbf{w}_n, b, \boldsymbol{\xi}\right) = \frac{1}{2}\left\|\otimes_{n=1}^{N}\mathbf{w}_n\right\|^2 + C\sum_{m=1}^{M}\xi_m \\ \text{s.t. } & y_m\left(\underline{\mathbf{X}}_m \bar{\times}_1 \mathbf{w}_1 \cdots \bar{\times}_N \mathbf{w}_N + b\right) \geqslant 1 - \xi_m, \quad \boldsymbol{\xi} \geqslant 0, \\ & m = 1, \ldots, M. \end{aligned} \tag{3-16}$$

Where $\boldsymbol{\xi}$ is the vector of all slack variables e.g. $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_M]^T \in \mathbb{R}^M$. The STM consists out of N quadratic programming sub problems, which can be denoted by:

$$\begin{aligned} &\min_{\mathbf{w}_n, b, \boldsymbol{\xi}} \frac{1}{2}\left\|\mathbf{w}_n\right\|^2 \prod_{1 \leqslant i \leqslant N}^{i \neq n}\left(\left\|\mathbf{w}_i\right\|^2\right) + C\sum_{m=1}^{M}\xi_m \\ \text{s.t.} \quad & y_m\left(\mathbf{w}_n^{\mathrm{T}}\left(\underline{\mathbf{X}}_m \bar{\times}_{i \neq n} \mathbf{w}_i\right) + b\right) \geqslant 1 - \xi_m, \quad \boldsymbol{\xi} \geqslant 0, \\ & m = 1, \ldots, M. \end{aligned} \tag{3-17}$$

These N optimization problems do not have a closed-form solution and hence the problem is no longer convex. In order to solve the STM an alternating projection algorithm has to be used. This iterative technique, however, is very time consuming. The alternating projection algorithm for STM fixes all $\mathbf{w}$ except one and solves optimization problem 3-17. Whenever the following condition holds:

$$\sum_{n=1}^{N}\left(\left|\left(\mathbf{w}_t^{(n)}\right)^{\mathrm{T}}\mathbf{w}_{t-1}^{(n)}\right|\left\|\mathbf{w}_t^{(n)}\right\|_F^{-2} - 1\right) < \varepsilon, \tag{3-18}$$

where ε is a predefined threshold, a solution has been found.

The computational complexity of this STM is $\mathcal{O}(M^2 N T \prod_{n=1}^{N} I_n)$ where M is the total number of tensor samples, N the order of tensor $\underline{\mathbf{X}}_m \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, I_n is the size of dimension n and T is the loop number. The storage complexity of STM is $\mathcal{O}(M \prod_{n=1}^{N} + \sum_{n=1}^{N} I_n + 1)$ [23]. The storage and computational complexity are thus higher for STM when comparing it to SVM.

Z. Hao et al. [23] compared the performance of the STM to the classcial SVM on 12 experimental datasets. 9 of them are second-order face recognition datasets and 3 of them are third-order gait recognition datasets. In terms of test accuracy, STM only had better performance than SVM on one of the datasets. This was the case for one of the three third-order datasets, STM failed to converge for the remaining two. The training time of SVM is less on all 12 datasets compared to STM, which was expected when comparing the computational complexity of the methods. Another disadvantage of STM is the fact that $\underline{\mathbf{W}}$ is a rank-1 tensor which has a significant impact on the expressive power of the STM, resulting in an usually unsatisfactory classification accuracy for many real-world data problems [24].

3-2-2 Support Higher-order Tensor Machines

In order to overcome some of the shortcomings of the initial STM, Hao et al. introduced a novel linear Support Higher-order Tensor Machine (SHTM) which combines the merits of C-SVM and tensor rank-one decomposition (Canonical Polyadic (CP) decomposition) [23]. SHTM uses more compact R rank-one tensors instead of the original tensor, saving storage space and computational time. The SHTM optimization problem 3-19 can be solved by using a sequential minimization algorithm instead of an alternating projection algorithm, hence making it faster to compute. The resulting optimization problem of the SHTM method is displayed in equation 3-19.

$$\begin{aligned}
\max_{\boldsymbol{\alpha}} \quad & \sum_{m=1}^{M} \alpha_m - \frac{1}{2} \sum_{i,j=1}^{M} \sum_{p=1}^{R} \sum_{q=1}^{R} \alpha_i \alpha_j y_i y_j \prod_{n=1}^{N} \left\langle \mathbf{x}_{ip}^{(n)}, \mathbf{x}_{jq}^{(n)} \right\rangle \\
\text{s.t.} \quad & \sum_{m=1}^{M} \alpha_m y_m = 0, \\
& 0 \leq \alpha_m \leq C, \quad m = 1, \cdots, M
\end{aligned} \tag{3-19}$$

The decision hyperplane can now be denoted by:

$$y(X) = \text{sign}\left(\sum_{m=1}^{M} \sum_{p=1}^{R} \sum_{q=1}^{R} \alpha_m y_m \prod_{n=1}^{N} \left\langle \mathbf{x}_{mp}^{(n)}, \mathbf{x}_{q}^{(n)} \right\rangle + b \right). \tag{3-20}$$

The computational complexity of SHTM is $\mathcal{O}(M^2 R^2 \sum_{n=1}^{N} I_n)$, which is more efficient than SVM and STM. The storage complexity of SHTM is $\mathcal{O}((M+1)R \sum_{n=1}^{N} I_n + 1)$, which is also less than SVM and STM.

The performance of SHTM was evaluated on 9 second-order face recognition datasets and 3 third-order gait recognition datasets. The performance evaluation showed that the SHTM is more effective and efficient for tensor classification than C-SVM and STM. This superiority became more dominant for the third-order tensor datasets. The test accuracy of SHTM had an increase of up to 5% compared to SVM and a speed increment of up to 160× compared to SVM.

3-2-3 Dual Structure-preserving Kernels

He et al. proposed an approach to supervised tensor learning, called Dual Structure-preserving Kernel (DuSK) [25]. Using CP factorization a structure-preserving kernel is learned in the tensor product feature space. From now on, when DuSK is mentioned, it refers to the entire classification method and not just the kernel for convenience.

The paper states that it is possible to factorize tensor data directly in the feature space as if in the original space. This is mathematically defined as:

$$\phi : \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \mathbf{x}_r^{(n)} \rightarrow \sum_{r=1}^{R} \prod_{n=1}^{N} \otimes \phi\left(\mathbf{x}_r^{(n)}\right). \tag{3-21}$$

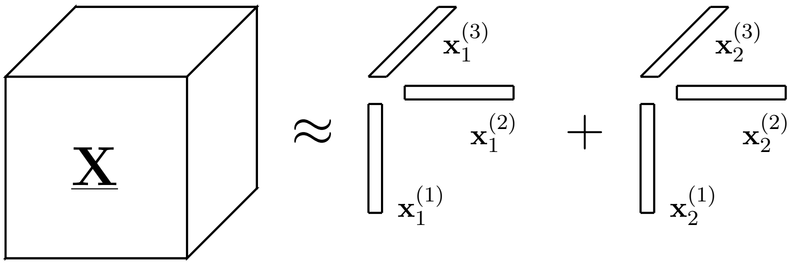
This can be interpreted as mapping the original data into tensor feature space and applying CP decomposition in the feature space itself. The DuSK kernel can now be denoted by:

$$\begin{aligned} &\kappa\left(\textstyle\sum_{r=1}^{R}\prod_{n=1}^{N}\otimes\mathbf{x}_r^{(n)}, \sum_{r=1}^{R}\prod_{n=1}^{N}\otimes\mathbf{y}_r^{(n)}\right) \\ &\quad = \textstyle\sum_{i=1}^{R}\sum_{j=1}^{R}\prod_{n=1}^{N}\kappa\left(\mathbf{x}_i^{(n)}, \mathbf{y}_j^{(n)}\right). \end{aligned} \tag{3-22}$$

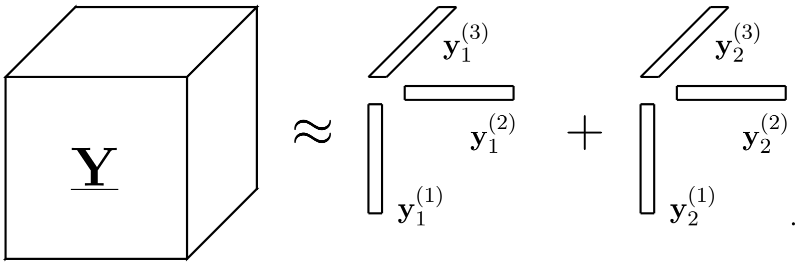
Since the paper uses the RBF kernel, equation 3-22 can be rewritten by substituting the kernel:

$$\begin{aligned} \kappa(\underline{\mathbf{X}}, \underline{\mathbf{Y}}) &= \sum_{i=1}^{R}\sum_{j=1}^{R}\prod_{n=1}^{N}\kappa\left(\mathbf{x}_i^{(n)}, \mathbf{y}_j^{(n)}\right) \\ &= \sum_{i=1}^{R}\sum_{j=1}^{R}\exp\left(-\sigma\sum_{n=1}^{N}\left\|\mathbf{x}_i^{(n)} - \mathbf{y}_j^{(n)}\right\|^2\right). \end{aligned} \tag{3-23}$$

This kernel is called the DuSK$_{RBF}$ kernel. This kernel is also used in the current state-of-the-art STM introduced in the next subsection. Given the importance of this kernel, it is necessary to conduct a more thorough examination of its characteristics. Therefore, a single entry of a kernel is computed symbolically. First, two third order samples are selected and their rank-2 CP-decomposition is computed. The ranks have intentionally been chosen low in order to maintain clarity and simplicity within the example. The resulting CP-decomposition of $\underline{\mathbf{X}}$ is:



and the resulting CP-decomposition of $\underline{\mathbf{Y}}$ is:



Using these samples the kernel from equation 3-23 is constructed:

$$\begin{aligned}
&\sum_{i=1}^{2}\sum_{j=1}^{2}\exp\left(-\sigma\sum_{n=1}^{3}\left\|\mathbf{x}_i^{(n)}-\mathbf{y}_j^{(n)}\right\|^2\right)\\
&=\exp\left(-\sigma\left\|\mathbf{x}_1^{(1)}-\mathbf{y}_1^{(1)}\right\|^2+\left\|\mathbf{x}_1^{(2)}-\mathbf{y}_1^{(2)}\right\|^2+\left\|\mathbf{x}_1^{(3)}-\mathbf{y}_1^{(3)}\right\|^2\right)\\
&+\exp\left(-\sigma\left\|\mathbf{x}_1^{(1)}-\mathbf{y}_2^{(1)}\right\|^2+\left\|\mathbf{x}_1^{(2)}-\mathbf{y}_2^{(2)}\right\|^2+\left\|\mathbf{x}_1^{(3)}-\mathbf{y}_2^{(3)}\right\|^2\right)\\
&+\exp\left(-\sigma\left\|\mathbf{x}_2^{(1)}-\mathbf{y}_1^{(1)}\right\|^2+\left\|\mathbf{x}_2^{(2)}-\mathbf{y}_1^{(2)}\right\|^2+\left\|\mathbf{x}_2^{(3)}-\mathbf{y}_1^{(3)}\right\|^2\right)\\
&+\exp\left(-\sigma\left\|\mathbf{x}_2^{(1)}-\mathbf{y}_2^{(1)}\right\|^2+\left\|\mathbf{x}_2^{(2)}-\mathbf{y}_2^{(2)}\right\|^2+\left\|\mathbf{x}_2^{(3)}-\mathbf{y}_2^{(3)}\right\|^2\right).
\end{aligned}\tag{3-24}$$

From the example it becomes clear that for every dimension the squared euclidean norm is computed for all possible combinations of CP-factors. This means that the number of exponentials that have to be computed and summed is equal to the CP-rank squared (R^2). When increasing the order of the tensor (and therefore the order of the CP-decomposition), the number of terms inside the exponentials scales linearly.

In the original paper [25] the performance of DuSK$_{RBF}$ was evaluated on three real-world functional magnetic resonance imaging (fMRI) datasets and compared to seven state-of-the-art SVM methods, namely: non-linear SVM with Gaussian-RBF kernel, Factor kernel, K_{3rd} kernel, linear SHTM, linear SVM, principle component analysis (PCA) + SVM and multilinear principal component analysis (MPCA) + SVM. The whole dataset is randomly sampled to split the data in 80% training and 20% test set. This process was repeated 50 times for all the methods and the reported accuracy is the mean of the performance on all 50 training/test splits. The method that outperforms all other methods on all the datasets is DuSK$_{RBF}$, with a maximum performance gain of 20% over competing methods.

The time complexity of computing a Gaussian RBF kernel matrix for vector applications is $\mathcal{O}(M^2\prod_{n=1}^{N}I_n)$. The time complexity of computing the kernel with the DuSK method is $\mathcal{O}(M^2R^2\sum_{n=1}^{N}I_n)$. Taking into account that typical tensor data is high dimensional while R is often small, it can be concluded that DuSK is computationally more efficient. Furthermore, the storage complexity is reduced from $\mathcal{O}(M\prod_{n=1}^{N}I_n)$ to $\mathcal{O}(M\sum_{n=1}^{N}I_n)$.

He et al. [26] extended DuSK to the kernelized CP (KCP) input. This method is called the Multi-way Multi-level Kernel (MMK) method. Kour et al. [27] extended the DuSK approach to the Tensor-Train (TT)-decomposition with enforced uniqueness and norm distribution. This method is called TT-MMK and has superior performance on most of the datasets used in the paper, with respect to the standard SVM, support tucker machine (STuM), MMK, K-STTM-prod and K-STTM-sum. Neither of the papers report the computational complexity of the methods, Kour et al. does however report the running time for a grid search on the ADNI dataset for both methods. MMK took approximately 17 minutes to complete and TT-MMK took approximately 3.5 hours to complete. Tensor Train Multi-way Multi-level Kernel (TT-MMK) will be discussed in the next section.

3-2-4 Tensor Train Multi-way Multi-level Kernel

Using the TT decomposition in a STM was first proposed by chen et al. [24]. This method is called the Support Tensor-Train Machine (STTM) and is a linear method. Later on chen et al. extended the STTM to the kernelized support tensor train machine (K-STTM) [18] which is a kernelized version of the STTM. TT-MMK is one of the newest proposed STM methods which uses a TT decomposition. From the benchmarks TT-MMK seems to have superior performance over all other kernelized STMs [27], which is why this novel approach is used in the thesis.

The TT-MMK method first decomposes the data into TTs using a newly proposed algorithm called the "Uniqueness Enforcing TT-SVD". The newly proposed algorithm fixes the signs of the singular vectors in the regular TT-SVD, in order to achieve a more stable learning method. The paper also shows that when the uniqueness Enforcing TT-SVD is used instead of the original TT-SVD algorithm, the performance is increased. The pseudo code for the uniqueness enforcing TT-SVD is given in algorithm 3.

Algorithm 3 Uniqueness Enforcing TT-SVD [27]

```
Input: $M$-dimensional tensor Tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 \dots I_M}$, relative error threshold $\epsilon$.
Output: Cores $\underline{\mathbf{G}}^{(1)}, \underline{\mathbf{G}}^{(2)}, \dots, \underline{\mathbf{G}}^{(M)}$ of the TT-approximation $\underline{\mathbf{X}}'$

Compute truncation parameter $\delta = \frac{\varepsilon}{\sqrt{M-1}} || \underline{\mathbf{X}} ||_F$
Initialize $\hat{\mathbf{Z}}_1 = \underline{\mathbf{X}}_{(1)}$, $R_0 \leftarrow 1$
for $m = 1$ to $M-1$ do
    $\mathbf{Z}_m \leftarrow \text{reshape}(\hat{\mathbf{Z}}_m, [R_{m-1} I_m, I_{m+1} \dots I_M])$
    Compute $\delta$-truncated SVD: $\mathbf{Z}_m = \mathbf{U}_m \mathbf{S}_m \mathbf{V}_m + \mathbf{E}_m, ||\mathbf{E}_m||_F \leq \delta$, where $\mathbf{U}_m$
    $= [u_1^{(m)}, u_2^{(m)}, \dots, u_{R_m}^{(m)}]$, $\mathbf{S}_m = [\theta_1^{(m)}, \theta_2^{(m)}, \dots, \theta_{R_m}^{(m)}]$, $\mathbf{V}_m = [v_1^{(m)}, v_2^{(m)}, \dots, v_{R_m}^{(m)}]$
    for $r_m = 1$ to $R_m$ do
        $i^*_{r_m} = \arg\max_{i=1,\dots,R_{m-1}I_m} |u^{(m)}_{i,r_m}|$ (with ties broken to first element)
        $\bar{u}^{(m)}_{r_m} := u^{(m)}_{r_m} / \text{sign}(u^{(m)}_{i^*_{r_m}, r_m})$,    $\bar{v}^{(m)}_{r_m} := v^{(m)}_{r_m} / \text{sign}(u^{(m)}_{i^*_{r_m}, r_m})$
        $\underline{\mathbf{G}}^{(m)}_{r_{m-1}, i_m, r_m} = u^{(m)}_{r_{m-1}+(I_m-1)R_{m-1}, r_m}, \bar{\mathbf{V}} = [\bar{v}_1^{(m)}, \bar{v}_2^{(m)}, \dots, \bar{v}_{R_m}^{(m)}]$
    end for
    $\hat{\mathbf{Z}}_{m+1} := \mathbf{S}_m \bar{V}_m^T$
end for
$\underline{\mathbf{G}}^{(M)} = \hat{\mathbf{Z}}_M$
```

After the TT decomposition, a TT-CP expansion is used in order to to gain a reliable CP-decomposition. This TT-CP expansion is given in equation 3-25 .

$$\sum_{r_0,\dots,r_M} \underline{\mathbf{G}}^{(1)}_{r_0,i_1,r_1}\underline{\mathbf{G}}^{(2)}_{r_1,i_2,r_2}\cdots\underline{\mathbf{G}}^{(M)}_{r_{M-1},i_M,r_M} = \sum_{r=1}^{R} \hat{\mathbf{H}}^{(1)}_{i_1,r}\hat{\mathbf{H}}^{(2)}_{i_2,r}\cdots\hat{\mathbf{H}}^{(M)}_{i_M,r} \tag{3-25}$$

where $r = r_1 + (r_2 - 1)R_1 + \cdots + (r_M - 1)\prod_{l=1}^{M-1}$. It can be seen that for a TT decomposition of a third order tensor of rank R, the CP expansion will have rank R^2. The computational complexity will go up quadratically for increasing rank. For higher order tensors the computational complexity will increase even more drastically, rendering the method infeasible for tensors of order 5 or more. Since many natural occurring tensor data is of order 4 at most, the method is applicable in most cases.

Hereafter, the TT-CP expansion is rescaled to ensure that the columns of the CP factors have equal norms. This is crucial since the DuSK kernel uses the same width parameters for all CP factors. Hence, this requires all CP factors to have the same order of magnitude. This norm equilibration is given in equation 3-26.

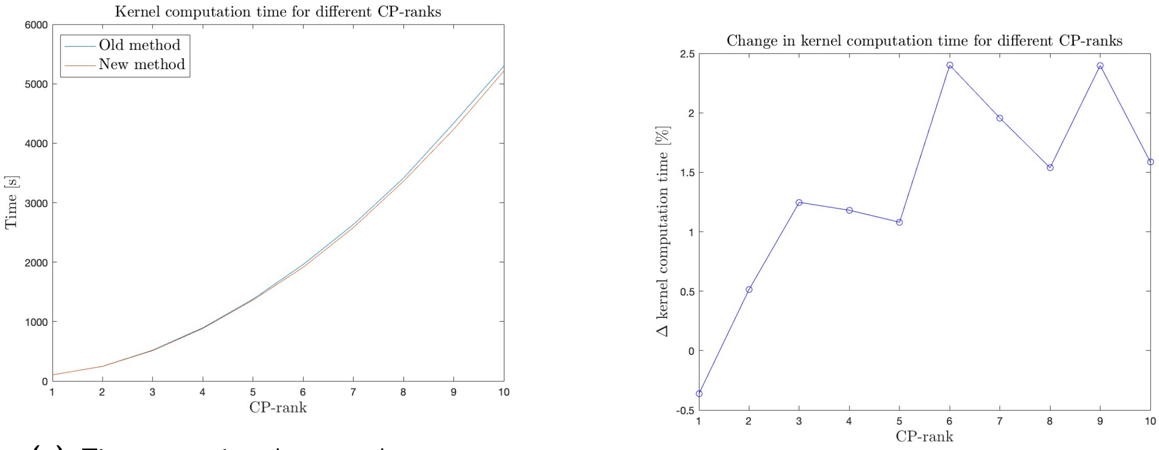
$$H_r^{(m)} := \frac{\hat{H}_r^{(m)}}{\left\|\hat{H}_r^{(m)}\right\|}\cdot n_r^{1/M}, \quad m = 1, 2, \cdots, M. \tag{3-26}$$

where

$$n_r = \left\|\hat{H}_r^{(1)}\right\|\cdots\left\|\hat{H}_r^{(M)}\right\|$$

is the total norm of each of the rank-1 tensors. After the equilibration of the TT-CP expansion, the DuSK$_{\mathrm{RBF}}$ kernel is used in order to train a model.

The code for both DuSK and TT-MMK is publicly available on Github. Since both DuSK and TT-MMK make use of the DuSK$_{\mathrm{RBF}}$ kernel, it is worth looking into the way it is implemented for both methods. It was found that DuSK and TT-MMK use the same code for the computation of the DuSK$_{\mathrm{RBF}}$ kernel. The code used for the construction of the kernel is improved slightly as can be seen in figure 3-3. In the experiment kernels are computed for 4000 third order samples and varying rank.



(a) Time comparison between the two implementations.

(b) Relative speed increase.

Figure 3-3: Comparison of DuSK$_{\mathrm{RBF}}$ kernel construction time for two implementations.

The modifications resulted in a maximum average speed increase for CP-ranks 6-10 of approximately **2%**. This increase in speed might seem insignificant but, due to the large amount of kernels that have to be computed for all the experiments conducted in chapter 5, it adds up to minutes of saved time per experiment. It is expected that for tensors of higher order (>3) and higher ranks (>10), the relative speed increase will be more significant. Slightly increasing the efficiency of an implementation could be crucial when eventually running the algorithm on embedded systems.

Chapter 4

Experiment setup

In this chapter the general classification pipeline for all the experiments will be discussed. First the dataset and the sampling strategy are presented. Secondly, the pre-processing steps that have been carried out are presented. Finally the general outline is given for all experiments and experiment specific flowcharts of the pipelines will be presented.

4-1 Dataset & sampling

The dataset that is used in this thesis is the CHB-MIT scalp dataset from Massachusetts Institute of Technology [28]. This dataset is one of the most widely used electroencephalogram (EEG) datasets in the literature. The CHB-MIT dataset contains 844 hours of continuous scalp EEG recordings with a total of 163 detected seizures from 23 patients. The dataset is recorded using the standardized 10-20 electrode positions using a sampling rate of 256 Hz. This dataset is heavily imbalanced due to ictal data being more rare than interictal data. In table 4-1 information on the subset of data that will be used in the experiments is presented.

patient	№ channels	№ seizures	Total seizure time [s]	Time interictal [h]	interictal/ictal ratio
chb01	23	7	442	40.55	±330
chb03	23	7	402	38.00	±340
chb08	23	5	919	20.1	±78

Table 4-1: Information on patient data included in the research.

In the table, it is evident that the data characteristics vary per patient. In order to process the large amount of interictal data, significant computational power is required. To be able to conduct experiments on a regular computer, a subset of the interictal data is used. This subset of interictal data is evenly sampled from all the recordings of a patients. This is done in order to get a representative distribution of the data over the day. The sampling is carried out by sliding a window of a certain length along the time axis.

The positive/ictal instances are sampled with an overlap in order to get as much positive instances from a single seizure. This is done using the same sliding window method as the interictal sampling with the same length. The sample length used in the experiments is 6 seconds with a 1 second overlap for the ictal class. The interictal class is samples are also 6 seconds long but without overlap. For all experiments the maximal amount of samples for the ictal class are used, hence the limiting factor determining the training and test set size is the maximum amount of ictal samples which can be sampled from a patient and the desired ratio between positive and negative samples. In order for the performance metrics to be representative for real-life applications, the test set needs to be imbalanced. For comparing the proposed methods a sampling ratio of 1/10 ictal/interictal is used and for comparing the proposed methods to a state-of-the-art seizure detector the interictal/ictal ratio from table 4-1 is used. This ratio is sampled from the results of the 1/10 ratio experiments.

4-2 Pre-processing

The first pre-processing step is filtering the data with a bandpass filter from 0.1 to 50 Hz. This is done in order to only retain the frequencies relevant for seizure detection. Hereafter the data is normalized using Z-score normalization, for which the formula is displayed in equation 4-1.

$$z = \frac{x - \mu}{\sigma} \tag{4-1}$$

Where x is a single entry, μ is the mean and σ is the standard deviation of the sample. By transforming the features to be on a similar scale, the performance and stability of the model should improve.

For a subset of experiments the Continuous Wavelet Transform (CWT) is applied to the samples as a pre-processing step. The CWT transforms the time-domain into time-frequency domain by using shifted and scaled forms of a mother wavelet function. Due to Heisenberg's uncertainty principle it is not possible to have high frequency and time resolution at a single point. By carefully selecting the mother wavelet function it is possible to choose the regions where high time and where high frequency resolution is desired. This makes the CWT a powerful tool for signal analysis. The mathematical formula of the CWT is given in equation 4-2.

$$X_w(a,b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t)\bar{\psi}\left(\frac{t-b}{a}\right) dt \tag{4-2}$$

Where $\bar{\psi}$ is the continuous mother wavelet, a is the frequency scale and b is the translation parameter. The CWT is computed using the **cwtft** function in MATLAB. **cwtft** function uses the Fast Fourier Transform (FFT) in order to compute the CWT.

Multi-channel EEG data is originally obtained as a second-order tensor (matrix) with dimensions *time* × *electrodes*. When applying a CWT the data will be tensorized, which results in a third-order tensor with dimensions *time* × *scales* × *electrodes*. The resulting tensor is displayed in figure 4-1 .

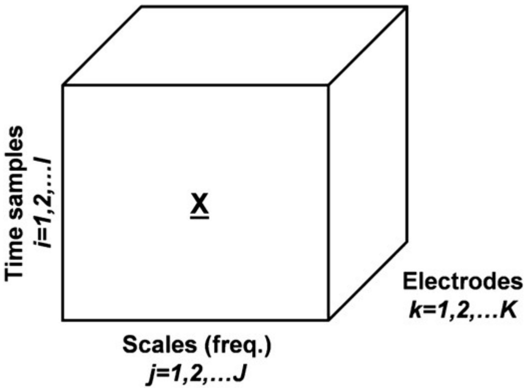


Figure 4-1: Resulting EEG tensor after wavelet transform [3].

There exist various different types of wavelet functions, but Latka et al. [29] showed that a Mexican-hat wavelet function captures epileptic events well. The tensor representation of EEG together with Canonical Polyadic (CP) decomposition and Tucker decomposition already showed promising results with regard to artifact extraction, seizure origin localization and artifact removal [3]. Experiment 1 is carried out to get a better understanding of the added value of performing the CWT in combination with Support Tensor Machine (STM) classification using EEG data.

4-3 Experiments

The general framework for all the experiments in each section is given in a summarizing flowchart. All the experiments use two types of validation techniques called leave one seizure out (LOSO) and leave one patient out (LOPO). LOSO uses all but one seizure to train and validates on the remaining seizure data. As a consequence the amount of folds depends on the amount of seizures recorded per patient. LOPO uses multi-patient data for training and validates on a never seen patient. The amount of folds depends on the amount of patients included in the experiment. The experiments carried out in this thesis include 3 patients, which means that two patients are used for training and one patient for testing. This results in three possible combinations of training set and test set. Increasing the amount of patients in the LOPO model naturally increases the model training time when the amount of samples per patient are held constant. The model complexity is also likely to increase in this scenario since the amount of support vectors to describe the separating hyperplane increases with more data. For computational reasons the research is limited to LOPO experiments using three patients.

Each experiment is executed 10 times in order to assess the robustness of the results. This is done by calculating a 95% confidence interval. Each repetition the interictal class is resampled resulting in a different training and testing set. The formula of the 95% confidence interval is shown in equation 4-3.

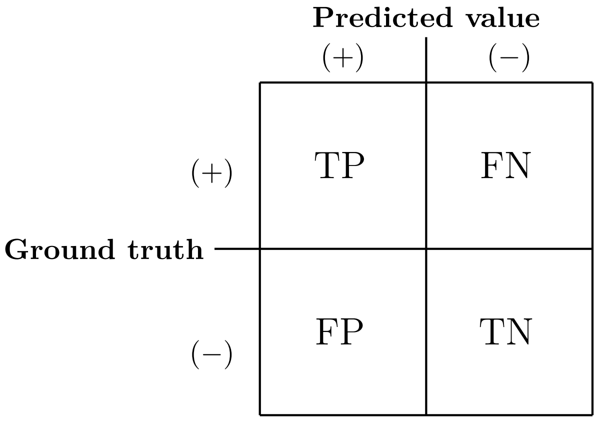
$$CI = \overline{x} \pm z\frac{s}{\sqrt{n}} \tag{4-3}$$

where CI is the confidence interval, $\overline{x}$ is the sample mean, z is the confidence level value ($\approx$ 1.96 for 95% confidence), s is the sample standard deviation and n is the sample size.

The optimal regularization parameter C and kernel parameter σ are found in the range of $2^{[-8,\ldots,8]}$ using integer powers of 2. Because values for σ smaller than 2^0 the results were sub-optimal, the values were discarded for most experiments to reduce the computation time of the experiments. The optimal parameters are selected by first validating the model on the training set. The model whose combination of parameters yield the highest accuracy is then selected and used for validation on the test set.

The training set is balanced (50/50, positive/negative) and contains the maximum possible amount of ictal data per patient. The test set has a ratio of 1/10 positive/negative for all experiments, since the performance must be representative for real use case scenario's. Ideally we want this test set to contain as much of the dataset as possible, but due to computational limitations (especially for the 3D case) the 1/10 ratio is used in most experiments. This ratio is sufficient in order to compare the models introduced in this thesis. As mentioned before: For comparison with the state-of-the-art, the results will be sampled in the ratio given in table 4-1 in order to better assess the performance of the proposed models with a state-of-the-art model.

Performance metrics are used to evaluate and compare machine learning models. All the metrics for binary classification are based on a type of ratio of the four possible outcomes. These four possible outcomes are displayed in the confusion matrix of figure 4-2.



		Predicted value	
		(+)	(−)
Ground truth	(+)	TP	FN
	(−)	FP	TN

Figure 4-2: The confusion matrix.

The four possible outcomes include:

- True Positive (TP) is the amount of positive class samples correctly classified.
- True Negative (TN) is the amount of negative class samples correctly classified.

- False Positive (FP) is the amount of negative class samples incorrectly classified. This factor represents the Type-I error.
- False Negative (FN) is the amount of positive class samples incorrectly classified. This factor represents the Type-II error.

All the relevant metrics used in this thesis can be calculated using these four possible outcomes.

Accuracy The accuracy is the ratio of correct predictions to the total number of predictions:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}. \tag{4-4}$$

The accuracy is a poor metric when working with unbalanced datasets. Since high accuracy can easily be obtained by only predicting a single class. Since the accuracy is still one of the most common metrics in literature, it is still evaluated in this thesis for comparison.

Precision The precision is the ratio of true positives and total positives predicted:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{4-5}$$

In the context of epileptic seizure detection, the precision metric indicates the correctly classified seizures out of all samples classified as a seizure. Naturally we want this value to approach 1, since this would indicate that that there are no false alarms (FP).

Recall Recall is the ratio of true positives to all the positives in the ground truth:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{4-6}$$

In the context of epileptic seizure detection, the recall metric indicates the correctly classified seizures out of all (ground truth) seizures. Naturally we want this value to approach 1, since this would indicate that the model is not missing any seizures (FN).

F1-score The F1-score is the harmonic mean of precision and recall:

$$F_1 = 2 * \frac{\text{prescision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{4-7}$$

For a model to have a F1-score approaching 1, the model needs to not have many false alarms (FP) and not miss any seizures (FN). Hence the F1-score is a good metric to evaluate model performance on unbalanced datasets.

These metrics will be used throughout the thesis in order to evaluate the performance of the different models.

4-3-1 Experiment 1: Classification using the CWT data representation

In figure 4-3 the general framework for experiment 1 using a CWT is presented. In these experiments the data is first transformed into the time-frequency domain using the CWT, yielding a third order tensor. The CWT is computed using 30 scales over a frequency range of 30 Hz. The hypothesis is that the CWT will enrich the data with frequency information which could result in better classification performance. This hypothesis has it's foundations in the fact that the Discrete Wavelet Transform (DWT) is used frequently in the literature as a technique to extract features from the data [30].

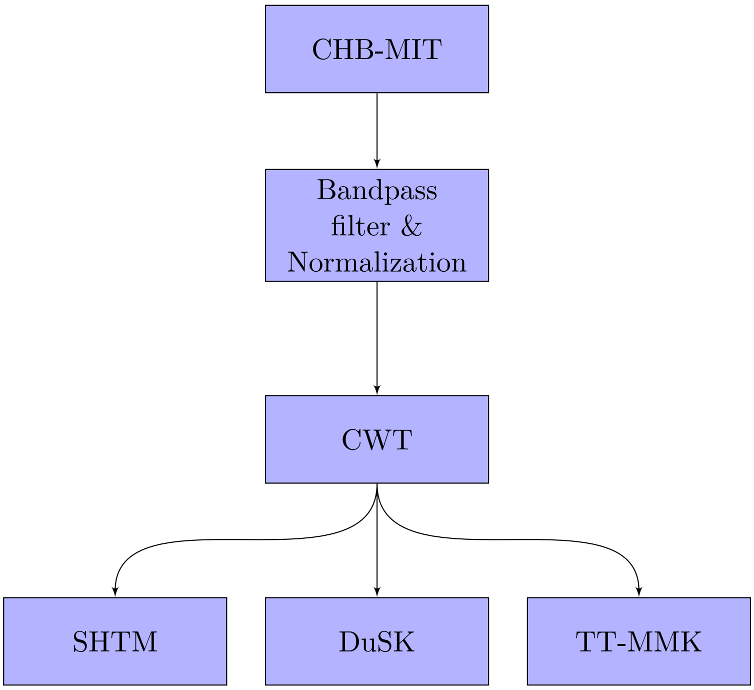


Figure 4-3: Experiment 1: Classification pipeline.

In figure 4-3 it can be seen that the data is pre-processed with a bandpass filter and is normalized using z-score normalization. Hereafter the CWT is applied to the data. Finally the three different methods are applied to the transformed data, where Support Higher-order Tensor Machine (SHTM) and Dual Structure-preserving Kernel (DuSK) use a Canonical Polyadic Decomposition (CPD) to decompose the data. Tensor Train Multi-way Multi-level Kernel (TT-MMK) uses the Tensor-Train Decomposition (TTD) to decompose the data.

4-3-2 Experiment 2: Classification with the original data representation

In figure 4-4 the general framework for experiment 2 using the original second-order tensors (matrices) is presented. When using 2D data the TTD is equivalent to the singular value decomposition (SVD) and the CP decomposition is a sub-optimal form of the SVD. Hence The SVD is used instead, since the SVD is optimal and faster to compute. This also means that TT-MMK is reduced to just using the DuSK kernel, hence it is no longer needed and only SHTM and DuSK are used in this experiment.

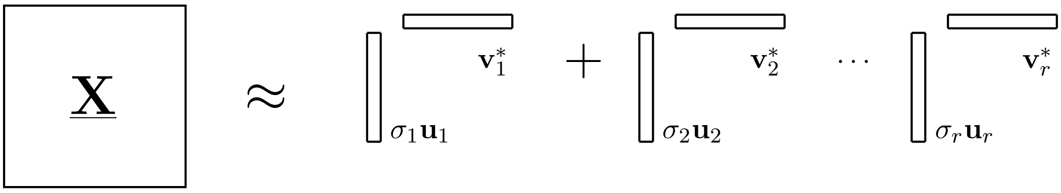
The SVD is a factorization of a matrix into two rotation matrices and a scaling matrix and can be denoted by:

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* \tag{4-8}$$

where $\mathbf{U}$ is an $m \times m$ complex unitary matrix, $\mathbf{\Sigma}$ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers (singular values) on the diagonal and $\mathbf{V}$ is an $n \times n$ unitary matrix. It is also possible to write equation 4-8 as a sum of outer-products, which is shown in equation 4-9:

$$\mathbf{M} = \sum_{i=1}^{r} \mathbf{U}_{(:,i)}\mathbf{\Sigma}_{(i,i)}\mathbf{V}^*_{(:,i)} \tag{4-9}$$

where $r \leq \min\{m, n\}$. This sum can be graphically displayed as:



where $(\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_r})$ and $(\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_r})$ are the orthonormal bases (vectors) contained within $\mathbf{U}$ and $\mathbf{V}$. The singular values $(\sigma_1, \sigma_2 \ldots \sigma_r)$ contained in $\mathbf{\Sigma}$ are absorbed into $(\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_r})$. From the figure the likeness to the CP-decomposition becomes more clear. Instead of using the CP-factors to construct the DuSK$_{\text{RBF}}$ kernel, the orthonormal bases in $\mathbf{U}$ and $\mathbf{V}$ are used.

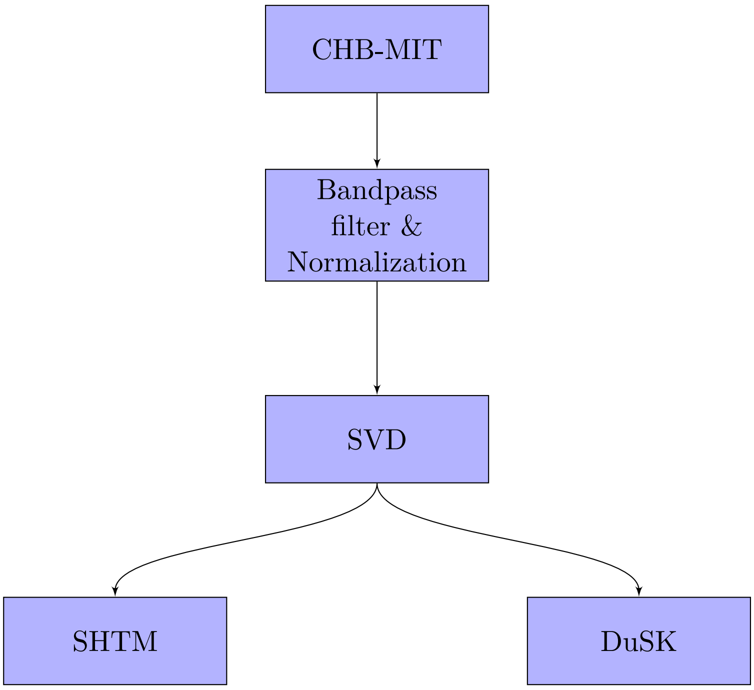
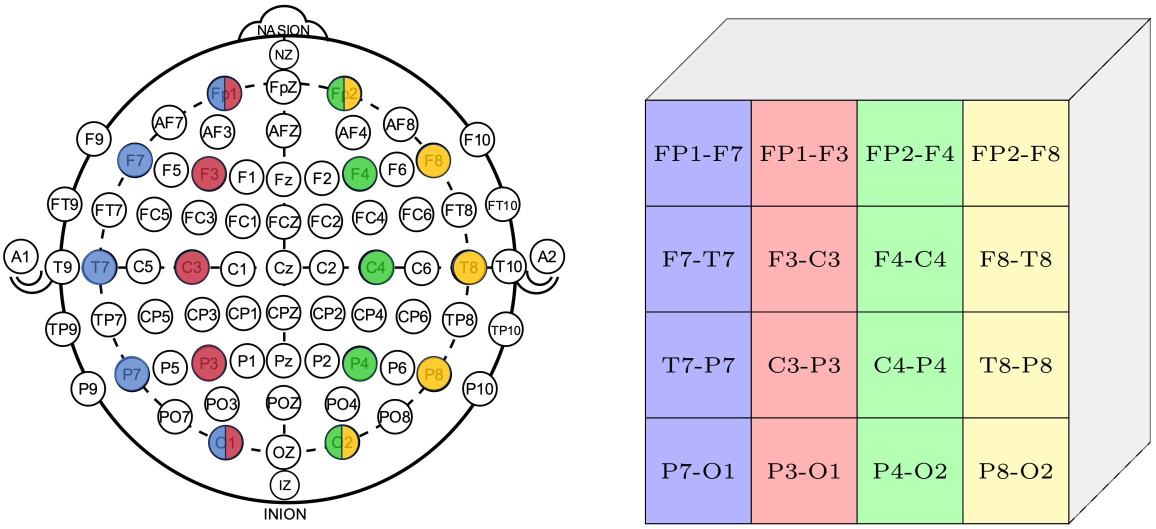


Figure 4-4: Experiment 2: Classification pipeline.

In figure 4-4 it can be seen that the data is pre-processed with a bandpass filter and is normalized using z-score normalization. Hereafter the SVD is applied to the data in order to decompose it. Now SHTM and DuSK are used in order to train a classification model.

4-3-3 Experiment 3: Classification using the tensorized data representation

In figure 4-6 the general framework for experiment 3 is presented. In these experiments the original 2D data is folded (tensorized) into a higher dimension. This enables the use of of the TT-MMK algorithm on the original data. There are many ways in which the data can be tensorized. One could quantize over the channel or time dimension and the extra dimension can have many different sizes (especially the time dimension). Tensorizing over the time-domain would essentially introduce shorter time instances and tensorizing over the channels would impose a spatial structure. First an experiment is conducted by tensorizing over the channel dimension. The chosen approach imposes a structure in the 3rd dimension which resembles the 10-20 montage as seen in figure 1-3. This operation will result in a 3rd order tensor of size 4 × 4 × time. Where the time dimension depends on the sampling window and the sampling frequency of the original data. For the experiments these two quantities are fixed, hence the dimensions of the size of the dimensions of the tensors will be: [4 4 window*sampling frequency] = [4 4 6*256] = [4 4 1536].



FP1-F7	FP1-F3	FP2-F4	FP2-F8
F7-T7	F3-C3	F4-C4	F8-T8
T7-P7	C3-P3	C4-P4	T8-P8
P7-O1	P3-O1	P4-O2	P8-O2

(a) Electrodes included in the tensorized experiments.

(b) Resulting tensor from tensorization.

Figure 4-5: Resemblance between scalp electrode setup and imposed 3^{rd} dimension using colour coded electrodes and fibers.

For the best performing method additional experiments are carried out for higher ranks. In order to do this for ranks higher than 4, the tensor needs to be reshaped. In order to assess what happens to the results when the tensor is reshaped into a more cubic shape, the data is tensorized to the following dimensions: [16 16 96]. This means that now not only the channel dimension, but also the time dimension is folded.

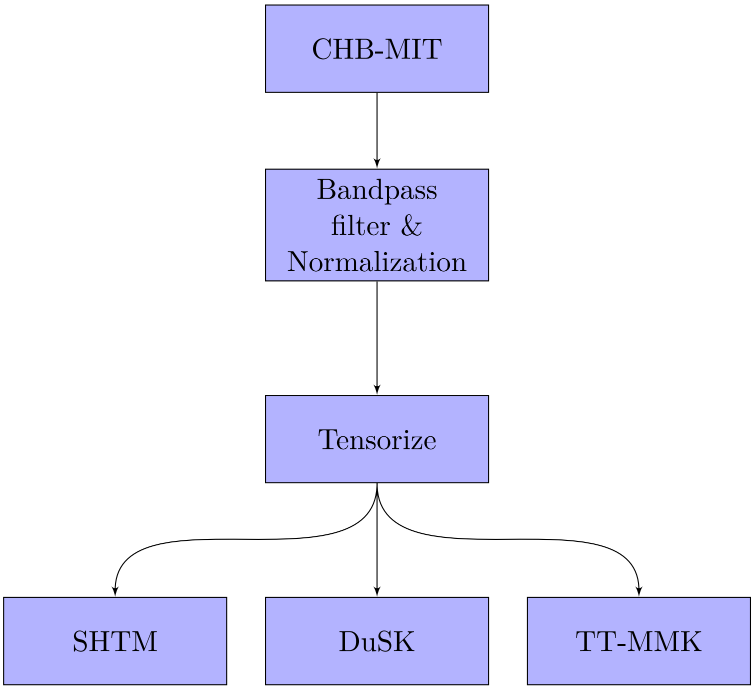


Figure 4-6: Experiment 3: Classification pipeline.

In figure 4-6 it can be seen that the data is pre-processed with a bandpass filter and is normalized using z-score normalization. Hereafter the data is tensorized to one of the dimensions mentioned before. Finally, the three different methods are applied tot the tensorized data.

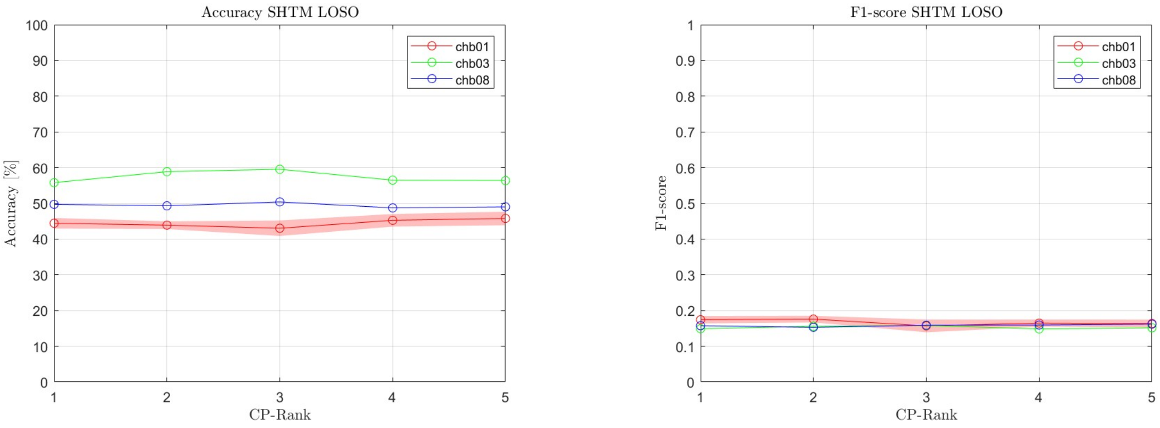
Chapter 5

Results & discussion

In this chapter the results of the different experiments introduced in chapter 4 will be presented and discussed. All the figures include three lines, which each correspond to a different model. For leave one patient out (LOPO) validation, the legend indicates the patients on which the model is trained. The remaining patient is hence used for validation. In some figures the confidence interval is narrow or sometimes non-existent. All experiments were conducted on a MacBook pro (2.7 GHz Dual-Core Intel Core i5) using MATLAB 2023a.

5-1 Results experiment 1: CWT data representation

SHTM The results of applying Support Higher-order Tensor Machine (SHTM) to the third order tensor data are displayed in figure 5-1a and 5-1b. From the figures it can be observed that the performance of a linear kernel applied to wavelet transformed EEG data is poor. The average accuracy lies around 50% which indicates a random predictor. The average F1-score of around 0.15 confirms this statement. Increasing the ranks does not influence the performance significantly. All this indicates that the data is poorly linearly separable and hence SHTM in combination with the wavelet transform is not suitable for patient specific models.

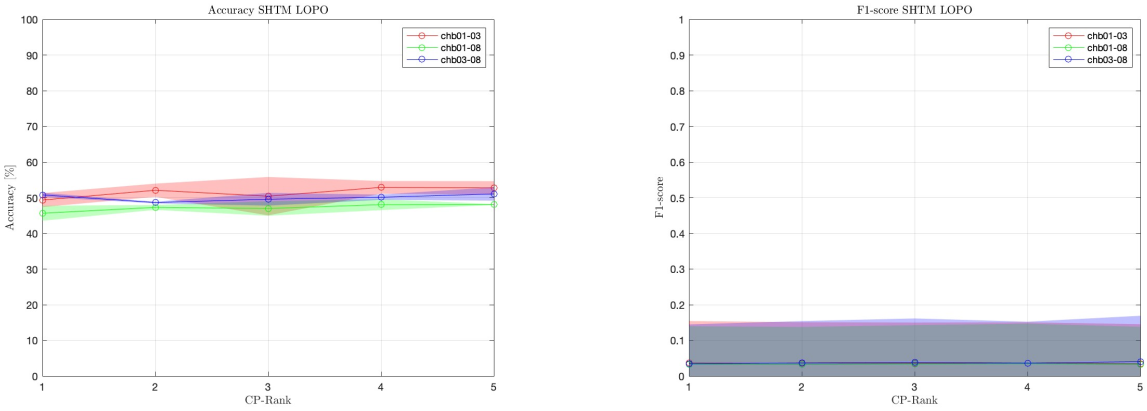


(a) Accuracy

(b) F1-score

Figure 5-1: Experiment 1: Accuracy and F1-score of SHTM using LOSO validation for different CP-ranks.

In figure 5-2a & 5-2b LOPO validation is carried out for SHTM. Similar results as in the leave one seizure out (LOSO) experiment can be observed. Both plots show an increase in confidence interval indicating less robustness to a change in interictal data. The accuracy and F1-scores are too low in order to take this approach into consideration. Hence, SHTM in combination with the wavelet transform is not able to classify the samples effectively in the patient independent case.

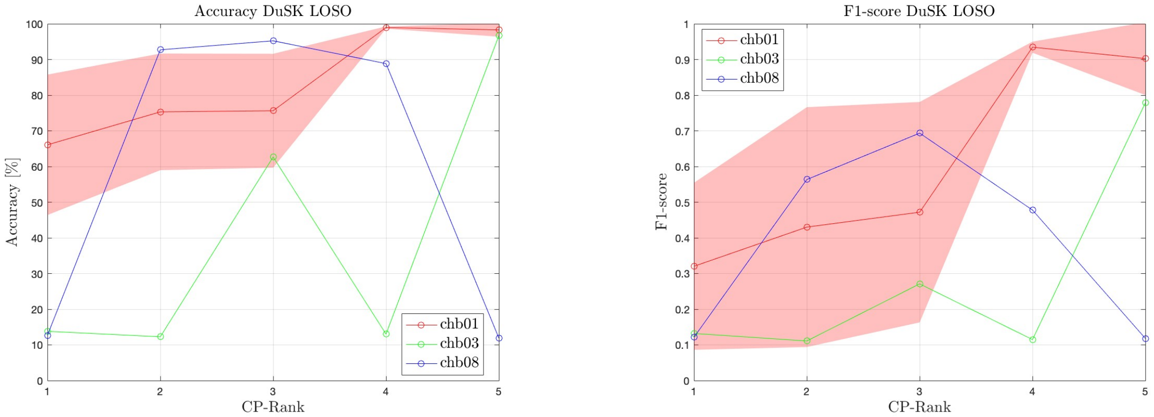


(a) Accuracy

(b) F1-score

Figure 5-2: Experiment 1: Accuracy and F1-score of SHTM using LOPO validation for different CP-ranks.

DuSK From both figures 5-9a and 5-9b it can be seen that the results are not consistent for all patients. The behaviour with increasing rank is unpredictable, especially for patients chb03 and chb08. The accuracy and F1-score for patient chb01 does however increase with increasing rank, but the confidence interval is rather large, indicating a large spread of performance for different sample instances. Despite the unpredictable performance of the models, the chb01 model does show promising performance for ranks 4 and 5.

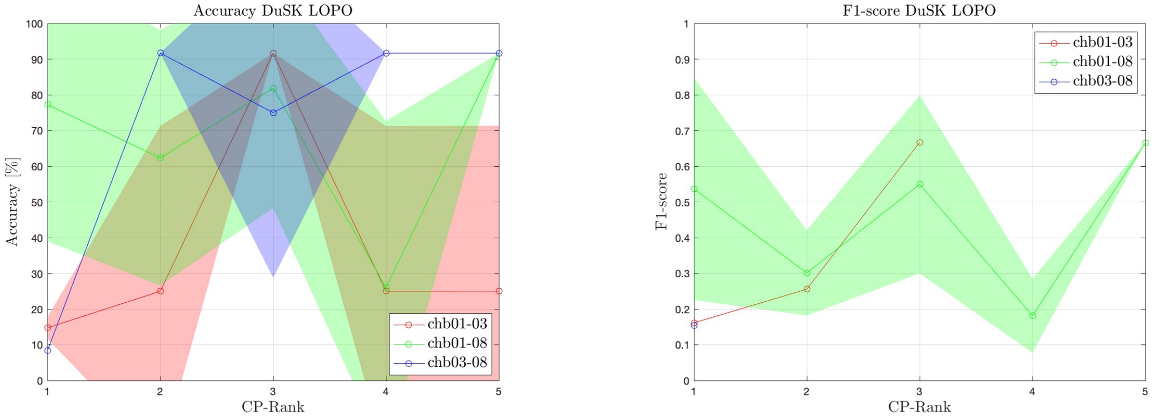


(a) Accuracy

(b) F1-score

Figure 5-3: Experiment 1: Accuracy and F1-score of DuSK using LOSO validation for different CP-ranks.

When looking at figure 5-4a it can be seen that the confidence intervals for LOPO validation are large and the accuracies follow a very unpredictable pattern with increasing rank. Figure 5-4b shows a large spread in the F1-scores of the chb01-08 model and is not able to display all F1-scores for all models, since it could not always be computed. From the figures it can be concluded that the current patient independent model is not a suitable candidate for being a good classifier. A possible reason for the poor performance could be the low amount of patients included in the training of the model. The generalizability could possibly be increased when extra patients are added to the training set. This would however increase the amount of model parameters if the amount of samples per patient are held constant.

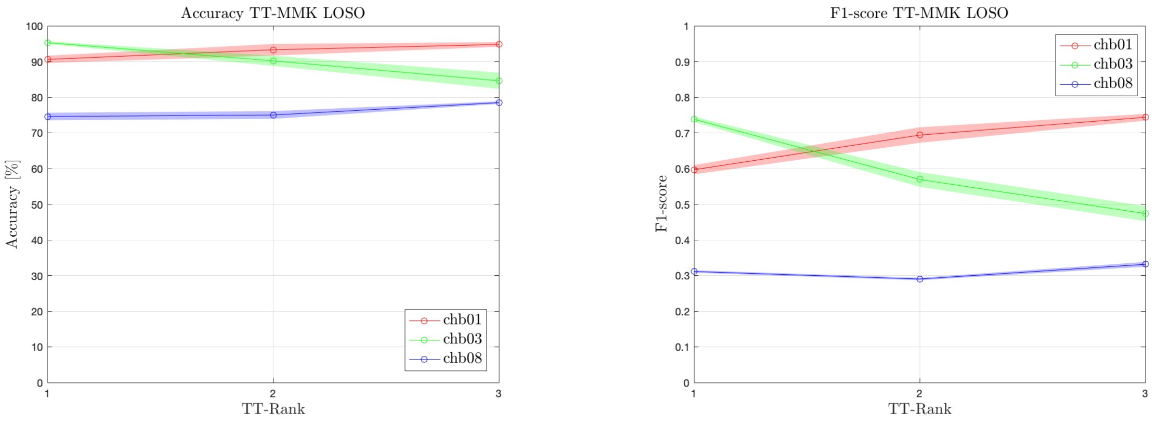


(a) Accuracy

(b) F1-score

Figure 5-4: Experiment 1: Accuracy and F1-score of DuSK using LOPO validation for different CP-ranks.

TT-MMK The patient specific performance metrics of Tensor Train Multi-way Multi-level Kernel (TT-MMK) are displayed in figures 5-5a and 5-5b. It can be observed that the confidence intervals are less wide when comparing it to Dual Structure-preserving Kernel (DuSK). It can be observed that the accuracy and F1-score are increasing when the TT-rank is increased for patients chb01 and chb08. However, this is not the case for patient chb03, which shows a a decrease in accuracy and F1-score with increasing rank. Despite this irregularity, TT-MMK seems to be the most stable Support Tensor Machine (STM) when for patient specific seizure classification when working with Continuous Wavelet Transform (CWT) transformed data.

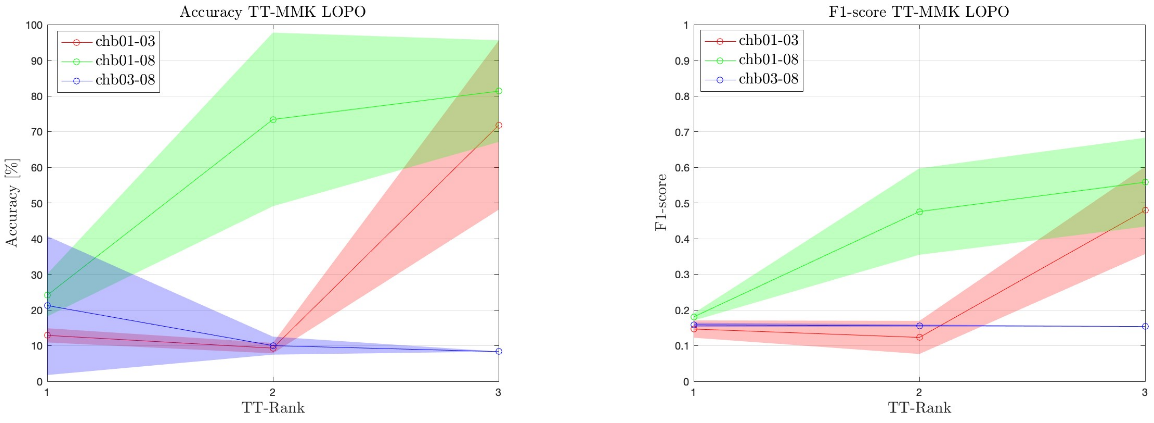


(a) Accuracy

(b) F1-score

Figure 5-5: Experiment 1: Accuracy and F1-score of TT-MMK using LOSO validation for different CP-ranks.

The patient independent performance metrics of TT-MMK, applied to the CWT transformed data, are displayed in figures 5-6a and 5-6b. Just like the patient independent models of DuSK, a large confidence interval with unpredictable performance is observed.



(a) Accuracy

(b) F1-score

Figure 5-6: Experiment 1: Accuracy and F1-score of TT-MMK using LOPO validation for different CP-ranks.

The experiments using SHTM yielded stable but poor performance. The experiments using DuSK had poor performance and stability of the results. The exception is the result of patient chb01 for which the model achieved high accuracy and F1-score. But because of the large confidence intervals and the poor performance on the other patients, the model overall is not performing as an ideal classifier for the problem at hand. The experiments using TT-MMK in the patient specific case yielded better and more stable results, especially in the patient specific experiment. However, the performance for the chb08 model and the chb03-08 model of TT-MMK is still on the lower side of the spectrum.

The average performance metrics for the highest scoring rank for each method and patient are displayed in tables 5-1 and 5-2 . The best performance metrics per model is highlighted in bold.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01	4	45.25%	0.0977	0.6737	0.1706
	chb03	2	58.86%	0.0934	0.4543	0.1550
	chb08	5	49.02%	0.0939	0.5915	0.1620
DuSK	chb01	4	**98.95%**	**0.9394**	**0.9299**	**0.9344**
	chb03	5	**96.67%**	**0.9091**	0.6806	**0.7784**
	chb08	3	**95.23%**	**0.7055**	**0.6821**	**0.6936**
TT-MMK	chb01	9	94.77%	0.6285	0.9118	0.7437
	chb03	1	95.27%	0.6912	**0.7927**	0.7380
	chb08	9	78.47%	0.2234	0.6419	0.3314

Table 5-1: Experiment 1: Comparison between models using LOSO validation.

From table 5-1 it can be observed that DuSK is the best performing model. However, caution must be exercised regarding the displayed average values due to the large uncertainty intervals and unstable behaviour when the ranks are changed slightly.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01-03	1	49.95%	0.0867	0.5246	0.1488
	chb01-03	4	46.30%	0.0818	0.5177	0.1413
	chb03-08	5	**50.60%**	**0.1008**	0.6069	**0.1729**
DuSK	chb01-03	3	**91.67%**	**1**	0.5	**0.6667**
	chb01-08	5	**91.62%**	**0.4987**	**0.9990**	**0.6653**
	chb03-08	1	8.33%	0.0833	**1**	0.1538
TT-MMK	chb01-03	9	71.79%	0.3338	**0.9643**	0.4792
	chb01-08	9	81.35%	0.4056	0.9875	0.5581
	chb03-08	1	21.15%	0.0901	0.8689	0.1580

Table 5-2: Experiment 1: Comparison between models using LOPO validation.

From table 5-2 it can be observed that DuSK is again the best performing model for the chb01-03 and chb01-08 models. TT-MMK has a slightly higher F1-score for the chb03-08 model. Just like with the LOSO results, caution must be exercised regarding the displayed average values due to the large uncertainty intervals and unstable behaviour when the ranks are changed slightly.

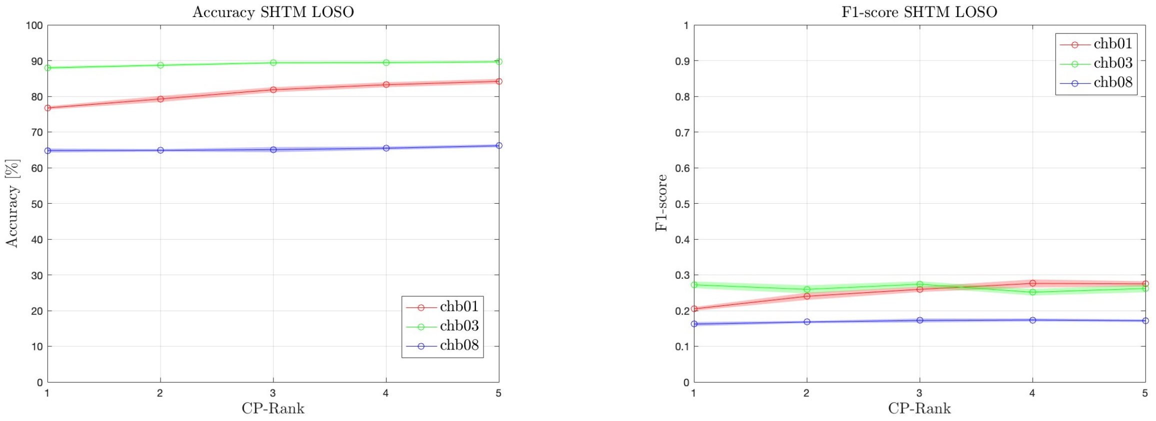
Using the findings from experiment 1, it is now possible to address the first research question which is:

"Do the proposed STMs possess classification capabilities when first applying the CWT?".

The current model setup using the CWT does possess good classification capabilities for one of the patients in the patient specific case, however the fact that it does not work for all patients does not make it a good general classifier in this configuration. For the patient independent case, the models do not possess significant classification capabilities in the current configuration. The patient independent model could potentially be improved by adding more patients.

5-2 Results experiment 2: Original data representation

SHTM The performance metrics of SHTM applied to the data matrices is displayed in figures 5-7a and 5-7b for the patient specific models. When comparing the accuracy with that of figure 5-1a it can be observed that the accuracy has increased. The F1-score observed in figure 5-7b has increased slightly but is still low. This indicates that the model has a bad performance on an imbalanced dataset. Hence this patient specific model does not satisfy the requirements.

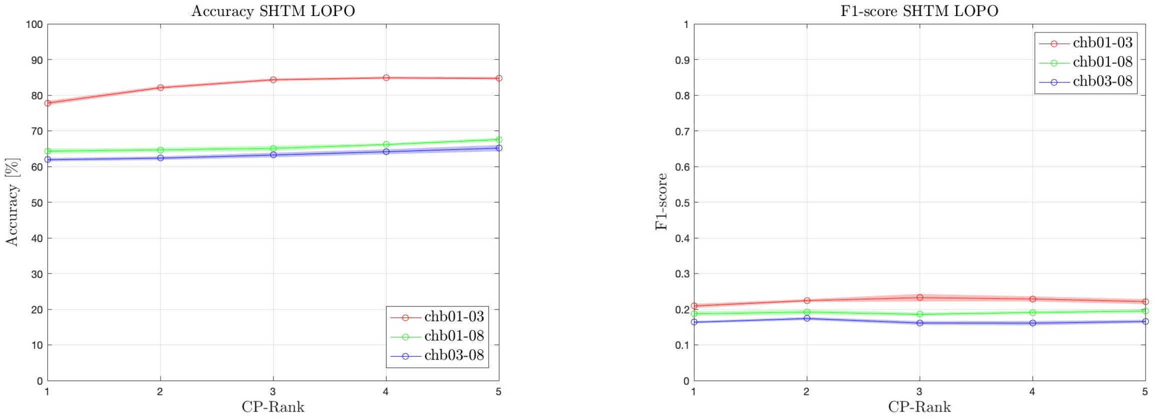


(a) Accuracy

(b) F1-score

Figure 5-7: Experiment 2: Accuracy and F1-score of SHTM using LOSO validation for different CP-ranks.

When evaluating the patient independent performance displayed in figures 5-8a and 5-8b the same arguments can be made as in the patient independent case. The accuracy has improved compared to figure 5-2a but the F1-scores are still low.

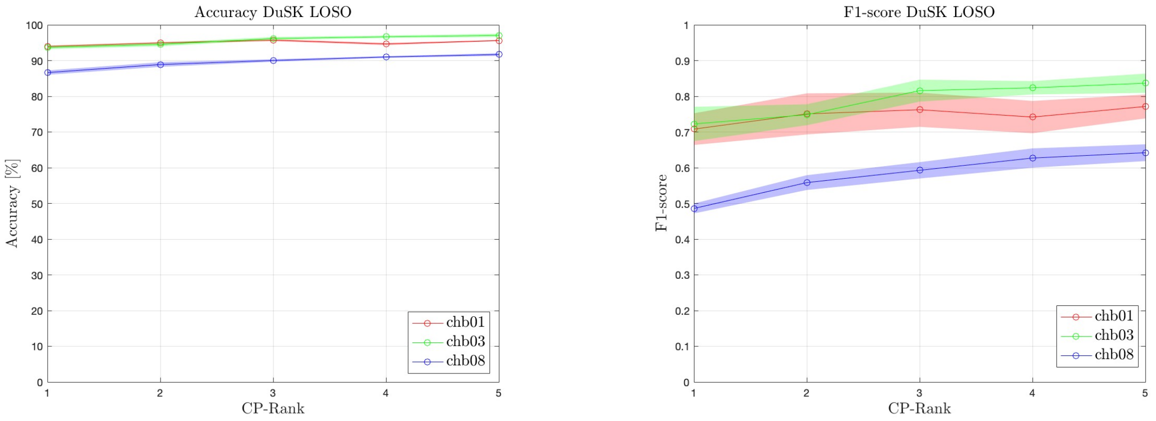


(a) Accuracy

(b) F1-score

Figure 5-8: Experiment 2: Accuracy and F1-score of SHTM using LOPO validation for different CP-ranks.

DuSK The patient specific DuSK model performance metrics are displayed in figures 5-9a and 5-9b. When comparing it to the SHTM results of the previous paragraph, the effect of the Gaussian kernel can now be clearly observed. The average accuracy has increased from ±80% to ±95% for a CP-rank of 5. Even more notable is the increase in average F1-score from ±0.25 to ±0.75 for a CP-rank of 5. When comparing it to the patient independent DuSK with CWT results of figures 5-3a and 5-3b, the overall performance is improved and it is more stable. When increasing the CP-rank the accuracy and F1-score increase on average, which is a more predictable path.

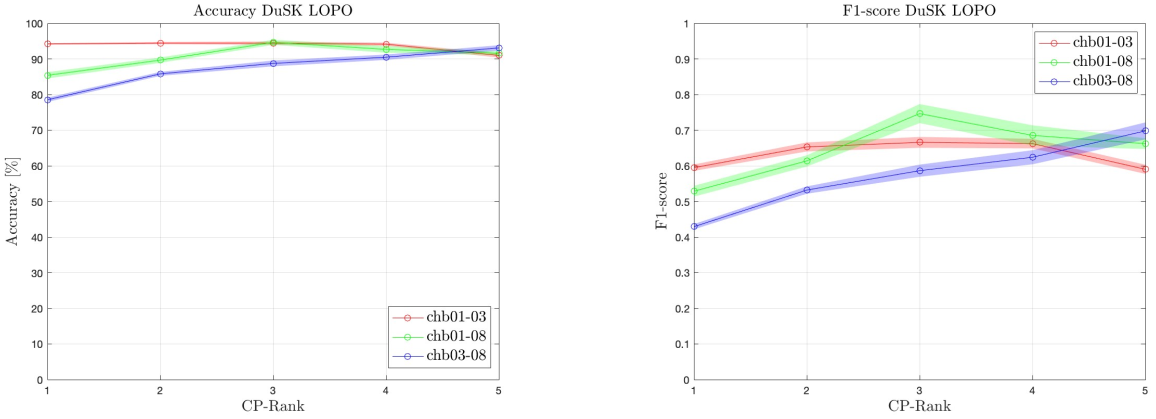


(a) Accuracy

(b) F1-score

Figure 5-9: Experiment 2: Accuracy and F1-score of DuSK using LOSO validation for different CP-ranks.

The patient independent DuSK model performance metrics are displayed in figures 5-9a and 5-9b. It can be seen that the overall patient independent performance is inferior to the patient specific DuSK performance. This is to be expected as this is a common trend observed in most studies. When comparing it to the SHTM case, the influence of the DuSK kernel on performance can be observed just like in the patient specific experiments. The average accuracy has increased from ±75% to ±90% for a CP-rank of 5. The average F1-score has increased from ±0.25 to ±0.6 for a CP-rank of 5.



(a) Accuracy

(b) F1-score

Figure 5-10: Experiment 2: Accuracy and F1-score of DuSK using LOPO validation for different CP-ranks.

The results of the experiment using the singular value decomposition (SVD) yields more consistent and overall better results than using a CWT together with a Canonical Polyadic Decomposition (CPD) or Tensor-Train Decomposition (TTD). Overall using DuSK with a SVD seems to have good classification performance.

In table 5-3 and 5-4 the results for the best ranks are displayed. The best performance metrics per model is highlighted in bold.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01	4	83.51%	0.2190	0.3912	0.2808
	chb03	3	89.45%	0.3295	0.2452	0.2807
	chb08	4	65.57%	0.1100	0.4407	0.1761
DuSK	chb01	5	**95.60%**	**0.9839**	**0.6538**	**0.7855**
	chb03	5	**97.05%**	**0.9797**	**0.7448**	**0.8461**
	chb08	5	**91.70%**	**0.5016**	**0.8524**	**0.6315**

Table 5-3: Experiment 2: Comparison between models using LOSO validation.

From table 5-3 it can be observed that DuSK produces the best performing patient specific models, which is to be expected since SHTM is a linear classifier.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01-03	4	84.97%	0.2003	0.2678	0.2291
	chb01-03	5	67.66%	0.1255	0.4830	0.1993
	chb03-08	2	62.54%	0.1057	0.4685	0.1725
DuSK	chb01-03	3	**94.42%**	**0.6668**	**0.6659**	**0.6659**
	chb01-08	3	**94.66%**	**0.6183**	**0.9423**	**0.7465**
	chb03-08	5	**93.08%**	**0.5494**	**0.9577**	**0.6980**

Table 5-4: Experiment 2: Comparison between models using LOSO validation.

From table 5-4 it can be observed that DuSK also produces the best performing patient independent models. The average F1-score for the patient independent models is lower when comparing it to the patient specific models.

Using the findings from experiment 2, it is now possible to address the second research question which is:

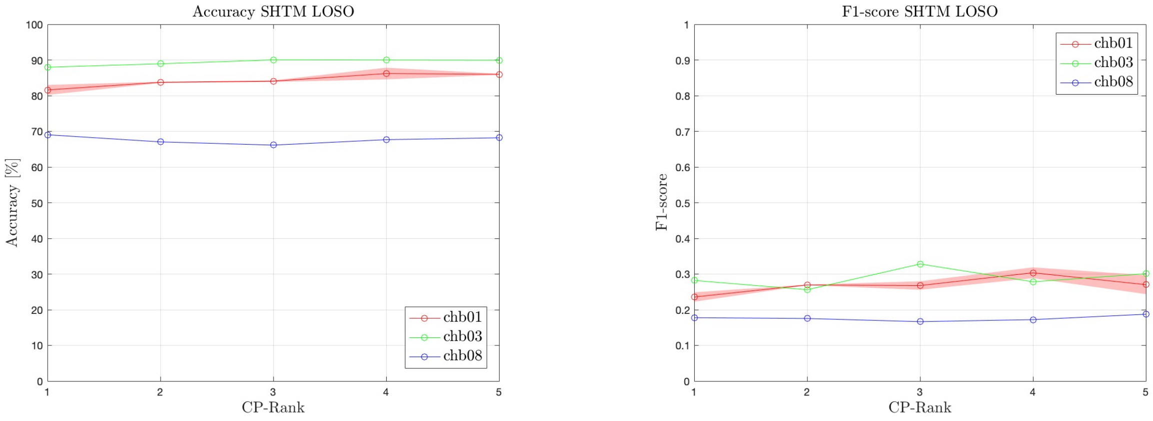
"Do the proposed STMs possess classification capabilities when working with the original electroencephalogram (EEG) data?".

It can be concluded that both SHTM and DuSK possess classification capabilities when working with the original EEG data. However, only DuSK has the potential of being a viable seizure detector.

5-3 Results experiment 3: Tensorized data representation

By imposing a three-dimensional structure onto the two-dimensional data, consistent with the 10-20 system, the idea is that it may be possible to augment the data with additional information. The imposed structure for the following experiments is [4, 4, window*sampling frequency] = [4 4 1536].

SHTM The patient specific results from applying SHTM to the tensorized data are displayed in figures 5-11a and 5-11b. The average accuracy and F1-score is higher when comparing it to the patient specific SHTM with CWT. It's performance is also slightly increased when comparing it to the patient independent SHTM model using a SVD. Indicating that the linear classifier is able to effectively extract the imposed structural information. The F1-scores remain on the low side and therefore the linear patient specific model is not a suitable candidate classifier. Nevertheless, this experiment does give insight into the added value of the imposed structure.

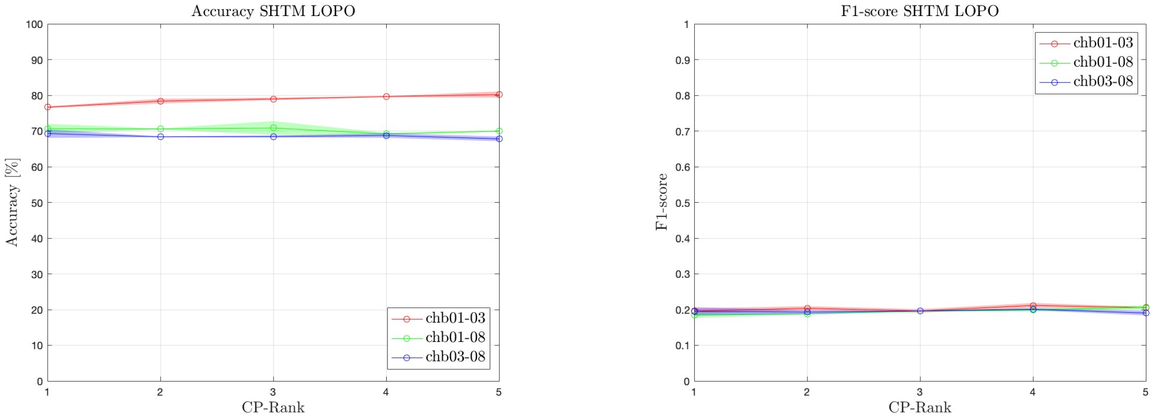


(a) Accuracy

(b) F1-score

Figure 5-11: Experiment 3: Accuracy and F1-score of SHTM using LOSO validation for different CP-ranks.

The patient independent results from applying SHTM to the tensorized data are displayed in figures 5-12a and 5-12b. The average accuracy and F1-score are higher when comparing it to the patient independent SHTM with CWT. It's performance is also slightly increased when comparing it to the patient independent SHTM model using a SVD just like the patient specific model.



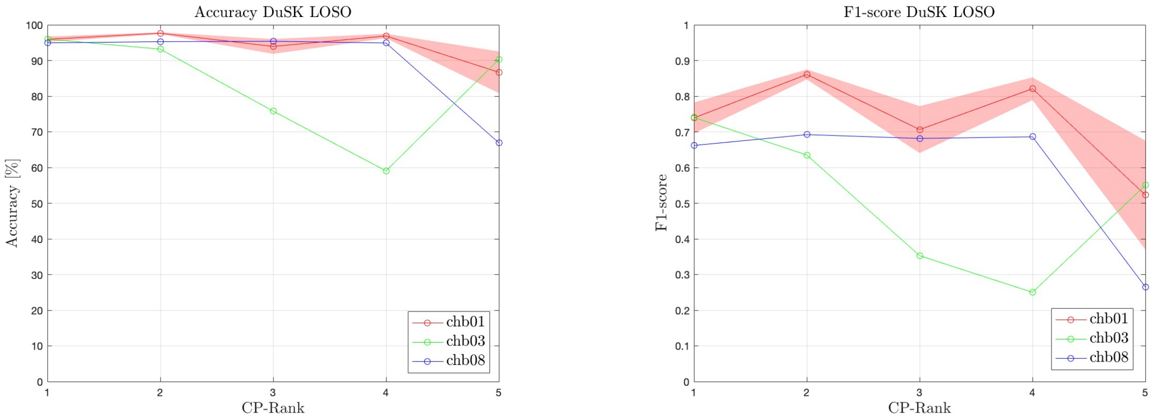
(a) Accuracy

(b) F1-score

Figure 5-12: Experiment 3: Accuracy and F1-score of SHTM using LOPO validation for different CP-ranks.

Overall, the F1-scores using SHTM remain too low for SHTM to be a viable candidate for a real-world epileptic seizure classifier. This is unfortunate in terms of computational complexity, since a linear kernel is faster to compute than a Gaussian kernel. Nevertheless, the SHTM experiments provides a valuable insight into the added value of a kernel when working with EEG data.

DuSK The patient specific results from applying DuSK to the tensorized data are displayed in figures 5-11a and 5-11b. It can be observed that the results are not consistent for every patient. The performance for patient chb01 and patient chb08 is reasonable up to a rank of 4, hereafter the accuracy and F1-score drop significantly. For patient chb03 the opposite performance characteristic is observed. The performance drops up until rank 4 and increases significantly when increased to rank 5. This kind of unstable performance was also observed in the first experiment using the CWT. This indicates that possibly not only the CWT is contributing to the unstable performance, but the Canonical Polyadic (CP)-decomposition using the alternating least-squares (ALS) algorithm might not be the right tool for seizure classification.

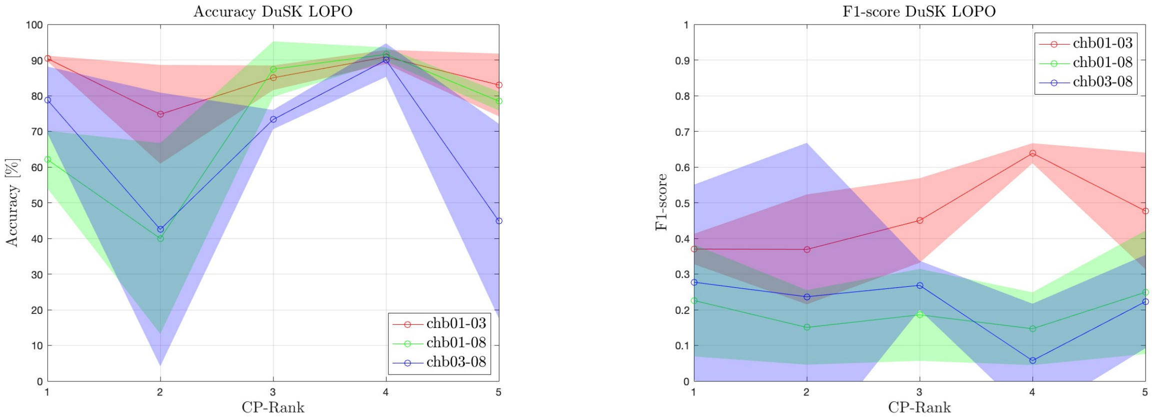


(a) Accuracy

(b) F1-score

Figure 5-13: Experiment 3: Accuracy and F1-score of DuSK using LOSO validation for different CP-ranks.

The patient independent results from applying DuSK to the tensorized data are displayed in figures 5-12a and 5-12b. Just like the patient specific results of DuSK, the results have some inconsistencies. Furthermore, the confidence interval of both the accuracy and the F1-score show a large standard deviation over the iterations. This implies that the results depend more heavily on changes in the sampled subset of negative class and/or the CP decomposition.



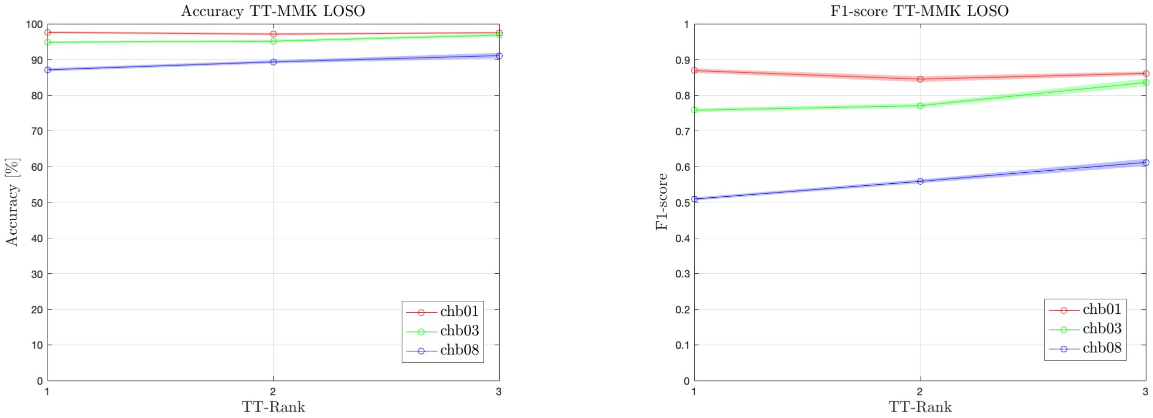
(a) Accuracy

(b) F1-score

Figure 5-14: Experiment 3: Accuracy and F1-score of DuSK using LOPO validation for different CP-ranks.

When comparing the results for DuSK in this section with the results of DuSK from section 5-1 where the CWT is applied first, it can be noted that the unreliability of the results is not only caused by the CWT but most likely also due to the CP decomposition. The results obtained with DuSK in section 5-2 using the SVD are on average superior to the results obtained in this section. Moreover, the stability of the results is also significantly better.

TT-MMK The patient specific results from applying TT-MMK to the tensorized data are displayed in figures 5-15a and 5-15b. It can be observed that the F1-scores are the highest of all experiments so far. The confidence intervals are very narrow, indicating robustness to a change in samples.

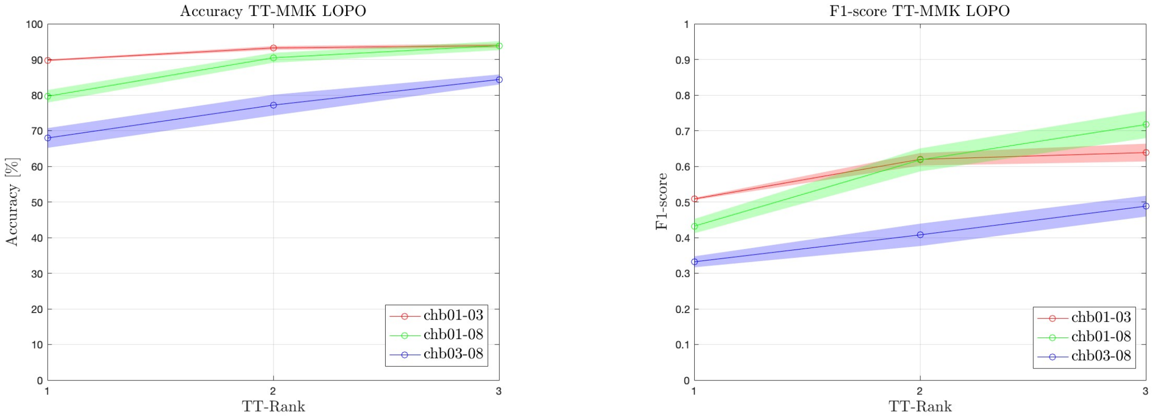


(a) Accuracy

(b) F1-score

Figure 5-15: Experiment 3: Accuracy and F1-score of TT-MMK using LOSO validation for different CP-ranks.

The patient independent results from applying TT-MMK to the tensorized data are displayed in figures 5-16a and 5-16b. It can be observed that the results are more stable and predictable when comparing it to the TT-MMK results of section 5-1. Indicating that the CWT in it's current form is negatively impacting the stability and robustness of the classification results. Furthermore, the performance metrics are significantly better than the TT-MMK with CWT.



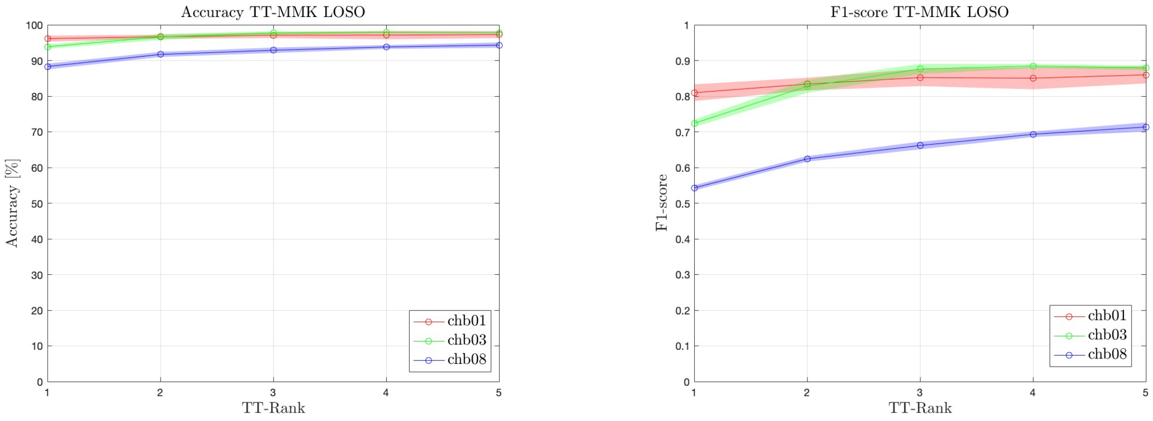
(a) Accuracy

(b) F1-score

Figure 5-16: Experiment 3: Accuracy and F1-score of TT-MMK using LOPO validation for different CP-ranks.

TT-MMK applied to the tensorized data has thus far showed the most promising consistency and performance overall when comparing it to all other methods. For this reason additional experiments are conducted in order to further assess the method. In the first set of experiments, the performance including TT-ranks 4 and 5 is computed. In order to add these extra ranks, the data representation has to be modified since a rank 5 edge between a 4 × 4 frontal slice cannot be computed. The data is therefore tensorized to dimensions [16 16 96].

TT-MMK with new data representation with higher TT-ranks The results of the patient specific experiments using the new data representation and added TT-ranks are displayed in figures 5-17a and 5-17b. When comparing it to the patient specific results with the [4 4 1536] data representation displayed in figures 5-15a and 5-15b, it can be seen that the overall performance for lower TT-ranks is higher for the [4 4 1536] setup. When using the [16 16 96] data representation and the TT-rank is increased to 5, the overall performance is superior.

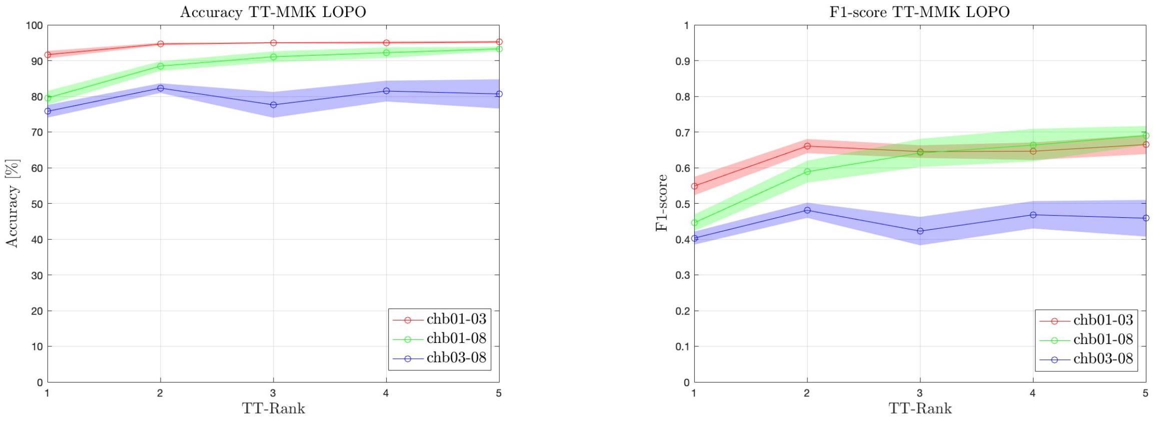


(a) Accuracy

(b) F1-score

Figure 5-17: Experiment 3: Accuracy and F1-score of TT-MMK using LOSO validation for different CP-ranks and a different data representation.

The results of the patient independent experiments using the new data representation and added TT-ranks are displayed in figures 5-18a and 5-18b. The average performance increase, observed for the patient specific TT-MMK models for the new data representation and higher ranks, is not observed for the patient independent case. The [4 4 1536] representation still has the best performance of the two.



(a) Accuracy

(b) F1-score

Figure 5-18: Experiment 3: Accuracy and F1-score of TT-MMK using LOPO validation for different CP-ranks and a different data representation.

Changing the data representation and adding extra ranks can be beneficial for a patient specific model. It does however not increase the overall performance of the patient independent model. As mentioned before, adding additional patients to the LOPO models might increase the performance.

The results from the experiments using the tensorized data representation are displayed in tables 5-5 and 5-6 for LOSO and LOPO respectively. The best performance metrics per model is highlighted in bold.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01	4	86.20%	0.2611	0.3641	0.3036
	chb03	3	90.03%	0.3746	0.2919	0.3281
	chb08	5	68.19%	0.1193	0.4370	0.1874
DuSK	chb01	4	95.95%	0.7025	0.9088	0.7924
	chb03	5	94.68%	0.6179	0.9112	0.7364
	chb08	5	**94.65%**	**0.6564**	0.7240	0.6885
TT-MMK	chb01	9	**97.26%**	**0.7722**	**0.9728**	**0.8688**
	chb03	16	**97.89%**	**0.8026**	**0.9853**	**0.8845**
	chb08	25	94.31%	0.6067	**0.8675**	**0.7136**

Table 5-5: Experiment 3: Comparison between models using LOSO validation.

It can be observed in table 5-5 that the patient specific models using TT-MMK have the highest F1-scores for all patients. This indicates that TT-MMK is able to detect patterns in the raw data better than the other methods. It is also significantly more stable than DuSK rendering it a potential good STM for patient specific EEG tensor classification.

model	patient	CP-rank	accuracy	precision	recall	F1
SHTM	chb01-03	4	79.64%	0.1562	0.3280	0.2116
	chb01-03	5	69.96%	0.1324	0.4694	0.2066
	chb03-08	4	68.75%	0.1276	0.4713	0.2008
DuSK	chb01-03	4	90.86%	0.4845	**0.9599**	0.6385
	chb01-08	5	78.46%	0.1671	0.4875	0.2489
	chb03-08	1	78.81%	0.2033	0.4453	0.2772
TT-MMK	chb01-03	9	**93.85%**	**0.6271**	0.6511	**0.6386**
	chb01-08	9	**93.85%**	**0.5855**	**0.9292**	**0.7172**
	chb03-08	9	**84.36%**	**0.3360**	**0.8934**	**0.4882**

Table 5-6: Experiment 3: Comparison between models using LOPO validation.

From table 5-6 it can be observed that also the patient independent TT-MMK models have the best performance for all patients. It can now be concluded that TT-MMK using either the [4 4 1536] or [16 16 96] setup has the highest F1-score for all patients in both the LOSO and the LOPO table. This indicates that TT-MMK is the superior method when working with tensorized EEG data for both LOSO and LOPO validation.

Using the findings from experiment 3, it is now possible to address the third research question which is:

"Does tensorizing the data by reshaping, increase the performance when comparing it to the performance on the original data?

It can be concluded that by tensorizing the data the performance of the patient specific models can be increased when comparing it to the experiments of section 5-2 where the original data is used. For the patient independent case tensorizing the data does not necessarily increase the overall performance of the models in the current LOPO configuration. Adding extra patients to the LOPO model might increase the performance.

The fact that the imposed electrode structure in the third dimension increases patient specific performance but does not improve performance for patient independent models confirms existing knowledge about epileptic seizures. Specifically, it highlights the variability in seizure types and their occurrence in different brain regions for each patient. Adding information about electrode location provides information on seizure characteristics, which is beneficial for patient specific seizure classification but not necessarily for patient independent seizure classification.

5-4 Comparison & Discussion

First the performance of the proposed methods are compared to one another by taking the best performing method for each model from each of the previous sections. The comparison is divided into LOSO and LOPO validation metrics. Hereafter, the the best performing method overall is compared to a state-of-the-art seizure detection algorithm.

5-4-1 Comparison of proposed methods

LOSO The metrics of the best performing models for LOSO validation are displayed in table 5-7. It can be observed that TT-MMK with tensorization is the best performing model for patients chb03 and chb08 and that DuSK with CWT is the best performing model for patient chb01. From section 5-1, it became evident that the results for the DuSK with CWT model were not consistently stable for the patients chb03 and chb08. Even though the model is performing the best for the LOSO case for patient chb01, it cannot be used for the other patients since the F1-scores are not satisfactory and due to the unpredictable behaviour when increasing the rank.

The exact reason why the model performs so well on one patient and is under performing and unstable for other patients is hard to point out. A possible reason could be the way that the training models are evaluated in order to obtain the optimal C and σ parameter. The current method of performing a grid search for the C and σ parameter, chooses the combination for which the model has the highest accuracy on the training set. In practice there were combinations which gave the same accuracy, hence a combination was chosen at random. The model might be very sensitive to the choice of hyperparameters when evaluated on the test set. Possibly resulting in the observed performance. A different method for optimizing the parameters for C and σ might result in better and more stable performance for DuSK with CWT.

The unstable results could also be attributed to the CP decomposition that is used on the data. The reason for this being the same unstable performance observed in section 5-3 for the DuSK models. These models do not use a CWT and hence the isolated effect of classification using the CP decomposition is observed. The ALS method used to compute the CP decomposition is not guaranteed to converge to a global minimum or even a stationary point [15]. This could result in sub-optimal tensor approximations of the original data for some of the ranks and therefore sub-optimal classification performance.

The results for TT-MMK with tensorization follow a more predictable and stable line when the rank is increased. The confidence intervals for all patients is also small, indicating robust performance when other samples are used. Even though, DuSK with CWT has a higher F1-score of 0.93 for patient chb01, the F1-score of TT-MMK with tensorization is also commendable. For these reasons the TT-MMK with tensorization models are used for comparison with the state-of-the-art presented in the next subsection.

model	patient	rank	accuracy	precision	recall	F1
DuSK w/ CWT	chb01	4	**98.95%**	**0.9394**	0.9299	**0.9344**
	chb03	5	**96.67%**	**0.9091**	0.6806	0.7784
	chb08	3	**95.23%**	**0.7055**	0.6821	0.6936
DuSK w/ SVD	chb01	5	95.60%	0.9839	0.6538	0.7855
	chb03	5	97.05%	0.9797	0.7448	0.8461
	chb08	5	91.70%	0.5016	0.8524	0.6315
TT-MMK w/ tensorization	chb01	9	97.26%	0.7722	**0.9728**	0.8688
	chb03	16	97.89%	0.8026	**0.9853**	**0.8845**
	chb08	25	94.31%	0.6067	**0.8675**	**0.7136**

Table 5-7: Comparison between best proposed method from all experiments for LOSO validation.

LOPO The metrics of the best performing models for LOPO validation are displayed in table 5-8. The best performing chb01-03 model is trained using DuSK with CWT but the F1-score is just slightly better than the F1-score of the DuSK with SVD method. Because of the higher robustness and stability of the models trained with DuSK with SVD this model is preferred over the slightly better DuSK with CWT model. For the chb01-08 and chb03-08 models the best method is also DuSK with SVD.

The fact that the method using a second-order tensor has the best performance is a bit underwhelming. A possible reason why this behaviour is observed could lie in it's ability to generalize better to unseen data. Ictal and interictal EEG data characteristics can differ greatly between patients, which is why LOPO validation often has lower scores than it's LOSO counterpart. The results of both LOSO and LOPO indicate that by tensorizing EEG data (3^{rd}-order and higher) and by using a tensor decomposition it becomes more suitable for patient specific models. This essentially means that the tensor representations are more prone to overfitting which positively impacts the LOSO performance. The downside of this phenomena is the fact that it is less competent in correctly classifying data of unseen patients. Conversely, the second-order methods underperform on LOSO validation and outperform on LOPO validation.

model	patient	rank	accuracy	precision	recall	F1
DuSK w/ CWT	chb01-03	3	91.67%	**1**	0.5	**0.6667**
	chb01-08	5	91.62%	0.4987	**0.9990**	0.6653
TT-MMK w/ CWT	chb03-08	1	21.15%	0.0901	0.8689	0.1580
DuSK w/ SVD	chb01-03	3	**94.42%**	**0.6668**	**0.6659**	0.6659
	chb01-08	3	**94.66%**	**0.6183**	0.9423	**0.7465**
	chb03-08	5	**93.08%**	**0.5494**	**0.9577**	**0.6980**
TT-MMK w/ tensorization	chb01-03	9	93.85%	0.6271	0.6511	0.6386
	chb01-08	9	93.85%	0.5855	0.9292	0.7172
	chb03-08	9	84.36%	0.3360	0.8934	0.4882

Table 5-8: Comparison between best proposed method from all experiments for LOPO validation.

5-4-2 Comparison to state-of-the-art

Epileptic seizure detection using algorithms is a very active field of research, as a result there exist many papers on the subject, which all claim a certain performance. Unfortunately, many papers do not describe all parts of the classification pipeline in much detail and neglect to mention all the relevant performance metrics. This makes a one-on-one comparison between methods difficult. The paper selected for comparison in describes the classification pipeline and performance metrics well, but the classification pipeline does differ in some ways to the proposed algorithm in this thesis.

The method which is used for comparison is described in a paper by Gómez et al. [31]. The paper proposes an algorithm to detect epileptic seizures using an imaged-EEG representation of brain signals. The proposed detection algorithm is a convolutional neural network which work especially well on imaged data representations. One of the main differences is that the paper presents a method which is trained on both scalp and intracranial EEG data. Whereas the models in this thesis are solely trained on scalp EEG data.

Another difference is that the paper uses the complete CHB-MIT dataset for training and testing unlike the experiments conducted in this thesis, which use a subset of the complete dataset. The paper hence uses the interictal/ictal ratio's described in table 4-1. In order to make a more accurate comparison, the results are sampled in accordance to these ratio's. This sampling is repeated 10 times of which the average performance is used for comparison. In table 4-1 the mean value of the performance metrics is displayed together with the sample standard deviation. The sample standard deviation is calculated using the following formula:

$$s = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}} \tag{5-1}$$

where x_i is a single calculated performance metric, $\bar{x}$ is the sample mean of the performance metric and n is the total number of iterations (n=10).

The last main difference is that the LOPO models in the paper are trained on all but one patient and the models trained in this thesis are trained on two patients and validated on the remaining patient. This could be an advantage for the convolutional neural network (CNN) if the collection of all but one patients contains a better representation of what an average seizure looks like or what non-seizure data looks like. It could also be advantageous for our methods if the data in the selected training patients exhibit greater similarity to each other compared to the patients that are not included. However, we will only ascertain this when all patients are included in the patient-independent study.

Just like the proposed method in this thesis, the method of Gómez et al. does not need any estimation of pre-selected features and uses a minimal amount of pre-processing steps. In table 5-9 the performance metrics of the model by Gómez et al. and the best performance metrics of the models designed in this thesis are displayed.

model	patient	CP-rank	accuracy	precision	recall	F1	№ of parameters
Gómez et al. CNN [31]	chb01	-	**99.9%**	**0.8688**	0.674	0.7591	314.000
	chb03	-	**99.66%**	0.7697	0.8355	0.8012	314.000
	chb08	-	**99.11%**	0.2401	0.2064	0.222	314.000
TT-MMK w/ tensorization	chb01	9	97.32±0.85%	0.7691±0.06	**0.9814±0.01**	**0.8614±0.03**	**828**
	chb03	16	97.86±0.17%	**0.8051±0.01**	**0.9826±0.01**	**0.8850±0.01**	**768**
	chb08	16	94.26±0.53%	**0.6073±0.03**	**0.8713±0.01**	**0.7154±0.02**	**1816**

Table 5-9: Comparison of best proposed method to state-of-the-art for LOSO validation.

From table 5-9 it can be observed that the proposed method outperforms the CNN for the selected patients when purely comparing the F1-scores. However, critical examination of the comparison must be maintained due to the inherent differences in the test set, despite the sampling strategy. The comparison does however gives a positive indication of the possibilities the model has, especially when taking into account the massive reduction of number of model parameters. The accuracy of the CNN is however still higher than the models proposed in this thesis. This is caused by the high number of true negatives, which are not included in the other performance metrics. Since the true negatives are the least important metric this difference in accuracy between the metrics is not that relevant.

model	(train) patients	rank	accuracy	precision	recall	F1	№ of parameters
Gómez et al. CNN [31]	01-03	-	**99.87%**	**0.8323**	**0.6963**	**0.7582**	314.000
	03-08	-	**99.3%**	0.2368	0.6219	0.3430	314.000
	01-08	-	**98.44%**	0.3108	0.1798	0.2279	314.000
DuSK w/ SVD	01-03	3	94.50±0.12%	0.6712±0.01	0.6658±0.01	0.6684±0.01	**1596**
	01-08	3	94.78±0.16%	**0.6227±0.01**	**0.9453±0.01**	**0.7507±0.01**	**2644**
	03-08	5	93.55±0.45%	**0.5676±0.02**	**0.9575±0.01**	**0.7126±0.02**	**2584**

Table 5-10: Comparison of best proposed method to state-of-the-art for LOPO validation.

From table 5-10 it can be observed that the proposed method outperforms the CNN for the models chb01-08 and chb03-08 when evaluating the average F1-score. These models have high recall when comparing it to the CNN, which indicates that there are few false positives (false alarms). The CNN does, however, outperform the proposed method with the chb01-03 model. If more patients are added to the LOPO models in future research, the proposed models might be able to outperform all of the CNN models presented in the paper. Overall, the results show promising potential for the DuSK with SVD model for patient independent seizure classification.

Using the findings from the comparison to a state-of-the-art method, it is now possible to address the final research question which is:

"Are the newly proposed methods able to rival the state-of-the-art methods in terms of performance (Accuracy & F1-score) and model complexity?" .

Although a one-to-one comparison cannot be fully drawn, it becomes evident that the both the LOPO and LOSO method exhibit the potential to rival state-of-the-art methods with fewer model parameters. All the patient specific models have higher F1-score and 2 out of the three patient independent models have higher F1-score. This patient independent performance is achieved by using only two patients for training while the CNN is trained on all but one patient. This could indicate that by adding more patients to the LOPO models the proposed method will be able to outperform the CNN on all the models.

The accuracy of the CNN is higher for all models, but as discussed, this is less relevant than the observed F1-scores because of the imbalanced nature of the dataset. The higher amount of test data used by the CNN is driving the high accuracy score, primarily due to the true negatives. As seen in chapter 4, the true negatives are not included in the recall and precision metric.

Chapter 6

Conclusions & recommendations

6-1 Conclusions

This thesis proposes a novel approach to the epileptic seizure detection problem using Support Tensor Machine (STM)s. Three types of experiments have been conducted using different representations for the electroencephalogram (EEG) data. The STMs that have been used in the experiments are Support Higher-order Tensor Machine (SHTM), Dual Structure-preserving Kernel (DuSK) and Tensor Train Multi-way Multi-level Kernel (TT-MMK). Where SHTM uses a linear kernel and DuSK and TT-MMK make use of the DuSK_{RBF} kernel.

The first experiment uses a time-frequency domain representation of the data by first applying a Continuous Wavelet Transform (CWT). The resulting representation of the data is a third order tensor with dimensions [channels, time, scales]. The results showed poor performance in terms of stability and F1-score for both patient specific (leave one seizure out (LOSO)) and patient independent validation (leave one patient out (LOPO)). The only exception was the patient specific DuSK model for patient chb01 which showed an increasing performance with increasing rank and a high F1-score for CP-rank 4. It was observed that the current setup of the CWT together with the selected STMs possess poor classification capabilities on EEG data for most patients. Therefore it can be concluded that CWT does not enrich the data in a way that is always beneficial for the proposed STMs in the current setup. However, due to the very good performance observed for the patient specific model of patient chb01, it cannot be conclusively determined that the CWT is not beneficial at all for enriching data for STM learning.

The second experiment uses the original representation of the CHB-MIT data. The representation is two-dimensional, therefore the singular value decomposition (SVD) was used in order to decompose the data. The DuSK results show promising performance for both the patient specific and the patient independent models. When comparing the results to those of experiment 1, an increase in stability and predictability of the performance is observed with increasing rank. Being able to use the SVD in order to decompose the data has the computational advantage over the experiments which use a tensor decomposition, since the SVD is faster and more reliable to compute.

The final experiments use a tensorized data representation mimicking the placements of electrodes on the scalp in the added dimension. The results showed superior performance of the TT-MMK method for this experiment, yielding the best results using both LOSO and LOPO validation. The TT-MMK model trained on the tensorized data is even the best model overall for LOSO validation. This result indicates that the imposed structure of the electrodes in the data helps identifying patient specific seizure characteristics. The patient independent models trained using this new data representation could not outperform the DuSK with SVD model. This indicates that by adding patient specific information about a seizure profile, does not help when classifying seizures of an unseen patient. It can be concluded that when tensorizing the data the models perform better for LOSO validation, but when using the original second-order tensors the model is able to generalize better to unseen patients, causing the DuSK with SVD model to outperform the TT-MMK with tensorization model.

The best performing patient specific method (TT-MMK with tensorization) and the best performing patient independent method (DuSK with SVD) were compared to a state-of-the-art convolutional neural network (CNN). The TT-MMK with tensorization had higher average F1-scores then the CNN for all the patient specific models. DuSK with SVD outperformed the CNN for two of the three models. Although a one-to-one comparison cannot be fully drawn, it can be concluded that both the TT-MMK with tensorization model and the DuSK with SVD model have the potential to rival state-of-the art methods using significantly fewer model parameters.

6-2 Recommendations for future research

Despite the successes achieved using the combination of CWT and tensor decompositions in other aspects concerning EEG data analysis like seizure localisation, it's benefits for seizure classification did not stand out in this research. Besides the patient specific DuSK model for patient chb01 from section 5-1, the performance of a model using CWT underperforms when comparing it to the other experiments. Moreover, the models showed a large spread in results when repeating the experiment and the models performance could sometimes decrease by 80% when increasing the rank, which is not desirable behaviour for a real-life classifier. It is obvious that the current approach using the CWT is not the most optimal, but because of the outlier observed, there is hope that the method using a combination of the CWT and a STM could prove effective eventually. Future research might include new methods to better exploit the tensor structure of CWT transformed EEG data in a more efficient way. An approach could be quantizing the tensor, which is tensorization to a high order with very small dimension size, in order to decompose it to a Tensor-Train (TT) which can then be used in a different learning machine.

Due to computational limitations, only a subset of the total data available per patient is used for validation. While the results presented in this thesis give a good indication of the performance, it can not be compared directly to studies which use the complete dataset for validation. Further research might include more data in order to give a more definite answer to the question whether the performance holds for the complete dataset.

The training and testing of the patient independent models is done using only three patients due to computational limitations. Most research in the literature use all but one patient for training and the remaining patient data for testing, which makes fairly comparing the patient independent models presented in this thesis to the models in the literature difficult. Future research could add more patients to the patient independent model which may determine whether the addition of extra patients to the training set can enhance the performance of the patient independent models.

The TT-MMK method first uses the TT-SVD algorithm to construct a TT and then converts it to a Canonical Polyadic (CP) decomposition. If a third order tensor is used the resulting CP-rank from a rank R TT-decomposition is R^2. This scales even faster for higher order tensors, rendering the method too computationally intensive for higher order tensors. Further research may explore the possibilities to learn directly form the TT representation in order to make the use of higher order tensors more computationally efficient.

All of the aforementioned points collectively can serve as a solid foundation for a follow-up thesis study on epileptic seizure classification using EEG data and STMs.

Bibliography

[1] Sanguk Ryu and Inwhee Joe. A hybrid densenet-lstm model for epileptic seizure prediction. *Applied Sciences*, 11(16):7661, 2021.

[2] Andrzej Cichocki, Namgil Lee, Ivan V. Oseledets, Anh Huy Phan, Qibin Zhao, and Danilo P. Mandic. Low-rank tensor networks for dimensionality reduction and large-scale optimization problems: Perspectives and challenges PART 1. *CoRR*, abs/1609.00893, 2016.

[3] Evrim Acar, Canan Aykut-Bingol, Haluk Bingol, Rasmus Bro, and Bülent Yener. Multiway analysis of epilepsy tensors. *Bioinformatics*, 23(13):i10–i18, 2007.

[4] Anthony Ngugi, Christian Bottomley, Immo Kleinschmidt, Ley Sander, and Charles Newton. Estimation of burden of active and life-time epilepsy: A meta-analytic approach. *Epilepsia*, 51:883–90, 05 2010.

[5] Patrick Kwan and Martin J. Brodie. Early identification of refractory epilepsy. *New England Journal of Medicine*, 342(5):314–319, 2000. PMID: 10660394.

[6] 124. Electrode locations of international 10-20 system for eeg (electroencephalography) recording, 2010. [Online; accessed 14-August-2023].

[7] Der Lange. Generalized 3 hz spike and wave discharges in a child with childhood absence epilepsy, 2006. [Online; accessed 14-August-2023].

[8] Robert S Fisher and Jerome J Engel Jr. Definition of the postictal state: when does it start and end? *Epilepsy & Behavior*, 19(2):100–104, 2010.

[9] Ziwei Wang and Paolo Mengoni. Seizure classification with selected frequency bands and eeg montages: a natural language processing approach. *Brain Informatics*, 9(1):11, 2022.

[10] J Gotman. Automatic recognition of epileptic seizures in the eeg. *Electroencephalography and Clinical Neurophysiology*, 54(5):530–540, 1982.

[11] Srikanth Cherukuvada and R Kayalvizhi. A review on eeg based epileptic seizures detection using deep learning techniques. In *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 966–973, 2022.

[12] Wen Li, Guangming Wang, Xiyuan Lei, Duozheng Sheng, Tao Yu, and Gang Wang. Seizure detection based on wearable devices: A review of device, mechanism, and algorithm. *Acta Neurologica Scandinavica*, 146(6):723–731, 2022.

[13] NeuroPace. Rns system patient manual, 2020.

[14] Roger Penrose. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1:221–244, 1971.

[15] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[16] Mark Fannes, Bruno Nachtergaele, and Reinhard F Werner. Finitely correlated states on quantum spin chains. *Communications in mathematical physics*, 144:443–490, 1992.

[17] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[18] Cong Chen, Kim Batselier, Wenjian Yu, and Ngai Wong. Kernelized support tensor train machines. *Pattern Recognition*, 122:108337, 2022.

[19] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, sep 1995.

[20] Johan Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural Processing Letters*, 9:293–300, 06 1999.

[21] John Platt. Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft, April 1998.

[22] Dacheng Tao, Xuelong Li, Weiming Hu, Stephen Maybank, and Xindong Wu. Supervised tensor learning. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[23] Zhifeng Hao, Lifang He, Bingqian Chen, and Xiaowei Yang. A linear support higher-order tensor machine for classification. *IEEE Transactions on Image Processing*, 22(7):2911–2920, 2013.

[24] Cong Chen, Kim Batselier, Ching-Yun Ko, and Ngai Wong. A support tensor train machine. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[25] Lifang He, Xiangnan Kong, Philip S Yu, Xiaowei Yang, Ann B Ragin, and Zhifeng Hao. Dusk: A dual structure-preserving kernel for supervised tensor learning with applications to neuroimages. In *Proceedings of the 2014 SIAM International Conference on Data Mining*, pages 127–135. SIAM, 2014.

[26] Lifang He, Chun-Ta Lu, Hao Ding, Shen Wang, Linlin Shen, Philip S Yu, and Ann B Ragin. Multi-way multi-level kernel modeling for neuroimaging classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 356–364, 2017.

[27] Kirandeep Kour, Sergey Dolgov, Martin Stoll, and Peter Benner. Efficient structure-preserving support tensor train machine. *Journal of Machine Learning Research*, 24(4):1–22, 2023.

[28] Ali Hossam Shoeb. *Application of machine learning to epileptic seizure onset detection and treatment.* PhD thesis, Massachusetts Institute of Technology, 2009.

[29] Miroslaw Latka, Ziemowit Was, Andrzej Kozik, and Bruce J West. Wavelet analysis of epileptic spikes. *Physical Review E*, 67(5):052902, 2003.

[30] Poomipat Boonyakitanont, Apiwat Lek-uthai, Krisnachai Chomtho, and Jitkomut Songsiri. A review of feature extraction and performance evaluation in epileptic seizure detection using eeg. *Biomedical Signal Processing and Control*, 57:101702, 2020.

[31] Catalina Gómez, Pablo Arbeláez, Miguel Navarrete, Catalina Alvarado-Rojas, Michel Le Van Quyen, and Mario Valderrama. Automatic seizure detection based on imaged-eeg signals through fully convolutional networks. *Scientific reports*, 10(1):21833, 2020.

Glossary

List of Acronyms

EEG	electroencephalogram
SVM	support vector machine
STM	Support Tensor Machine
STMs	Support Tensor Machines
LS-SVM	least-squares support vector machine
QP	quadratic programming
STuM	support tucker machine
CPD	Canonical Polyadic Decomposition
STTM	Support Tensor-Train Machine
SHTM	Support Higher-order Tensor Machine
K-STTM	kernelized support tensor train machine
TTD	Tensor-Train Decomposition
TT	Tensor-Train
ALS	alternating least-squares
PCA	principle component analysis
DWT	discrete wavelet transform
RNS	responsive neurostimulation
DBS	deep brain stimulation
VNS	vagus nerve stimulation
fMRI	functional magnetic resonance imaging
DuSK	Dual Structure-preserving Kernel
CP	Canonical Polyadic
MPCA	multilinear principal component analysis
CNN	convolutional neural network

CWT	Continuous Wavelet Transform
MMK	Multi-way Multi-level Kernel
LOPO	leave one patient out
LOSO	leave one seizure out
TT-MMK	Tensor Train Multi-way Multi-level Kernel
SVD	singular value decomposition
SVDs	singular value decompositions
FFT	Fast Fourier Transform
DWT	Discrete Wavelet Transform
SMO	Sequential Minimal Optimization
KKT	Karush-Kuhn-Tucker