Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Circuit Theory

Influence of artifacts in microelectrode recordings on the subcortical structure classification

Aneta Ambrosova

Supervisor: Ing. Eduard Bakštein, Ph.D.
Field of study: Medical electronics and bioinformatics
May 2022

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Ambrosová Aneta**	Personal ID number:	**492183**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Circuit Theory**		
Study program:	**Medical Electronics and Bioinformatics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Influence of artifacts in microelectrode recordings on the subcortical structure classification

Bachelor’s thesis title in Czech:

Vliv artefaktů v mikroelektrodových záznamech na klasifikaci subkortikálních struktur

Guidelines:

Deep brain stimulation is a treatment technique for neurological conditions such as the Parkinson's disease. During the surgery, high accuracy of microelectrode placement is necessary, for which manual analysis of microelectrode physiological recordings is typically used. Automatic classification of these nuclei from clinical data is affected by presence of frequent signal artifacts. The aim of the thesis is to find optimal signal preprocessing pipeline for the purpose of nuclei classification.
Tasks:
1 - Study available methods for signal preprocessing, artifact removal and nuclei classification from microelectrode recordings.
2 - Implement a set of classifiers for nuclei classification
3 - Find an optimal preprocessing pipeline for microelectrode recordings, with respect to classification accuracy of your classifiers, implemented in step 2.

Bibliography / sources:

[1] Moran, A., Bar-Gad, I., Bergman, H., & Israel, Z. (2006). Real-time refinement of subthalamic nucleus targeting using Bayesian decision-making on the root mean square measure. Mov Disord, 21(9), 1425–1431. https://doi.org/10.1002/mds.20995
[2] Bakštein, E. (2016) Deep Brain Recordings in Parkinson’s Disease: Processing, Analysis and Fusion with Anatomical Models, doctoral thesis, Czech Technical Univeristy in Prague
[3] Bakštein, E., Sieger, T., Wild, J., Novák, D., Schneider, J., Vostatek, P., Urgošík, D., & Jech, R. (2017). Methods for automatic detection of artifacts in microelectrode recordings. Journal of Neuroscience Methods, 290, 39–51. https://doi.org/http://dx.doi.org/10.1016/j.jneumeth.2017.07.012

Name and workplace of bachelor’s thesis supervisor:

Ing. Eduard Bakštein, Ph.D. Analysis and Interpretation of Biomedical Data FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **02.02.2022** Deadline for bachelor thesis submission: **20.05.2022**

Assignment valid until: **30.09.2023**

Ing. Eduard Bakštein, Ph.D. Supervisor’s signature	doc. Ing. Radoslav Bortel, Ph.D. Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce her thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt

Student’s signature

Acknowledgements

Firstly, I would like express my gratitude to my supervisor, Ing. Eduard Bakštein, Ph.D., who patently and kindly guided my throughout this thesis.

Furthermore, I would like to thank my family for their continuous love and support.

Declaration

I declare that this work is all my own work, and I have cited all sources I have used in the bibliography.

Prague, May , 2022

Abstract

Deep brain stimulation (DBS) is a neurochirurgical method consisting of continuous electrical stimulation to specific subcortical brain structures. Today, DBS is an established treatment for Parkinson's disease. The neurologist manually annotates the microelectrode recordings recorded during DBS surgery to identify the ideal position of the DBS electrode. Microelectrode recordings often contain unwanted artefacts that could significantly influence further signal processing.

This bachelor's thesis aims to implement and evaluate a set of classifiers for automatic brain nuclei detection in microelectrode recordings. The focus is on the influence of undesirable artifacts in microelectrode recordings and on the methods of their removal. The nuclei classifier could advise the surgical team intraoperatively and in real-time.

Firstly, classification by thresholding based on one of the microelectrode features, normalized RMS power, was evaluated. The classification was performed after two different methods removed artefacts from the signals. Proper artifact removal has an immense impact on the accuracy of classification. Secondly, the nucleus was classified using Random Forest (RF) with tuned hyperparameters and the Support Vector Machine (SVM) algorithm with a balanced training dataset. The normalized RMS thresholding achieved an accuracy of 90.71 %. In comparison, the RF and SVM algorithms achieved an accuracy of 90.89 % and 90.93 %, respectively.

Keywords: microelectrode recordings, classification, Parkinson's disease, deep brain stimulation, subthalamic nucleus

Supervisor: Ing. Eduard Bakštein, Ph.D.

Abstrakt

Hlboká mozgová stimulácia je neurochirurgická metóda, pozostávajúca z elektrickej stimulácie špecifickej subkortikálnej štruktúry. Je to jedna z metód liečby Parkinsonovej choroby. Počas operácie hlbokej mozgovej stimulácie neurológ manuálne anotuje nahrané mikroelektródové záznamy aby identifikoval ideálnu pozíciu stimulačnej elektródy. Tieto záznamy často obsahujú nežiaduce artefakty, ktoré môžu významne ovplyvniť následné spracovanie a vyhodnotenie signálov.

Táto bakalárska práca je zameraná na implementáciu a vyhodnotenie sady klasifikátorov pre automatické rozpoznávanie mozgových jadier v záznamoch. Cieľom práce je sledovať vplyv nežiaducich artefaktov a dopad ich odstránenia na ich na následné vyhodnotenie. Automatická klasifikácia mozgového jadra by mohla počas operácie uľahčiť lekárom manuálnu prácu.

Najskôr bola implementovaná klasifikácia mozgového jadra prahovaním založeným na jednom z príznakov mikroelektródových záznamov, normalizovanej energii RMS. Pred klasifikáciou boli zo záznamov odstránené artefakty. Bolo ukázané, že presné odstránenie artefaktov má veľký vplyv na presnosť klasifikácie. Následne bola implementovaná klasifikácia pomocou algoritmu Random forest s doladenými parametrami a algoritmu Support Vector Machine (SVM) so selekciou príznakov. Klasifikácia prahovaním dosiahla presnosť 90.71 %. Random forest dosiahol presnosť 90.89 % a SVM presnosť 90.93 %.

Klíčová slova: mikroelektródové záznamy, klasifikácia, Parkinsonova choroba, hlboká mozgová stimulácia, subthalamické jadro

Překlad názvu: Vplyv artefaktov v mikroelektródových záznamoch na klasifikáciu subkortikálnych štruktúr

Contents

List of abbreviations **1**

1 Introduction **3**

1.1 Problem description 3

1.2 Aims of the thesis 3

2 Medical background **5**

2.1 Parkinson's disease 5

2.2 Parkinson's disease treatment . . . 6

2.3 Deep brain stimulation 6

2.4 Microelectrode recordings 7

2.5 Artifacts in microelectrode recordings 8

3 Methods **11**

3.1 Classifier 11

3.1.1 Random forest 12

3.1.2 Random forest hyperparameters 14

3.1.3 Support Vector Machine 17

3.1.4 Features 19

3.1.5 Feature selection 22

3.2 Artifact detection 22

3.3 Criteria for classification comparison 25

3.4 Structure of the suggested solution 27

4 Results **29**

4.1 Data description 29

4.2 Artifact detection 30

4.3 Artifact detection methods and their impact on STN detection by NRMS thresholding 35

4.3.1 NRMS thresholding by ROC curves 35

4.3.2 NRMS thresholding by the best threshold 40

4.3.3 STN detection compared to artifact detection 41

4.4 Random forest STN classification 42

4.4.1 Random forest hyperparameters 42

4.4.2 The settings of artifact detection methods and their impact on RF classification 44

4.5 Support Vector Machine STN classification 45

4.5.1 The settings of artifact detection methods and their impact on SVM classification 46

4.6 Comparison of STN classifiers . . 47

4.6.1 Comparison of NRMS thresholding, RF and SVM classifier 47

5 Conclusion **49**

Bibliography **51**

A Technical details **57**

Figures

2.1 MER signal with no artifacts and its power spectral density estimation 8
2.2 MER signal with artifacts and its power spectral density estimation . . 9

3.1 Decision tree 13
3.2 Random forest 13
3.3 ROC curve of STN classification by NRMS thresholding 26

4.1 Artifact removal by maxDiffPSD method with different winLen and threshold settings 31
4.2 Artifact removal by COV method with different winLen and threshold settings 32
4.3 Artifact detection heatmaps for maxDiffPSD method 33
4.4 Heatmaps of lost data for maxDiffPSD method 34
4.5 Artifact detection heatmaps for COV method 34
4.6 Heatmaps of lost data for COV method 34
4.7 AUC heatmap for maxDiffPSD STN classification 36
4.8 Youden-J heatmap for maxDiffPSD STN classification 36
4.9 Fraction of data kept for maxDiffPSD STN classification . . . 37
4.10 Fraction of non-zero signals for maxDiffPSD STN classification . . . 37
4.11 AUC heatmap for COV STN classification 38
4.12 Youden-J heatmap for COV STN classification 38
4.13 Fraction of data kept for COV STN classification 39
4.14 Fraction of non-zero signals for COV STN classification 39
4.15 Accuracy heatmap for maxDiffPSD STN thresholding . . . 40
4.16 Accuracy heatmap for COV STN thresholding 41

Tables

3.1 Table of RF hyperparameters and their default values 14
3.2 Confusion matrix 25

4.1 Table of artifact classification performance for different methods and parameters 33
4.2 Table of STN classification performance for different methods and parameters 35
4.3 RF evaluation with different number of trees after maxDiffPSD removal 43
4.4 RF evaluation with different number of trees after COV removal 43
4.5 RF evaluation with different hyperparameters after maxDiffPSD removal 43
4.6 RF evaluation with different hyperparameters after COV removal 44
4.7 Impact of maxDiffPSD STN detection parameters on RF classification 45
4.8 Impact of COV STN detection parameters on RF classification . . . 45
4.9 RF STN classification on unseen data 45
4.10 Impact of maxDiffPSD STN detection parameters on SVM classification 46
4.11 Impact of COV STN detection parameters on SVM classification . 46
4.12 SVM STN classification on unseen data 46
4.13 Comparison of NRMS thresholding, RF and SVM classifier 47

List of abbreviations

DBS	Deep brain stimulation
MER	Microelectrode recording
PD	Parkinson’s disease
STN	Subthalamic nucleus
Th	Thalamus
SNr	Substantia nigra
GPi	Globus Pallidus
OCD	Obsessive-compulsive disorder
MRI	Magnetic resonance imagining
SVM	Support vector machine
RF	Random forest
RMS	Root mean square
NRMS	Normalized Root mean square
PSD	Power spectral density
ANE	Average nonlinear energy
CL	Curve length
ZC	Zero crossing
ROC	Receiver operatin characteristic curve
AUC	Area under the curve
TP	True positive
FP	False positive
FN	False negative
TN	True negative
TPR	True positive rate
FNR	False negative rate
Sen	Sensitivity
Spe	Specificity
J	Youden’s J statistics
Acc	Accuracy
Train Err	Training error
Test Err	Test error

th	Threshold
winLen	Window length
maxDiffPSD	Maximal spectral difference method
COV	Autocorrelation method
NumTrees	Number of trees
InBagFraction	In bag fraction (Sample size)
SampleWithReplacement	Sampling with replacement
NumPredictorsToSample	Number of predictors to sample
MinLeafSize	Minimum number of observations
MaxNumSplits	Maximal number of decision splits
MergeLeaves	Merging the leaves
PredictorSelection	Selection of predictors
PruneCriterion	Prunning criterion
SplitCriterion	Splitting criterion
NumPredictors	Number of predictors
NumObservations	Number of observations

Chapter 1
Introduction

Deep brain stimulation (DBS) is a method used in neurosurgery consisting of delivering continuous electrical stimulation to a specific brain structure. Stimulation is delivered through electrodes implanted during surgery and connected to a neuropacemaker or a stimulator. Nowadays, DBS is an established treatment for chronic movement disorders such as Parkinson's disease. Significant movement symptoms of Parkinson's disease, such as tremors, stiffness (rigidity) and slowness of movement (bradykinesia), are treated by targeting the subthalamic nucleus with the stimulation. The specialist identifies the ideal position during surgery by recording and manually annotating the extracellular microelectrode recording (MER). Optimal results can be obtained by precisely identifying the exact location of the target nucleus.

1.1 Problem description

MER signals commonly contain undesirable artifacts, and their optimal detection plays a vital role as they may influence further signal processing. They might affect over 25 % of the recording length. One of the primary sources of artifacts is the electromagnetic interference from electrical appliances in the operating room. This thesis aims to analyse microelectrode recordings and develop and validate a classifier for automatically detecting nuclei in the microelectrode recordings of DBS targeted at patients with Parkinson's disease. Before the nuclei classification takes place, the aritfacts are removed from the dataset to ensure better classifier performance. The classifier could advise the surgical team in real-time and help determine the electrode's position. As the manual evaluation of MER is prone to human error and can take a long time, this could greatly serve the surgical team.

1.2 Aims of the thesis

To classify STN successfully, artifacts should be removed from the dataset. The first aim of this thesis is to study available methods for signal preprocessing, artifact removal and STN classification from MER. A part of this thesis focuses on two different artifact removal methods, those methods are evaluated using different method settings, and consequently, the best artifact removal approach should be described.

Another task in this thesis is to implement and evaluate a set of STN classifiers on artifact-free data, and the best classifier shall be derived. The algorithms chosen for this task are Random Forest and Support Vector Machine. Lastly, an optimal preprocessing pipeline for MER with respect to classification accuracy should be defined.

Chapter 2

Medical background

This chapter is focused on the medical background of Parkinson's disease and serves as an introduction to the problems discussed in this thesis. The thesis aims to classify a subthalamic nucleus from microelectrode recordings gathered during deep brain stimulation operation. Deep brain stimulation (2.3) is a method of treatment for Parkinson's disease, which is described in Section 2.1. Other treatment methods are described in 2.2. A detailed description of microelectrode recordings is in Section 2.4. Artifacts found in MER, what causes them and what problems they cause are described in Section 2.5.

2.1 Parkinson's disease

Parkinson's disease (PD) is a progressive neurological disorder characterised by a great number of the motor and non-motor symptoms. It is named after James Parkinson, the London doctor who first reported the symptoms in 1817. PD has a significant clinical impact on patients and affects their quality of life (Karthick et al. 2020). When the disease progresses to its advanced stage, the complications and symptoms are devastating.

According to Feigin (2017) prevalence of Parkinson's is approximately 6.2 million people worldwide. Most people who get PD are over 60, but 10 % are under 50. Research revealed that the disease develops from the interplay of environment and genetics. Slightly more men than women are affected by the disease. Established environmental risk factors are exposure to pesticides, prior head injury, and rural living just as well-water drinking. Factors which are associated with decreased risk are tobacco smoking, coffee drinking, anti-inflammatory drug usage (aspirin, ibuprofen and naproxen) and alcohol consumption (Noyce et al. 2012). However, smoking does not protect against PD. A case-control study performed by Ritz (2014) showed that PD patients could quit smoking more easily than controls, which is compatible with a decreased responsiveness to nicotine during the prodromal phase of PD.

There is no definitive test for diagnostics, and that is why PD must be diagnosed based on clinical criteria. Significant motor symptoms include essential tremor, the most common and easily recognised symptom. Tremors are unilateral and occur at a frequency between 4 and 6 Hz. Hand tremors are described as supination–pronation ("pill-rolling") tremors that spread from one hand to the other (Jankovic 2008). Other characteristic motor symptoms mentioned by Jankovic (2008) are bradykinesia (slowness of movement), rigidity (increased resistance) and loss of postural reflexes, and he also mentioned non-motor symptoms like autonomic dysfunction, cognitive/neurobehavioral abnormalities, sleep disorders and sensory abnormalities such as anosmia, paresthesias and pain. Non-motor symptoms are a common feature of PD and are often underappreciated.

The disease is mainly caused by progressive degradation of cells producing dopamine in the substantia nigra (SNr) - structure located in the mesencephalon. Dopamine is a neurotransmitter playing a significant role in the control of motor functions. A consequence of loss of dopaminergic function is a lack of dopamine and an imbalance between direct (facilitatory) and indirect (inhibitory) pathways through the basal ganglia - several large clusters of neurons located within the cerebral hemisphere and the upper brainstem. The result is decreased motor activity characteristic of PD. Research also supports the evidence that PD involves a global network dysfunction at multiple levels in the nervous system (Scarmeas 2001; Bloem, Okun, and Klein 2021). As mentioned in a paper by Tolosa (2006), Parkinson's disease is not considered a single disease entity, and the term may be used with different meanings. Parkinsonism is a term used to describe a group of neurological problems, and its causes are numerous. PD is differentiated from other related parkinsonism diseases by its features' presence and specific presentation. Today, the actual cause of PD is still unknown, and there is no recognised cure for it.

2.2 Parkinson's disease treatment

The therapeutic methods are aimed at alleviation of the PD symptoms. The PD medication helps control the symptoms in the early stage of the disease. The most potent medication for PD is levodopa - a dopamine precursor transformed to dopamine in the body. However, the effectiveness of these drugs declines in the advanced stage (Groiss et al. 2009). Apart from therapeutic interventions under active research, such as gene therapy and cell transplants, the DBS is the only widely used surgical intervention for PD as of today. DBS to the subthalamic nucleus (STN) is proven to improve the quality of life in patients suffering from advanced PD (Karthick et al. 2020).

2.3 Deep brain stimulation

Deep brain stimulation (DBS) has been used to treat hundreds-of-thousands patients worldwide suffering from PD and essential tremors. It has been approved for the treatment of dystonia and is currently undergoing clinical trials as a treatment for depression and obsessive-compulsive disorder (OCD). DBS is a source of a great deal of hope for patients with PD, possibly providing a new life to those suffering from it (Gardner 2013).

The DBS consists of delivering continuous electrical stimulation to neural brain structure through surgically implanted electrodes. The success of DBS depends on the selection of appropriate candidate patients and on the accurate targeting of target nuclei. Optimal pretargeting using methods like Magnetic resonance imagining (MRI), Ventriculography, Electrophysiological exploration done by microelectrodes, and Intraoperative stimulation reduces the number of passes, risk of bleeding and the duration of surgery (A. L. Benabid 2003). Implantation of electrodes to identified target is usually done under local anaesthesia. When the best track has been identified by assessing clinical response to DBS by a skilled neurologist, the microelectrode is removed and replaced by a chronic lead. The lead is then fixed to the skull by, e.g., plate and screw, plug or dental cement (A. L. Benabid et al. 2009). Stimulation electrodes are then connected to a stimulation device, which closely resembles a heart pacemaker.

The choice of target depends on disease type, and its symptoms, different targets of stimulation provide different clinical effects. It is a general trend that treating the STN improves all symptoms of PD, making it the best target for stimulation. Another commonly targeted structure is Globus Pallidus (GPi), which specifically improves dyskinesias (A. L. Benabid 2003). The pulse generator settings such as frequency, pulse width and amplitude can be adjusted post-operatively and are optimally tuned for each patient. Optimal clinical results were obtained by stimulus frequency 120-180 Hz, pulse width 60-200 μs and 1-5 V stimulation amplitude (Breit, Schulz, and A.-L. Benabid 2004). The best candidate patients are those with a short beneficial duration from levodopa doses and patients who have considerable motor benefit from oral medication (Perlmutter and Mink 2006).

2.4 Microelectrode recordings

The neurophysiological technique of microelectrode recordings (MER, μEEG, microEEG) of very fine neuronal activity is used to refine target position during the surgery for accurate DBS electrode placement. The electrode impedance determines its voltage response to current flow during stimulation. If the surface area of the electrode is larger, then the impedance value it will measure in the physiological frequency range is lower. Microelectrode's tip diameter is typically 2-4 μm and has impedance larger than 0.5 MOhm. Such dimensions are comparable to dimensions of neurons and, therefore, can isolate single neural activity and define the properties of individual neurons in the closest proximity of the microelectrode. The background noise of neuronal activity in areas more distant from the electrode is also present in the recordings (Gross et al. 2006). Typical sampling frequencies for MER are about 20 kHz.

MER is often carried out prior to implantation of the stimulation electrode to obtain more accurate information on the target's location. A set of electrodes (commonly 5) is shifted into the brain using a microdrive. Neuronal activity is recorded at every position, and based on those recordings, a qualified neurologist evaluates the signals and identifies the margins of the target structure and the optimal position of the stimulation electrode. This process is usually based on visual and auditory examination of the signals. Primary markers are neuronal firing pattern and amplitude of neuronal background - both are higher in areas such as STN with high neuronal density (Bakstein, Sieger, et al. 2016). MER signals commonly contain different types of artifacts.

2.5 Artifacts in microelectrode recordings

Recordings obtained using microelectrodes with tip size around one μm may contain unwanted signal artifacts. Two examples of MER signals without and with artifacts can be found in Figures 2.1, 2.2. Electrode susceptibility to mechanical shifts and electromagnetic interference is caused by its small size and low source signal voltage. Great care must be taken to distinguish artifacts to be able to interpret MER reliably (Stacey et al. 2013). Their presence might have a dramatic impact on further signal processing. The main source of artifacts in MER is the electromagnetic interference from the electrical appliances in the operating room, and it can cause extensive damage to the MER signals. Other sources are DBS electrode vibration and patients' or doctors' movement or speech. Therefore artifacts are commonly found in DBS MER data, and they may affect more than 25 % of the recording length. Details on existing methods of artifact detection are described in Section 3.2.

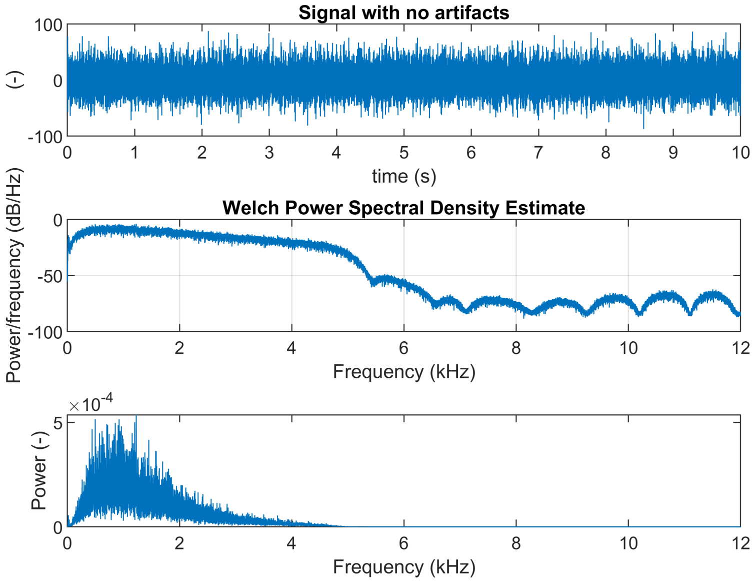


Figure 2.1: MER signal with no artifacts and its power spectral density estimation

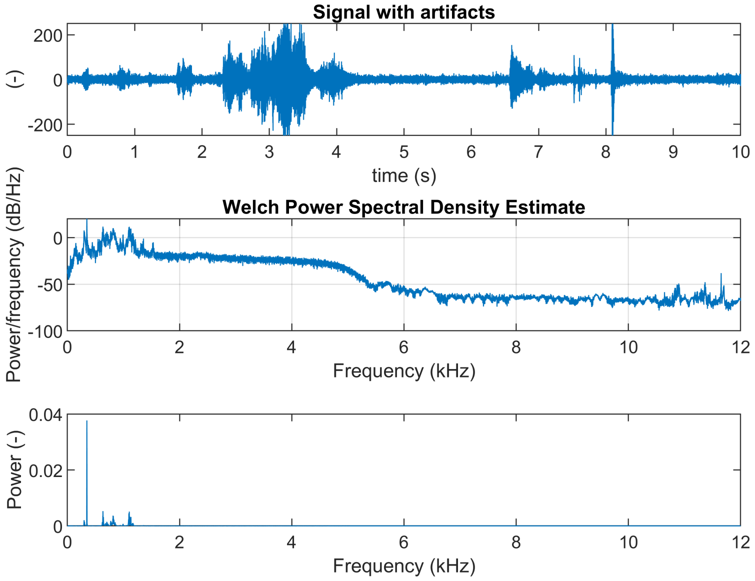


Figure 2.2: MER signal with artifacts and its power spectral density estimation

Chapter 3

Methods

This thesis aims to evaluate a set of STN classifiers. STN is classified using features computed from MER signals. Artifact detection plays a vital role in this task, as undesirable artifacts might impact the classification results.

This chapter deals with different methods which were used in this thesis. Section 3.1 contains a brief description of machine learning and methods of classification. Subsection 3.1.1 describes one of the machine learning algorithms - Random forest, and Section 3.1.3 is dedicated to the Support Vector Machine algorithm. Subsection 3.1.4 describes what is a feature and presents different MER features and how to compute them, followed by Section 3.1.5 about feature selection.

Section 3.2 discusses different methods for detection of artifacts in MER. Section 3.3 presents criteria which were used to compare different classification methods and Section 3.4 suggests a solution for the thesis problems which were already presented in the Problem description (1.1)

3.1 Classifier

Machine learning is a part of artificial intelligence that studies computer algorithms that enable systems to learn and improve automatically through experience. Machine learning distinguishes between two phases: training phase (also known as learning phase) based on training data and the testing phase, which is performed on the test data after training the model. The goal of the training phase is to optimize the training parameters. During the test phase, the learning model is tested with data set that has not been seen before, and its performance is evaluated. The ability of a system to classify previously unseen samples is called generalization.

Learning takes place if performance is improved after making observations about the world. According to Russel (2012), there are three types of learning: unsupervised, reinforcement and supervised. In unsupervised learning, there is no teacher and algorithms (for instance, K-means and Principal component analysis) typically form potentially useful clusters. Another type of learning is reinforcement learning, when the agent learns from a series of reinforcements. Those could be rewards or punishments, and examples of such algorithms are Q-learning and SARSA. The last group of algorithms is called supervised learning. A teacher is present and provides output to observed input, and the agent learns an input to an output mapping function. Perceptron, Support vector machine (SVM) and Random forest represent this group. This thesis is based on supervised learning algorithms, as there are input data and output-category labels annotated by neurologists.

One sub-problem a pattern classification task deals with is choosing the suitable feature extraction. It is crucial to find features that are useful, fast to compute and discriminating (they preserve useful information, enabling differentiation) to build an effective algorithm. Calculated features are then used as their input. The feature extraction is a part of data pre-processing. If important information is thrown away during pre-processing, the overall accuracy of the system can suffer (Bishop 2006). Features are usually numeric, often forming an n-dimensional feature vector.

Bishop (2006) also mentions other sub-problems, such as noise handling, overfitting and model selection. Noise is a non-fundamental property of pattern caused by randomness in the outside world or sensors. Overfitting is a situation when a complex model classifies specific samples perfectly but performs poorly on different data.

3.1.1 Random forest

A random forest (RF), first introduced by Breiman (2001), is a supervised algorithm that uses multiple decision trees which operate as an ensemble. A binary decision tree is a structure based on a sequential decision process made by asking a series of questions. A decision tree is considered relatively fast and more intuitive than other algorithms. One of the reasons is that instead of observing weights, one can look at the questions asked.

The decision tree has a flowchart structure in which every node represents a question; each branch represents an outcome of this question when evaluated, and each leaf node represents a class label. Figure 3.1 illustrates a single decision tree. Growing a tree involves deciding what conditions to use for splitting and in what order. The given dataset is evaluated and split using the chosen criterion function. Three different criterion functions are described in 4.4.1. Choosing the best split is crucial because it determines the structure of a decision tree and its performance. Our goal is to reduce the number of splits needed, making the decision path shorter. After the tree is grown, the classification takes place. The classification process starts at the root, where a feature is evaluated, and one of the subsequent branches is selected. This procedure is repeated until the final leaf is reached (Bonaccorso 2017).

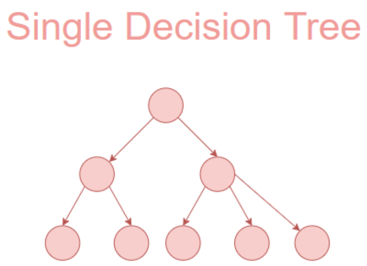


Figure 3.1: Decision tree

To build the forest, the training set is divided into smaller sets. One of the algorithms used to divide the training set is Bootstrap aggregating (or bagging). Training set cases are sampled by random with replacement. This sample will be the training dataset for growing one of the individual decision trees. Bagging improves the stability of the algorithm and helps to avoid overfitting. Also, a random subset of variables (features) is used for each tree. Each tree is grown to the largest possible extent. To classify a new object, the input vector is put down each of the trees in the forest. This results in different predictions by different trees, some weaker and some stronger than others. The most common approach to interpreting these results is to consider the majority vote correct (Breiman and Cutler 2004). Figure 3.2 contains a visualization of the RF algorithm consisting of single decision trees. Random Forest ensembles are a divide-and-conquer approach based on the idea that a group of weak learners be combined and form a strong learner.

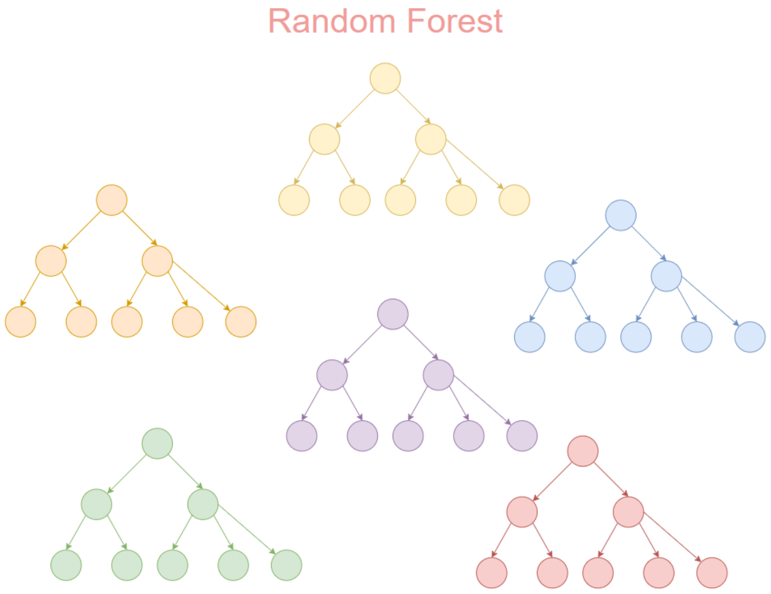


Figure 3.2: Random forest

3.1.2 Random forest hyperparameters

A set of hyperparameters controls the structure of the forest and each tree in the RF algorithm. Default values of hyperparameters were derived from the TreeBagger function (MATLAB 2020a). Table 3.1 summarizes those hyperparameters and is followed by more detailed description.

Hyperparameter	**Default value**
InBagFraction	1
Cost	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
SampleWithReplacement	on
NumPredictorsToSample	$\sqrt{NumPredictors}$
MinLeafSize	1
Prior	empirical
MaxNumSplits	$NumObservations - 1$
MergeLeaves	on
PredictorSelection	allsplits
PruneCriterion	error
Surrogate	off

Table 3.1: Table of RF hyperparameters and their default values

NumTrees

The first hyperparameter is the number of trees in a random forest (NumTrees). Theoretically, more trees are always better. However, according to Probst and Boulesteix (2017), the most significant performance improvement can be achieved when growing the first 100 trees. The number of trees needed to obtain optimal performance is dependent on the properties of the dataset and may also depend on hyperparameters and the computational time increases linearly with an increasing number of trees (Probst, Wright, and A.-L. Boulesteix 2019).

InBagFraction

The second hyperparameter to consider is the sample size (InBagFraction). It determines the fraction of input data to sample with replacement from the input data for training of each tree. When the sample size decreases, the trees are more diverse, and their correlation is lower. This results in better prediction accuracy when aggregating the trees, but the single tree accuracy declines since fever observations are used for training (Probst, Wright, and A.-L. Boulesteix 2019). The default value at the TreeBagger function is set to 1.

Cost

Next hyperparameter is the cost function (Cost), defined as a square matrix C, where $C(i, j)$ is the cost of classifying a point into class j if its true class is i. The default value is $C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

SampleWithReplacement

Another hyperparameter is the setting of replacement (SampleWithReplacement). The TreeBagger function samples with replacement by default.

NumPredictorsToSample

The number of randomly drawn candidate variables (NumPredictorsToSample) is the number of random variables (predictors, features) to consider at each split point. When the value of candidate variables is lower, the trees are more different and less correlated and yield better stability when aggregated. However, the trees tend to perform worse on average, as they are built based on suboptimal variables (Probst, Wright, and A.-L. Boulesteix 2019). This means a trade-off between stability and accuracy of single trees has to be made. The default value at TreeBagger classification is $\sqrt{numPredictors}$, where $numPredictors$ is the number of predictor variables.

MinLeafSize

The minimum leaf size (MinLeafSize) is the next parameter, describing the minimum number of observations per tree lead. In the mentioned function, it is set to 1 by default.

Prior

The prior probability (Prior) is set to empirical by default, which means the class probabilities are determined from class frequencies in the training data labels. If set to uniform, the class probabilities are set to equal.

MaxNumSplits

Maximal number of decision splits (MaxNumSplits), also called branch nodes, is by default set to $NumObservations - 1$, where $NumObservations$ is the number of observations.

MergeLeaves

Merging the leaves (MergeLeaves) that come from from the same parent node and that yields a sum of risk values larger or equal to the risk associated with the parent node is set to on by default.

PredictorSelection

A further important parameter is the choice of the algorithm used to select the best split predictor at each tree node (PredictorSelection).

It can be set to Standard CART (A Classification And Regression Tree), which chooses the split predictor that maximizes the split-criterion gain over all viable splits of all predictors (Breiman, Friedman, et al. 1984).

Or a curvature test can be used. It is a statistical test determining whether two variables are unassociated. It chooses the split predictor, which minimizes the significant P-value of the chi-square independence tests (curvature tests) between each predictor and the response (Sinica et al. 1997).

Another option is to use the interaction test. It is a statistical test determining whether there is no interaction between a pair of predictor variables and the response variable. It selects the split predictor that minimizes the P-value of the chi-square independence tests between each predictor and the response and, at the same time, minimizes the P-value of the chi-square independence tests between each pair of predictors and response (Loh 2002).

PruneCriterion

Two criteria can be used to prune the subtrees (PruneCriterion). The first criterion is the Node Error, the fraction of misclassified classes at a node. The node error is $1 - p(j)$ if j is the class with the largest number of training samples at a node. The second criterion is the impurity measure, which depends on the choice of the SplitCriterion.

SplitCriterion

The splitting rule (SplitCriterion) is the next hyperparameter. MATLAB's function offers three criteria: Gini's diversity index, maximum deviance reduction (also known as cross-entropy) and the twoing rule.

In this case, the Gini impurity index is the default. This is a cost function which gives an idea of how good a split is depending on how mixed the classes are in the two groups created by the split. Put in other words, it is a measure of node impurity. It can be calculated as

$$1 - \sum_i p^2(i) \tag{3.1}$$

where the sum is through the classes i at the node and $p(i)$ is the fraction of classes with class i that reach the specific node. Given a sample, it measures the probability of misclassification if a label is chosen randomly using the probability distribution of the branch. Its values range from 0 to 0.5. A perfect split would have a Gini index of 0, while the worst-case split would result in a Gini index of 0.5 (Bonaccorso 2017).

Deviance (also known as cross-entropy) is based on information theory. It can be calculated as

$$-\sum_i p(i) \log_2 p(i) \tag{3.2}$$

where $p(i)$ is defined the same as for the Gini impurity index. A pure node has null deviance, and deviance is maximum when the classes are uniformly distributed (which means there are still many steps until the final classification).

Unlike the previous functions, twoing rule does not measure node purity. Let $L(i)$ be the fraction of members of class i in the left child node after a split and $R(i)$ be the fraction of members of class i in the right child node after a split. Let P(L) be the fraction of the observations that split to the left and P(R) be the fraction of the observations that split to the right. Compute

$$P(L)P(R)(\sum_i |L(i) - R(i)|)^2 \tag{3.3}$$

and select the split criterion, which maximizes the expression result. If it is large, each child node is purer after the split.

Surrogate

The last hyperparameter considered in this thesis is the surrogate decision splits flag (Surrogate). The parameter is set to off by default. MATLAB function fitctree finds no more than ten surrogate splits at each branch node when the parameter is set to on. This can use considerable time and memory. If the parameter is set to a positive integer, the fitctree function finds no more than the specified number of surrogate splits. This can improve the accuracy of predictions for data with values missing.

3.1.3 Support Vector Machine

Support-vector machine (SVM) is a machine learning model developed by Cortes and Vapnik (1995). The objective of this classifier is to find a hyperplane in N-dimensional space that distinctly classifies given data points.

It is based on a linear classifier, which classifies data into two classes as:

$$\hat{y} = sign(\boldsymbol{w} \cdot \boldsymbol{x} + b) \tag{3.4}$$

where a set of labeled training patterns $(y_1, \boldsymbol{x}_1), \ldots, (y_N, \boldsymbol{x}_N)$ is given and $y_i \in \{-1, 1\}, i = 1 \ldots N$. Many possible hyperplanes could be selected to separate data points into two classes. The objective of SVM is to find a hyperplane that has the maximal margin (the distance between data points from different classes). Data is linearly separable if there exists a vector $\boldsymbol{w}$ and a scalar b such that all elements in the training set meet the condition

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1, \quad i = 1, \ldots, N \tag{3.5}$$

SVM finds the optimal hyperplane

$$\boldsymbol{w_0} \cdot \boldsymbol{x} + b_0 = 0 \tag{3.6}$$

which determines the direction $\frac{\boldsymbol{w}}{|\boldsymbol{w}|}$ which maximizes the distance between projections of the training vectors. The maximal distance is $\frac{2}{||\boldsymbol{w}||}$. The maximal margin optimization problem can be formulated as:

$$(\boldsymbol{w}^*, b^*) = \underset{\mathbf{w},b}{\operatorname{argmin}} \frac{1}{2}||\boldsymbol{w}||^2 \tag{3.7}$$

$$\text{subject to: } y_i(\boldsymbol{w} \cdot \boldsymbol{x} + b) \geq 1, \quad \forall i = \{1, \ldots, N\} \tag{3.8}$$

Support vectors are vectors $\boldsymbol{x_i}$ closest to the hyperplane. The problem of finding an optimal hyperplane can also be expressed as

$$L = \underset{\alpha}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x_i} \cdot \boldsymbol{x_j} \right\} \tag{3.9}$$

$$\text{subject to} \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad \forall i \in \{1, 2, \ldots, N\} \tag{3.10}$$

using Lagrangian (α_i's are non-negative multipliers), where N is the number of training data. This equation can be rewritten as

$$L = \underset{\alpha}{\operatorname{argmax}} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \right\} \tag{3.11}$$

$$\text{subject to} \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad \forall i \in \{1, 2, \ldots, N\} \tag{3.12}$$

$K(\boldsymbol{x_i}, \boldsymbol{x_j})$ is the so-called kernel function, which maps the original non-linear observations into a higher dimensional space to separate the data points (also called the dimension lifting). With an appropriate nonlinear mapping, data from two categories can always be separated by a hyperplane (Duda, Hart, and Stork 2012). Every semi-positive definite symmetric function can be a kernel. Commonly used kernel functions are:

Linear kernel

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = \boldsymbol{x_i} \cdot \boldsymbol{x_j} \tag{3.13}$$

Polynomial kernel of power p

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (1 + \boldsymbol{x_i} \cdot \boldsymbol{x_j})^p \tag{3.14}$$

Gaussian kernel, or Radial Basis Function (RBF) where σ is a free parameter

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{-||\boldsymbol{x_i} - \boldsymbol{x_j}||^2/2\sigma^2} \tag{3.15}$$

Two-layer perceptron

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = tahn(\alpha \boldsymbol{x_i} \cdot \boldsymbol{x_j} + \beta) \tag{3.16}$$

To solve the equation 3.11, we set the following:

$$\frac{\partial L}{\partial \boldsymbol{w}} = 0, \quad \frac{\partial L}{\partial b} = 0 \tag{3.17}$$

and it can be derived that

$$\boldsymbol{w} = \sum_{i \in S} a_i y_i \boldsymbol{x_i} \tag{3.18}$$

$$b = \frac{1}{N_S} \sum_{i \in S} \left(a_i - \sum_{j \in S} a_j y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \right) \tag{3.19}$$

where S is a set of N_S support vectors.

Sometimes, the training data can not be separated into different classes without error. In this case, we might want to separate the data with a minimal number of errors. That is when slack variables are introduced. This model is called Soft Margin SVM. The previous model without slack variables is also called Hard Margin SVM.

The new minimization task has a form

$$(\boldsymbol{w}^*, b^*) = \underset{\mathbf{w}, b}{\text{argmin}} \frac{1}{2} ||\boldsymbol{w}||^2 + C \sum_{i=1}^{N} \xi_i \tag{3.20}$$

$$\text{subject to: } y_i(\boldsymbol{w} \cdot \boldsymbol{x_i} + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = \{1, \ldots, N\} \tag{3.21}$$

where no penalty is paid if the point $(\boldsymbol{x}, y)$ fulfills the condition $y(\boldsymbol{w} \cdot \boldsymbol{x} + b) \geq 1$ and penalty $C \cdot \xi$ is paid if the condition is relaxed to $y(\boldsymbol{w} \cdot \boldsymbol{x} + b) \geq 1 - \xi$.

This problem can be also expressed using Lagrangian

$$L = \underset{a}{\text{argmax}} \left\{ \sum_{i=1}^{N} a_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \right\} \tag{3.22}$$

$$\text{subject to} \sum_i a_i y_i = 0 \tag{3.23}$$

$$0 \leq a_i \leq C, \quad \forall i \in \{1, 2, \ldots, N\} \tag{3.24}$$

In this thesis, the RBF kernel function was used. MATLAB's function fitcsvm (MATLAB 2020c) was used to implement the algorithm.

3.1.4 Features

STN is the main target of PD treatment. The features used for STN classification are computed from MER recordings. The typical recording length is 10 s, and the sampling frequency is 24 kHz. The density of neurons is much higher in STN. This can be observed as the amplitude of the MER signal is much higher once the electrode is near STN. More STN classification models have already been implemented, and most of them embrace this attribute of MER signals.

A paper by Moran (2006) describes a procedure for STN classification that could advise the surgical team intraoperatively and in real-time. They used normalized Root mean square (NRMS) power calculated from MER to implement a Bayesian classifier. A paper by Zaidel et al. (2009) used Hidden Markov Model based on observed NRMS and Power spectral density (PSD) to detect STN in MER. A paper by Guillén et al. (2011) characterized MER by Support Vector Machines (SVM) classifier. The features they used in the paper were Curve length (CL), Threshold, Number of peaks (Peaks), Root mean square (RMS), Average nonlinear energy (ANE) and Zero crossings (ZC). The description of features used in this thesis follows.

RMS (Root Mean Square) power of signal

The RMS value of a signal can be calculated as the square root of the arithmetic mean of the squares of each signal value (Guillén et al. 2011). It was calculated as :

$$RMS(X) = \sqrt{\frac{\sum_{i=1}^{N} x_i^2}{N}} \tag{3.25}$$

where N is the number of samples in the signal.

NRMS (Normalized Root Mean Square) power of signal

The NRMS value was calculated by dividing each RMS power in one trajectory by the mean RMS of the first five electrode positions recorded in the same trajectory. This was computed as

$$NRMS(X) = \frac{RMS(X)}{\frac{1}{5}\sum\limits_{i \in F5} RMS(i)} \tag{3.26}$$

where $F5$ are the first five electrodes in the same trajectory as X. This is possible since all recordings start 10 mm before STN, so we can safely assume that the first five electrode positions (depths) are outside the STN. This normalization will secure comparability of power values throughout all electrodes and all patients (Moran et al. 2006).

Power Spectral Density (PSD)

Power spectral density (PSD) measures have also been used for STN classification. Different parts of the spectrum might be informative for STN detection. PSD of the signal was estimated using Welch's method. Signal was divided into segments with window length 1 s and 50 % overlapping windows. The number of discrete Fourier transform points was 24000, equal to the window length. Afterwards, in order to quantify the oscillatory patterns, spectrum was divided into three frequency bands. Band α (also referred to as low band) with frequencies $3 - 12Hz$, band β with frequencies $13 - 30Hz$ and band γ with frequencies $31 - 100Hz$. All indices were computed as a ratio between the average power in the frequency band and the average power in the spectrum (Cagnan et al. 2011).

Average Nonlinear Energy (ANE)

Average Nonlinear Energy was calculated as :

$$ANE = \frac{1}{N-2}\sum_{i=2}^{N-1} x_i^2 - x_{i-1}x_{x+1} \tag{3.27}$$

where $x_i \in \boldsymbol{X} = x_1, x_2, \ldots, x_N$ and N is the number of samples in the signal (Wong et al. 2009).

Curve Length (CL)

Curve length was calculated as:

$$CL = \sum_{i=1}^{N-1} |x_{i+1} - x_i| \tag{3.28}$$

where $x_i \in \boldsymbol{X} = x_1, x_2, \ldots, x_N$ and N is the number of samples in the signal (Wong et al. 2009).

Zero Crossing (ZC)

Zero crossing was calculated as:

$$ZC = \frac{1}{2}\sum_{i=1}^{N-1} sgn(x_{i+1}) - sgn(x_i) \tag{3.29}$$

where $x_i \in \boldsymbol{X} = x_1, x_2, \ldots, x_N$ and N is the number of samples in the signal (Wong et al. 2009).

Peaks

The number of peaks was computed as:

$$Peaks = \frac{1}{2}\sum_{i=1}^{N-2} max\{0, |sgn[x_{i+1} - x_i] - sgn[x_{i+2} - x_{i+1}|\} \tag{3.30}$$

where $x_i \in \boldsymbol{X} = x_1, x_2, \ldots, x_N$ and N is the number of samples in the signal (Guillén et al. 2011).

Values of CL, ZC and Peaks were normalized (divided by the number of samples in the signal) because some signals may have a different length than others.

First STN classification method used is called NRMS thresholding. This method uses NRMS power derived by normalizing RMS power. RF and SVM were evaluated using feature selection of RMS, NRMS, PSD, ANE, CL, ZC and Peaks. All features were obtained after the artifacts were automatically removed from the MER by either maxDiffPSD or COV method.

3.1.5 Feature selection

The complexity of classification depends on a number of features d, which form the d-dimensional space. Feature selection methods are intended to reduce the number of features in the dataset, mainly to reduce the computational complexity. Unnecessary features may be deleted from the dataset, and the number of features may be reduced to a sufficient minimum. Reducing the number of features can reduce the computational cost and, in some cases, improve the performance of a model.

Two main dimension reduction methods are feature selection and feature extraction. Feature selection consists of selecting the subset of features which provide the most information from the original feature set. Feature extraction finds a new set of features which are a combination of the original feature set (Dougherty 2013).

The best subset of features contains the least sufficient number of features. Two approaches to selection are forward selection and backward selection. Forward selection starts with zero variables, and variables that decrease a suitable error metric are added. Forward selection stops when further variable addition does not decrease the error metric anymore. Backward selection starts with all the variables, and variables are removed from the set. In each step, the variable which decreases the error the most is removed. Backward selection stops when further variable removal does not decrease the error sufficiently. In both approaches, the process is costly and finding the optimal subset is not guaranteed.

Wrapper feature selection methods create many models with feature subsets and select the features which perform the best according to a chosen metric. Wrapper methods usually apply cross-validation to select the features. Other types of methods are called filter methods, which use a chosen function to measure the characteristics of the supplied data to select the best features (Dougherty 2013).

In this thesis, forward feature selection using the wrapper method was used. MATLAB's function sequentialsfs was used to implement the selection of features. This function creates candidate subsets by adding the features which were not yet selected to the subset. A 10-fold validation is performed with different subsets of supplied training and testing data on each candidate subsets. A performance metric, the so-called criterion function, is also supplied to the sequentialsfs function. Each candidate feature subset is evaluated in accordance with the criterion function. A subset of features which best predicts the data is selected and returned (MATLAB 2020b).

3.2 Artifact detection

Microelectrode recordings (2.4) commonly contain signal artifacts. The presence of artifacts (2.5) in MER is unwanted and could have a dramatic impact on further signal processing such as spike detection and sorting (Bakštein et al. 2017). Those methods are used to separate the activity of neurons close to the electrode tip from the background activity and then cluster, grouping the recorded spikes into clusters based on the similarity of their shapes (Martinez et al. 2009). Widely-used spike detection algorithms are susceptible to artifact presence.

The first and the essential step in handling artifacts is correct identification. The most general approach is to label artifacts manually. The automatic methods I am going to present based on autocorrelation or maximum-spectral difference were first introduced in articles (Bakstein, Schneider, et al. 2015; Bakštein et al. 2017).

Autocorrelation method

The autocorrelation-based approach is a technique that extends the existing stationary segmentation method by Aboy (2006) and Falkenberg (2003). The stationary segmentation method locates the longest stationery segment of the MER signal, which can be considered the most accurate representation of the characteristics of the neural structure. MER signal is normalized and segmented into non-overlapping frames of the same duration. Next, a ratio of the variances of the covariance values for two neighbouring segments is calculated and is used to measure distance. This test statistic is used to establish transition instances. Compared to the threshold and values that are greater are labelled as transition instances, establishing a boundary between stationary segments. Then the longest wide stationery segment is selected. Bakstein (2015) extends this method by forming a distance matrix which contains the distance between all potential segment pairs. Matrix is then scanned, and the longest uninterrupted segment is returned. The evident benefit of this approach is that non-contiguous subsets of the signal may be returned, which is convenient for manual annotation comparison.

Autocorrelation method algorithm:

1. Signal is normalized:

$$x_n = \frac{x - \mu_x}{\sigma_x} \tag{3.31}$$

2. Signal X is divided into m non-overlapping segments

$$\{X\}_{l=1}^{m} = \{X_1, X_2, \ldots, X_k, \ldots, X_m\} \tag{3.32}$$

Those segments are transformed by autocorrelation function and stationary wavelet transform.

3. Then variance of all segments is computed as:

$$v_i = var\{\gamma(X_i)\}, i \in \langle 1, n \rangle, \tag{3.33}$$

4. Next, variance of adjacent segments is computed according to:

$$d_{ij} = \frac{max(v_i, v_j)}{min(v_i, v_j)}, i \in \langle 1, n-1 \rangle, j = i+1 \tag{3.34}$$

5. Distance matrix is formed:

$$D = \begin{pmatrix} 0 & d_{12} & \dots & d_{1,n-1} \\ d_{21} & 0 & \dots & d_{2,n-1} \\ \vdots & \vdots & \ddots & \dots \\ d_{n-1,1} & d_{n-1,2} & \dots & 0 \end{pmatrix} \tag{3.35}$$

6. Values d_{ij} are compared with threshold θ

$$d_{ij} = \begin{cases} 1, \text{if } d_{ij} > \theta \\ 0, \text{if } d_{ij} < \theta \end{cases} \tag{3.36}$$

Then a greedy algorithm is used to search matrix D for the longest continuous segment.

Maximum spectral difference method

Secondly, the maximum spectral difference method can be used. This method was also presented in papers (Bakstein, Schneider, et al. 2015; Bakštein et al. 2017) and is based on the MER power spectral density and the assumption that artifact-free band-pass filtered signals have smooth PSD and, on the contrary, most signals containing artifacts have uneven PSD.

Clean signal segments from training set of N signals $\boldsymbol{X} = \{X_1, X2, \dots, X_N\}$ and corresponding annotated artifacts $\boldsymbol{a} = \{a_1, a_2, \dots, a_N\}$ are used to compute mean spectrum C

$$\boldsymbol{C} = \frac{1}{\sum_{i=1}^{N} a_i} \cdot \sum_{j=1}^{N} a_j \boldsymbol{Pj}, \tag{3.37}$$

Normalized PSD of signal segment X_j of length 1 s is denoted as $\boldsymbol{Pj}$. It is estimated using Welch's method and divided by its sum. Consequently, the sum of all spectral bins in $\boldsymbol{Pj}$ is equal to one.

$$\boldsymbol{Pj} = \frac{psd(X_j))}{\sum psd(X_j)} \tag{3.38}$$

Next $\boldsymbol{C}$ is compared to all signals in the training set and maximum absolute difference is computed according to:

$$d = \max_{m=1 \dots M} |\boldsymbol{Pj}(m) - \boldsymbol{C}(m)| \tag{3.39}$$

where $m = 1 \dots M$ denotes all spectral bins.

The value which maximizes performance on training data is set as an optimal threshold for d.

3.3 Criteria for classification comparison

The performance of parameter settings for artifact removal methods was evaluated by following parameters.

Receiver Operating Characteristic Curve (ROC)

ROC is a performance measurement for classification problems. It is a graph of false-positive rate on the x-axis versus true positive rate on the y-axis binary classified for different threshold values between 0 and 1. The input values are classified as true or false, and we can define a confusion matrix (also known as a contingency table) containing four observations as shown in figure 3.2.

		True class	
		Positive	**Negative**
Predicted class	**Positive**	TP	FP
	Negative	FN	TN

Table 3.2: Confusion matrix

Four other observations can be defined as:

- true positive rate, also known as sensititivity (Sen)

$$\text{Sensititivity} = TPR = \frac{TP}{P} = \frac{TP}{TP+FN} = 1 - FNR \tag{3.40}$$

- false positive rate

$$FPR = \frac{FP}{N} = \frac{FP}{FP+TN} = 1 - TNR \tag{3.41}$$

- false negative rate

$$FNR = \frac{FN}{P} = \frac{FN}{FN+TP} = 1 - TPR \tag{3.42}$$

- true negative rate, also known as specificity (Spe)

$$\text{Specificity} = TNR = \frac{TN}{N} = \frac{TN}{TN+FP} = 1 - FPR \tag{3.43}$$

The graph is sometimes being called the sensitivity versus (1-specificity) plot. ROC space is defined by FPR and TPR and the perfect method would score a point with coordinates (0,1), which represents 100 % sensitivity and 100 % specificity. On the other hand, a model with no skill can be represented by coordinates (0.5, 0.5). A ROC curve of STN classification by simple NRMS thresholding can be seen in Figure 3.3.

The Area under the curve (AUC) measures the two-dimensional area underneath the ROC curve, and its value is 0.5 to 1. The MATLAB function perfcurve was used to compute the ROC curve and AUC.

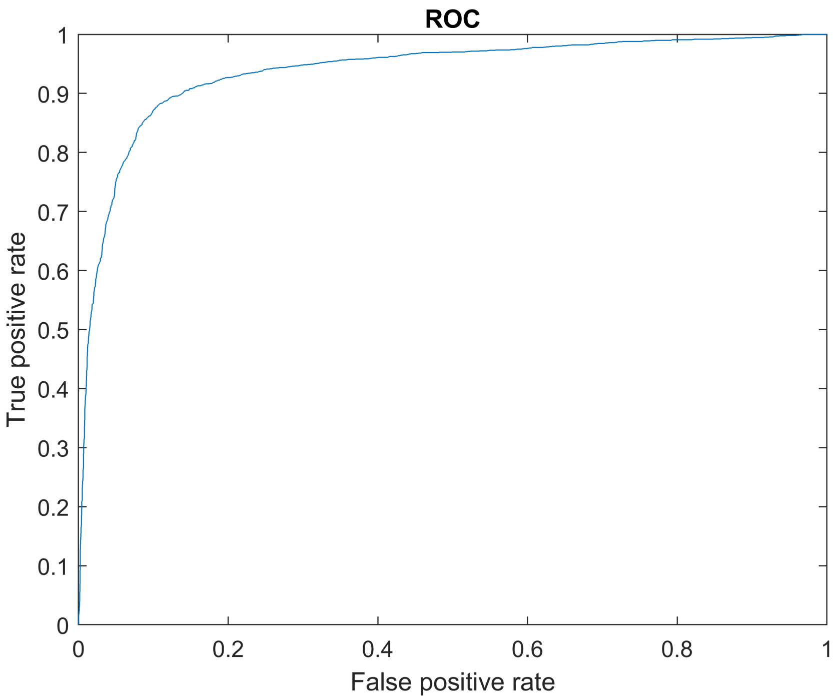


Figure 3.3: ROC curve of STN classification by NRMS thresholding

The optimal operating point of the ROC

There are various ways how to obtain the optimal operating point of the ROC curve. One way is to find the slope S first, using

$$S = \frac{Cost(P|N) - Cost(N|N)}{Cost(N|P) - Cost(P|P)} * \frac{N}{P} \tag{3.44}$$

Where $Cost(N|P)$ is the cost of misclassification of a positive class as a negative class and cost $Cost(P|N)$ is the cost of misclassification of a negative class as a positive class.
The P is defined as

$$P = TP + FN \tag{3.45}$$

and N as

$$N = TN + FP \tag{3.46}$$

.

In the scope of this thesis, the misclassification costs for both classes are the same. The point is found by moving the straight line with slope S from the upper left corner of the ROC plot down and to the right. The movement stops when the line intersects the ROC curve. Matlab function perfcurve was used to compute the optimal operating point.

Youden's J statistics (Youden index, J)

Youden's J statistics is defined as

$$J = \text{Sensitivity} + \text{Specificity} - 1 \tag{3.47}$$

the maximum of Youden's J statistics is often used

$$J_{max} = max_i\{\text{Sensitivity}(t) + \text{Specificity}(t) - 1\} \tag{3.48}$$

where t denotes the threshold, for which J is maximal. (Ranganathan et al. 2019). In the case of STN detection task, point t is the optimal operating point of the curve.

Accuracy (Acc)

The accuracy of a classifier is calculated as ratio of number of correct predictions to the total number of input samples

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.49}$$

For binary classification, accuracy can be also computed in terms of positives and negatives which were defined in Table 3.2

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.50}$$

The Matlab function classperf was used to implement the accuracy calculation.

Training error (Train Err)

Training error is the error obtained by calculating the classification error of a model on the same data the model was trained on.

Test error (Test Err)

Test error is obtained by running the trained model on a set of data that the model has previously never seen before.

3.4 Structure of the suggested solution

The aim of this thesis is to implement a set of STN classifiers and compare their performance. Firstly, either maxDiffPSD or COV methods remove artifacts from the dataset. Various settings of those methods are tested, and the best method settings are obtained. Afterwards, the STN is classified by one of the classifiers.

The Bachelor thesis by Konicarova (2021) concludes an outcome that NRMS is the greatest feature for the detection of STN by thresholding. The result was derived from the feature set containing RMS, NRMS, PSD, ANE CL, ZC and Peaks. It is also easy to compute and is notably improved by COV artifact removal. Also, the automatic artifact removal methods showed the best results compared to the raw signal and manual annotation.

Based on those results, the classifier, which consists of NRMS thresholding, was chosen to be suitable for the following STN detection task aimed at searching for the best parameter settings. This could be classification using MATLAB's perfcurve function - simple NRMS thresholding by various thresholds or by the threshold with the best results. Owing to the simplicity of the NRMS thresholding STN classifier, a large number of parameter value combinations for the artifact detection methods can easily be compared. In the next step, the performance of more complex classifiers (SVM and RF) in the STN classification task using pre-selected artifact detection parameter sets from the previous step are evaluated. RF was evaluated using all previously mentioned features. SVM was evaluated using feature selection. Those classifiers are then compared, and the best classifier for this task is selected.

Chapter 4

Results

One of the aims of this thesis is to implement and compare a set of classifiers for nuclei classification on a given dataset. This dataset is briefly described in Section 4.1. Section 4.2 is dedicated to optimal settings of automatic artifact detection methods, and those settings are compared to optimal settings for STN detection task in Section 4.3.3. Section 4.3 is dedicated to setting the best parameters for automatic artifact detection and following STN classification using NRMS thresholding by a set of thresholds using ROC curves (4.3.1) or by the selected best threshold (4.3.2). Afterwards, classification by the Random forest algorithm with feature selection was performed. Section 4.4.1 is dedicated to setting the best hyperparameters for RF classification. Those hyperparameters are then used in Section 4.4.2 to derive the ideal artifact detection settings for following RF classification. Then, those settings are tested on unseen data. Section 4.5 is dedicated to STN detection by the SVM algorithm with feature selection. The ideal artifact detection settings for following SVM classification are derived. Results of both are compared to simple NRMS thresholding in Section 4.6.1.

4.1 Data description

We used extracellular microelectrode recordings (MER) of 28 patients with Parkinson's disease to test the classifier. Recordings were collected during electrophysiological exploration for deep brain stimulation surgery. The typical recording length is 10 s, sampling frequency was 24 kHz. In practice, the MER signal consists of spiking activity of most nearby neuronal units and background electrical activity that comes from several sources, including more remotely located units, measurement noise, and various artefacts (Dolan et al. 2009).

The signal's id contains the patient's id, trajectory id, electrode and electrode id depth. Recording depth carries information about how close to the expected target a given recording position is located, and there are typically five electrodes per trajectory (central, medial, lateral, posterior, anterior). An experienced neurologist manually annotated data during surgery. Neurologist annotated data as STN, SNr, Th or unknown. The subthalamic nucleus (STN) is the target, substantia nigra (SNr) can be found below STN, and Th stands for the thalamus. The dataset contains 2195 areas annotated as STN, 244 as SNr, 261 as Th, and the rest of the areas (5551) are annotated as "other". This thesis is focused on detection of STN nuclei versus all others.

Additional set of annotation labels focuses on artifacts. The neurologist annotated the presence of artifacts and the exact type of observed artifacts. Of 8322 signals, 5342 were annotated. Observed artifacts contain mechanical movement artifacts, low-frequency interference, electromagnetic interference, "irritated neuron" artifact and others.

4.2 Artifact detection

The artifact classification of Autocorrelation and Maximal spectral method was evaluated with different parameter settings. Two selected parameters to vary were window length (winLen) and threshold (sometimes referred to as th). In other words, in this section, we used the manual labels of signal sections and tested the accuracy of different artifact classifier settings. Microelectrode recordings with automatic artifact detection are available in Figures 4.1 and 4.2.

When winLen decreases, the maxDiffPSD method can recognize small parts of the clean signal, and as it increases, the recognition is less detailed. When the threshold decreases, a huge amount of data is removed from the test set, often whole signals, and the amount of kept data increases with the increasing threshold value. This can be observed in Figure 4.1. It contains two different signals containing signal artifacts and the artifact detection with various winLen and threshold settings. The detected artifacts are red, and the clean signal has a blue color.

When winLen decreases, the chunks removed by COV method are shorter, no signals are entirely removed, and the amount of kept data is higher. When winLen increases, the method removes large pieces of signals and sometimes even whole signals are removed from the test set. As the threshold increases, the amount of data kept increases. This can be observed in Figure 4.2. It contains two different signals containing signal artifacts and the artifact detection with various winLen and threshold settings.

Accuracy and Youden-J of the classification were compared. Each signal second was divided into four segments. The presence of a single segment labelled by method as an artifact resulted in the whole second being labelled as one.

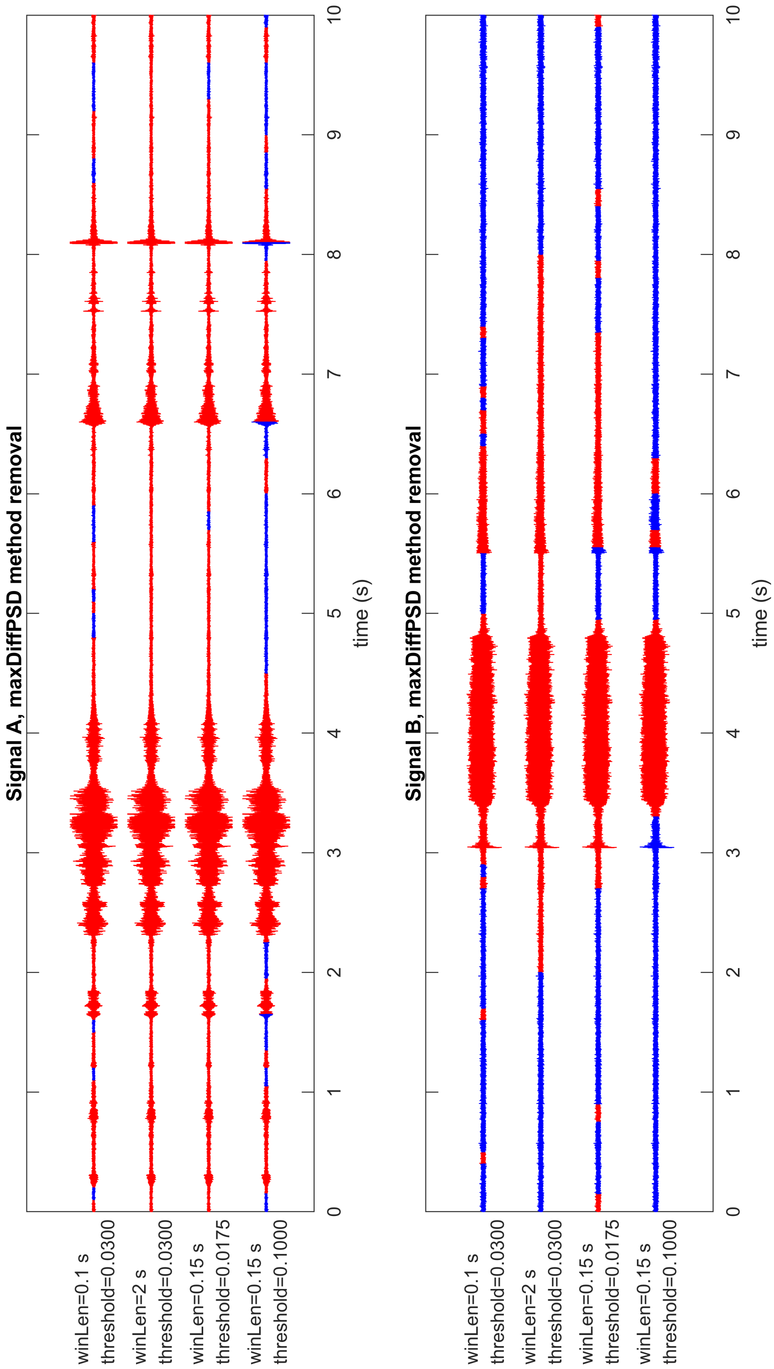


Figure 4.1: Artifact removal by maxDiffPSD method with different winLen and threshold settings demonstrated on signal A and signal B, detected artifact sections are in red

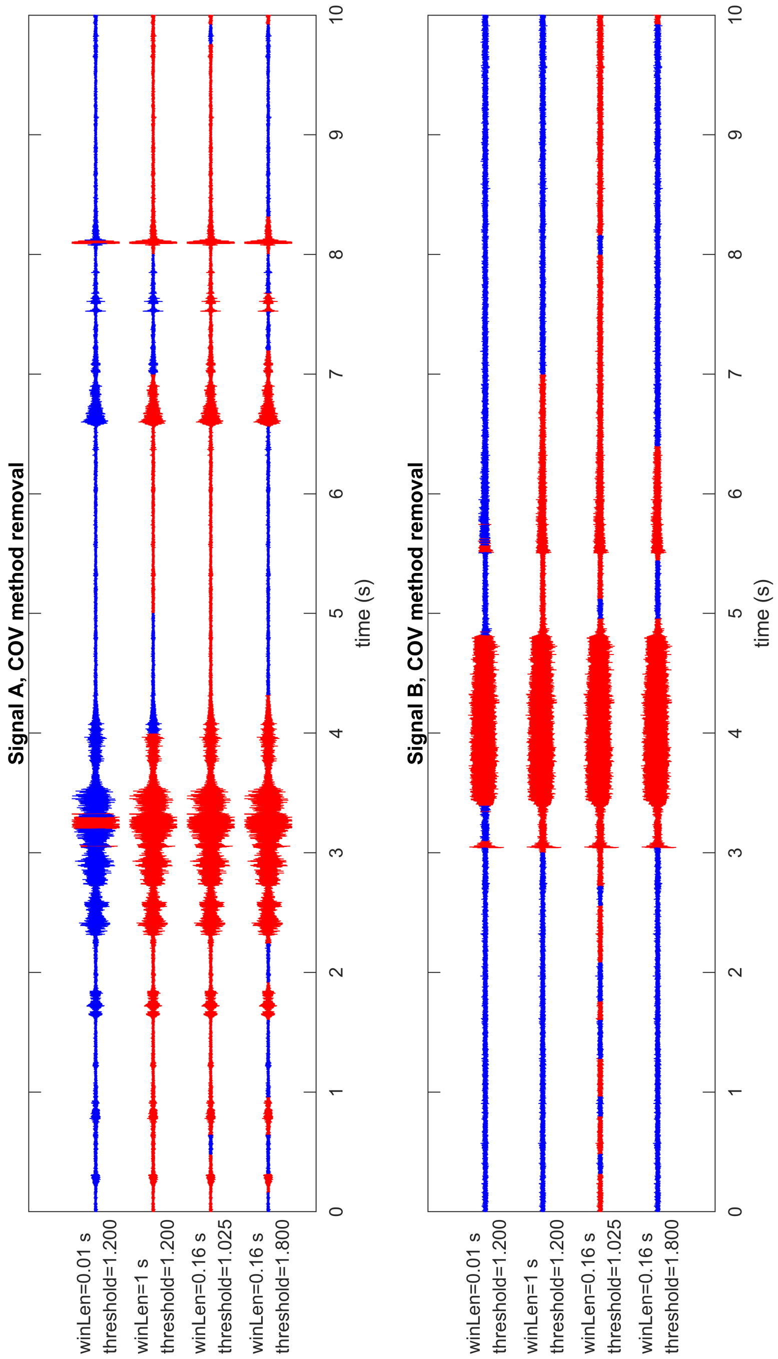


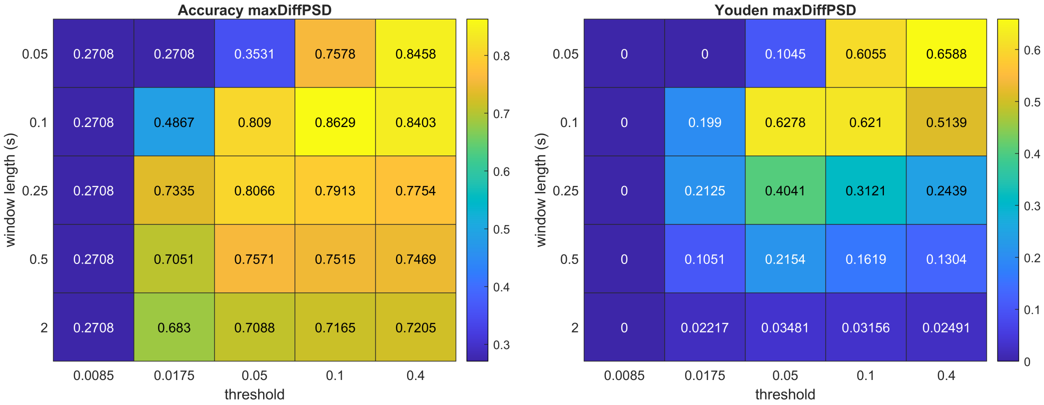
Figure 4.2: Artifact removal by COV method with different winLen and threshold settings demonstrated on signal A and signal B, detected artifact sections are in red

The detection results were compared with manual annotation, and the accuracy and maximum Youden-J of the artifact classification were computed. Six examples of results of artifact detection can be found in Table 4.1. Results with high Accuracy and Youden were chosen.

method	th	winLen (s)	Accuracy	J	raw signals[3]	non-zero signals[4]
maxDiffPSD	0.04	0.05	0.8458	0.6588	0	0
maxDiffPSD	0.1	0.1	0.8629	0.621	0.8635	0.8957
maxDiffPSD	0.05	0.25	0.8066	0.4041	0.8046	0.9316
COV	1.2	0.16	0.7663	0.2334	0.875	0.996
COV	1.4	0.16	0.7744	0.3408	0.9097	0.999
COV	1.4	0.33	0.7805	0.2823	0.8626	0.994

Table 4.1: Table of artifact classification performance for different methods and parameters

The maxDiffPSD performance can be seen in Figures 4.3, 4.4, and the COV performance is shown in 4.5, 4.6.



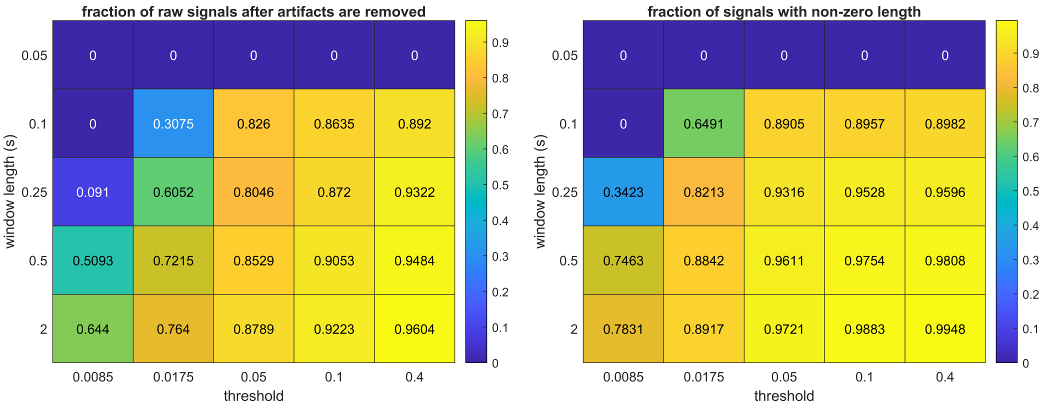
(a) : Accuracy heatmap

(b) : Youden heatmap

Figure 4.3: Artifact detection heatmaps for maxDiffPSD method

[3] Fraction of raw signals after artifacts are removed

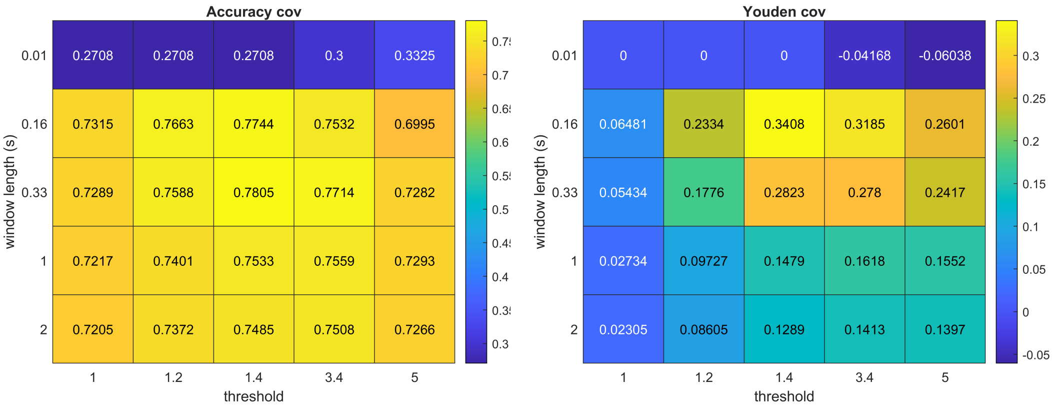
[4] Fraction of signals with non-zero length after artifacts are removed



(a) : Fraction of data kept

(b) : Fraction of non-zero signals

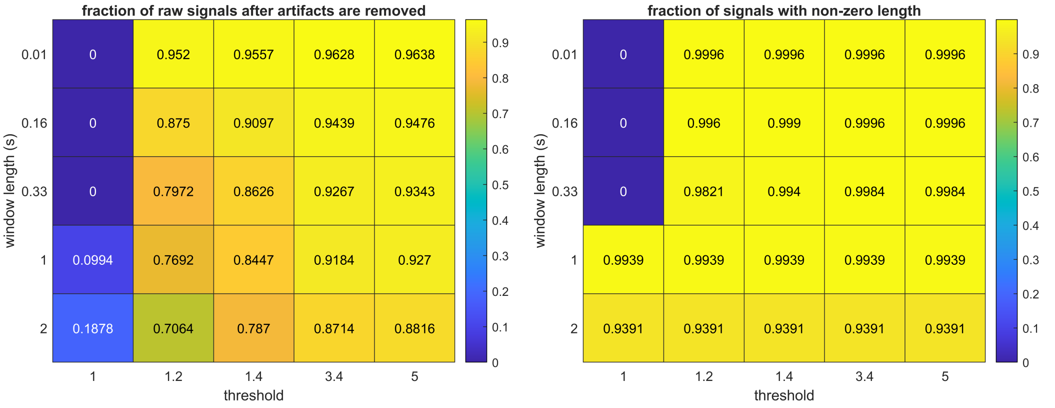
Figure 4.4: Heatmaps of lost data for maxDiffPSD method



(a) : Accuracy heatmap

(b) : Youden heatmap

Figure 4.5: Artifact detection heatmaps for COV method



(a) : Fraction of data kept

(b) : Fraction of non-zero signals

Figure 4.6: Heatmaps of lost data for COV method

4.3 Artifact detection methods and their impact on STN detection by NRMS thresholding

4.3.1 NRMS thresholding by ROC curves

Both the Autocorrelation method and the Maximal spectral method previously described in Section 3.2 were used to remove artifacts on a test dataset. The aim was to compare how accurately can the methods classify STN with different parameter settings. The chosen parameters were detection threshold (th) and time-window length (winLen).

After the artifacts were automatically removed from signals by one of the methods with a predetermined set of parameters, the NRMS features were computed, the results were compared with manual annotation. The STN classification was evaluated in AUC and the Youden-J statistics of the optimal operating point of the ROC was computed. This is also referred to as NRMS thresholding by a set of thresholds using ROC curves. It is essential to emphasize the amount of data lost due to artifact removal, as results with a low amount of preserved data may be biased. Some signals were even removed as a whole. When such a situation occurred, the NRMS feature was calculated only from signals with non-zero length.

Seven examples of results of STN detection can be found in Table 4.2. Parameters with high AUC and Youden performance were chosen, and detection was done for the raw signal too. The classification was performed on training data.

The full maxDiffPSD performance evaluation can be seen in Figures 4.7, 4.8, 4.9, 4.10. From corresponding AUC and Youden values it is clear, that artifact removal has a positive impact on classification accuracy.

method	th	winLen (s)	AUC	J	raw signals[1]	non-zero signals[2]
raw signal	-	-	0.8042	0.2629	100	100
maxDiffPSD	0.005	0.66	0.9233	0.6773	0.3167	0.6216
maxDiffPSD	0.0085	0.66	0.9232	0.6508	0.5358	0.7821
maxDiffPSD	0.03	0.66	0.9103	0.5391	0.7532	0.9151
maxDiffPSD	0.0085	1	0.9176	0.6496	0.5514	0.7661
COV	1.025	0.01	0.9222	0.6759	0.8037	1
COV	1.05	0.16	0.9183	0.6631	0.5926	1
COV	1.4	0.08	0.9152	0.6064	0.9239	1
COV	1.2	0.33	0.9135	0.6017	0.7811	0.9885

Table 4.2: Table of STN classification performance for different methods and parameters

The COV performance evaluation can be seen in Figures 4.11,4.12,4.13,4.14

[1] Fraction of raw signals after artifacts are removed

[2] Fraction of signals with non-zero length after artifacts are removed

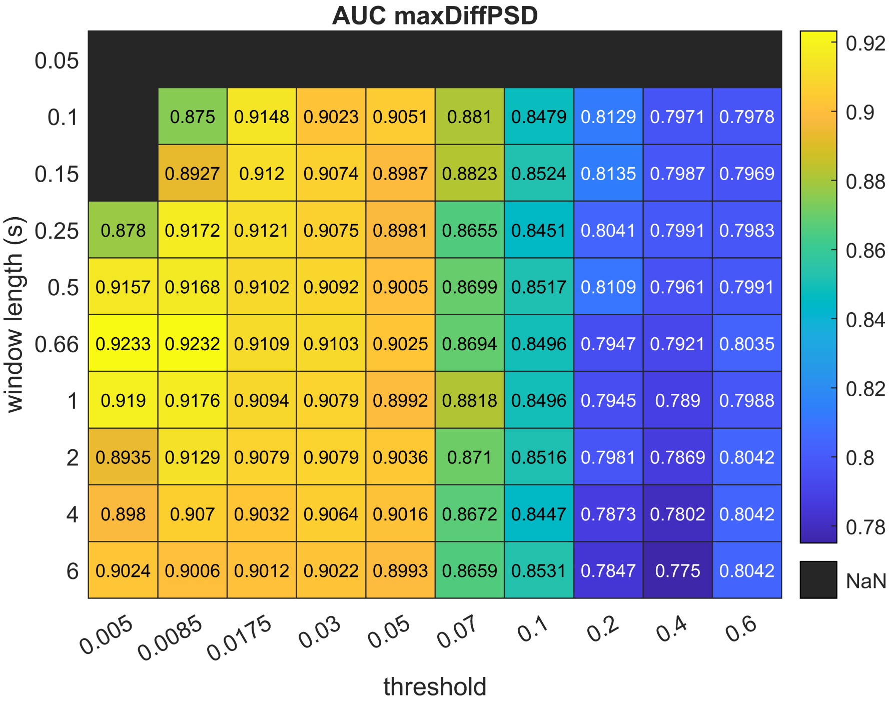


Figure 4.7: AUC heatmap for maxDiffPSD STN classification

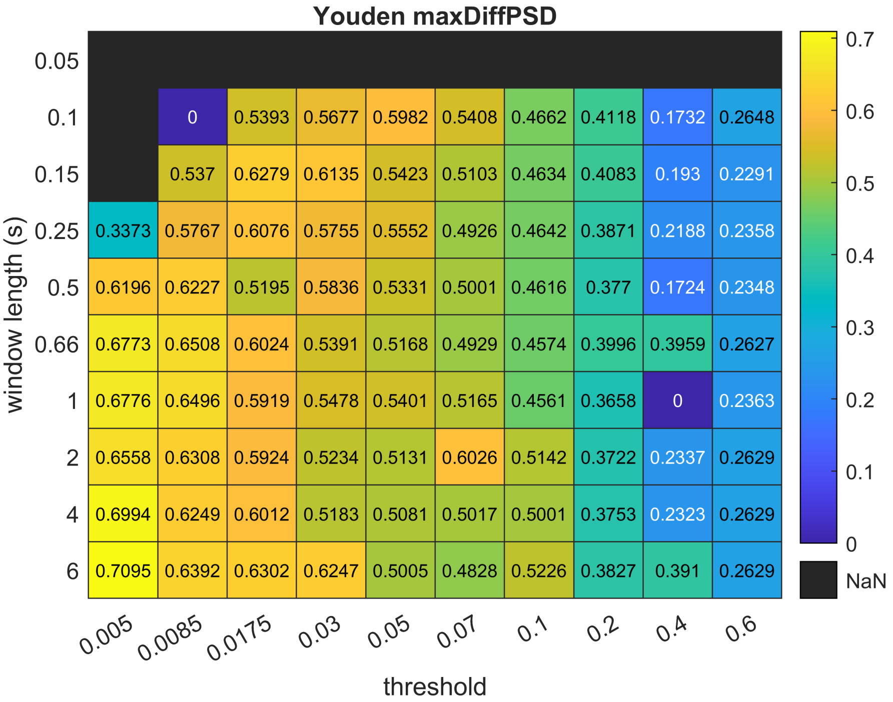
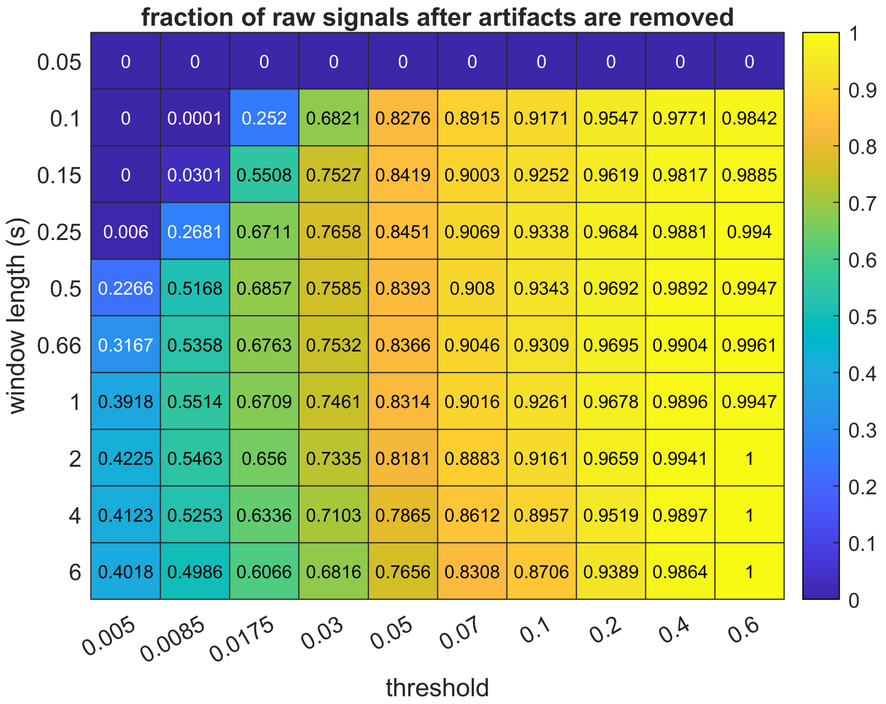
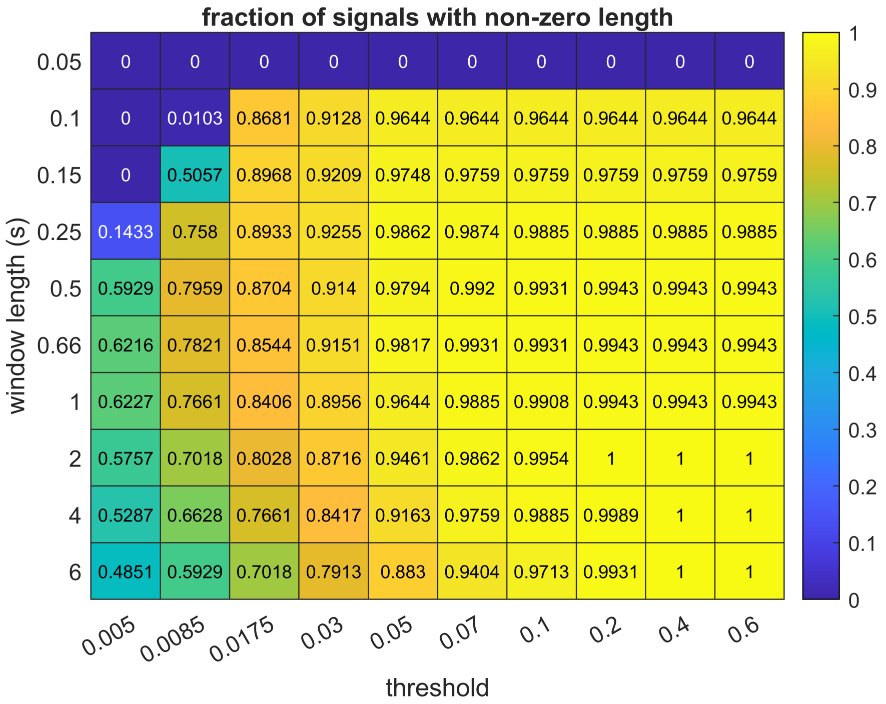


Figure 4.8: Youden-J heatmap for maxDiffPSD STN classification



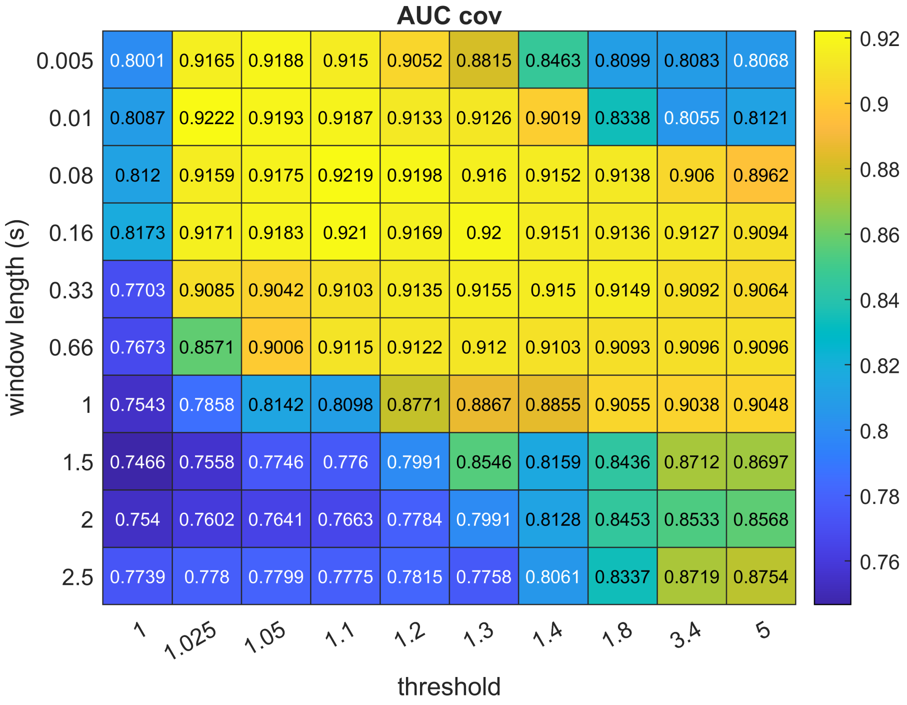
window length (s) \ threshold	0.005	0.0085	0.0175	0.03	0.05	0.07	0.1	0.2	0.4	0.6
0.05	0	0	0	0	0	0	0	0	0	0
0.1	0	0.0001	0.252	0.6821	0.8276	0.8915	0.9171	0.9547	0.9771	0.9842
0.15	0	0.0301	0.5508	0.7527	0.8419	0.9003	0.9252	0.9619	0.9817	0.9885
0.25	0.006	0.2681	0.6711	0.7658	0.8451	0.9069	0.9338	0.9684	0.9881	0.994
0.5	0.2266	0.5168	0.6857	0.7585	0.8393	0.908	0.9343	0.9692	0.9892	0.9947
0.66	0.3167	0.5358	0.6763	0.7532	0.8366	0.9046	0.9309	0.9695	0.9904	0.9961
1	0.3918	0.5514	0.6709	0.7461	0.8314	0.9016	0.9261	0.9678	0.9896	0.9947
2	0.4225	0.5463	0.656	0.7335	0.8181	0.8883	0.9161	0.9659	0.9941	1
4	0.4123	0.5253	0.6336	0.7103	0.7865	0.8612	0.8957	0.9519	0.9897	1
6	0.4018	0.4986	0.6066	0.6816	0.7656	0.8308	0.8706	0.9389	0.9864	1

Figure 4.9: Fraction of data kept for maxDiffPSD STN classification



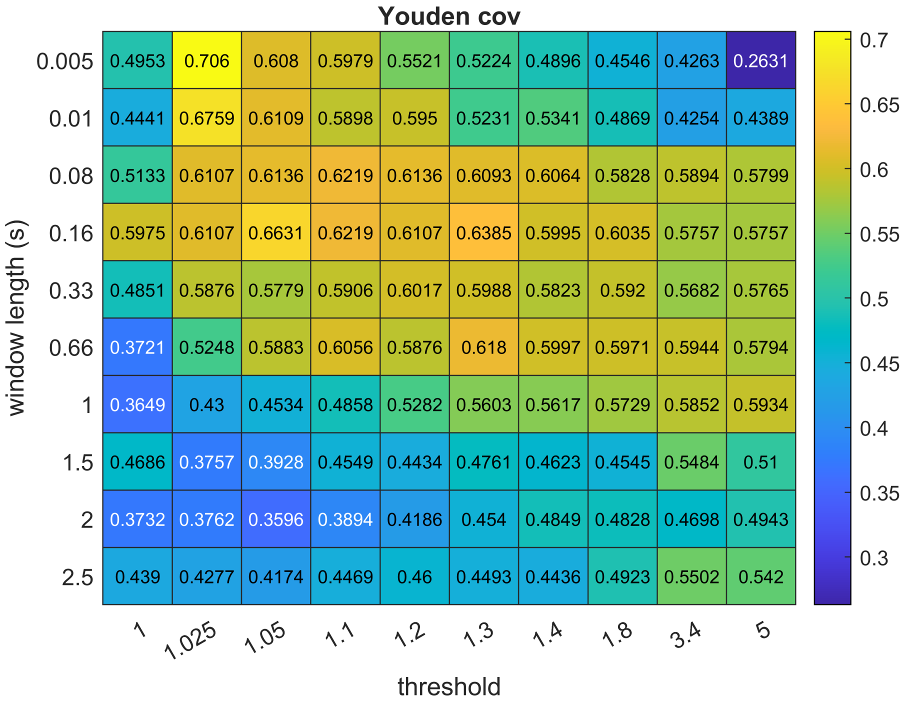
window length (s) \ threshold	0.005	0.0085	0.0175	0.03	0.05	0.07	0.1	0.2	0.4	0.6
0.05	0	0	0	0	0	0	0	0	0	0
0.1	0	0.0103	0.8681	0.9128	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644
0.15	0	0.5057	0.8968	0.9209	0.9748	0.9759	0.9759	0.9759	0.9759	0.9759
0.25	0.1433	0.758	0.8933	0.9255	0.9862	0.9874	0.9885	0.9885	0.9885	0.9885
0.5	0.5929	0.7959	0.8704	0.914	0.9794	0.992	0.9931	0.9943	0.9943	0.9943
0.66	0.6216	0.7821	0.8544	0.9151	0.9817	0.9931	0.9931	0.9943	0.9943	0.9943
1	0.6227	0.7661	0.8406	0.8956	0.9644	0.9885	0.9908	0.9943	0.9943	0.9943
2	0.5757	0.7018	0.8028	0.8716	0.9461	0.9862	0.9954	1	1	1
4	0.5287	0.6628	0.7661	0.8417	0.9163	0.9759	0.9885	0.9989	1	1
6	0.4851	0.5929	0.7018	0.7913	0.883	0.9404	0.9713	0.9931	1	1

Figure 4.10: Fraction of non-zero signals for maxDiffPSD STN classification



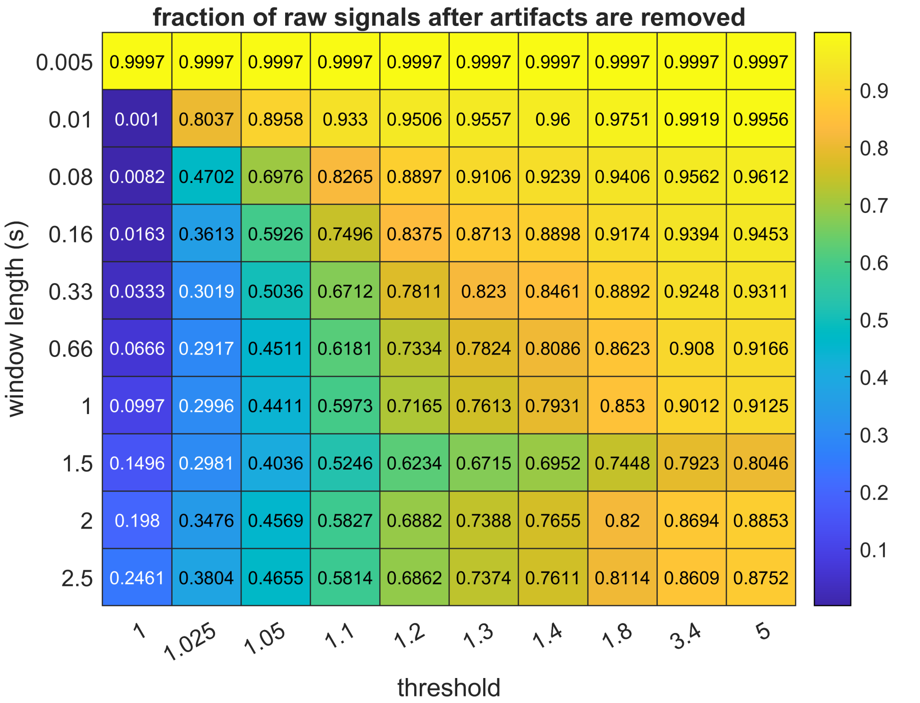
window length (s)	1	1.025	1.05	1.1	1.2	1.3	1.4	1.8	3.4	5
0.005	0.8001	0.9165	0.9188	0.915	0.9052	0.8815	0.8463	0.8099	0.8083	0.8068
0.01	0.8087	0.9222	0.9193	0.9187	0.9133	0.9126	0.9019	0.8338	0.8055	0.8121
0.08	0.812	0.9159	0.9175	0.9219	0.9198	0.916	0.9152	0.9138	0.906	0.8962
0.16	0.8173	0.9171	0.9183	0.921	0.9169	0.92	0.9151	0.9136	0.9127	0.9094
0.33	0.7703	0.9085	0.9042	0.9103	0.9135	0.9155	0.915	0.9149	0.9092	0.9064
0.66	0.7673	0.8571	0.9006	0.9115	0.9122	0.912	0.9103	0.9093	0.9096	0.9096
1	0.7543	0.7858	0.8142	0.8098	0.8771	0.8867	0.8855	0.9055	0.9038	0.9048
1.5	0.7466	0.7558	0.7746	0.776	0.7991	0.8546	0.8159	0.8436	0.8712	0.8697
2	0.754	0.7602	0.7641	0.7663	0.7784	0.7991	0.8128	0.8453	0.8533	0.8568
2.5	0.7739	0.778	0.7799	0.7775	0.7815	0.7758	0.8061	0.8337	0.8719	0.8754

Figure 4.11: AUC heatmap for COV STN classification



window length (s)	1	1.025	1.05	1.1	1.2	1.3	1.4	1.8	3.4	5
0.005	0.4953	0.706	0.608	0.5979	0.5521	0.5224	0.4896	0.4546	0.4263	0.2631
0.01	0.4441	0.6759	0.6109	0.5898	0.595	0.5231	0.5341	0.4869	0.4254	0.4389
0.08	0.5133	0.6107	0.6136	0.6219	0.6136	0.6093	0.6064	0.5828	0.5894	0.5799
0.16	0.5975	0.6107	0.6631	0.6219	0.6107	0.6385	0.5995	0.6035	0.5757	0.5757
0.33	0.4851	0.5876	0.5779	0.5906	0.6017	0.5988	0.5823	0.592	0.5682	0.5765
0.66	0.3721	0.5248	0.5883	0.6056	0.5876	0.618	0.5997	0.5971	0.5944	0.5794
1	0.3649	0.43	0.4534	0.4858	0.5282	0.5603	0.5617	0.5729	0.5852	0.5934
1.5	0.4686	0.3757	0.3928	0.4549	0.4434	0.4761	0.4623	0.4545	0.5484	0.51
2	0.3732	0.3762	0.3596	0.3894	0.4186	0.454	0.4849	0.4828	0.4698	0.4943
2.5	0.439	0.4277	0.4174	0.4469	0.46	0.4493	0.4436	0.4923	0.5502	0.542

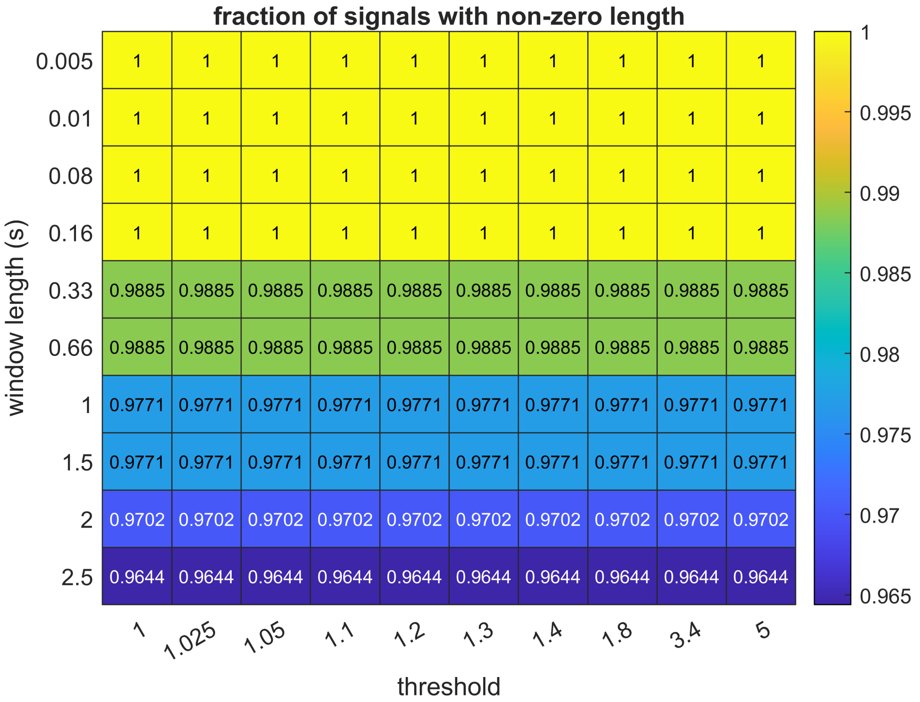
Figure 4.12: Youden-J heatmap for COV STN classification



fraction of raw signals after artifacts are removed

window length (s) \ threshold	1	1.025	1.05	1.1	1.2	1.3	1.4	1.8	3.4	5
0.005	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997
0.01	0.001	0.8037	0.8958	0.933	0.9506	0.9557	0.96	0.9751	0.9919	0.9956
0.08	0.0082	0.4702	0.6976	0.8265	0.8897	0.9106	0.9239	0.9406	0.9562	0.9612
0.16	0.0163	0.3613	0.5926	0.7496	0.8375	0.8713	0.8898	0.9174	0.9394	0.9453
0.33	0.0333	0.3019	0.5036	0.6712	0.7811	0.823	0.8461	0.8892	0.9248	0.9311
0.66	0.0666	0.2917	0.4511	0.6181	0.7334	0.7824	0.8086	0.8623	0.908	0.9166
1	0.0997	0.2996	0.4411	0.5973	0.7165	0.7613	0.7931	0.853	0.9012	0.9125
1.5	0.1496	0.2981	0.4036	0.5246	0.6234	0.6715	0.6952	0.7448	0.7923	0.8046
2	0.198	0.3476	0.4569	0.5827	0.6882	0.7388	0.7655	0.82	0.8694	0.8853
2.5	0.2461	0.3804	0.4655	0.5814	0.6862	0.7374	0.7611	0.8114	0.8609	0.8752

Figure 4.13: Fraction of data kept for COV STN classification



fraction of signals with non-zero length

window length (s) \ threshold	1	1.025	1.05	1.1	1.2	1.3	1.4	1.8	3.4	5
0.005	1	1	1	1	1	1	1	1	1	1
0.01	1	1	1	1	1	1	1	1	1	1
0.08	1	1	1	1	1	1	1	1	1	1
0.16	1	1	1	1	1	1	1	1	1	1
0.33	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885
0.66	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885	0.9885
1	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771
1.5	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771	0.9771
2	0.9702	0.9702	0.9702	0.9702	0.9702	0.9702	0.9702	0.9702	0.9702	0.9702
2.5	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644	0.9644

Figure 4.14: Fraction of non-zero signals for COV STN classification

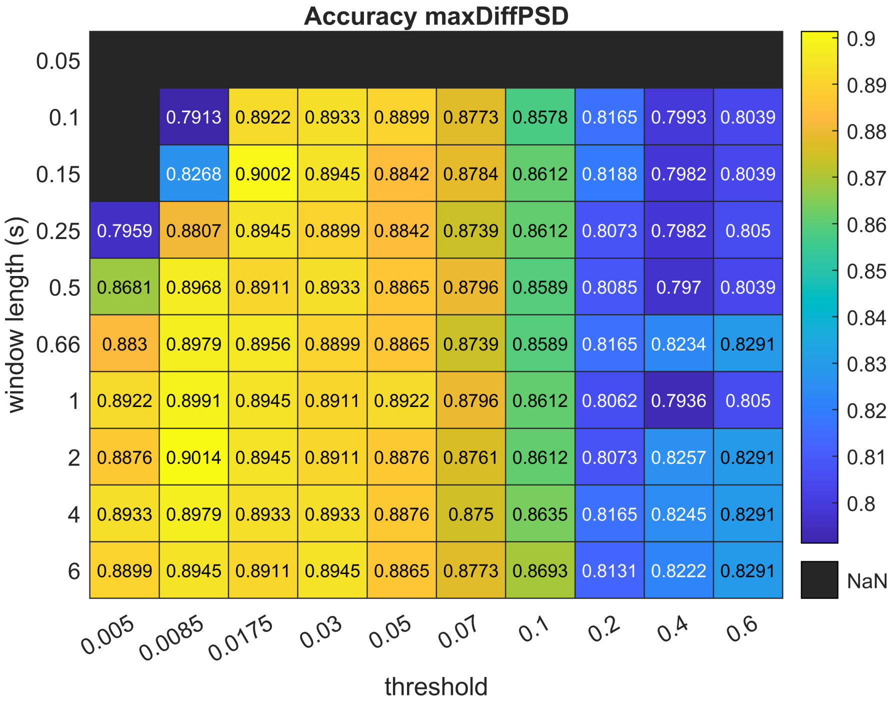
4.3.2 NRMS thresholding by the best threshold

As results from the previous Section 4.3.1 about STN detection parameters could be hard to compare to the results of artifact detection and RF classifier performance, the NRMS thresholding was additionally implemented. The best threshold value to classify signals containing STN was derived from MATLAB's perfcurve function. After the best threshold θ was found, data were classified as :

$$position = \begin{cases} 1, & \text{if } NRMS > \theta. \\ 0, & NRMS < \theta. \end{cases} \tag{4.1}$$

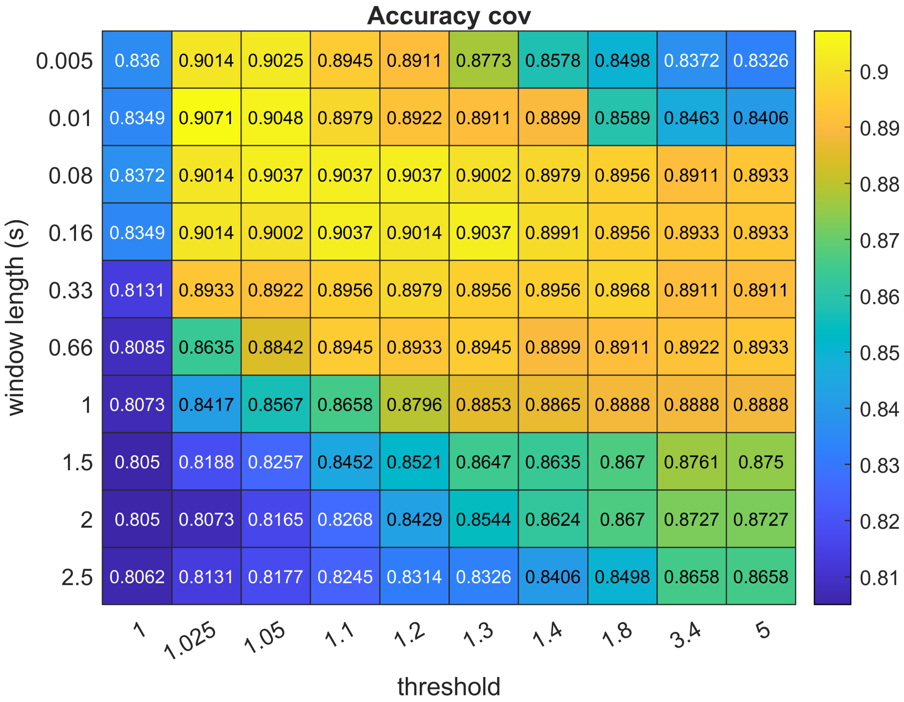
where 1 labels the signal as STN and 0 labels it as outside of STN.

The accuracy values of such STN classification are shown in Figure 4.15 for maxDiffPSD method and in Figure 4.16 for COV method.



window length (s) \ threshold	0.005	0.0085	0.0175	0.03	0.05	0.07	0.1	0.2	0.4	0.6
0.05										
0.1		0.7913	0.8922	0.8933	0.8899	0.8773	0.8578	0.8165	0.7993	0.8039
0.15		0.8268	0.9002	0.8945	0.8842	0.8784	0.8612	0.8188	0.7982	0.8039
0.25	0.7959	0.8807	0.8945	0.8899	0.8842	0.8739	0.8612	0.8073	0.7982	0.805
0.5	0.8681	0.8968	0.8911	0.8933	0.8865	0.8796	0.8589	0.8085	0.797	0.8039
0.66	0.883	0.8979	0.8956	0.8899	0.8865	0.8739	0.8589	0.8165	0.8234	0.8291
1	0.8922	0.8991	0.8945	0.8911	0.8922	0.8796	0.8612	0.8062	0.7936	0.805
2	0.8876	0.9014	0.8945	0.8911	0.8876	0.8761	0.8612	0.8073	0.8257	0.8291
4	0.8933	0.8979	0.8933	0.8933	0.8876	0.875	0.8635	0.8165	0.8245	0.8291
6	0.8899	0.8945	0.8911	0.8945	0.8865	0.8773	0.8693	0.8131	0.8222	0.8291

Figure 4.15: Accuracy heatmap for maxDiffPSD STN thresholding



window length (s) \ threshold	1	1.025	1.05	1.1	1.2	1.3	1.4	1.8	3.4	5
0.005	0.836	0.9014	0.9025	0.8945	0.8911	0.8773	0.8578	0.8498	0.8372	0.8326
0.01	0.8349	0.9071	0.9048	0.8979	0.8922	0.8911	0.8899	0.8589	0.8463	0.8406
0.08	0.8372	0.9014	0.9037	0.9037	0.9037	0.9002	0.8979	0.8956	0.8911	0.8933
0.16	0.8349	0.9014	0.9002	0.9037	0.9014	0.9037	0.8991	0.8956	0.8933	0.8933
0.33	0.8131	0.8933	0.8922	0.8956	0.8979	0.8956	0.8956	0.8968	0.8911	0.8911
0.66	0.8085	0.8635	0.8842	0.8945	0.8933	0.8945	0.8899	0.8911	0.8922	0.8933
1	0.8073	0.8417	0.8567	0.8658	0.8796	0.8853	0.8865	0.8888	0.8888	0.8888
1.5	0.805	0.8188	0.8257	0.8452	0.8521	0.8647	0.8635	0.867	0.8761	0.875
2	0.805	0.8073	0.8165	0.8268	0.8429	0.8544	0.8624	0.867	0.8727	0.8727
2.5	0.8062	0.8131	0.8177	0.8245	0.8314	0.8326	0.8406	0.8498	0.8658	0.8658

Figure 4.16: Accuracy heatmap for COV STN thresholding

4.3.3 STN detection compared to artifact detection

It would appear that when the parameters for one of the methods are being chosen, the task of detection should be taken into account, as the best settings for artifact detection and detection of STN differ.

The STN detection has achieved higher J values than the artifact detection task. The J was computed from the optimal operating point of the ROC curve in the STN detection and confusion matrix in the artifact detection task.

The STN detection is surprisingly sensitive to any unwanted artifact presence. The highest classification results were achieved when a large amount of data was removed before the STN classification.

COV method has a better performance for STN classification task and maxDiffPSD method has a better performance for artifact detection task.

In the STN detection task, maxDiffPSD method has the accuracy of 0.9002 when winLen is set to 0.15 s, and the threshold is set to 0.0175. With those settings, the J statistics has a value of 0.6279.

In artifact detection task, the maxDiffPSD method has the highest accuracy of 0.8629 when the threshold is set to 0.1 and winLen is set to 0.1. With those settings, the J statistics has a value of 0.621.

The COV method has the highest accuracy of 0.9071 in the STN classification task when winLen is set to 0.01 s, and the threshold is 1.025. With those settings, the J statistics has a value of 0.6759. It should be also taken into account that with the mentioned settings, 80.37 % of raw signal data were preserved, and all signals have non-zero length.

When it comes to artifact detection, this method has the highest accuracy of 0.7805 with winLen 0.33 s and threshold 1.4. The highest J is accomplished by setting winLen to 0.16 s and having the same threshold.

4.4 Random forest STN classification

The STN classification by the Random forest algorithm was implemented. Firstly, the dataset was divided into 8-fold cross-validation set containing signals from 24 patients and a test set containing signals from the remaining 4 patients. Afterwards, the hyperparameters of RF were tuned, and the performance of the RF algorithm, which classified data previously cleared from artifacts by one of the artifact removal methods with different parameter settings, was evaluated. Those actions were taken on the cross-validation set, and the results were eight times cross-validated. Subsequently, the best hyperparameters and the best artifact removal parameter settings were used to classify the STN by RF algorithm on the test set.

4.4.1 Random forest hyperparameters

The influence of different RF hyperparameter settings was tested. Testing was performed so that only one of the hyperparameters was changed, and the rest of the hyperparameters kept their original value. Data from 24 patients were used in this part of the thesis. It was divided into training and testing data with a ratio of 7:1. The results were eight times cross-validated.

Firstly, artifacts were removed from the dataset using either maxDiffPSD or COV method. A set of features described in Section 3.1.4 was used. The STN was classified by RF algorithm with a various number of trees, algorithm was implemented using MATLAB's TreeBagger function. Results of classification after maxDiffPSD removal and COV removal can be seen in Table 4.3 and Table 4.4, respectively. The parameters for maxDiffPSD and COV artifact removal were chosen based on results in Section 4.3. For maxDiffPSD method, threshold was set to 0.0085 and window length to 0.66 s. For COV method, threshold was set to 1.025 and window length to 0.01 s.

After the artifact removal, STN was classified by the RF algorithm with the number of trees with the best accuracy and yet different hyperparameter settings. Results of classification after maxDiffPSD removal and COV removal can be seen in Table 4.5 and Table 4.6, respectively. Every row of the table contains one hyperparameter whose value has changed. The first row of the table contains the classification results with all RF hyperparameters set to do default.

Num Trees	Acc	J	Sen	Spe	Train Err	Test Err
10	0.9123	0.7161	0.7564	0.9597	0.0182	0.0877
20	0.9162	0.7304	0.7688	0.9616	0.0154	0.0838
100	0.9105	0.7250	0.7718	0.9532	0.0131	0.0895
200	0.9118	0.7302	0.7771	0.9531	0.0126	0.0882
500	0.9161	0.7357	0.7774	0.9583	0.0118	0.0839

Table 4.3: RF evaluation with different number of trees after maxDiffPSD removal

Num Trees	Acc	J	Sen	Spe	Train Err	Test Err
10	0.9159	0.7263	0.7642	0.9620	0.0072	0.0841
20	0.9169	0.7375	0.7790	0.9585	0.0026	0.0831
100	0.9238	0.7606	0.7997	0.9609	0.0002	0.0762
200	0.9230	0.7567	0.7956	0.9611	0.0001	0.0770
500	0.9232	0.7568	0.7954	0.9614	0.0001	0.0768

Table 4.4: RF evaluation with different number of trees after COV removal

Hyperparam	Value	Acc	J	Sen	Spe	Train Err	Test Err
-	**all default**	0.9097	0.7208	0.7673	0.9535	0.0133	0.0903
InBagFraction	**0.5**	0.9125	0.7297	0.7746	0.9551	0.0132	0.0875
Cost	$\begin{bmatrix} \mathbf{0} & \mathbf{1} \\ \mathbf{5} & \mathbf{0} \end{bmatrix}$	0.8409	0.6925	0.8578	0.8347	0.0720	0.1591
SampleWithReplacement	**Off**	0.9123	0.7226	0.7662	0.9564	0.0120	0.0877
NumpredictorsToSample	**2**	0.9156	0.7275	0.7662	0.9612	0.0112	0.0844
NumpredictorsToSample	**6**	0.9138	0.7213	0.7613	0.9600	0.0149	0.0862
NumpredictorsToSample	**22**	0.9123	0.7149	0.7538	0.9612	0.0151	0.0877
MinLeafSize	**2**	0.9116	0.7249	0.7701	0.9548	0.0186	0.0884
Prior	**Uniform**	0.8953	0.7453	0.8310	0.9143	0.0297	0.1047
MaxNumSplits	**(NumObservations)/2**	0.9154	0.7245	0.7632	0.9613	0.0138	0.0846
MaxNumSplits	**(NumObservations)/4**	0.9118	0.7340	0.7821	0.9519	0.0122	0.0882
MergeLeaves	**Off**	0.9151	0.7319	0.7733	0.9587	0.0114	0.0849
PredictorSelection	**Curvature**	0.9176	0.7367	0.7769	0.9598	0.0139	0.0824
PredictorSelection	**Interaction-curvature**	0.9153	0.7251	0.7638	0.9612	0.0155	0.0847
SplitCriterion	**Gini**	0.9103	0.7295	0.7788	0.9507	0.0130	0.0897
SplitCriterion	**Deviance**	0.9173	0.7384	0.7784	0.9600	0.0129	0.0827
SplitCriterion	**Twoing**	0.9091	0.7248	0.7743	0.9505	0.0131	0.0909
Surrogate	**On**	0.9138	0.7377	0.7840	0.9537	0.0112	0.0862

Table 4.5: RF evaluation with different hyperparameters after maxDiffPSD removal

Hyperparam	Value	Acc	J	Sen	Spe	Train Err	Test Err
-	all default	0.9239	0.7606	0.7995	0.9611	0.0005	0.0761
InBagFraction	0.5	0.9219	0.7519	0.7908	0.9611	0.0005	0.0781
Cost	$\begin{bmatrix}0 & 1\\ 5 & 0\end{bmatrix}$	0.9144	0.7758	0.8413	0.9346	0.0142	0.0856
Sample-WithReplacement	Off	0.9214	0.7541	0.7948	0.9594	0.0002	0.0786
Numpredictors-ToSample	2	0.9194	0.7454	0.7852	0.9602	0.0005	0.0806
Numpredictors-ToSample	6	0.9199	0.7531	0.7967	0.9564	0.0005	0.0801
Numpredictors-ToSample	22	0.9191	0.7551	0.8010	0.9541	0.0004	0.0809
MinLeafSize	2	0.9225	0.7592	0.7993	0.9600	0.0081	0.0775
Prior	Uniform	0.9174	0.7679	0.8233	0.9446	0.0035	0.0826
MaxNumSplits	(Num-Observations)/2	0.9200	0.7519	0.7942	0.9577	0.0004	0.0800
MaxNumSplits	(Num-Observations)/4	0.9225	0.7591	0.8000	0.9591	0.0005	0.0775
MergeLeaves	Off	0.9221	0.7547	0.7943	0.9604	0.0006	0.0779
PredictorSelection	Curvature	0.9212	0.7575	0.7998	0.9577	0.0021	0.0788
PredictorSelection	Interaction-curvature	0.9189	0.7513	0.7953	0.9560	0.0018	0.0811
SplitCriterion	Gini	0.9228	0.7598	0.8003	0.9596	0.0006	0.0772
SplitCriterion	Deviance	0.9200	0.7541	0.7973	0.9567	0.0005	0.0800
SplitCriterion	Twoing	0.9244	0.7609	0.7988	0.9622	0.0005	0.0756
Surrogate	On	0.9205	0.7493	0.7895	0.9598	0.0004	0.0795

Table 4.6: RF evaluation with different hyperparameters after COV removal

On the dataset cleared from artifacts by the maxDiffPSD method, the RF algorithm had the best performance when PredictorSelection was set to Curvature. The RF algorithm had the best performance on the dataset cleared from artifacts by the COV method when SplitCriterion was set to Twoing. In the case of maxDiffPSD, the number of trees was set to 20 and in the case of the COV method to 100 trees.

4.4.2 The settings of artifact detection methods and their impact on RF classification

The performance of the RF algorithm, which classified data previously cleared from artifacts by one of the artifact removal methods with different parameter settings, was evaluated. Data from 24 patients were used in this part of the thesis. They were divided into training and testing data with a ratio of 7:1. The results were eight times cross-validated.

Firstly, artifacts were removed from the dataset using either the maxDiffPSD method or COV method with different threshold and window length values. The results from Section 4.3.2 about STN detection parameters were used to choose the best threshold and window length values. The results from 4.4.1 were used to set the RF hyperparameters for each method.

PredictorSelection was set to Curvature and number of trees was 20 when the artifacts were removed by maxDiffPSD method. SplitCriterion was set to

Twoing and number of trees was 100 when artifacts were removed by COV method. Evaluation results are shown in Tables 4.7, 4.8, respectively.

th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
0,0050	**0,6600**	0.8959	0.6854	0.7399	0.9456	0.0468	0.1041
0,0300	**0,6600**	0.9143	0.7318	0.7755	0.9562	0.0098	0.0857
0,0085	**0,6600**	0.9164	0.7365	0.7781	0.9584	0.0147	0.0836
0,0085	**1,0000**	0.9128	0.7271	0.7710	0.9561	0.0126	0.0872

Table 4.7: Impact of maxDiffPSD STN detection parameters on RF classification

th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
1.0250	**0.0100**	0.9209	0.7530	0.7938	0.9592	0.0003	0.0791
1.0500	**0.1600**	0.9180	0.7523	0.7985	0.9538	0.0006	0.0820
1.4000	**0.0800**	0.9203	0.7545	0.7974	0.9571	0.0004	0.0797
1.2000	**0.3300**	0.9198	0.7554	0.7992	0.9563	0.0006	0.0802

Table 4.8: Impact of COV STN detection parameters on RF classification

Lastly, the artifact detection and RF settings with the best performance results were used on the unseen signals from the remaining four patients. COV method with threshold 1.025 and window length 0.01 s was used to remove the artifacts. Afterwards, RF classification with SplitCriterion set to Twoing and 100 trees was used to classify STN. Final classification evaluation is shown in Figure 4.9.

method	th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
COV	**1.0250**	**0.0100**	0.9089	0.7829	0.8434	0.9395	0.0003	0.0911

Table 4.9: RF STN classification on unseen data

4.5 Support Vector Machine STN classification

The STN was classified by the Support Vector Machine algorithm. First, the dataset was divided into 8-fold cross-validation set containing signals from 24 patients and a test set containing signals from the remaining 4 patients. Then, the performance of the SVM algorithm, which classified data previously cleared from artifacts by one of the artifact removal methods with different parameter settings, was evaluated. Afterwards, the best artifact removal parameter settings were used to classify the STN by SVM classifier on the test set.

4.5.1 The settings of artifact detection methods and their impact on SVM classification

Support Vector Machine algorithm was also used to detect STN in MER signals. Data from 24 patients were used in this part of the thesis. It was divided into training and testing data with a ratio of 7:1. The results were eight times cross-validated.

Before actual classification, feature selection was applied to a set of features described in 3.1.4. This action aimed to reduce the number of input variables and eliminate the redundant features. Forward wrapper feature selection was used. The criterion function used, which evaluated the subsets of features, was an error of the SVM classifier with the RBF kernel.

The performance of the SVM algorithm, which classified data previously cleared from artifacts by one of the artifacts removal methods with different parameter settings, was evaluated. Evaluation results are shown in Tables 4.10 and 4.11. The final classification on unseen signals from four remaining patients is shown in Figure 4.12

th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
0.0050	**0.6600**	0.8770	0.5478	0.5859	0.9619	0.1263	0.1230
0.0300	**0.6600**	0.9121	0.7322	0.7822	0.9500	0.0840	0.0879
0.0085	**0.6600**	0.9141	0.7251	0.7693	0.9558	0.0837	0.0859
0.0085	**1.0000**	0.9158	0.7375	0.7831	0.9544	0.0833	0.0842

Table 4.10: Impact of maxDiffPSD STN detection parameters on SVM classification

th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
1.0250	**0.0100**	0.9194	0.7580	0.8043	0.9536	0.0729	0.0806
1.0500	**0.1600**	0.9040	0.7062	0.7589	0.9472	0.0729	0.0960
1.4000	**0.0800**	0.9190	0.7649	0.8154	0.9495	0.0786	0.0810
1.2000	**0.3300**	0.9198	0.7554	0.7992	0.9563	0.0006	0.0802

Table 4.11: Impact of COV STN detection parameters on SVM classification

th	winLen (s)	Acc	J	Sen	Spe	Train Err	Test Err
1.2000	**0.3300**	0.9093	0.8067	0.8866	0.9202	0.0773	0.0907

Table 4.12: SVM STN classification on unseen data

4.6 Comparison of STN classifiers

4.6.1 Comparison of NRMS thresholding, RF and SVM classifier

The cross-validation set containing signals from 24 patients was used to derive the best settings for RF and SVM algorithms. The results were eight times cross-validated. The process of obtaining the best settings for classification by the Random forest algorithm is described in 4.4.1 and 4.4.2. The same process for the SVM algorithm is described in 4.5. The best settings were then used to classify the STN on the test set.

The test set contained previously unseen signals from four patients. For the RF, the artifacts were removed from the dataset by the COV method with a 1.025 threshold and window length of 0.01 s and the hyperparameter SplitCriterion was set to Twoing. For the SVM the threshold 1.2 and window length 0.33 s was used.

Results on the test set of both RF classification and SVM classification were compared to the results of NRMS thresholding with the best parameter settings from Section 4.3.2 and are presented in Table 4.13. Simple NRMS thresholding has satisfactory results for STN classification. Higher accuracy can be achieved by RF and SVM classifiers. The biggest benefit RF and SVM classifiers bring is their sensitivity, which is considerably higher than the sensitivity of the NRMS thresholding.

Classifier	Artifact detection			Results			
	method	th	winLen (s)	Acc	J	Sen	Spe
NRMS thresholding	**COV**	1.0250	0.0100	0.9071	0.6759	0.7207	0.9553
RF	**COV**	1.0250	0.0100	0.9089	0.7829	0.8434	0.9395
SVM	**COV**	1.2000	0.3300	0.9093	0.8067	0.8866	0.9202

Table 4.13: Comparison of NRMS thresholding, RF and SVM classifier

Table 4.13, which contains values of accuracy, sensitivity and specificity for NRMS thresholding, RF and SVM classification, shows that SVM had the highest accuracy, Youden-J and sensitivity, while the specificity of NRMS thresholding was higher than the specificity of the rest.

Chapter 5
Conclusion

The goal of this thesis was to implement and evaluate a set of classifiers for automatic detection of the subthalamic nucleus in microelectrode recordings. Microelectrode recordings were recorded during deep brain stimulation surgery performed on patients with Parkinson's disease. The focus of the thesis was on the influence of microelectrode artifacts and their precise detection and removal. Moreover, an optimal processing pipeline for microelectrode recordings with respect to the accuracy of implemented classifiers for identifying the brain nuclei was found.

After an initial study of signal preprocessing methods, artifact removal and nuclei classification from microelectrode recordings, a method evaluating the influence of correct artifact detection on STN detection was implemented. Two different artifact detection methods (autocorrelation and maximal spectral difference method) were compared using different method settings. Normalized RMS power was the feature according to which the nuclei were classified using the ROC curve. Respective AUC and Youden values were computed, and the accuracy of the classification was derived by using the best threshold to classify the data. Overall, the COV method achieved the highest accuracy of 90.71 % with threshold and window length parameters set to 1.025 and 0.01 s, respectively. The accuracy of STN detection was then compared to plain artifact detection using one of the mentioned methods.

Subsequently, two other STN classifiers were implemented and evaluated on a set of computed features. The first algorithm was the Random Forest (RF), for which the best hyperparameter settings were derived. Secondly, the Support Vector Machine (SVM) was evaluated using feature selection. Selection of features computed from corresponding MER signals was applied to reduce the number of input features and remove redundant values.

Classifiers were evaluated on a set consisting of 8322 microelectrode recordings and were compared with respect to achieved accuracy, Youden-J, sensitivity and specificity. It was concluded that simple normalized RMS thresholding could acquire satisfactory results if the artifacts are removed from the set correctly. Slightly better classification results can be achieved by RF or

SVM classifiers. NRMS thresholding can achieve an accuracy of 90.71 %, RF achieved an accuracy of 90.89 %, and the SVM algorithm achieved the best results of 90.93 % accuracy. The most significant benefit RF and SVM classifiers bring is their sensitivity, which is considerably higher than NRMS thresholding.

To conclude, the subthalamic detection task can be solved using relatively easy implementation. The precise removal of artefacts has the has high impact on nuclei classification performance, as opposed to the choice of classifier or hyperparameter settings. If artifacts are present, they can significantly influence the classification results. When artifact removal methods are set, the detection task should be kept in mind, as different settings achieve the best results for artifact detection and nuclei detection. For artifact detection, the maxDiffPSD method has an accuracy of 86.29 % when the threshold is set to 0.1 and the window length is set to 0.1 s. In the STN detection task, the COV method has an accuracy of 90.71 % when the threshold is set to 1.025 and the window length is set to 0.01 s.

Even though the goal of this thesis was achieved, there is plenty of room for improvement. If a larger dataset consisting of manual nuclei annotation from different neurologists could be obtained, it would be possible to study the classification deeper. A different set of classifiers could be used to detect the nuclei and could be compared. Also, a different set of features could be computed. Consequently, a new artifact removal method could be implemented and compared to COV and maxDiffPSD spectral method.

Bibliography

Aboy, Mateo and J. Haakon Falkenberg (June 2006). “An Automatic Algorithm for Stationary Segmentation of Extracellular Microelectrode Recordings”. In: *Medical & Biological Engineering & Computing* 44 (6), pp. 511–515. ISSN: 0140-0118. DOI: 10.1007/s11517-006-0052-2. URL: http://link.springer.com/10.1007/s11517-006-0052-2.

Anna, Konicarová Carmen (2021). *Klasifikace subthalamického jádra z mikroelektrodových záznamů*. URL: https://dspace.cvut.cz/handle/10467/94427.

Bakstein, Eduard, Jakub Schneider, et al. (Aug. 2015). “Supervised segmentation of microelectrode recording artifacts using power spectral density”. In: vol. 2015-November. IEEE, pp. 1524–1527. ISBN: 978-1-4244-9271-8. DOI: 10.1109/EMBC.2015.7318661. URL: https://ieeexplore.ieee.org/document/7318661/.

Bakstein, Eduard, Tomas Sieger, et al. (2016). “Probabilistic Model of Neuronal Background Activity in Deep Brain Stimulation Trajectories”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9832 LNCS, pp. 97–111. ISSN: 16113349. DOI: 10.1007/978-3-319-43949-5_7. URL: http://link.springer.com/10.1007/978-3-319-43949-5_7.

Bakštein, Eduard et al. (Oct. 2017). “Methods for automatic detection of artifacts in microelectrode recordings”. In: *Journal of Neuroscience Methods* 290, pp. 39–51. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2017.07.012. URL: https://linkinghub.elsevier.com/retrieve/pii/S0165027017302492.

Benabid, Alim Louis (Dec. 2003). “Deep brain stimulation for Parkinson’s disease”. In: *Current Opinion in Neurobiology* 13 (6), pp. 696–706. ISSN: 09594388. DOI: 10.1016/j.conb.2003.11.001. URL: https://linkinghub.elsevier.com/retrieve/pii/S0959438803001739.

Benabid, Alim Louis et al. (Jan. 2009). “Deep brain stimulation of the subthalamic nucleus for the treatment of Parkinson’s disease.” In: *The Lancet. Neurology* 8 (1), pp. 67–81. ISSN: 1474-4422. DOI: 10.1016/S1474-4422(08)70291-6. URL: http://www.ncbi.nlm.nih.gov/pubmed/19081516.

Bishop, Christopher M (2006). "Pattern Recognition and Machine Learning (Information Science and Statistics)". In.

Bloem, Bastiaan R, Michael S Okun, and Christine Klein (June 2021). "Parkinson's disease". In: *The Lancet* 397 (10291), pp. 2284–2303. ISSN: 01406736. DOI: 10.1016/S0140-6736(21)00218-X. URL: https://linkinghub.elsevier.com/retrieve/pii/S014067362100218X.

Bonaccorso, Giuseppe (2017). "Machine Learning Algorithms Reference guide for popular algorithms for data science and machine learning". In: URL: www.packtpub.com.

Breiman, Leo (Oct. 2001). "Random Forests". In: *Machine Learning 2001 45:1* 45 (1), pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324. URL: https://link.springer.com/article/10.1023/A:1010933404324.

Breiman, Leo and Adele Cutler (2004). *Random forests - classification code*. URL: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_software.htm.

Breiman, Leo, Jerome H. Friedman, et al. (Jan. 1984). "Classification and regression trees". In: *Classification and Regression Trees*, pp. 1–358. DOI: 10.1201/9781315139470/CLASSIFICATION-REGRESSION-TREES-LEO-BREIMAN-JEROME-FRIEDMAN-RICHARD-OLSHEN-CHARLES-STONE. URL: https://www.taylorfrancis.com/books/mono/10.1201/9781315139470/classification-regression-trees-leo-breiman-jerome-friedman-richard-olshen-charles-stone.

Breit, Sorin, Jorg B. Schulz, and Alim-Louis Benabid (Oct. 2004). "Deep brain stimulation". In: *Cell and Tissue Research* 318 (1), pp. 275–288. ISSN: 0302-766X. DOI: 10.1007/s00441-004-0936-0. URL: http://link.springer.com/10.1007/s00441-004-0936-0.

Cagnan, Hayriye et al. (Aug. 2011). "Automatic subthalamic nucleus detection from microelectrode recordings based on noise level and neuronal activity." In: *Journal of neural engineering* 8 (4), p. 046006. ISSN: 1741-2552. DOI: 10.1088/1741-2560/8/4/046006. URL: http://www.ncbi.nlm.nih.gov/pubmed/21628771.

Cortes, Corinna and Vladimir Vapnik (Sept. 1995). "Support-vector networks". In: *Machine Learning* 20 (3), pp. 273–297. ISSN: 0885-6125. DOI: 10.1007/BF00994018. URL: http://link.springer.com/10.1007/BF00994018.

Dolan, Kevin et al. (July 2009). "Automatic noise-level detection for extracellular micro-electrode recordings". In: *Medical & Biological Engineering & Computing* 47 (7), pp. 791–800. ISSN: 0140-0118. DOI: 10.1007/s11517-009-0494-4. URL: http://link.springer.com/10.1007/s11517-009-0494-4.

Dougherty, Geoff (Aug. 2013). *Pattern recognition and classification: An introduction*. Vol. 9781461453239. Springer New York, pp. 1–196. ISBN: 9781461453239. DOI: 10.1007/978-1-4614-5323-9.

Duda, Richard O., Peter E. Hart, and David G. Stork (2012). "Pattern Classification". In: 2012, p. 680. URL: http://books.google.com/books?hl=en&lr=&id=Br33IRC3PkQC&pgis=1.

Falkenberg, J.H. et al. (2003). "Segmentation of extracellular microelectrode recordings with equal power". In: vol. 3. IEEE, pp. 2475–2478. ISBN: 0-7803-7789-3. DOI: 10.1109/IEMBS.2003.1280417. URL: http://ieeexplore.ieee.org/document/1280417/.

Feigin, Valery L et al. (Nov. 2017). "Global, regional, and national burden of neurological disorders during 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015". In: *The Lancet Neurology* 16 (11), pp. 877–897. ISSN: 14744422. DOI: 10.1016/S1474-4422(17)30299-5. URL: https://linkinghub.elsevier.com/retrieve/pii/S1474442217302995.

Gardner, John (Oct. 2013). "A history of deep brain stimulation: Technological innovation and the role of clinical assessment tools". In: *Social Studies of Science* 43 (5), pp. 707–728. ISSN: 0306-3127. DOI: 10.1177/0306312713483678. URL: http://journals.sagepub.com/doi/10.1177/0306312713483678.

Groiss, S.J. et al. (Nov. 2009). "Review: Deep brain stimulation in Parkinson's disease". In: *Therapeutic Advances in Neurological Disorders* 2 (6), pp. 379–391. ISSN: 1756-2864. DOI: 10.1177/1756285609339382. URL: http://journals.sagepub.com/doi/10.1177/1756285609339382.

Gross, Robert E. et al. (June 2006). "Electrophysiological mapping for the implantation of deep brain stimulators for Parkinson's disease and tremor". In: *Movement Disorders* 21 (S14), S259–S283. ISSN: 0885-3185. DOI: 10.1002/mds.20960. URL: https://onlinelibrary.wiley.com/doi/10.1002/mds.20960.

Guillén, P et al. (Aug. 2011). "Characterization of subcortical structures during deep brain stimulation utilizing support vector machines." In: *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference* 2011, pp. 7949–52. ISSN: 2694-0604. DOI: 10.1109/IEMBS.2011.6091960. URL: http://www.ncbi.nlm.nih.gov/pubmed/22256184.

Jankovic, J (Apr. 2008). "Parkinson's disease: clinical features and diagnosis". In: *Journal of Neurology, Neurosurgery & Psychiatry* 79 (4), pp. 368–376. ISSN: 0022-3050. DOI: 10.1136/jnnp.2007.131045. URL: https://jnnp.bmj.com/lookup/doi/10.1136/jnnp.2007.131045.

Karthick, P.A. et al. (Sept. 2020). "Automated detection of subthalamic nucleus in deep brain stimulation surgery for Parkinson's disease using microelectrode recordings and wavelet packet features". In: *Journal of Neuroscience Methods* 343, p. 108826. ISSN: 01650270. DOI: 10.1016/j.jneumeth.2020.108826. URL: https://linkinghub.elsevier.com/retrieve/pii/S0165027020302491.

Loh, Wei-Yin (2002). "REGRESSION TREES WITH UNBIASED VARIABLE SELECTION AND INTERACTION DETECTION". In: *Statistica Sinica* 12, pp. 361–386. URL: http://lib.stat.cmu.

Martinez, Juan et al. (Nov. 2009). "Realistic simulation of extracellular recordings". In: *Journal of Neuroscience Methods* 184 (2), pp. 285–293.

ISSN: 01650270. DOI: 10.1016/j.jneumeth.2009.08.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S0165027009004506.

MATLAB (2020a). *Create bag of decision trees - MATLAB (Version 2020a).* Natick, Massachusetts. URL: https://www.mathworks.com/help/stats/treebagger.html.

— (2020b). *Sequential feature selection using custom criterion - MATLAB sequentialfs.* URL: https://www.mathworks.com/help/stats/sequentialfs.html.

— (2020c). *Train support vector machine (SVM) classifier for one-class and binary classification - MATLAB fitcsvm.* URL: https://www.mathworks.com/help/stats/fitcsvm.html.

Moran, Anan et al. (Sept. 2006). "Real-time refinement of subthalamic nucleus targeting using Bayesian decision-making on the root mean square measure." In: *Movement disorders : official journal of the Movement Disorder Society* 21 (9), pp. 1425–31. ISSN: 0885-3185. DOI: 10.1002/mds.20995. URL: http://www.ncbi.nlm.nih.gov/pubmed/16763982.

Noyce, Alastair J et al. (Dec. 2012). "Meta-analysis of early nonmotor features and risk factors for Parkinson disease." In: *Annals of neurology* 72 (6), pp. 893–901. ISSN: 1531-8249. DOI: 10.1002/ana.23687. URL: http://www.ncbi.nlm.nih.gov/pubmed/23071076%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3556649.

Perlmutter, Joel S. and Jonathan W. Mink (July 2006). "DEEP BRAIN STIMULATION". In: *Annual Review of Neuroscience* 29 (1), pp. 229–257. ISSN: 0147-006X. DOI: 10.1146/annurev.neuro.29.051605.112824. URL: https://www.annualreviews.org/doi/10.1146/annurev.neuro.29.051605.112824.

Probst, Philipp and Anne Laure Boulesteix (May 2017). "To tune or not to tune the number of trees in random forest?" In: *Journal of Machine Learning Research* 18, pp. 1–8. ISSN: 15337928. DOI: 10.48550/arxiv.1705.05654. URL: https://arxiv.org/abs/1705.05654v1.

Probst, Philipp, Marvin N. Wright, and Anne-Laure Boulesteix (May 2019). "Hyperparameters and tuning strategies for random forest". In: *WIREs Data Mining and Knowledge Discovery* 9 (3), e1301. ISSN: 1942-4787. DOI: 10.1002/widm.1301. URL: https://onlinelibrary.wiley.com/doi/10.1002/widm.1301.

Ranganathan, Shoba et al. (2019). *Encyclopedia of Bioinformatics and Computational Biology.* Ed. by Shoba Ranganathan et al. Vol. 3. Academic Press, pp. 938–952. ISBN: 978-0-12-811432-2.

Ritz, Beate et al. (Oct. 2014). "Parkinson disease and smoking revisited: ease of quitting is an early sign of the disease." In: *Neurology* 83 (16), pp. 1396–402. ISSN: 1526-632X. DOI: 10.1212/WNL.0000000000000879. URL: http://www.ncbi.nlm.nih.gov/pubmed/25217056%20http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4206154.

Russel, Stuart and Peter Norvig (2012). *Artificial intelligence—a modern approach 3rd Edition.* ISBN: 0136042597. DOI: 10.1017/S0269888900007724.

Scarmeas, N. (Oct. 2001). “Cortical Basal Ganglionic Degeneration”. In: *Science of Aging Knowledge Environment* 2001 (1), 1dn–1. ISSN: 1539-6150. DOI: 10.1126/sageke.2001.1.dn1. URL: http://sageke.sciencemag.org/cgi/doi/10.1126/sageke.2001.1.dn1.

Sinica, Published In Statistica et al. (1997). “Split Selection Methods for Classification Trees”. In: URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.23.1815.

Stacey, William C. et al. (Aug. 2013). “Potential for unreliable interpretation of EEG recorded with microelectrodes”. In: *Epilepsia* 54 (8), pp. 1391–1401. ISSN: 00139580. DOI: 10.1111/epi.12202. URL: https://onlinelibrary.wiley.com/doi/10.1111/epi.12202.

Tolosa, Eduardo, Gregor Wenning, and Werner Poewe (Jan. 2006). “The diagnosis of Parkinson’s disease.” In: *The Lancet. Neurology* 5 (1), pp. 75–86. ISSN: 1474-4422. DOI: 10.1016/S1474-4422(05)70285-4. URL: http://www.ncbi.nlm.nih.gov/pubmed/16361025.

Wong, S et al. (Apr. 2009). “Functional localization and visualization of the subthalamic nucleus from microelectrode recordings acquired during DBS surgery with unsupervised machine learning.” In: *Journal of neural engineering* 6 (2), p. 026006. ISSN: 1741-2552. DOI: 10.1088/1741-2560/6/2/026006. URL: http://www.ncbi.nlm.nih.gov/pubmed/19287077.

Zaidel, Adam et al. (Sept. 2009). “Delimiting subterritories of the human subthalamic nucleus by means of microelectrode recordings and a Hidden Markov Model”. In: *Movement Disorders* 24 (12), pp. 1785–1793. ISSN: 08853185. DOI: 10.1002/mds.22674. URL: https://onlinelibrary.wiley.com/doi/10.1002/mds.22674.

Appendix A

Technical details

This Appendix contains the description of files implemented during this Bachelor's thesis. All of them were implemented in MATLAB R2020a. The repository contains those folders :

- `thresholdingNRMS`:
  - implementation of the NRMS thresholding STN classification
  - the implementation can be run using the file `nrmsValidate.m`
- `SVM` :
  - implementation of the STN classification using SVM using feature selection
  - the implementation can be run using the file `svmValidate.m`
- `RF`:
  - implementation of the STN classification using RF
  - the implementation can be run using the file `rfValidate.m`