Czech
Technical
University
in Prague

Faculty of Electrical Engineering
Department of Computer Science

Diploma thesis

Automatic artifact detection in micro-EEG signals

Bc. Tomáš Grubhoffer

January 2016
Thesis supervisor: Ing. Eduard Bakštein

České vysoké učení technické v Praze
Fakulta elektrotechnická

katedra počítačů

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Tomáš Grubhoffer**

Studijní program: Otevřená informatika
Obor: Umělá inteligence

Název tématu: **Automatická detekce artefaktů v mikro-EEG signálech**

Pokyny pro vypracování:

1) Seznamte se s problematikou mikroelektrodových záznamů, souvisejících s hlubokou mozkovou stimulací [1].

2) Vytvořte systém pro automatickou detekci a segmentaci artefaktů pro signály získané pomocí intrakraniálních mikroelektrod.

3) Vlastnosti modelu otestujte na databázi mikroelektrodových záznamů pacientů s Parkinsonovou nemocí a porovnejte chování Vámi navrženého modelu s publikovanými přístupy [2-4] napříč signály z různých jader.

Seznam odborné literatury:

[1] J.H. Falkenberg, J. McNames, M. Aboy, K.J. Burchiel: "Segmentation of extracellular microelectrode recordings with equal power", in proceeding of: Engineering in Medicine and Biology Society, 2003.

[2] Aboy M, Falkenberg JH.: "An automatic algorithm for stationary segmentation of extracellular microelectrode recordings. "Med Biol Eng Comput. 2006 Jun;44(6):511-5

[3] Guarnizo, C.; Orozco, A.A.; Castellanos, G.,: "Microelectrode Signals Segmentation Using Stationary Wavelet Transform," BioMedical Engineering and Informatics, 2008.

[4] Israel, Z, Burchiel, KJ: "Microelectrode Recordings in Movement Disorder Surgery", Thieme New York, USA, 2004, ISBN 1-58890-172-4

Vedoucí: Ing. Eduard Bakštein

Platnost zadání: do konce letního semestru 2015/2016

České vysoké učení technické v Praze
Fakulta elektrotechnická
L.S.
-4-

doc. Ing. Filip Železný, Ph.D.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 26. 3. 2015

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....

Podpis autora práce

Acknowledgement

I would like to thank my supervisor for his knowledge, advices and patience during the whole project. Also, I would like to thank my family and my girlfriend for their moral support during the study.

Abstrakt

Tato práce popisuje algoritmus pro detekci artefaktů v mikro EEG signálech (MER), které se používají pro lokalizaci jader v mozku užívanou při hluboké mozkové stimulaci (DBS) pro pacienty s Parkinsonovou chorobou.
Představujeme metodu pro detekci artefaktů, která používá rozhodovací strom. Pravidla pro rozhodovací strom byla vytvořena na příznacích založených na časovém průběhu a frekvenčním spektru MER signálů. Náš klasifikátor jsme porovnali s metodami pro detekci artefaktů, které existují v dostupné literatuře. Přesnost rozhodovacího stromu byla 90,38% na trénovací databázi a 86,33% na testovací databázi. Na validačních datech byla rovna 86,33%. Ostatní metody dosáhly přesnosti kolem 77% na trénovací databázi a 80% na testovací databázi.

Abstract

This thesis proposes an algorithm for artifact detection in microelectrode recordings (MER) which are used for the localization of nuclei in Deep Brain Stimulation (DBS) for Parkinson's disease patients. We present a method for artifact detection which uses decision tree. Rules of the decision tree were based on features based on observed temporal and spectral properties of MER artifacts. We have compared our classifier with methods for artifact detection which already exist in the available literature. The accuracy of the decision tree was 90,38% on the training dataset and 86,33% on the testing dataset. On the validation dataset it was equal to 86,33%. Other methods achieved accuracy of about 77% on the training dataset and 80% on the testing dataset.

Contents

List of Figures

List of Tables

1 Introduction

Parkinson's disease (PD) is one of the most common neurodegenerative disorders. There does not exist any cure for PD yet, but drugs containing dopamine can be used as a treatment for PD patients. However, when the PD advance, this treatment becomes less effective.

The deep brain stimulation procedure (DBS) of the subthalamic nucleus (STN) is used as a treatment of advanced PD. One of the most important stages of implantation of the DBS aparatus is the localization of nuclei, which will be used as a target of the stimulation. During the implantation the surgeon records brain activity of the patient using microelectrodes. These records are called micro-electrode recordings (MER). More information about the PD and the structure of the MER is provided in Chapters 2 and 3 of this Thesis. Apart from localization during the surgery, MER signals are often used for a variety of research purposes - it is after all a very rare occasion to record signals directly from the human brain and study its activity at a high level of detail.

One of the problems of the MER processing is dealing with the noise and artifacts, which cause devaluation of the MER signal. They are caused for example by electromagnetic interference from the mains and electrical devices in the operating theatre. Some of the noise can be already removed during the recording using a high-pass filter (for frequencies, which are above 500 Hz) and a low-pass filter (for frequencies below 5kHz). However, some of the noise could still remain in the MER and could affect the nuclei positioning accuracy and also the results of subsequent signal analyses in research studies. For example the spike sorting algorithm, which is used for the detection of individual neurons contributing to the signal could become inaccurate when noise peaks are present in the signal. Detailed description of the noise and the tests of the filtration of this noise is given in the Chapter 4.

The detection of the artifacts in the MER is commonly performed manually, but it can be used for smaller databases only. Also, the manual detection of artifacts may become inaccurate due to inconsistency between detection carried out manually by different people.

Several methods of automated detection of the artifacts already exist in available literature. Some of the methods use statistical tests as a detector of discrepancies in the MER signal. An example of automatic detection is given in [1], where authors split MER into smaller windows, calculated Root Mean Square Value (RMS) and then use analysis of variance algorithm (ANOVA) to compare RMS values and determine whether the MER contains an artifact or not.

Another method is based on finding the longest stationary segment in the processed signal. In this approach, the MER is divided into non-overlaping segments for which a variance is calculated to a ratio. After that, the ratio is compared between neighboring segments and if the value of the ratio exceeds a threshold, then the segment is annotated. An example of stationary segmentation is given in [2, 3] and [4]. The only difference between cited approaches is the method used for the calculation of the variance. An extension of cited methods is presented in [5], where author is comparing all segments between each other (segments may not be neighboring). Exact steps of these methods can be found in Chapter 5.

This thesis describes another method of the detection of the artifacts in the MER. We decided to create a decision tree, which can be used to determine whether an artifact is present in the MER or not. Decision tree is a very common classifier in data processing. The process of learning is described in Chapter 6.

For the decision tree we had to extract features from the raw MER signals. The features were then used as rules in the decision tree. These features were focused on the characteristic behavior of the artifacts in the MER. Features were divided into three groups - 1) Features extracted from the original input, 2) Features extracted from the signal power spectrum and 3) Features extracted from the signal after gaussian filtration. Detailed description of these features can be found in Chapter 8.

We also needed data from which the decision tree could be learned and evaluated. We have used signals which were recorded during the microelectrode exploration phase of DBS surgery. In a group of several researchers we performed initial manual annotation of these recorded signals. The final annotation was based on the majority voting. The steps of the annotation are described in Chapter 7. We have prepared 5739 annotated signals, which can be used for learning the decision tree. The learning data were separated into three datasets - 1) Training dataset (60% of data), 2) Testing dataset (30%) and 3) Validation dataset (10%) which was used as the final validation of decision tree classification accuracy.

In Chapter 9 we present the results of our tests performed on the generated decision tree. We also present a comparsion with the methods that use stationary segmentation.

2 Parkinson's disease and Deep brain stimulation

2.1 Parkinson's Disease

Parkinson's disease (PD) [6, 7] is a neurodegenerative disorder of the central nervous system. It affects the motoric system of the body. In short, PD is caused by death of cells in the substantia nigra (SNr) which are producing neurotransmitter called dopamine. With less dopamine, person has a lower ability to control his movements, body or emotions.

The main symptoms in the early course of PD are related to the motoric system of the body. Most common symptoms include ”tremor” (shaking movement) and with later stages of the disease, psychical problems, like thinking or behavioral problems, may arise. Other symptoms are related to sleep, sensoric or emotional problems. The parkinson's disease is more prevalent among older people (>50 years) but it may also occur with adults (21-40). Then it is called Young onset parkinson's disease.

Nowadays, there is no cure for PD, there are just treatment options such as medication or surgery to manage the symptoms described above. Treatments for this disease can be efective especially for the motoric symptoms of the disease. Treatments for the early state of the disease usually consists of consuming dopamine agonists. In later phases of PD the medication treatments may become ineffective. In that case, the deep brain stimulation (DBS) can be used for further treatment of PD.

2.2 Deep Brain Stimulation

Deep brain stimulation (DBS)[8, 9, 10, 11] is a neurosurgical therapy for intractable movement disorders and is being explored in a growing number of neurological and behavioral disorders. The DBS procedure consists of the implantation of a brain pacemaker and electrodes. RTG image of a DBS system is shown in Figure 1. The brain pacemaker sends electrical impulses through the electrodes and using that, it stimulates specific area of the brain. The development of DBS has rapidly grown over the last years.

Understanding of the electrical circuitry in the brain is very important for DBS. Using this knowledge we are able to localize specific nodes of the brain which are responsible for neurosurgical disorders. These nodes can be used as targets for DBS. Nowadays, the common targets for DBS are the subthalamic nucleus (STN) for PD, the globus palidus internus (GPi) for dystonia and PD, and the ventralis inermedius nucleus of the thalamus (VIM) for essential tremor. Appropriate candidate for the DBS surgery is a person with later stage of PD. The expected benefit of DBS must outweight the potential risk associated with the surgical implantation of DBS.

The DBS surgery itself consists of two stages - planning (preoperative) stage and intraoperative stage. In the planning stage of the surgery an anatomical point of the

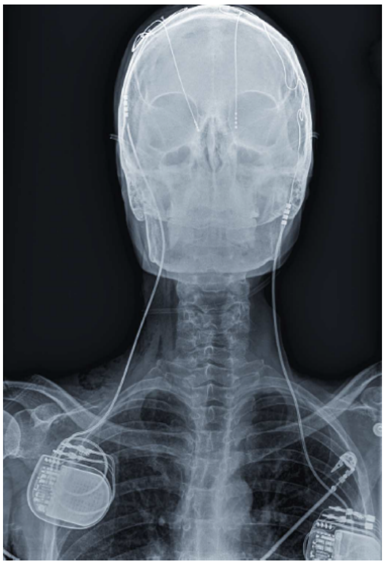


Figure 1: Example of an implantation of a DBS device [8]

brain is defined in 3-D space. The targeting of this point is based on a Cartesian coordinate system, which defines specific point in 3-D space. This involves visualization of deep nuclei using high definition magnetic resonance imaging (MRI). It is important to plan the surgery to minimize damage caused to the brain during the implantation itself. In the second stage, the DBS lead is placed in the defined nuclei using stereotactic apparatus as shown in Figure 2.

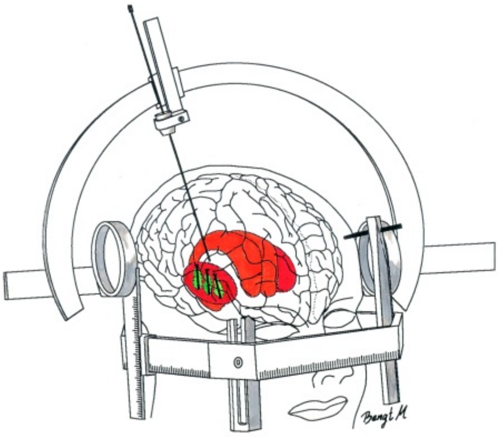


Figure 2: Stereotactic surgery head device [12]

To confirm the correct location of the target structure, a set of microelectrodes is used to record neuronal activity in the neighborhood of the assumed target. These recordings represents brain activity around the electrodes. The correct placement of the leads is confirmed when the microelectrode records display characteristic patterns of targeted nuclei.

3 Micro-EEG Signal

In this chapter, we would like to describe Microelectrode recordings (MER) in more detail. MER signals [13, 14], sometimes called extracellular recordings, are used for medical and scientific purposes. In medicine, MER are used for targeting nucleis (especially the STN) during DBS implantation (as we described in previous chapter) or for epilepsy treatment. In neuroscience, MER signals are used to study brain activity on the level of individual neurons.

In our case, the MER signals used for our work come from real patients during implantation of DBS devices. The signals were measured using five electrodes during the surgery and recorded using sampling rate of 24 kHz. These signals were commonly 10s long.

3.1 Neuron activity

For further description of extracellular recordings, we need to define electrical activity of a neuron. A neuron is a cell which is the core component of the nervous system. This cell is electrically excitable and it processes information through electrical and chemical signals. These signals are carried by synapses between neurons. Neurons are connected to each other to form neural networks. Standard neuron consists of dendrites which propagates the electrochemical signal from another neurons, soma which is the body of a neuron, nucleus and axon which transmits the electrical information to different neurons. There are several models of electrical activity of a neuron. The most widely known are Hodgkin–Huxley's [15], which is conductance based model used to simulate a single neuron activity, or Izhikevich's spiking neuron model [16] illustrated in Figure 3, which is used to simulate spiking of a neuron.

The electrical activity of a neuron differs through the nuclei of the brain mostly by the firing patterns, which can be characterized for example by firing frequency or occurence of bursts (very fast sequence of spikes) - see Figure 3 for examples of firing.

3.2 MER structure

The MER signals represent the brain activity around the electrode. All neurons in the close vicinity of the electrode contribute to the extracellular recordings. Based on the amplitudes in the signal, the extracellular recordings can be separated into individual components. These components are the spiking activity, the background noise and the external noise.

The spiking activity represents the fires of single neurons, which we have described above. There can be none to several close neurons depending on the situation around the electrode. The distance and the orientation of the neurons affects the recorded

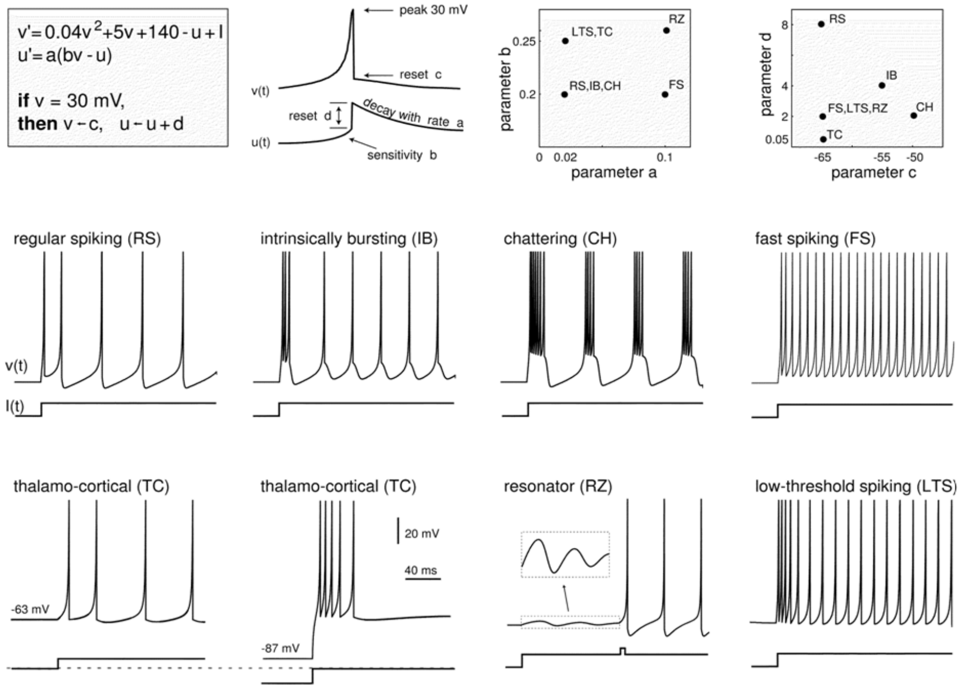


Figure 3: Izhikevich's model of spiking neurons
The examples of the known types of neurons are presented in this picture from [16]. These neurons can be easily simulated using four parameters (a,b,c and d). By using these parameters the shape and the frequency of spikes can be managed.

spike shape which can then be used for spike sorting algorithm, which classify spikes of neurons with similar behavior or shape. Based on these similarities we are able to define which brain nuclei is presented by the extracellular recording. The structure of MER signal can be viewed in Figure 4.

The backround noise can be defined as static neural background activity in the brain (long-distance neighborhood). Apart from the background noise, the MER signal may contain also non-physiological noise, caused by mechanical electrode shifts or electromagnetic interference. We refer to this phenomenon as external noise or artifacts further in the text. It is also important to mention that the external noise has nothing in common with the neuronal background of the brain.

3.3 Preprocessing and Analysis Methods

In here we would like to specify preprocessing and analysis steps undertaken before the start of the external noise analysis. As mentioned above, signals were recorded from real patients using parallel insertion of five microelectrodes. These recordings were filtered using band-pass filter which was set to let frequencies between 500 Hz and 5 khz through. These recordings were then sampled at 24 kHz. Next step of the signal preprocessing procedure is to separate single units from the background noise. Spike

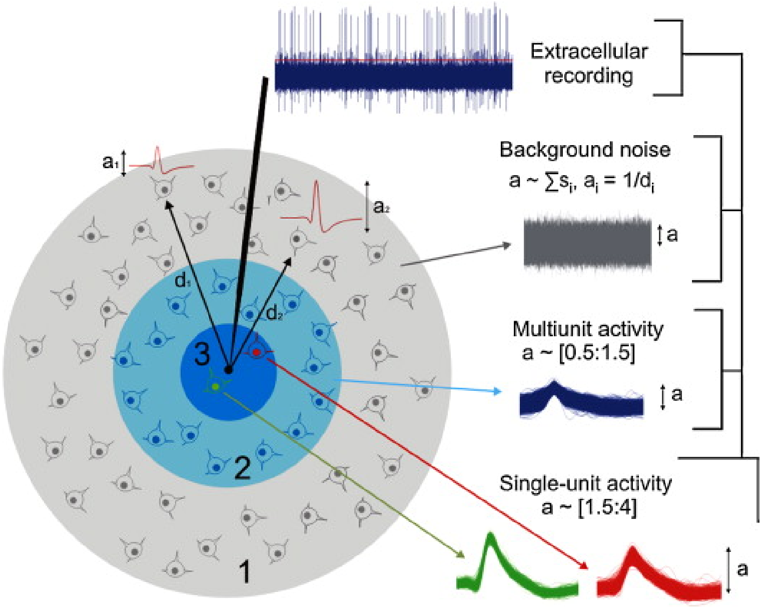


Figure 4: Structure of extracellular recording

This figure represents a model of brain activity in [14]. In our concept of the brain activity we have merged multi-unit and background activities into bigger group called spiking (neural) activity. The background noise represents noise activity in areas further away from the electrode. In our model, we added one more group called external noise which represents noise which is not caused by activity in the brain.

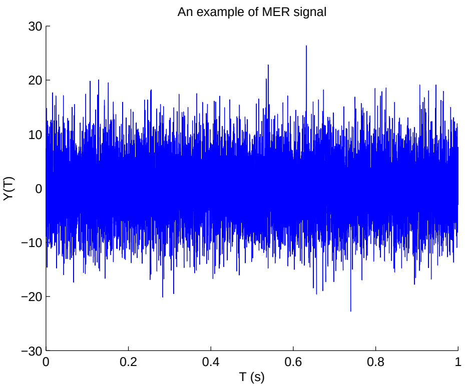


Figure 5: An example time course of processed signal

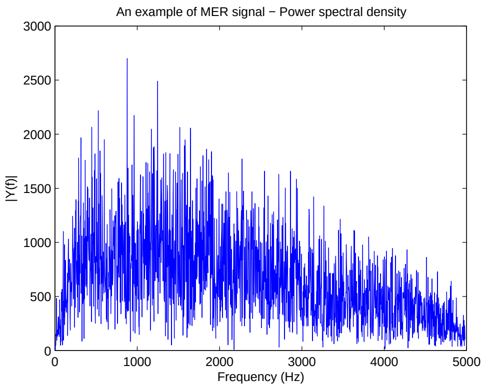


Figure 6: Power spectral density of the example signal

detection algorithm [17] is used for this separation. This algorithm simply detects spikes of individual neurons. In most cases, it is done by using an amplitude threshold which is calculated as a multiplicity of standard deviation of the raw signal samples. After the spikes of individual neurons are detected, the spike sorting algorithm is used (eg. WaveClust algorithm [18]). Spike sorting algorithm divides spikes into groups based on their shape. Spikes in different groups then represent action potentials of different neurons. Spike sorting is commonly used for evaluation of single neuron activity, mainly for research purposes (for example evaluation of changes of firing patterns in different areas of the brain or before/after stimulus). In the case of DBS targetting, spike sorting procedure is typically not required. Instead, a trained neurosurgeon evaluates the signal subjectively to identify positions where the electrodes enter or exit the target structure.

4 External noise in MER signals

This chapter is focused on external noise in extracellular recordings. As mentioned in previous chapter, these recordings consist of neural activity in the brain (in example single unit activity of nearby neurons and background noise) and external noise from the environment. In here, we would like to specify external noise behavior and characteristics.

4.1 Noise sources

The main source of external noise in MER signals is the electromagnetic interference from the mains and electrical devices in the operating theatre. These devices are also emiting electromagnetic waves which cause interference with MER signals. This effect can cause massive damage on MER signals during the implantation, and it occurs as high frequency component in the signal. This type of noise can be seen as a clear peak in the frequency spectrum of the signal. However, 50Hz signal can be notch filtered during the DBS operation. Another type of the noise are low frequency signals - for example small vibrations caused by the moving of the DBS electrodes. This type of external noise is demonstrated by peaks in lower frequencies of the power spectrum of the signal. Another example of low frequency noise is the talking of the doctor or other backround sound - in this case the noise is probably mechanically transmitted (the electrodes are shaking). The next source of the noise is the DBS apparatus itself. Especially the DBS electrodes can irritate the area around them and because of that, the neurons in the neighborhood area start to fire more rapidly. This type of noise is not so common but it is very hard to distinguish from physiological activity. Therefore, we did not focus on it during our research.

4.2 Description of MER artifacts

For further processing of the extracellular recordings we decided to separate types of noise into smaller groups with similar features. In following text we will call noise types artifacts. The main reason for the division was a better understanding of the individual artifacts. Groups were created based on similar behavior of artifacts in the extracellular recordings and were based on possible filtration of individual artifact types. Another reason was the frequency of appereance of the individual artifacts. We have created following artifact groups - power artifact, frequency artifact and baseline artifact. For the unspecified noise we created additional artifact group called other. We present each artifact group in the following paragraphs and provide detailed description, their specific behavior in signal and filtration tests aimed at their removal. For better understanding of the presented pictures, we recommend to compare the signals containing artifacts with the clean extracellular recordings described in previous chapter .

4.2.1 Power Artifact

The first group we have defined is the power artifact. This group consists of three similar noise types - 1) mechanical noise in the backround of the DBS procedure, 2) short-term electromagnetic artifact caused by electrical devices and 3) another specific types such as sounds - for example talking of the doctor. Features of these types of noise are very similar to each other, therefore we have merged them into one bigger group. The main feature of this artifact is high energy of the signal (artifact is usually represented as high signal gain). Thanks to this property, the power artifacts can be easily detected by visual inspection of the signal. Second approach for the detection of this artifact is to inspect the signal power spectrum. In the most cases, the power artifact consists of several strong frequency components, which can be easily detected by visual inspection of the signal power spectrum. An example of signal containing power artifact and its power spectrum is given in Figure 7.

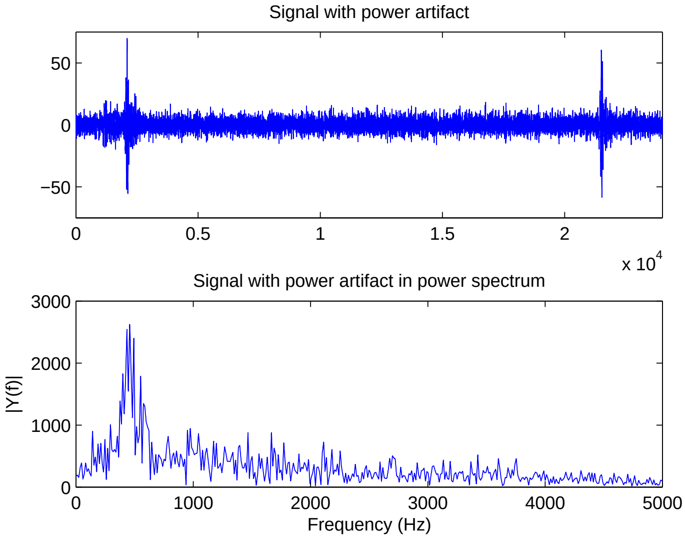


Figure 7: Signal with the power artifact waveform and power spectrum

As seen in the picture above, the power artifact is mostly short-term.

4.2.2 Frequency Artifact

Another artifact group we specified are the technical artifacts at stable frequency, which will be abbreviated to FREQ in the rest of this text. Source of the FREQ noise are the devices used during the operation. These devices can emit electromagnetic waves which cause appereance of high frequency signals in MER signal - for example 450Hz or 500Hz signals are very common. It is very hard to detect FREQ artifacts by visual inspection of the original MER signal. Better detection can be made in the signal power spectrum where the artifact is dominated by narrow peaks of high power, superimposed on the typical MER spectrum. Another common approach of detection is to listen to the original signal - FREQ artifact should be hearable. Example of artifact in the signal and the power spectrum is given in Figure 8.

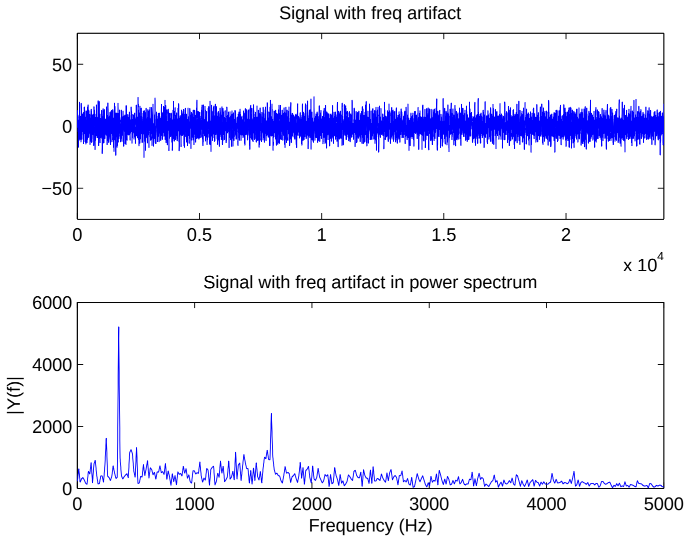


Figure 8: Signal with the FREQ artifact waveform and power spectrum

After deeper investigation, there were found few similar types of this artifact. One of the features which differs among the frequency artifacts is the number of narrow peaks in the power spectrum. The MER signals can contain several FREQ artifacts. Due to the similarity with the power artifact, this artifact caused problems in the initial annotation.

4.2.3 Baseline Artifact

The last group of artifacts are the baseline artifacts. The main source of the baseline artifact is the mechanical floating of the DBS electrodes. baseline artifacts are described as low frequency signals. The frequency of these signals is between 0 and 200 Hz (based on the preprocessing methods described above). Characteristic behavior of this type of artifact can be seen in Picture 9.

The peaks in the power spectrum of the signal with baseline artifact has similar shapes as the signals with frequency or power artifacts. Peaks are present in lower frequencies. This is the main difference from the other artifacts. During the initial processing of the data, we discovered that the occurence of the baseline artifact is often accompanied by other artifacts. For example strong frequency artifacts could invoke baseline artifacts.

4.2.4 Other Artifacts in MER signals

We would like to mention a few events in the signal which were not clearly classified in any one of previous groups. In most cases, the artifacts in this group are presented as a

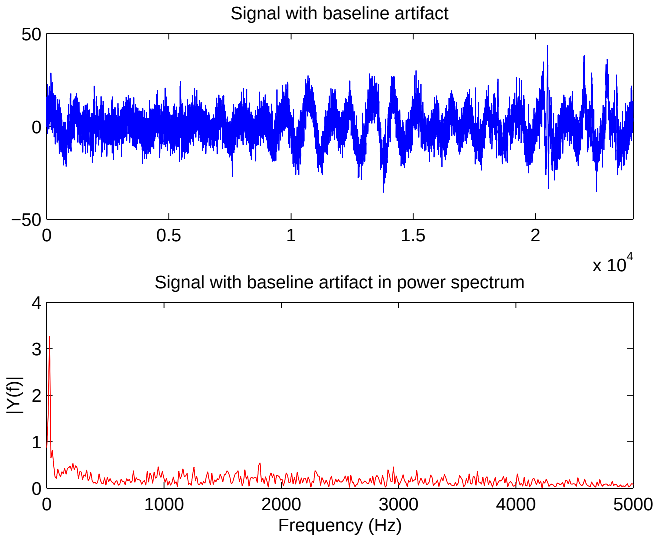


Figure 9: Signal with baseline artifact waveform and power spectrum

combination of the artifacts described above. In additional special case of other events are the neurons which are irritated by the DBS electrodes. It behaves as a fast change of frequency of neuron spikes in the waveform. Irritated neurons are very uncommon and so it is very hard to detect these types of events in the signal and it can be very often mistaken for clean signals. Therefore, we have not focused on these events while designing the automatic artifact detector.

4.3 Filtration of MER Artifacts

As mentioned above, one of the reasons of the artifact separation was to check the possibility of filtration. The main part of the filtering tests was focused on the frequency spectrum.

4.3.1 Filtration of Power artifact

The power artifact is represented by high signal gain, often outstretched over the wide band. We determined that the power artifacts are unfilterable, because it is localized in the time domain rather than in the signal power spectrum.

4.3.2 Filtration of Frequency artifact

The main symptom of FREQ are the sharp peaks in signal power spectrum on specific frequencies. The frequencies of these artifacts are usually between 100Hz and 1kHz. Peaks under 100Hz are considered as a symptom of baseline artifacts. Often, we encounter steady noise at varying frequencies around 400 Hz, which can not be easily filtered using a fixed filter. The problem is that the energy of these artifacts is very big and this method could damage and cause devaluation of the original signal. Theoretically the frequency artifacts could be removed by a comb filter (for example set on 50Hz windows), but similarly to the previously described approach, this method could cause devaluation of the original MER signal.

4.3.3 Filtration of Baseline artifact

As mentioned in previous sections, the baseline artifact is defined as a low frequency signal which is present in MER signal. This feature was the first to suggest that it would be possible to filter the baseline artifact from the original signal. We prepared two options of filtration for this type of artifact. The first one is to cut low frequencies from the signal using filtration in the frequency spectrum. This method consists of performing fast fourier transform, removing low frequencies from the spectrum (respectively setting the values of spectral bins corresponding to these frequencies) and performing inverse fourier transform. Second proposed method is to design a standard high-pass filter in time domain. This solution could potentially be much faster than the previous method. After initial tests, we discovered that this method could cause more devaluation to the original signal. The main problem was the initial ”filling” of the filter, which may cause artifacts at both ends of the signal.

We have tested both of these methods on artifically created baseline artifact. At first we selected a signal which did not contain any artifact and summed it with artifically generated baseline artifact. The baseline artifact was simulated by the following function:

$$y = 4\sin(2F_n\pi t) \tag{1}$$

$$F_n = 6/T\,[Hz] \tag{2}$$

$$T = 24000\,[s] \tag{3}$$

where F_n is the frequency of sinus signal, T is the duration of the signal (same as the original signal). The waveform of signal prepared by the sum of the baseline artifact and MER signal can be seen in Figure 10.

For better understanding of the created baseline artifact behavior in power spectrum we also present the spectrum of the created signal in Figure 11.

Now, since we have prepared signal with artifical baseline artifact, we started to test our methods. For the first method of filtration we have calculated the power spectrum of the signal (by fast fourier transformation) and when we had the power spectrum, we set the values of the spectrum to zero for frequencies from 0 to 80Hz. After that, we performed inverse fourier transform. The results of this method are given in Figure 12. As we discovered, this method was very effective. There was no considerable damage caused to the original signal meaning the loss of information. For the second method we prepared a high-pass filter. For the design of this filter we have used $fdatool$ function implemented in matlab. In the $fdatool$ GUI, the user is able to enter filter variables and instantly display the filter response. We have set the values as follows:

$$Fs = 24000, FStop = 10, FStart = 80, AStop = 80, APass = 0.4. \tag{4}$$

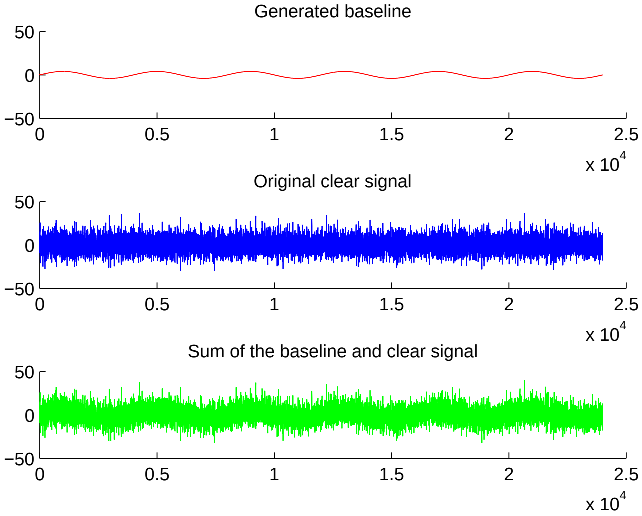


Figure 10: Manual generation of Baseline artifact

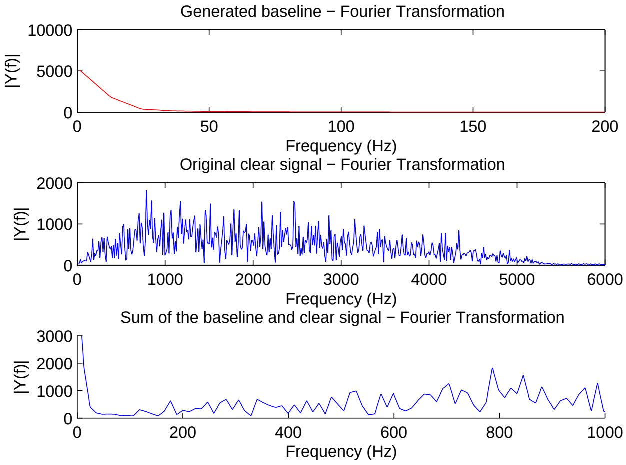


Figure 11: Manual generation of Baseline artifact (Fourier Transform)

The magnitude response for this filter can be seen in Figure 13. By exporting this filter into our code we were able to test the response on artifically created signal with baseline artifact. The results can be viewed in Figure 14.

The problem of this method was the amount of damage dealt to the signal while filling up the filter. This behavior can be seen at the start of the signal where the values were set to zero. At the end, we have compared results of filtration between both methods and the original clear signal. We visualized the signal in the power spectrum where the differencies between methods and the original signal are presented in Figure 15.

The results for both methods were similar. The fourier transform cut is slower than the high-pass filter but thanks to the filter filling, the second method caused more de-

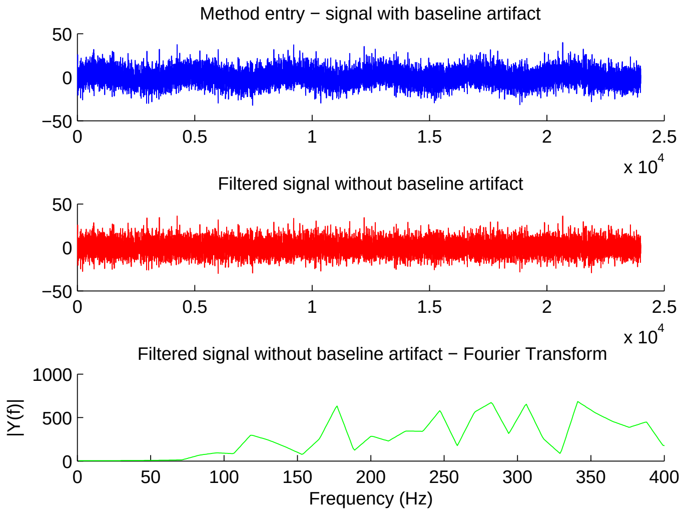


Figure 12: Filtration: First Method - Fourier Transform cut

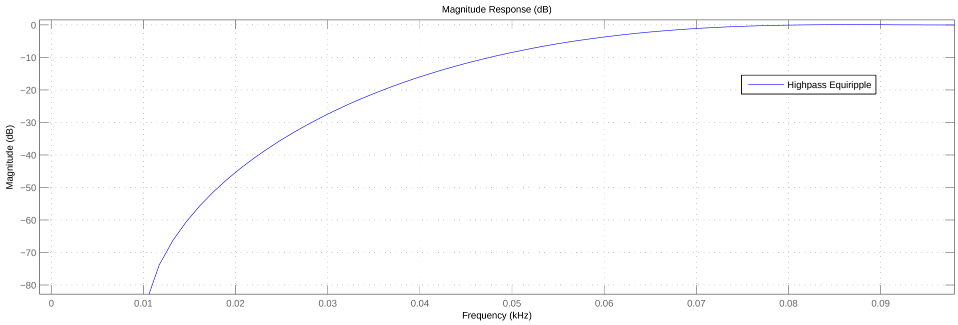


Figure 13: Filtration: Second Method - Filter response

valuation to the original signal. Considering properties of both methods (and especially the fact that temporal filtering is much faster in the case of long signals), we would prefer the high-pass filter, which can be added behind the system measuring MER signals during the DBS operation.

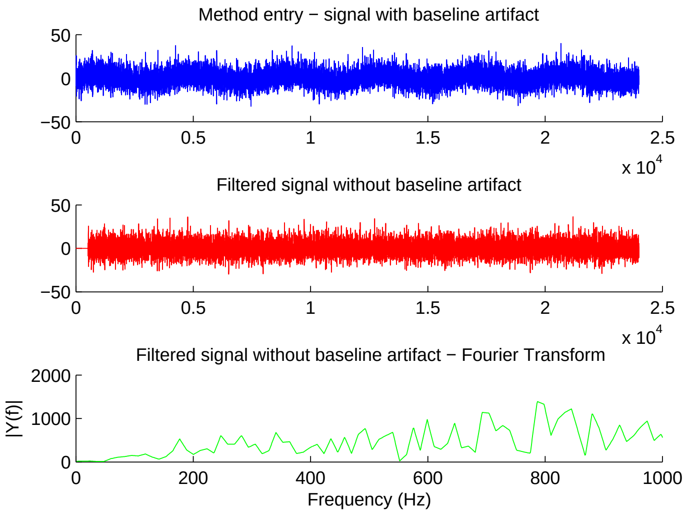


Figure 14: Filtration: Second Method - Results

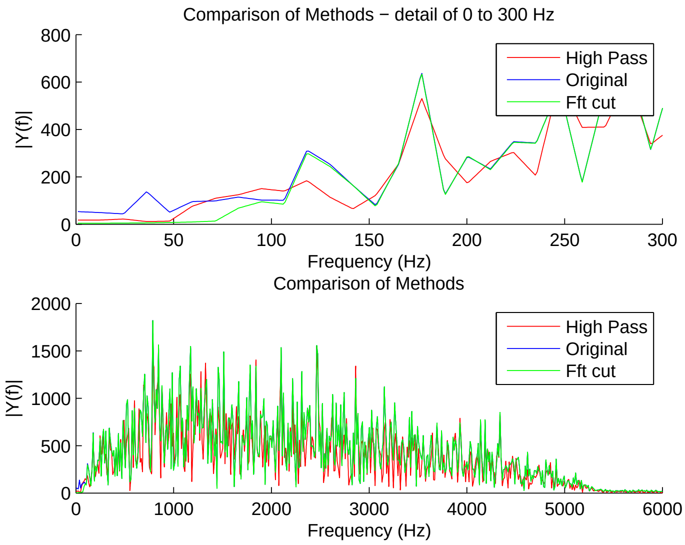


Figure 15: Filtration: Method comparison

5 Existing Methods for automatic detection

Several methods for artifact detection already exist in available literature. The simplest way of artifact classification is manual detailed inspection of the signal. This method can detect the strongest appearances of the artifacts. Weaker artifacts can be easily overlooked and are identified only with high effort. Even though this method is widely adopted in the literature, this approach was unsuitable due to the high number of recordings in our database (more than 16000 10-second recordings). For that reason we have focused on the automatic classification.

In this chapter we would like to present existing methods of automatic noise detection in the extracellular recordings. These methods can be divided into two groups - simple statistical test and signal segmentation.

5.1 Statistics Method

The first method which we want to present is noise detection using the Root Mean Square (RMS) measure [1]. It is a simple statistical test which uses RMS value to detect discrepancies in the signal. The RMS value is calculated as follows:

$$RMS(\hat{X}) = \sqrt{\frac{\sum\limits_{i=1}^{n} X_i^2}{n}}. \tag{5}$$

Where $\hat{X}$ is sampled analog signal (to 24kHz, same as in our case), X_i^2 is squared value of each sample and n is the total number of samples. Authors sampled signal to one hundred non-overlapping windows of the signal, each one with 20 miliseconds duration.

The signal is processed by two-step signal stability test. At first, the sessions with amplitudes exceeding a threshold are considered as signals containing an artifact and are rejected. In the second, step the ANOVA (analysis of variance algorithm) is used to compare RMS values of the 20 miliseconds non-overlaping windows from the first and the last two seconds of the signal. If there are significant differencies then the signal is rejected. This algorithm is very simple, but because of comparation of the first and the last seconds of the signal, it is also not very precise.

5.2 Stationary segmentation

Another approach of automatic artifact detection is based on finding the longest stationary segment in the processed signal. At first, the input signal is segmented into smaller non-overlaping segments. Then for each of the segments, the variance is calculated to form a ratio. Then the ratio for all neighboring segments is checked and if it exceeds given threshold, the border of these segments is marked as a transition. In the last step, the algorithm returns the longest signal segment with no marked transitions.

The first method which uses stationary segmentation is [2] presented by Falkenberg and Aboy. Their approach is based on the variance of the autocovariance approach.

First, MER signal (denoted as x) is normalized and sampled into S non-overlapping segments.

$$x_n = \frac{x - \mu_x}{\sigma_x}. \tag{6}$$

Where μ_x is signal mean and σ_x is the standard deviation of the signal. The non-overlapping frames $\{Z_l\}_{l-=1}^{S}$ have equal time duration t specified by the user. The set of segments is defined as follows.

$$\{Z_l\}_{l-=1}^{S} = \{Z_1, Z_2, ..., Z_S\}. \tag{7}$$

Next step is the calculation of the autovariance. For all of the frames $\{Z_k\}_{l-=1}^{S}$, the autocorelation sequence $r_x(l)$ is estimated using the biased estimator of autocorelation $\hat{r}_x(l)$.
For each of the segments, the variance of the covariance $\hat{\gamma}_x(0)$ is estimated.

$$var\{\hat{\gamma}_x(0)\} = \frac{2}{L} \sum_{l=-\infty}^{\infty} \hat{\gamma}_x^2(l) = \frac{2\sigma^4}{L} \sum_{l=-\infty}^{\infty} \hat{r}_x^2(l). \tag{8}$$

The result of the variance calculation is a vector with the variances of the autocovariances v for each of the segments from $\{Z_k\}_{l-=1}^{S}$.

$$v = [var\{\hat{\gamma}_{Z_1}(0)\}, var\{\hat{\gamma}_{Z_2}(0)\}, ..., var\{\hat{\gamma}_{Z_P}(0)\}]^T = (v_1, v_2, ..., v_p)^T. \tag{9}$$

After that, the algorithm uses statistics F-test to compare variances of all of the neighboring segments.

$$F = (\frac{v_{N_1}}{v_{D_1}}, ..., \frac{v_{N_k}}{v_{D_k}}, ..., \frac{v_{N_{P-1}}}{v_{D_{P-1}}})^T = (F_1, F_2, ..., F_k, ..., F_{P-1})^T. \tag{10}$$

$$v_{N_k} = max(v_k, v_{k+1})_{k=1}^{P-1}. \tag{11}$$

$$v_{D_k} = min(v_k, v_{k+1})_{k=1}^{P-1}. \tag{12}$$

When the F value for neighboring segments is bigger than the threshold value F_C, the algorithm records a transition for neighboring segments which it represents. The last step of the algorithm is the selection of the longest segment with no transitions. The output signal can be used for further processing.

Similar approach was used by Guarnizo [4]. The only difference between this approach and the method described above is the calculation of the variance for the segment. Here, the variance is not calculated from autocorelation function but from the stationary wavelet transform (SWT) of the signal. Guarnizo used Haar's wavelet (see Figure 16) for the transformation. After the transformation of the signal, the approach is same as in [2]. Variances for each neighboring segments are calculated and the longest

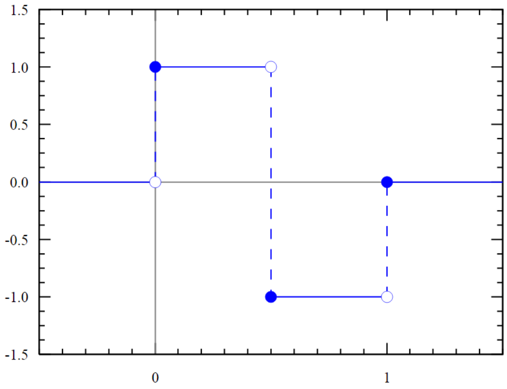


Figure 16: Haar's wavelet

signal segment with no transitions is found.

An extension of described algorithms is presented in [5]. In the previous methods only neighboring segments were compared. Here, all possible segment pairs are compared with each other to form a distance matrix D.

$$D = \begin{pmatrix} 0 & d_{12} & \cdots & d_{1,m-1} \\ d_{21} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ d_{m-1,1} & d_{m-1,2} & \cdots & 0 \end{pmatrix} \tag{13}$$

Values which exceed given threshold are then replaced by ones, the rest by zeros. Then, by using a greedy algorithm, the longest signal segment is found in the distance matrix (the longest sequence of zeros).

The comparison of results of the method presented by Falkenberg, the method presented by Guarnizo and our classification algorithm can be found in Chapter 9.

6 Automatic detection using Decision tree

In this chapter we would like to describe our classification method for detecting artifacts in MER signals. For classification we have used well-known and popular clasifier called Decision tree. Decision tree is a common method used for data classification due to its simplicity and wide support across data processing and statistical software packages. Another reason why are decision trees popular is that they do not behave as a black-box method. Thanks to that, the decision tree is very comprehensible.

Decision tree [19, 20, 21, 22] is a logic tree structure which combines a sequence of simple tests. Each branch represents logical outcome of a feature test, and each leaf represents class label predictions. Logical rules in decision tree are very simple for example in comparsion with the weights calculation in neural networks. An example of decision tree is given in Figure 17.

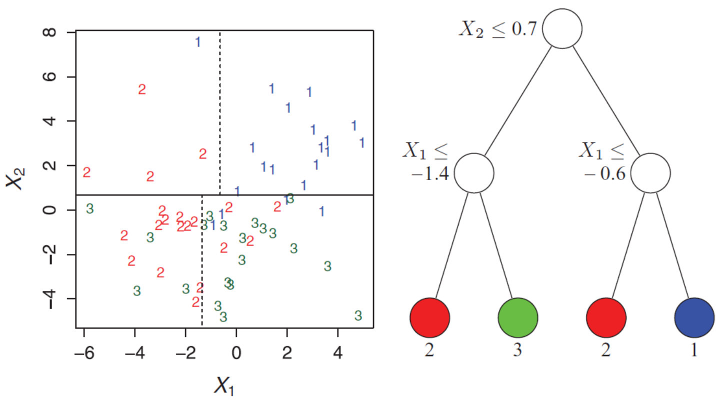


Figure 17: An example of simple decision tree

This is an example of simple decision tree taken from [22]. On the left side we have data set with measured parameters X_1 and X_2. We would like to classify this dataset into three groups (red, blue, green). Using decision tree data can be classified into these groups by simple logical rules.

To make our classifier simpler, we focused on the binary classification - whether the signal contains or does not contain any artifacts. In future implementation we would like to create a classifier, which will also determine the type of detected artifact in the signal as described in previous chapters based on generated logical rules.

The input into the descision tree structure is a set of parameters calculated for a signal. The output is the estimation whether the signal contains or does not contain artifacts.

6.1 Methods used for Decision tree learning

One of the main problems of the decission tree creation is the over-fitting - generated tree will classify data into groups with low error rate for given training database, but with high error for testing database. Several methods exist to avoid the over-fitting problem. Method which we have used is the tree pruning. Tree pruning is an algorithm, which determines which leaf nodes of the tree decrease the classification accuracy of the dataset. There exists two pruning methods - pre-pruning and post-pruning.

Pre-pruning is a method which determines whether some of the branches should not be terminated. This happens during the generation of the decision tree. A termination rule is required for this method. Post-pruning is a method which removes branches to increase the classification after the build phase of the decision tree.

We have used post-prunning method in our algorithm, because it does not need any termination rule. We determined the optimal prune level of the tree T by using the 10-fold cross-validation [23] for determining the minimum cost subtree [24], which will be used as the final classifier. In the cross-validation method, the data are split into k sets $S = S_0, ..., S_k$ each named as fold. These sets have same size. The decision tree learning algorithm is performed k times - data set $j = 1, ..., k$ is taken and used as the testing set, the other sets are merged and used as the training set for the tree.

For the finding of the minimum cost subtree all subtrees are defined as $T < T_{max}$. The number of leafs of T is defined as $|\widetilde{T}|$. Let $\alpha \geq 0$ be the real number called the complexity parameter. The minimum-cost subtree $T(\alpha)$ is found by minimizing the cost complexity measure $R_\alpha(T)$:

$$R_\alpha(T(\alpha)) = min_{T<T_{max}} R_\alpha(T). \tag{14}$$

Where $R_\alpha(T)$ represents the cost complexity measure and is calculated as follows:

$$R_\alpha(T) = R(T) + \alpha|\widetilde{T}|. \tag{15}$$

With more leafs, the complexity of the tree becomes higher. The complexity parameter α is used for setting the importance which will be put on the size of the tree. The cost complexity is then penalized by the error rate. This error rate needs to be minimized when prunning the tree. If there are only finitely many subtrees, the minimum cost subtree exists for any α.

The pre-processing of the input data is also an option to increase decision tree accuracy. For example, finding the optimal number of the paramateres used for decision tree learning. Another option is to split the training dataset into two parts. Normally, the training set is used for both initial tree generation, and for the minimum cost subtree finding. For the split training data, the first part is used to build an initial decision tree and the second part is then used for finding the minimum cost subtree. We have compared results for both of these methods in the Chapter 9.

6.2 Learning of the Decision tree

The tree learning consists of selection of parameters used for the feature tests and actual spliting conditions - in our case threshold values. Selection of the parameters is the most important part of the decision tree generation. In our case, the parameters were based on actual features of the MER signals and artifacts. It is important to select parameters which are the most informed for given set of data. With parameters which are less informed, the decision tree will become less acurate in the classification. Selected parameters are described in Chapter 8. At first, we need to calculate values of the parameters for the training set of the decision tree. In short, two matrices P and A are required for decision tree learning. P can be described as:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{pmatrix}. \tag{16}$$

P is a matrix which contains calculated values of m parameters p for n MER signals. Second matrix A describes the anotations a for all of the training signals:

$$A = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix}. \tag{17}$$

$a \in \{0, 1\}$, where 0 stands for clean signal and 1 for signal which contains artifact.

The spliting conditions are generated during the building phase of the decision tree. They are calculated from given training set to classify it the most precisely. The most informed parameters are selected by building algorithm for the upper parts of the tree, less informed parameters are used for the bottom parts. The cross-validation method, described in the previous section, is used during the generation to avoid over-fitting on this training set. Based on that, the actual values of the spliting conditions are recalculated or replaced by another feature test of different parameters. In the end of the tree generation, we perform post-pruning to make the logical classification more precise.

7 Data

In this chapter, we would like to describe the process of database preparation for the automatic classificator. This process consists of three steps - the data acquisition from patients, the manual annotation of measured data and training and testing data selection. These three steps are described in following sections in more detail.

7.1 Data Description

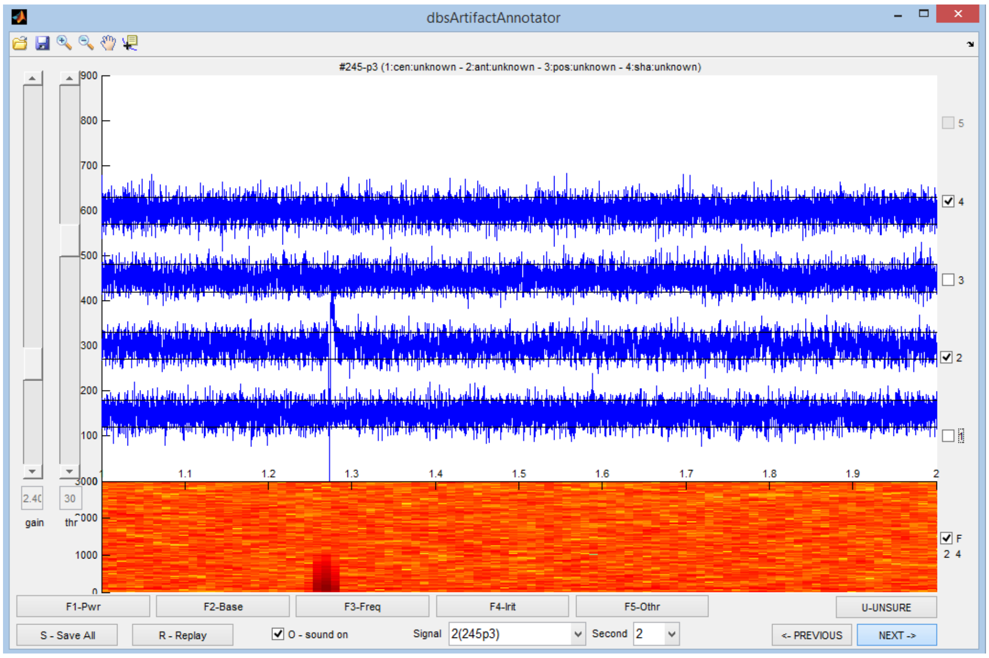
Signals were taken from Parkinson's disease patients during the microelectrode exploration phase of DBS surgery. The phases of DBS were already described in Chapter 2. Patients were awake and resting during the implantation of the electrodes and meanwhile, the surgeon recorded signals from the basal ganglia. The source nucleus of each signal was identified and labeled. The measured nuclei were: STN (subthalamic nucleus), SNr (substantia nigra), Th (thalamus) and "unknown". The "unknown" nucleus stands for other undiferentiated parts of the basal ganglia. In total, the initial dataset consisted of 18384 signals. These signals were available for further processing. Most of them (12672) were taken from the "unknown" nucleus, and 4319 of the signals were taken from the TN nucleus (STN). The recording length of most signals was 10s (some of the signals were shorter) and the sampling frequency was 24 kHz.

7.2 Database Annotation

At this point, we would like to describe the annotation (manual labelling of artifacts) of the initial dataset. A group of 8 people investigated a selected sample of MER data in detail. Based on the observations, initial artifact groups were created. These groups were already described in Chapter 4. Artifact groups were based on similar characteristics of microelectrode recordings and on the posibility of their filtration. Initial artfact groups were: power, baseline, frequency and undefined, which stood for unclear artifacts such as combinations of others.

With all of the artifacts described, we were prepared for the annotation of the signals. The problem was that the manual investigation of all of the microelectrode signals could be very slow - it was necessary to investigate the signals one by one and inspect each signal separately. For faster investigation of the microelectrode recordings an unique Matlab GUI tool was created as seen in Figure 18.

In this GUI, the user was able to select a database of the signals, which he would like to annotate. Each member of the annotation group had his own database which partially overlapped with other users' databases. Where multiple annotations were available (overlapping data), the final annotation was based on the majority voting of all users.



☑ O - sound on

Signal 2(245p3)

Second 2

Figure 18: DBS Artifact Annotator User Interface

Each user was able to select each of the microelectrode recordings from his database based on the specific id (id of the patient and electrode position). When a recording was selected, signals for all of the electrodes used for DBS of selected patient and position were displayed in the GUI. Below the signals from the electrodes a spectrogram was displayed. Spectrum was very useful for visual artifact detection. For example the frequency noise (from electrical devices) could be easily detected. By using check marks on the right side of the electrodes signals, the user could select ones which he wanted to be displayed in the spectrum. The user could annotate the selected electrodes signals' seconds by clicking the buttons with appropriate name of the found artifact. Also, by using these buttons, the user was able to deannotate these signals. The user can also play the signal back as audio, which is a common method used by neurophysiologists during MER investigation. Keyboard shortcuts were included in the GUI to make the annotation even faster. The current annotation of selected database could be saved into a mat file and then restored from it.

Now to the annotation itself. 5739 signals were given to the annotation group. That gives 57390 signals seconds which were processed and annotated. Each member of the group got a part of these data together with a control set (950-1240 seconds of MER) for the better control of the annotation. After the annotation, seconds from the control set were checked across the group and annotated using the majority voting. That means that the final artifact annotated to the signal must have been found by more users (five in our case). Seconds which were not a part of the joint database were classified as the user annotation.

Here, we would like to present the annotation results of the group. As was already mentioned, the annotation group consisted of eight people which were trained on the visual and audio features of all of the artifacts. Each member of this group had his own dataset, which he had to annotate. 950 seconds of distributed data were the same for all of the users. On this dataset we have measured specificity (true negative rate), sensitivity (true positive rate) and total accuracy of the individual users against the majority voting. These values for the classification group can be viewed in Table 1.

Member of the group	1	2	3	4	5	6	7	8
Specificity	0,99	0,96	0,94	1,00	0,90	0,91	0,86	0,90
Sensitivity	0,08	0,92	0,97	0,64	0,98	0,96	0,75	0,97
Accuracy	0,78	0,95	0,95	0,97	0,92	0,92	0,84	0,92

Table 1: Annotation results (8 members against the majority voting)

Member 1 had a very low sensitivity on the majority voting set, which caused poor accuracy. Member 4 had also a very low sensitivity, but his accuracy was good. Member 7 had a poor accuracy and it seemed that the artifact annotation was consistent (member annotated same number of artifacts). The values of specificity, sensitivity and accuracy were good for the rest of the users. Because of that we decided to exclude members 1,4 and 7 from the annotation group. Their signals were distributed between the rest of the group. Values for the new classification group can be viewed in Table 2. In Figure 19 we present the member agreement on the individual artifacts. In Figure 20 we present the distribution of the annotation for each of the member.

Member of the group	2	3	5	6	8
Specificity	97.7%	93.4%	95.8%	96.3%	90.9%
Sensitivity	83.9%	90.6%	96.1%	82.4%	96.1%
Accuracy	94.0%	92.6%	95.9%	92.5%	92.3%

Table 2: Annotation results (5 members against the majority voting)

As can be seen in Figure 20 the most common artifact in the signals was the FREQ. The total number of baseline and power artifacts was very similar. The least common group of artifacts was the irritated neuron group. The irititated neuron was very rarely found in the data. Also it was very difficult to detect this kind of artifact in the signals due to its hight similarity to physiological neuronal activity. There were several discussions on this artifact type and it was decided not to focus on it. As a result, the automatic artifact detector was not able to detect the irritated neuron artifact.

The number of the other artifacts was also very low - members usually annotated the other artifact as the major artifact which was present in the signal (for example FREQ). Another problem were the weak occurences of some artifacts which were

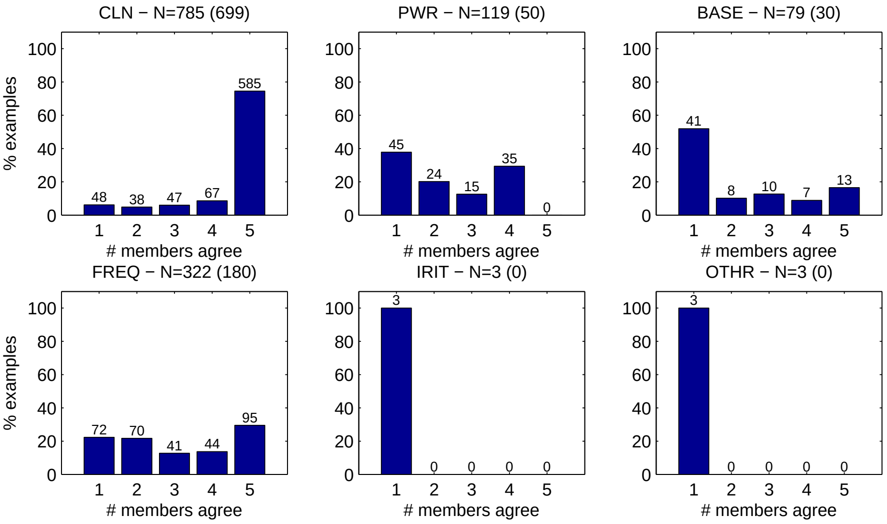


Figure 19: Member agreement over majority voting set

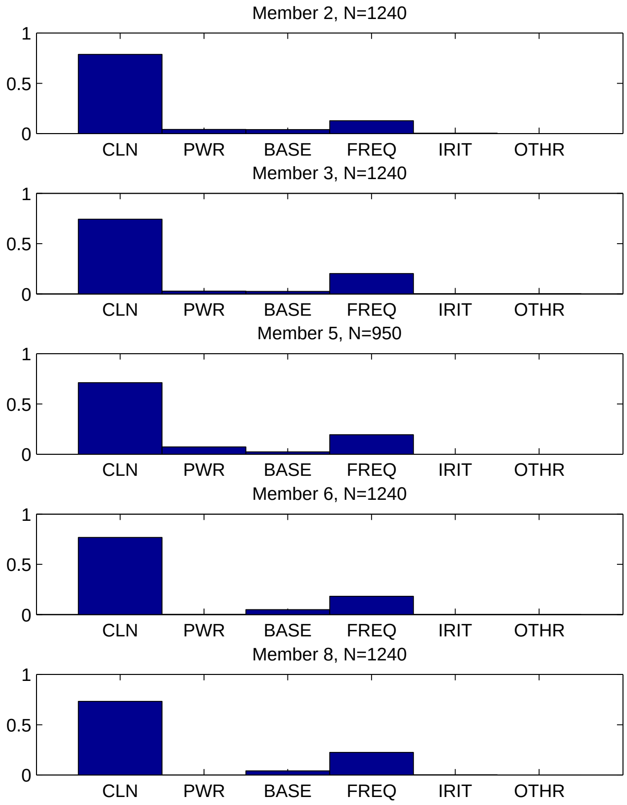


Figure 20: Member annotation of majority voting set

annotated as clean by some of the users and as artifacts by others. Based on the majority voting, the signal could remain clean even though there was a obvious presence of the artifact. All of these discrepancies can negatively affect the decision tree learning.

Closer analysis of the results of the annotation showed, that there were heavy discrepancies in annotation of the power artifact. The main problem was its similarity to the frequency artifact. In many cases, the power artifact was annotated as FREQ which lead to discrepancies in the database. Also, some of the clean signals were annotated as power artifacts because there was a peak in the signal, which was later identified as standard MER signal behavior - sparse neuron firing (in extreme cases just one in 10s recording).

In Figure 21 we present the total number of annotated seconds for all of the members of the annotation group.

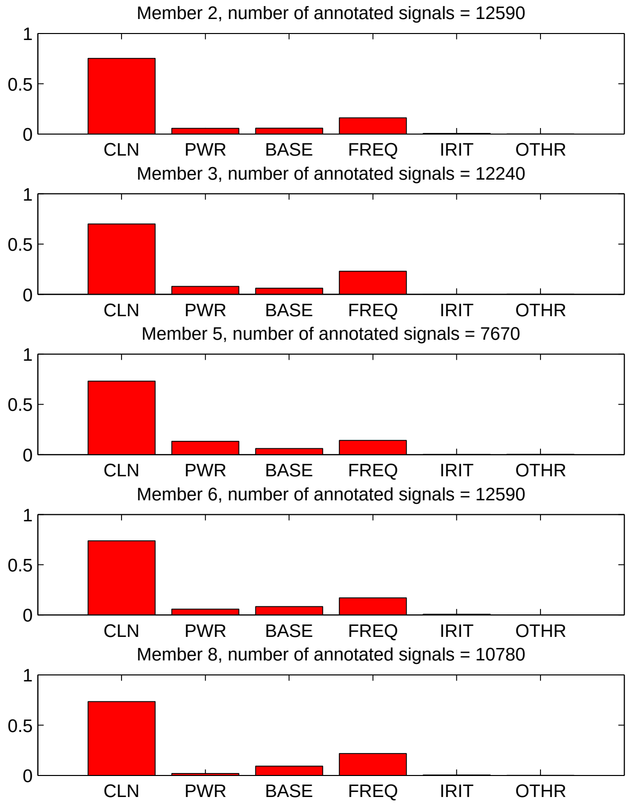


Figure 21: Member total annotation

7.3 Data Summary

In total, we have annotated 57390 channel seconds. In Table 3 we present the total number of annotated artifact groups.

Artifact Type	Count of annotated channel seconds
Total	57390
Clean signals	41849
Power	3619
Baseline	4095
Frequency	10649
Iritated Neuron	201
Uknown	39

Table 3: Annotation summary

As we have already mentioned, the most common artifact found by the annotation group was the FREQ artifact. The amount of power and baseline artifact was very similar. In Table 4 we also present the nuclei from which the annotated signals were taken.

Nucleus	Number of signals
STN	2321
SNr	111
Th	145
Unknown	3162

Table 4: Annotated signals nuclei

With the prepared data we were able to start searching for features of artifacts for the use in the automatic classifier.

8 Features of the MER artifacts

As we described in Chapter 6, the most important part of decision tree building is the preparation of the features which will be used as the rules. In this chapter, we would like to present features extracted from MER signals and artifacts, which we described in Chapter 4.

8.1 Features for noise detection

All of the basic properties can be evaluated by visual inspection or by sound inspection, which is a usual method during DBS implantation. Our goal was to convert these visual features into simple rules which will be then used for decision tree learning. The features should be as simple as possible - they should specify the characteristic artifact behavior. We focused on basic signal processing.

Features of artifacts were found in a small database of example signals, taken from the large database based on their annotation labels. Due to the detailed inspection, we have rechecked the annotation of selected signals to confirm the presence of an artifact. Features were based on the characteristic properties and behavior of signals with artifacts.

We divided our features into three categories - 1) features based on original MER signal, 2) features based on the signal power spectrum, and 3) features which were based on Gaussian filtration. All of the features have its own shortcut which was used in the code and in the visualisation of the decision tree.

8.2 Features extracted from the MER signal

The first category describes features prepared from the original MER signal. Standard behavior of the artifacts in MER signal was already described in Chapter 4.

8.2.1 Power of the signal

First feature which we have prepared as a rule for the decision tree was power of the signal. The power is defined as square of the Root mean square level (RMS) [25] which is calculated as follows:

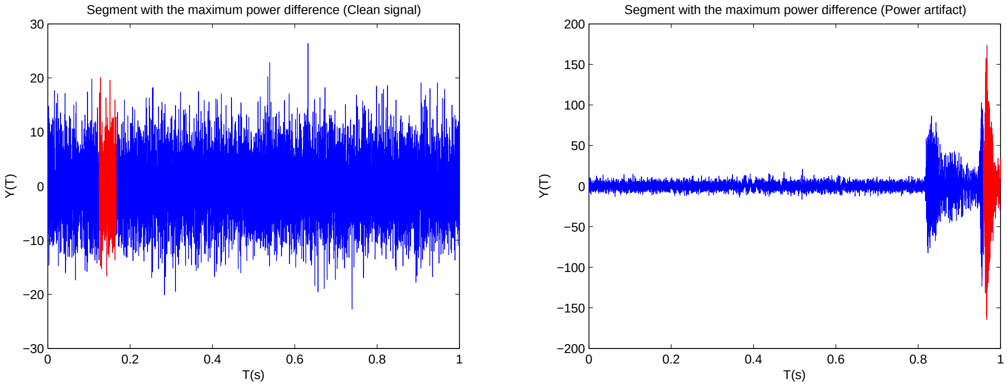
$$X_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|x_n|^2}. \tag{18}$$

$$X_{Power} = X_{RMS}^2. \tag{19}$$

The total power of the signal is represented by the TPOWSIG feature. This value can highly differ across the one-second segment of clean signal. Because of that, we decided to calculate the power value for 24 segments of a smaller size. We have selected 24 segments, because one second has 24000 samples. Each segment then had 1000 samples. For each of these segments we have calculated the power and then the difference between the power of each of the segment and the power of the complete signal. The value of POWSIG feature was equal to the maximum from this set of calculated values.

We have also measured the difference in the power between all of the segments. Feature DPOWSIG is defined as the maximum difference across all of the segments. The features POWSIG and DPOWSIG helped us with the detection of strong artifacts such as power artifact, where the values of power could highly differ across the signal.

An example of the calculation of these values is given in Figure 22. In this Figure we present a comparsion of power features calculation between clean signal and signal containing power artifact. The values for this example can be seen in Table 5.



(a) Clean signal

(b) Signal with the power artifact

Figure 22: Calculation of power features. Segment with the highest difference from the total power

Artifact/Feature	POWSIG	DPOWSIG	TPOWSIG
Clean signal	1.19	9.05	29.73
Power artifact	12.08	2537,37	211,09

Table 5: Values of features based on the power of the signal

The biggest difference in power was in segments 4 and 5 in the clean signal, and in segments 5 and 24 in the signal with the power artifact. The value of POWSIG should be close to 1 for the signals with no artifacts (the power of the segment is equal with the power of the signal). Differences in the values of power features between clean signal and signal with power artifact were obvious.

The power features were the only ones which were used to measure properties of the original MER signals. We have tried to generate another feature such as the maximum of the signal, but the results were not as good as for the power features.

8.3 Features in the Signal Power Spectrum

Another category of features was focused on the signal power spectrum [25]. Mainly, these features were focused on the peaks in the spectrum, which are one of the symptoms of an artifact. All of the defined artifacts have some characteristic behaviour in the signal power spectrum. For example baseline artifact can reach high peaks in lower frequencies (0-300Hz) and power or frequency artifact can reach high power in the spectrum.

For further description of the features, we denoted the input signal as x. The fast fourier transform (FFT) X is then calculated as follows:

$$X(k) = \sum_{t=0}^{N-1} x(t) e^{-j(\frac{2\pi}{N})tk}. \tag{20}$$

Where N is the FFT size, which was in our case equal to 8192, t is sample. The spectrum of the signal S is defined as

$$S = abs(X). \tag{21}$$

Due to the symmetric properties of the spectrum, we used only the first half of the spectrum for feature finding and also for calculation of the values of these features.

We further preprocessed S to get better results of the classification with our features. We subtracted the standard shape of MER signal power spectrum. This step for example helped with the baseline artifact detection. Peaks which are characteristic for baseline artifact are always present in lower frequencies (between 0 and 300Hz). However, those peaks can reach lower values than the power spectrum between 500Hz and 5kHz. Due to that, we deducted a polynomial curve from the original power spectrum. This polynomial curve is calculated by fitting a polynomial function of degree five on the spectrum [25, 26]. Steps of this process can be viewed in Figures 23 and 24. The final spectrum was calculated as:

$$Spk = S - Spoly5. \tag{22}$$

Where Spk are spectral peaks, S is the original spectrum and $Spoly5$ is a polynomial curve fitted on the S.

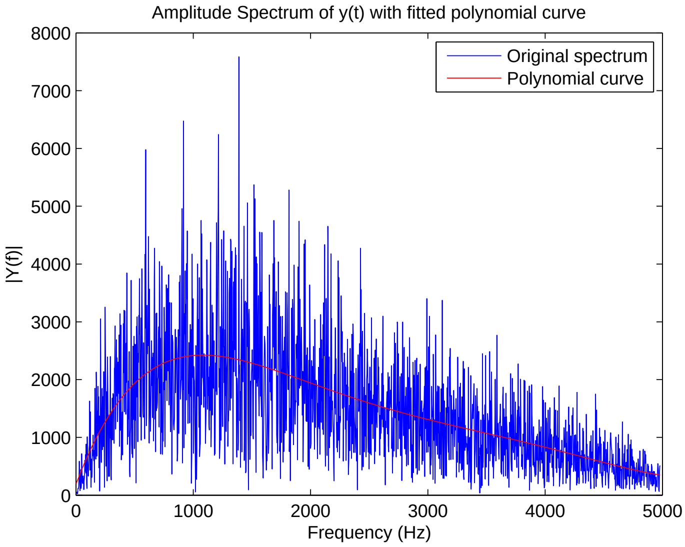


Figure 23: Power spectrum of MER signal with no artifacts together with generated polynomial curve

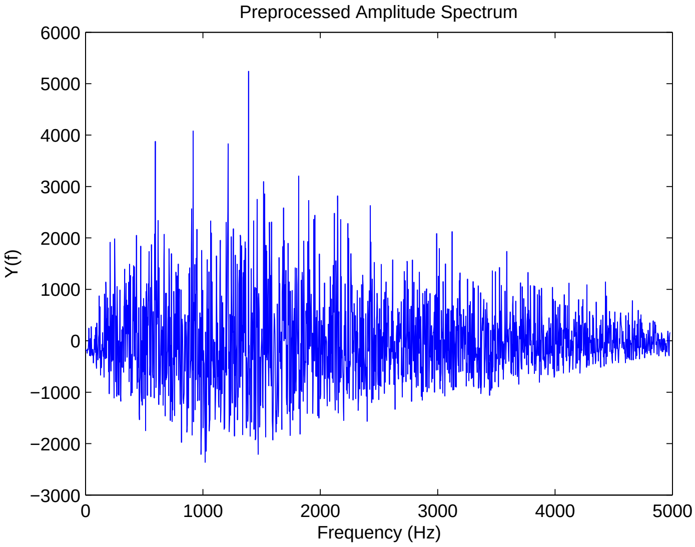
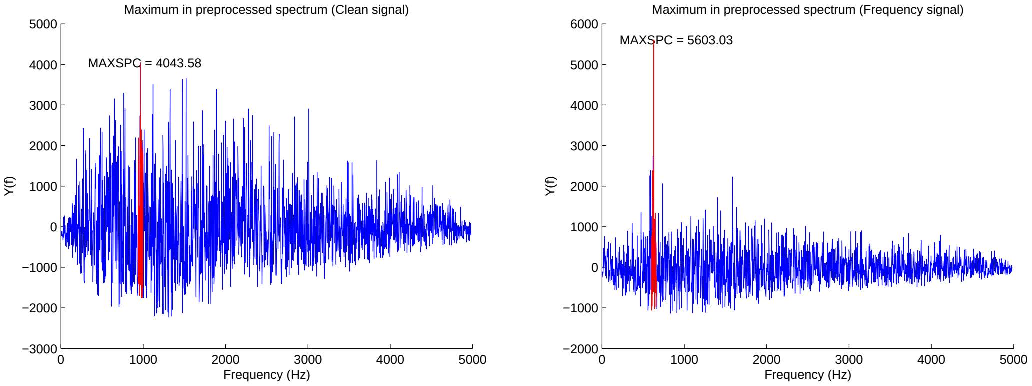


Figure 24: Preprocessed spectrum for feature finding

In the following sections we will describe our features which were created from the investigation of the preprocessed power spectrum of MER signals.

8.3.1 Maximum peak in spectrum

The first feature of the preprocessed power spectrum was the value of the maximal peak (MAXSPC). As we described in previous chapters, some of the artifacts are characteristic with their peaks in the signal power spectrum. For example power artifacts or FREQ artifacts can reach multifold high peaks in comparison with the signal without any artifacts. An example is given in Figure 25.



(a) Clean signal

(b) Signal with the frequency artifact

Figure 25: Maximum in the preprocessed signal spectrum

The value of this feature can be described as:

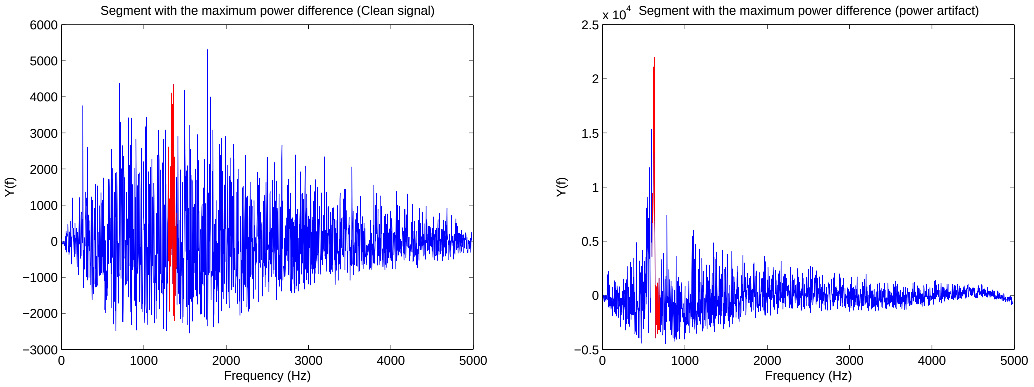
$$MAXSPC = max(Spk). \tag{23}$$

The main reason of creating the MAXSPC feature was to detect strong FREQ artifacts, but this feature had good results for the detection of power artifacts, which had similar behavior as FREQ artifacts. However, the maximum peaks in the spectrum of the power artifact were not as high as peaks in the spectrum of the FREQ artifacts so this feature was not so good for the detection of the baseline artifacts. The height of peaks in the signal power spectrum is not characteristic for the baseline artifacts. Some of the baseline artifacts could reach peaks high as the power signals but this is not common for this type of artifact. We have also tried to measure the sharpness of peaks in the spectrum using the diferential of the spectrum, but we have removed it after several tests on the dataset because the results of that feature were basically the same (worse, respectively) as the results of MAXSPC.

Another feature, which was related to the maximum peak was LOCSPC which compared values of the spectrum with the close neighborhood.

8.3.2 Power of the spectrum

Similarly as with the original MER signal, we decided to measure the power (TPOWSPC) of the preprocessed spectrum. For the same reasons as for the original MER signal, we decided to calculate the power for segments of a smaller size (POWSPC) - for the spectrum we selected segments of length of 100 Hz. For all of these segments we calculated the values of these parameters as the difference of the value of each of the segments and the total value of the spectrum. The value of this feature is equal to the maximum from this set of values. Basically, these features helped us to determine if the spectrum contained an unusual character such as high peak. The visualization of these features can be seen in Figure 26.



(a) Clean signal (b) Signal with the frequency artifact

Figure 26: Power in the preprocessed spectrum

8.3.3 Peak detection in the power spectrum

For the following features we needed to describe a peak of the spectrum. Peak is defined as a value in the spectrum which is higher than the multiplication of a constant and the maximum value of the spectrum. We performed several tests and after that, we set the value of this constant to 0.8:

$$P = Spk(Spk >= 0.8 \cdot max(Spk)). \tag{24}$$

Where P is the set of all peaks in the spectrum. One of the features that is based on this description of peak is the CNTSPC, which stands for the total number of peaks. It is calculated as the length of the vector P:

$$CNTSPC = length(P). \tag{25}$$

This feature was created to reflect characteristic behavior of clear signal in FFT. This clean signal should have a lot of similar peaks and also there should not be any high peaks. On the other hand, signals with artifacts should have bigger differencies between maximal peak and the rest of the signal. CNTSPC feature was very good for the detection of the FREQ and power artifacts, where peaks can reach extreme values. Good results were also measured for the baseline artifact.

We have also created several features which were based on the peaks of the preprocessed spectrum and which were focused mainly on the baseline artifact - most previously described features gave best results for the detection of the power or the FREQ artifact. As we discussed in the section of noise description, the baseline artifact was characterized by peaks in lower frequencies, mostly from 0 to 200Hz.

One of the features which is related to the peaks position in the spectrum is the mean position of peaks P in the spectrum S (MEANSPC). For the baseline artifact, this feature should return number under the 200Hz. Very similar features, which we prepared, were the position of the first peak in the spectrum (FIRSTSPC) and the position of the maximum of the spectrum (MAXPOSSPC). FIRSTSPC is calculated as the position of the peak at the first index of P. We discovered that the MAXPOSSPC and FIRSTSPC was not as good as, for example, the mean position of peaks. This happened because of the heights of peaks in the spectrum of the signal with no artifacts. These peaks can reach higher values than the peaks in lower frequencies of the signal with baseline artifact. An example of calculation of the features, which are based on the position of the peaks in the preprocessed spectrum is given in Figure 27.

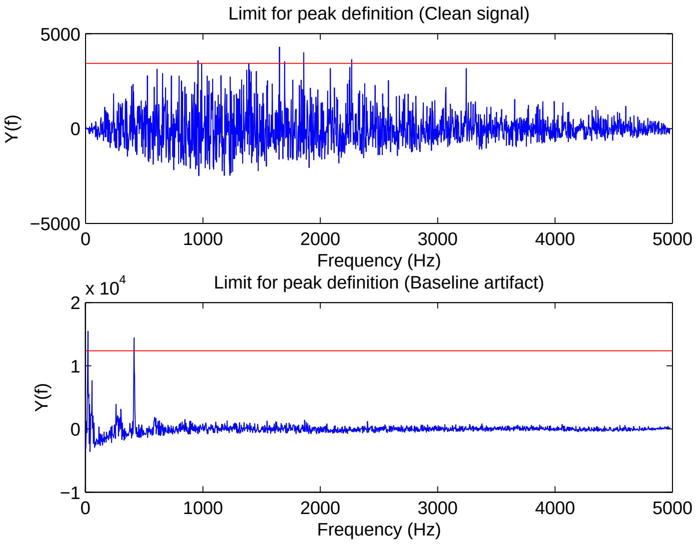


Figure 27: The limit for peak detection in preprocessed spectrum: for signal with no artifacts and signal with baseline artifact

The measured values of the features can be seen in Table 6.

Artifact/Feature	CNTSPC	MEANSPC	FIRSTSPC	MAXPOSSPC
No artifact	6	1637,87	958,11	1652,52
Baseline	2	216,82	20,51	20,51

Table 6: Values of features based on peak detection in preprocessed spectrum

The baseline artifact can be easily detected by using these features. The MEANSPC and FIRSTSPC features were the best for the detection of baseline artifacts.

8.4 Features from Gaussian filtration

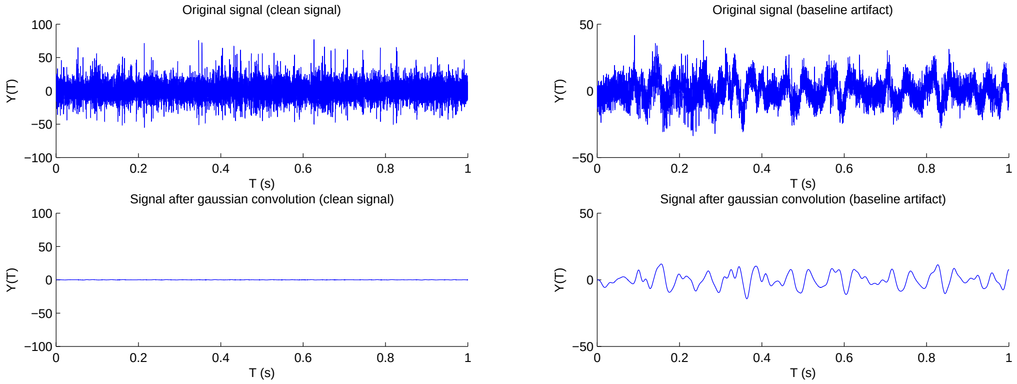
Another method, which we have used for searching of features, was the convolution of MER signal. For the convolution we need to create gaussian function v. We have calculated the gaussian window using $gaussWin$ function implemented in Matlab. The gaussian function was defined as:

$$v(t) = 4 * 10^{-3} e^{-\frac{(t-250)^2}{2*0.01^2}}. \tag{26}$$

After that we performed the convolution (w) as follows:

$$w(\tau) = \sum_t x(t) - v(\tau - t + 1). \tag{27}$$

In fact, the convolution represented low-pass filter. Most features which were created on this method, were focused on baseline artifact detection which, as we mentioned in previous chapters, is artifact with low frequency components in the signal. An example of this process is given in Figure 28, which displays described process for clean signal and signal with the baseline artifact.



(a) Clean signal

(b) Signal with baseline artifact

Figure 28: Convolution on the signal

8.4.1 Features after Gaussian filtration

Based on the sequence w, we calculated similar features as on the spectrum. First one was the maximum of the filtered signal (MAXGSS):

$$MAXGSS = max(w). \tag{28}$$

We have chosen this feature because of the characteristic behavior of the baseline artifact. Due to the convolution, only low frequencies remained in the signal.

Another feature based on this approach, was the count of the peaks after convolution (CNTGSS). Similarly to the count in the power spectrum, we set a threshold, which was calculated from the maximum. By using it, we determined the number of peaks in the signal. Similarly to the maximum of the convoluted signal, we have used this feature mainly for the classification of the baseline artifact.

The last feature which was created on the filtered signal, was the total power (TPOWGSS). This feature was created mainly for the detection of the baseline artifact.

8.5 Metrics for the quality of the features

To measure the quality of the features we have calculated receiver operator characteristics curves (ROC)[27]. ROC is a graphical approach for analysis of the performance of a classification. It is determined by the value of true positive rate (TPR) and false positive rate (FPR). When the classifier is applied, the following outcomes can arise out from the classificaton: true positive (TP) - data which were positive and were classified correctly, true negative (TN) - data which were positive and were misclassified, false positive (FN) - data which were negative and were classified correctly, and false negative (FN) - data which were negative and were misclassified. These outcomes were then stored in the confusion matrix. From this matrix we can calculate TPR and FPR by following equations:

$$TPR = \frac{TP}{TP + FN} \tag{29}$$

$$FPR = \frac{FP}{FP + TN} \tag{30}$$

In order to extract a single number, summarising discrimination capability of the given feature, we calculated the area under curve value (AUC) [28]. AUC is calculated by the integration of ROC. The value of AUC is between 0,5 and 1. 0,5 stands for the random guess classification. We have calculated AUC for all of the features described above. The AUC can be used as an estimate of the quality of the given features. AUC of our features can be viewed in Table 7.

As can be seen in the table above, the efficiency of our features were not as good as we expected. Features focused on the original signal gave us good results for the power

Feature/Artifact	FREQ	Power	Baseline	All artifacts
Original signal features				
DPOWSIG	0.5702	0.7899	0.6348	0.5664
TPOWSIG	0.5345	0.5915	0.7693	0.5630
POWSIG	0.6967	0.8938	0.6196	0.7172
Preprocessed spectrum features				
MAXSPC	0.6707	0.6894	0.5164	0.6396
LOCSPC	0.6285	0.6537	0.6644	0.5676
POWSPC	0.8350	0.8482	0.7646	0.8217
TPOWSPC	0.5006	0.5867	0.7348	0.5364
CNTSPC	0.5984	0.5665	0.6239	0.5980
MEANSPC	0.7636	0.7244	0.8443	0.7742
FIRSTSPC	0.6922	0.6356	0.8546	0.7178
MAXPOSSPC	0.7428	0.6965	0.8367	0.7550
Features after gaussian convolution				
MAXGSS	0.6067	0.6972	0.9552	0.7023
CNTGSS	0.6393	0.6290	0.8029	0.6740
TPOWGSS	0.5945	0.6689	0.9522	0.6890

Table 7: AUC values for the training database

artifact detection. The features calculating count of amplitudes (CNTSPC and CNTGSS) gave us the worst results, but the CNTGSS could be used for the baseline artifact detection with good results. All the other features gave us at least one good result for one of the artifacts - for example the POWSPC feature had good results for detection of FREQ and power artifacts, and the MAXGSS feature had very good results for the baseline detection. With prepared features we were able to generate and test our decision tree. The results of the decision tree making are given in the next part of this thesis.

9 Experimental results

In this chapter we would like to present the results of our method of artifact detection in MER signals in comparison with the existing methods described in Chapter 5 - with the Stationary segmentation of MER signal presented by Falkenberg [2] and Stationary wavelet transform segmentation presented by Guarnizo [4].

9.1 Experimental procedure

At first, we had to prepare the testing and training datasets which could be used for the learning and testing of our classifier. For our method of classification we decided to divide the whole dataset into three datasets - training dataset (60% of data), testing dataset (30% of data) which was used for the calibration of the decision tree rules, and validation dataset (10% of data) which was used as the final result of our classification. We wanted to preserve the real world proportion of artifacts which are present in MER signals. In total, we have used fourteen features, as described in Chapter 8, as the rules for the decision tree. Decision tree was generated in three steps. In the first step we have calculated the values for all of the features for the datasets. Next step was the initial generation of the decision tree. The last step was the pruning of the decision tree. As described in Chapter 6, two approaches were performed to build the decision tree. The first approach was to build an initial tree using the entire training dataset and then perform prunning of it by using the same dataset. The comparison of the cost of the subtrees with the number of leafs is given in Figure 29. In this Figure we also present the dependence of the number of the leafs on the standard error of the cost.

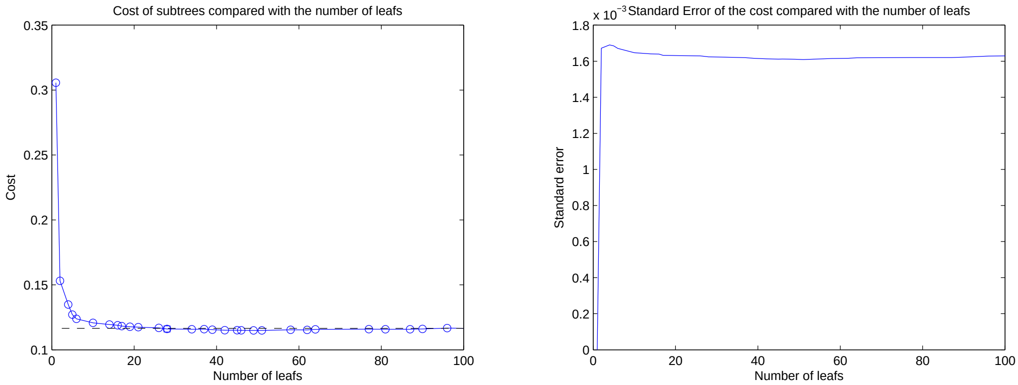


Figure 29: Cost and standard error of the cost depending on the number of the leafs of the subtree

The prune level was equal to 106 which generated minimum cost tree with 34 leafs. We denoted the generated tree as TREE A. The structure of TREE A is presented in Figure 30.

The second approach was to divide training dataset into two parts. The first part was used for the generation of the initial tree. Then, using the second part of the training dataset, we searched for the optimal prune level. We performed this approach too. The number of the leafs for generated decision tree was equal to 9. We denoted this tree as TREE B. The structure of TREE B can be seen in Figure 31. This approach helped us to generate less complex tree.

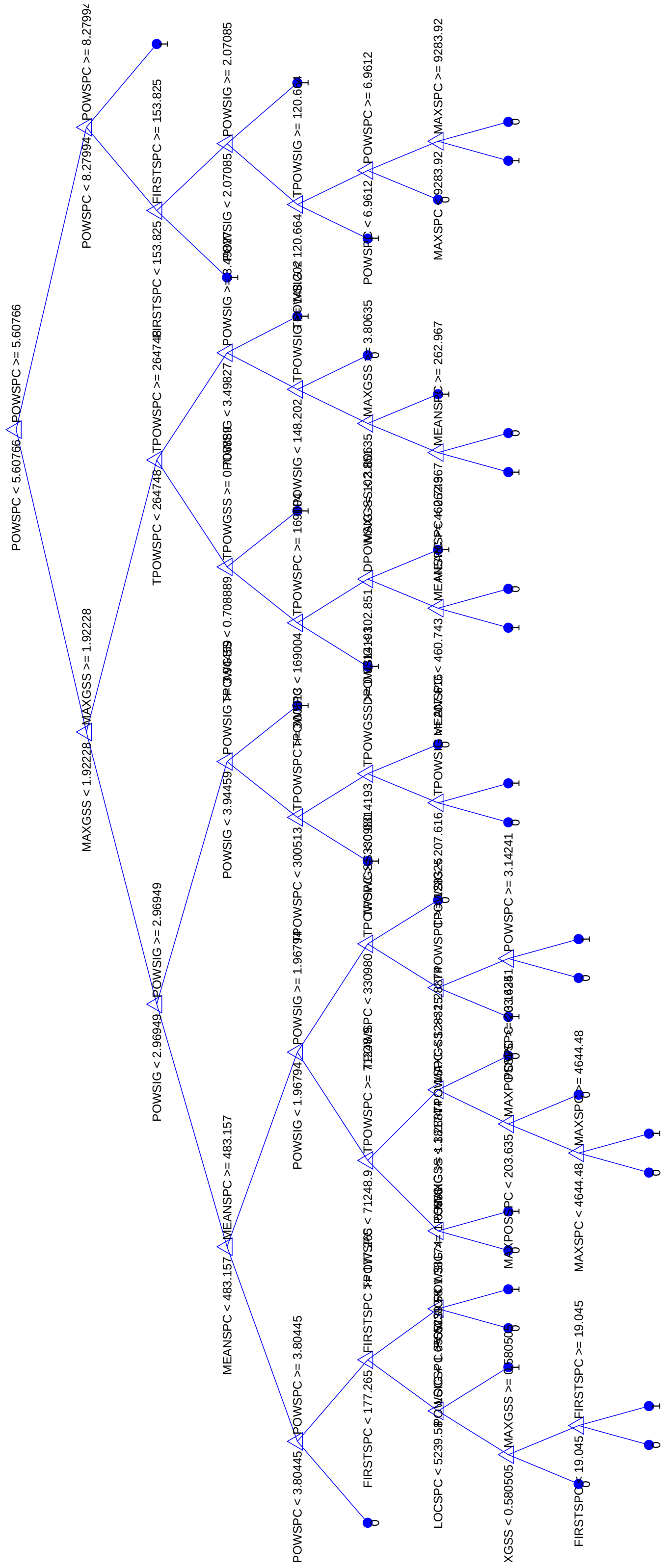


Figure 30: Generated decision tree - TREE A

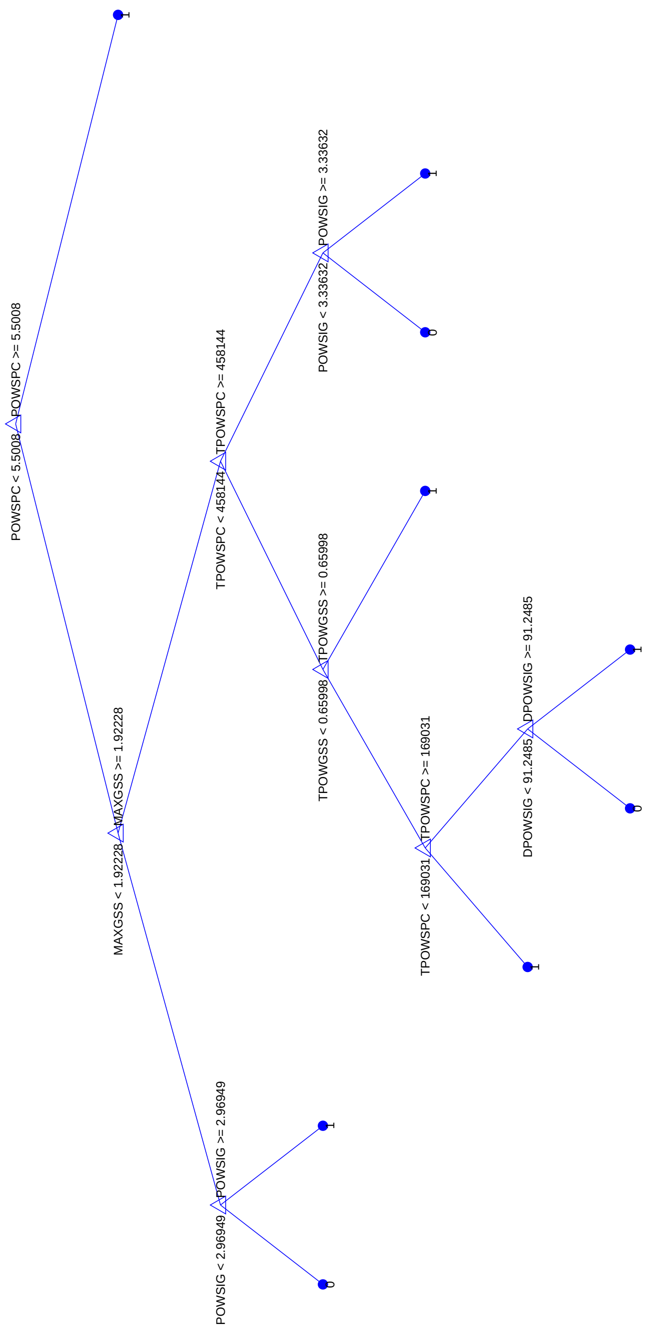


Figure 31: Generated decision tree - TREE B

Preliminary classification results using these decision trees were evaluated and used for the design of additional features. However, the results of the artifact detection were not as precise as expected. Because of that, we decided to perform Random undersampling technique.

9.1.1 Random undersampling

Random undersampling (RUS) [29] is a method where a random subset of the majority class samples is excluded from the training dataset to achieve balance between classes. Before using RUS the training dataset consisted of 24774 clean signal seconds and 10908 signals with artifacts. After RUS the number of clean signal seconds was equal to the number of artifacts (10908). Using the undersampled training dataset we performed both of the approaches of the decision tree generation again. Number of leafs for the tree generated using the first approach was equal to 57. Tree generated using the second approach had 31 leafs. Generated trees can be seen in Figures 32 and 33.

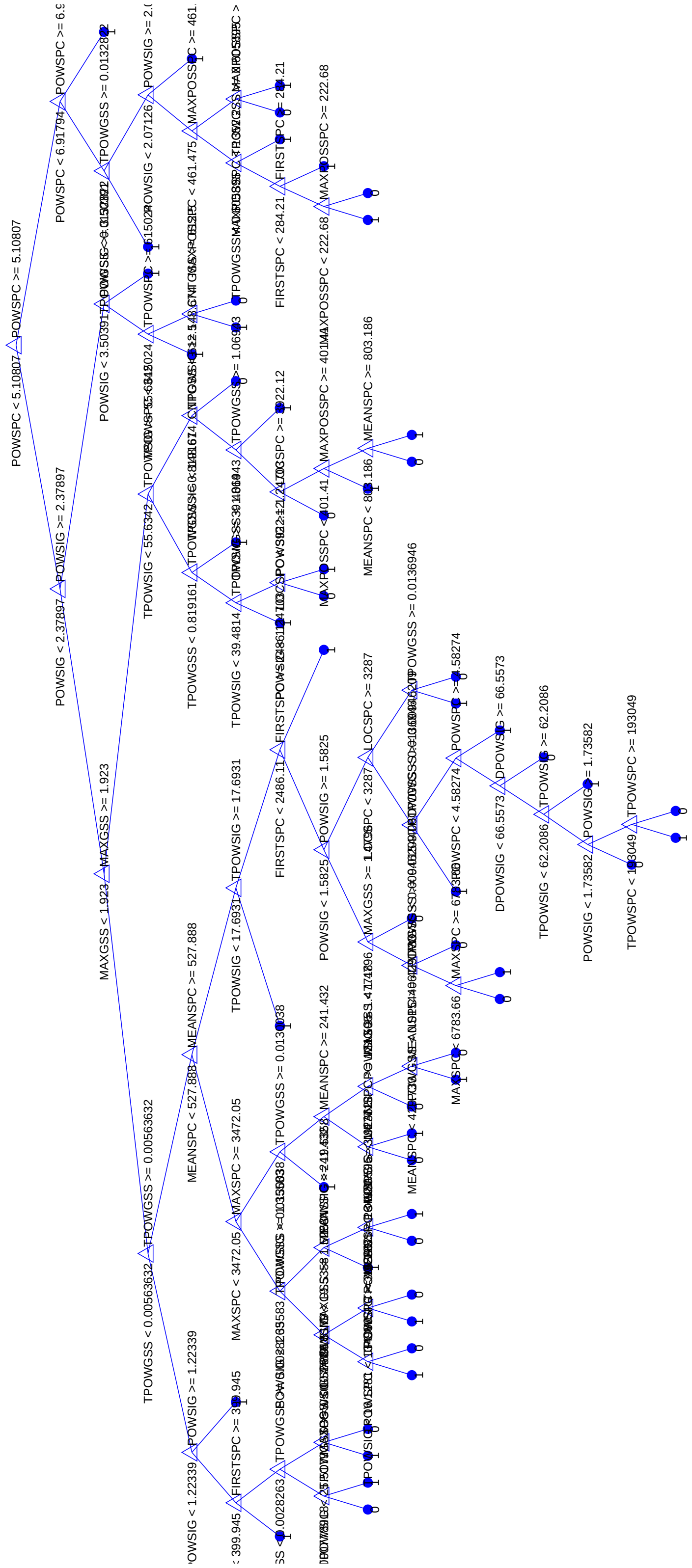


Figure 32: TREE A after RUS

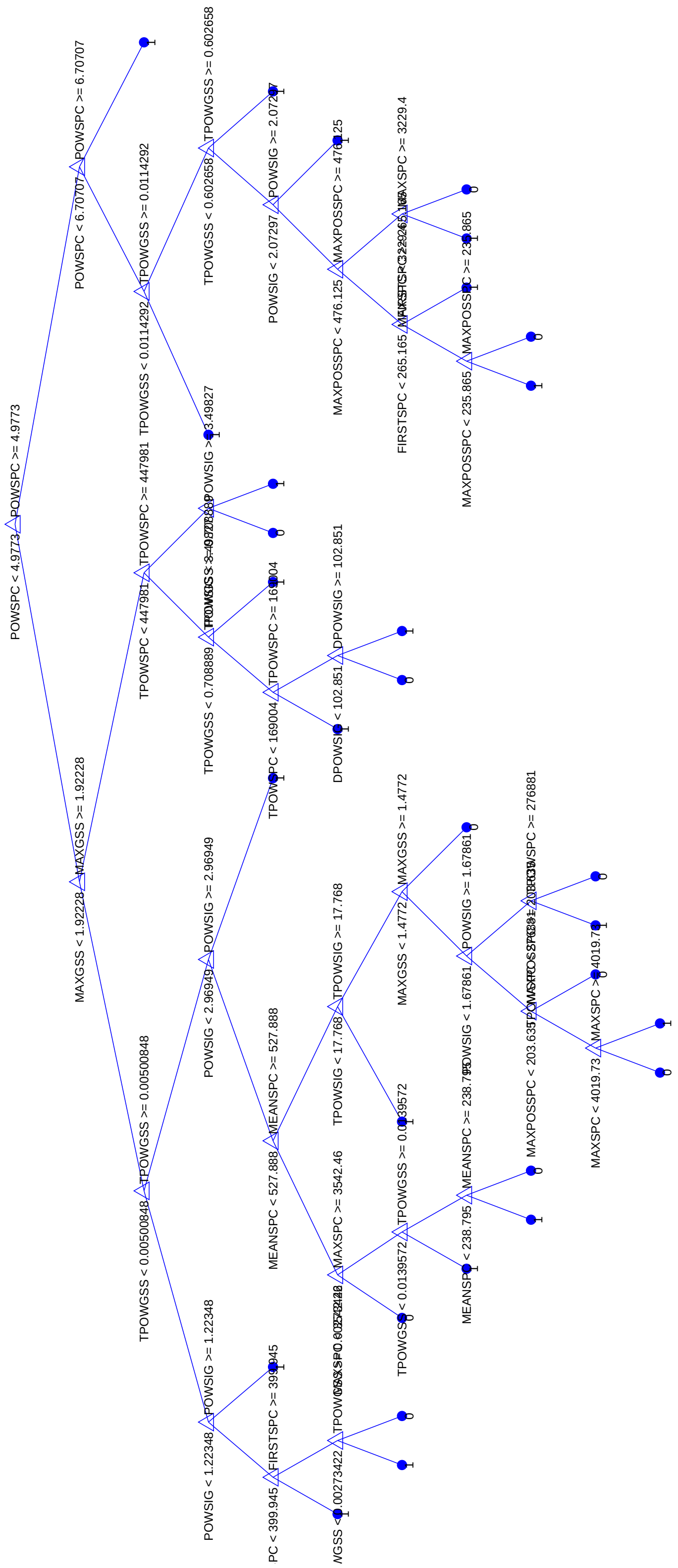


Figure 33: TREE B after RUS

9.2 Optimization of other methods

In this section, we would like to describe the process of optimization of the other methods, presented by Falkenberg [2] and Guarnizo [4]. Detailed description of these methods were presented in Chapter 5. These methods are very similar. The only difference between these methods is the calculation of the variance for the segments. The first method calculates the variance from the autocorelation function (COV) and the second method calculates the variance from the stationary wavelet transform (SWT). Both of these methods needed entire ten-second MER signals as an input.

Two parameters need to be defined for these methods. The first one is the length of the segments. We wanted to compare these methods with our algorithm, which uses one second of the MER as an input, so the length of the segments was set to one second.

The second parameter were the thresholds which were used to check the variance. Because of that, we have performed several runs with these methods to find the best threshold. As the training set, we have used the 60% of annotated signals (34430 seconds). The comparison of accuracies for different values of thresholds are given in Figures 34 and 35.

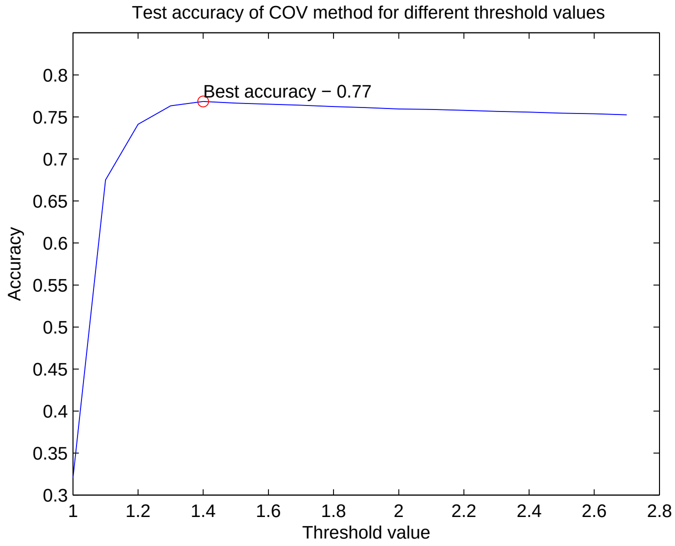


Figure 34: Training phase of the Stationary segmentation using autocorelation function method - Accuracy of different thresholds

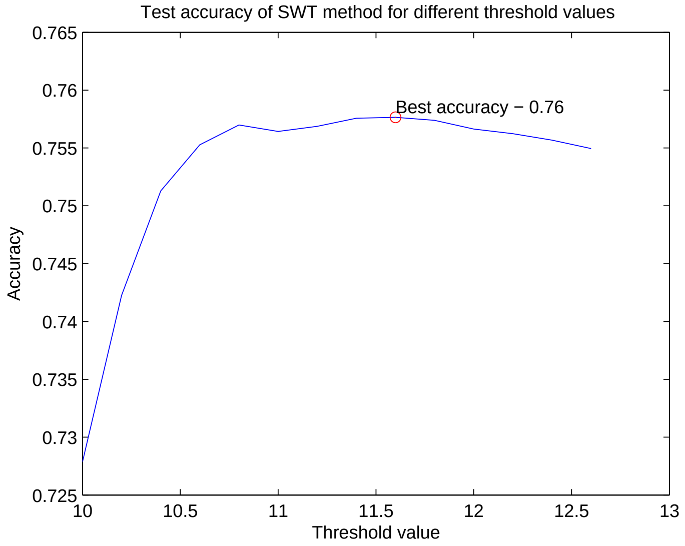


Figure 35: Training phase of the Stationary wavelet transform method - Accuracy of different thresholds

9.3 Results

In here we would like to present the results of the artifact detection using the generated decision trees (before and after RUS) and compare it with the existing methods which use stationary segmentation. We have calculated the accuracy, sensitivity (true positive rate) and specificity (true negative rate) for all of these methods. Tests were performed using the training and testing datasets. Results can be viewed in Tables 8 and 9.

Method	Sensitivity	Specificity	Train Accuracy
TREE A	0.7646	0.9493	89,28%
TREE A after RUS	0.8420	0.9247	88,33%
TREE B	0.7358	0.9428	87,95%
TREE B after RUS	0.8159	0.9250	87,05%
COV	0,3552	0,9479	77,48%
SWT	0,2876	0,9616	76,47%

Table 8: Comparsion of accuracy of all methods on training dataset

Method	Sensitivity	Specificity	Test Accuracy
TREE A	0.7618	0.9664	90,38%
TREE A after RUS	0.8263	0.8704	85,69%
TREE B	0.7640	0.9556	89,70%
TREE B after RUS	0.8162	0.8867	86,52%
COV	0,2778	0,9619	80,56%
SWT	0,2442	0,9733	80,68%

Table 9: Comparsion of accuracy of all methods on testing dataset

The accuracy of the decision tree built using the first approach (same training dataset for inicial tree build and for tree prunning) before RUS gave us the best results from all tested methods. The tree built by using the second approach had worse accuracy. Trees generated after RUS had better sensitivity which had big impact on the accuracy of the artifact detection. We selected TREE A for the validation test, because it had the best accuracy from all of the generated decision trees. It had about 10% better accuracy on the testing dataset than the methods using the stationary segmentation. The sensitivity of these methods was extremely low. Results of TREE A on the validation test are given in Table 10.

The total accuracy of the classifier on the validation dataset was good. However, the sensitivity had slightly decreased.

Method	Sensitivity	Specificity	Test Accuracy
TREE A	0,6770	0,9458	86,33%

Table 10: Accuracy of classifier on validation dataset

9.4 Discussion

The classifier TREE A had the best accuracy from all of the methods - 90,38% for the testing dataset and 86,33% for the validation dataset. The sensitivity of TREE A descreased on the validation set. However, the sensitivity on the validation set is still much better than sensitivities of methods which use stationary segmentation. Low sensitivity could be caused by selecting less accurate features of the artifacts or by the prunning of the decision tree, but it is necessary to perform prunning to avoid the overlearning on the training dataset. Also, the manual annotation of the datasets might be inaccurate and it could have negative impact on the results.

The results of the stationary segmentation methods were very inaccurate in artifact detection. The sensitivity of these methods was extremely low, owing probably to the high simplicity of the features used: the method was trained on artificial signals with properties changing in discrete steps. On the contrary, real-life signals contain various types of artifacts, including long-term FREQ artifacts, present throughout the whole signal. Such an artifact can not be detected by the stationary methods from the principle. However, this result might have been affected by the fact that we did not optimize the length of the segment.

The results of our classifier might be improved by creating additional features, which could specify standard behaviour of the artifacts more accurately (for example have better AUC values than the current features). Another improvement could be made by the checking of annotated data - for example perform additional annotation of data and comparing it with the previous annotation.

10 Conclusion

In this thesis, we have described the steps to create a classifier for the detection of artifacts in MER signal. The artifacts were separated into three groups - FREQ, power artifacts and baseline artifacts. Detailed characteristic behaviour of the artifacts was analysed in Chapter 4. We found out that some of the artifacts can be filtered before further processing of MER signal. For example the baseline artifact can be removed using a high-pass filter.

As a classifier we have decided to use the decision tree. The decision tree should test features of an input signal and using simple rules determine whether it contains artifact or not. For the preparation of our classifier we needed data, where were these artifacts present. Because of that, we have performed manual annotation of MER signals. In total, we annotated over 57390 seconds of MER signals. Most common artifact, which was present in annotated data, was the FREQ artifact (10649 seconds). The baseline and the power artifacts had similar distribution over the data (4095 and 3619). Using these data we have prepared initial features of the artifacts, which then were used for the preparation of rules of the decision tree. Features were prepared in three groups - 1) features prepared from the original signal, 2) features prepared from the preprocessed spectrum of the signal, and 3) features prepared from the signal after gaussian filtration.

We performed two approaches of decision tree generation. The first one used same training dataset for the generation of initial decision tree and then for the tree prunning. The second approach divided the training dataset into two parts. The first part was used for initial tree generation and the second part for the prunning. Based on the initial results, we decided to use random undersampling method to achieve a balance between the number of clean signals and artifacts. In total we have generated four different trees. The generated trees and results of them can be viewed in Chapter 9. As a final classifier we decided to use the decision tree built using the first method without random undersampling. The results of our classifier were good. The accuracy was 90,38% for the testing dataset and 86,33% for the validation dataset. We have compared our classifier with the other methods which were available in the literature. These methods used stationary segmentation of the signal to determine the longest segment with no artifacts. We have described these methods in Chapter 5. The accuracies for the testing dataset were 80,56% for COV method (variance is calculated from autocorelation function) and 80,68% for SWT method (variance is calculated from stationary wavelet transform). Also, our classifier had better sensitivity than these methods. The results of the classifier might be improved by preparing additional parameters, which will be used in the build phase of decision tree. Another option is to check the annotation of our datasets. In our future work, we would like to focus on the improvement of our decision tree. For example, we would like to detect each artifact types separately.

References

[1] Anan Moran, Izhar Bar-Gad, Hagai Bergman, and Zvi Israel. Real-time refinement of subthalamic nucleus targeting using bayesian decision-making on the root mean square measure. 21:1425–1431, 2006.

[2] J.H. Falkenberg, J. McNames, M. Aboy, and K.J. Burchiel. Segmentation of extracellular microelectrode recordings with equal power. 3:2475–2478 Vol.3, Sept 2003.

[3] Mateo Aboy and J.Haakon Falkenberg. An automatic algorithm for stationary segmentation of extracellular microelectrode recordings. *Medical and Biological Engineering and Computing*, 44(6):511–515, 2006.

[4] C. Guarnizo, A.A. Orozco, and G. Castellanos. Microelectrode signals segmentation using stationary wavelet transform. 2:450–454, May 2008.

[5] E. Bakstein, J. Schneider, T. Sieger, D. Novak, J. Wild, and R. Jech. Supervised segmentation of microelectrode recording artifacts using power spectral density. pages 1524–1527, Aug 2015.

[6] J Jankovic. Parkinson's disease: clinical features and diagnosis. *Journal of Neurology, Neurosurgery and Psychiatry*, 79(4):368–376, 2008.

[7] S.V. Sarma, M. Cheng, U. Eden, R. Hu, Z. Williams, E.N. Brown, and E. Eskandar. Modeling neural spiking activity in the sub-thalamic nucleus of parkinson's patients and a healthy primate. pages 2012–2017, Dec 2008.

[8] C.O. Oluigbo, A. Salma, and A.R. Rezai. Deep brain stimulation for neurological disorders. *Biomedical Engineering, IEEE Reviews in*, 5:88–99, 2012.

[9] Anan Moran, Izhar Bar-Gad, Hagai Bergman, and Zvi Israel. Real-time refinement of subthalamic nucleus targeting using bayesian decision-making on the root mean square measure. *Movement disorders*, 21(9):1425–1431, 2006.

[10] P.F. Grant and M.M. Lowery. Simulation of cortico-basal ganglia oscillations and their suppression by closed loop deep brain stimulation. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, 21(4):584–594, July 2013.

[11] J.H. Noble, R.D. Datteri, B.M. Dawant, and P.-F. D'Haese. Automatic segmentation of the optic tracts for computer assistance of deep brain stimulation procedures. pages 864–867, April 2010.

[12] Sven Möllers, Máté Döbrössy, and Guido Nikkhah. Transplantation of foetal ventral mesencephalic grafts in parkinson's disease: A still evolving concept with new regulatory challenges, towards new therapies for parkinson's disease, prof. david finkelstein (ed.), isbn: 978-953-307-463-4, intech, doi: 10.5772/16851.

[13] Zvi Israel and Kim J. Burchiel. *Microelectrode Recording in Movement Disorder Surgery.* Thieme New York, USA, 2004.

[14] Juan Martinez, Carlos Pedreira, Matias J. Ison, and Rodrigo Quian Quiroga. Realistic simulation of extracellular recordings. *Journal of Neuroscience Methods*, 184(2):285 – 293, 2009.

[15] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Bulletin of Mathematical Biology*, 52(1-2):25–71, 1990.

[16] E.M. Izhikevich. Simple model of spiking neurons. *Neural Networks, IEEE Transactions on*, 14(6):1569–1572, Nov 2003.

[17] I. Obeid and P.D. Wolf. Evaluation of spike-detection algorithms fora brain-machine interface application. *Biomedical Engineering, IEEE Transactions on*, 51(6):905–911, 2004.

[18] R. Quian Quiroga, Z. Nadasdy, and Y. Ben-Shaul. Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering. *Neural Comput.*, 16(8):1661–1687, August 2004.

[19] Liu Jian and Wang Yan-Qing. Research on application of decision tree in classifying data. *Intelligent Computation Technology and Automation (ICICTA), 2011 International Conference on*, 1:1098–1101, March 2011.

[20] LING LIU and M.TAMER ÖZSU. Decision tree. *Encyclopedia of Database Systems*, pages 765–765, 2009.

[21] S.B. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, 39(4):261–283, 2013.

[22] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[23] Cross-validation. In Claude Sammut and GeoffreyI. Webb, editors, *Encyclopedia of Machine Learning*, pages 249–249. Springer US, 2010.

[24] Minimal cost-complexity pruning. `https://onlinecourses.science.psu.edu/stat857/node/60`, 2016.

[25] Richard G. Lyons. *Understanding Digital Signal Processing.* Pearson Education, Boston, MA, USA, 1997.

[26] MathWorks polynomial curve fitting. `http://www.mathworks.com/help/matlab/ref/polyfit.html`. Accessed: 201511-30.

[27] Pang-Ning Tan. Encyclopedia of Database Systems: Receiver Operating Characteristics. pages 2349–2352, 2009.

[28] Area under curve. In Claude Sammut and GeoffreyI. Webb, editors, *Encyclopedia of Machine Learning*, pages 40–40. Springer US, 2010.

[29] N. Japkowicz. Learning from imbalanced data sets: A comparison of various strategies. pages 1–5, 2014.

CD content

Included CD contains text of this thesis in PDF format and the source code of our implementation in Matlab.

Directory	Description
text.pdf	the text of this thesis
code	contains the source code