Methods for Automatic Detection of Artifacts in Microelectrode Recordings

Eduard Bakšteín²⁴, Tomáš Sieger¹⁲, Jiří Wild³, Daniel Novák⁴, Jakub Schneider¹², Pavel Vostatek⁴, Dušan Urgošík⁵, Robert Jech⁵

¹Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
²National Institute of Mental Health, Klecany, Czech Republic
³Department of Neurology and Center of Clinical Neuroscience, First Faculty of Medicine and General University Hospital, Charles University, Prague, Czech Republic
⁴Department of Stereotactic Neurosurgery and Radiosurgery, Na Homolce Hospital, Prague, Czech Republic

Abstract

Background: Extracellular microelectrode recording (MER) is a prominent technique for studies of extracellular single-unit neuronal activity. In order to achieve robust results in more complex analysis pipelines, it is necessary to have high quality input data with a low amount of artifacts. We show that noise (mainly electromagnetic interference and motion artifacts) may affect more than 25% of the recording length in a clinical MER database.

New Method: We present several methods for automatic detection of noise in MER signals, based on i) unsupervised detection of stationary segments, ii) large peaks in the power spectral density, and iii) a classifier based on multiple time- and frequency-domain features. We evaluate the proposed methods on a manually annotated database of 5735 ten-second MER signals from 58 Parkinson’s disease patients.

Comparison with Existing Methods: The existing methods for artifact detection in single-channel MER that have been rigorously tested, are based on unsupervised change-point detection. We show on an extensive real MER database that the presented techniques are better suited for the task of artifact identification and achieve much better results.

Results: The best-performing classifiers (bagging and decision tree) achieved artifact classification accuracy of up to 89% on an unseen test set and outperformed the unsupervised techniques by 5-10%. This was close to the level of agreement among raters using manual annotation (93.5%).

Conclusion: We conclude that the proposed methods are suitable for automatic MER denoising and may help in the efficient elimination of undesirable signal artifacts.

Keywords: microelectrode recordings, artifact detection, external noise, supervised classification

Cite as


Note: This is the author’s version of the accepted article. The journal version is to be found at the publisher using the doi link above. Visit also http://neuro.felk.cvut.cz for supplementary matlab codes, sample data and information on our related research.

Highlights

- Artifacts are common in intra-operative micro electrode recordings (MER) - up to 25%.
- We propose a set of classifiers for automatic artifact detection.
- The methods are evaluated on a database of 5735 manually labeled MER signals.
- The best-performing classifiers achieved up to 89% test-set accuracy.
- Matlab source codes and sample data are available in the supplement.

Author’s version from August 1, 2017, distributed under

[License Icon]
1. Introduction

Extracellular microelectrode recording (MER) using electrodes with a tip size around 1 μm [1] is a basic technique for acquiring extracellular electric activity at the level of individual neurons (single-unit activity). Due to the small electrode size and the low voltage of the source signal, MER is susceptible to mechanical shifts and electromagnetic interference, resulting in signal artifacts [2]. While some components of the external noise can be filtered easily (e.g., 50 Hz or 60 Hz mains noise filtering using a notch or comb digital filter or sophisticated hardware design [3]), other may be more difficult to define and suppress.

In this study, we describe aspects of the most prevalent artifacts, as observed on an MER database, obtained from exploratory microelectrodes immediately before implantation of permanent electrode for deep brain stimulation (DBS) and we propose a set of classifiers for identifying them. As the DBS technique has been used routinely in movement disorder therapy for more than two decades [4] and MER is still used in the vast majority of DBS centers worldwide [5], DBS surgery also serves as a prominent source of human sub-cortical neuronal data for scientific purposes. Irrespective of the value that DBS MER signals may have for the research community, it is clinical considerations that mostly determine the procedure and put strain on the available time and instrumentation. Sources of undesired artifacts in DBS surgery include electrical appliances in the operating room, electrode vibration (after manual electrode shifting or touches applied to the microdrive or stereotactic frame), and movement or speech of the patient. Artifacts are therefore commonly found in DBS MER data, and we also illustrate this on data samples from 58 Parkinson’s disease patients.

The presence of artifacts in an MER signal may have a dramatic impact on subsequent signal processing. Its severity will depend on the particular processing pipeline, and also on the character of the artifact. In the case of single or multi-unit analysis, a spike detection method and spike sorting methods are used to separate the activity of neurons close to the electrode tip from the background activity (i.e., the summary activity of neurons further away from the electrode [6, 7]). Widely-applied extracellular spike detection methods use amplitude distribution to estimate an appropriate value of the detection threshold [8, 9, 10], and are therefore sensitive to the background noise level [11], and also to artifacts. The impact of artifacts on spike detection results may be mitigated to some extent by the use of a robust estimator of the neuronal background level [12] and subsequent removal of spikes of anomalous shape [9, 13]. However, external artifacts remain a significant source of undesirable noise in the whole (mostly statistical) analysis pipeline, and they may lead to a loss of sensitivity or to noise in the resulting spike trains.

1.1. Existing artifact detection approaches

The prevailing approach for obtaining an artifact-free dataset in MER-based studies relies on manual inspection and disposal of contaminated signal segments [14, 15]. However, this approach may be lengthy and may still not provide optimal results, as some artifacts cannot be identified solely from the time series plot, due to their low projection to signal envelope. In such cases, other modalities, including spectrogram and audio playback, are needed in order to identify all artifacts. Many researchers therefore use their own (semi) automatic methods, ranging from simple amplitude thresholding [16], through statistical testing of the amplitude distribution in short signal windows [17, 18], to sophisticated amplitude and power spectral density (PSD) based systems [19, 20, 21]. The detection thresholds are usually selected to match the investigator’s subjective evaluation, and the actual classification performance is not rigorously tested.

Two methods for identifying stationary MER segments (i.e., segments consistent in a selected signal feature) have been published in detail with a formal performance evaluation. The first of these methods is based on the variance of the autocorrelation function [22, 23], while the other method is based on the variance of the signal wavelet decomposition [24]. These methods, while suitable for detection of rapid changes in amplitude (as presented on simulated data in the original publications), seem to be less appropriate for motion artifacts and for electromagnetic interference [25], two of the most prevalent noise types observed on our DBS MER database.

1.2. Proposed artifact detection method

As we showed in our previous paper [25], much better detection results than for the existing solutions can be achieved by a simple linear classifier based on the power spectral density (PSD) of the signal. In this paper, we extend this approach by presenting an artifact detection model based on multiple time domain and spectral features. We tested a range of models based on decision trees, support vector machines (SVM) and boosting, and
we evaluated their performance together with previously published solutions on an extensive manually annotated DBS MER database.

2. Methods

2.1. Data collection

All data used in this study were collected during electrophysiological exploration for deep brain stimulation surgery at Na Homolce Hospital, Prague, Czech Republic. All patients were implanted either unilaterally or bilaterally, using one to five microelectrodes in a cruciform configuration (the "Ben-gun"), spaced at intervals of 2 mm around the central electrode. The signals were recorded using the Leadpoint recording system (Medtronic, MN) at a sampling rate of 24 kHz, and were band-pass filtered between 500-5000 Hz upon recording. The median recording length was 10 s, and signals shorter than 5 s were discarded. The data recording was part of an unaltered standard surgical procedure, with which all patients had agreed by signing the informed consent.

All annotation, classification and data handling was performed in MATLAB (The MathWorks, Inc., Natick, Massachusetts, United States).

2.2. Manual artifact data annotation

The data annotation was based on visual and auditory inspection of multi-channel signal time series. Additionally, the annotator was provided with a spectrogram heatmap, showing short-time Fourier transform spectra on a parallel time scale with the time series. All data were annotated in 1 s windows, as further experiments showed a very low level of agreement between raters in a scenario where the exact start and end of the artifacts was determined by each user. In cases where the data included multiple channels from electrodes recorded simultaneously, all channels were visualized in parallel for easier identification of movement artifacts, often spanning across multiple channels [26].

To achieve a high level of concordance, the team of five annotators (neurologists or engineers with long-term experience with MER processing) underwent repetitive annotation of a set of 20 multi-channel signals, and the discrepancies between raters were discussed after each of the three rounds. In the subsequent phase, the team annotated the whole database of 1676 multi-channel signals (i.e. about 330 signal sets per member), with a small proportion of signals shared among the team. In the subsequent evaluation, the shared set of signals was used to evaluate the level of agreement among raters.

2.2.1. Observed artifacts

For the purposes of annotation, and also for further analysis and evaluation, the observed artifacts were grouped into the following clusters (cluster acronyms and the percentage of signal seconds containing a given artifact on the cross-validation database are given in parenthesis)

- A mechanical movement artifact, manifested by short-time, high-power signal peaks, usually spread across the whole frequency spectrum. (POW, 4.8%), see Fig. 1 A) and C).
- Low-frequency interference below the mains frequency (50 Hz), causing visible variation in the signal offset or in the baseline (BASE, 7.6%)
- Electromagnetic interference at one or multiple stable frequencies, well localized in a narrow band(s) in the frequency spectrum and stable over time (FREQ, 17.6%). The frequency of the observed long-term interference often differed from the expected odd harmonics of the mains frequency (50 Hz, 150 Hz, 250 Hz etc.), see Fig. 1 B).
- An "irritated neuron": spiking activity of very high and variable amplitude and firing rate (IRRIT, 0.3%)
- Other artifacts that cannot be assigned into any of the groups above. (OTHER, 0.2%)

Each second of the MER signal may contain one artifact type, or several artifact types at the same time. A clean signal (CLN, 74.6%) is defined as a signal with no artifacts. Three examples of artifact-bearing MER signals can be found in Figure 1. Detailed tabulation of the artifact percentage in our cross-validation and test datasets can be found in Table A.1 in the Appendix. Twenty example MER signals with the different artifact types can be found in the online supplementary material.

2.3. Automatic classification methods

Due to the relatively low level of agreement on the exact artifact type among the annotators, all classifiers were designed only as two-class classifiers, trained to distinguish clean signals (CLN) from signals with all other artifact types.

2.3.1. Stationary segmentation methods

Two stationary segmentation methods have been described, by i) Falkenberg and Aboy et al. [22, 23], based on variance of the signal autocorrelation function and ii) by Guarnizo et al. [24], based on variance of the signal wavelet decomposition. In these methods, the signal
is first divided into short fixed-length segments. Subsequently, the variance ratio of the statistics of neighboring signal segments is computed, and is compared with a manually preset threshold. Points exceeding the threshold are marked as change points, and denote boundaries between stationary segments. Both methods focus solely on signal segmentation and do not imply which of the stationary segments contains a clean signal (or artifacts).

In our previous work [25], we presented an extension to these methods: instead of comparing the neighboring segments only, the method computes the distance matrix between all possible segment pairs and searches for the largest available component, connected by a sub-threshold path. Assuming higher stationarity in the clean signal segments, rather than in the artifact-bearing segments, the largest component is then marked as a clean signal, while the remaining signal sections are marked as artifacts. For further technical details, the reader is referred to [25], or to Appendix B of this text. The extended methods are abbreviated below as COV (autocorrelation-based approach) and SWT (wavelet-based approach).

For the purposes of performance evaluation, we optimized three parameters of each algorithm: i) the segment length (0.25, 0.33, 0.5 or 1 s), ii) the detection threshold, and iii) the number of segments within a one-second window labeled by the classifier as an artifact, which was necessary for marking the whole second as an artifact (this last point was necessary since the manual annotation labels were available for one-second windows only). Implementation of the COV method is available in the online supplementary material.

2.3.2. The maximum spectral difference method

A simple detection method, based on the power spectral density of MER signals, was also presented in our paper [25]. The basic assumption is that the power spectral density (PSD) of a clean band-pass filtered MER signal is smooth, unlike most signals with artifacts, which commonly contain high peaks and other disturbances. In the first step, a mean spectrum $C$ of clean signal segments is calculated from a set of $N$ training signals $X = \{X_1, X_2, ..., X_N\}$ with the corresponding artifact annotation $a = \{a_1, a_2, ..., a_N\}$, with $a_i$ equal to 1 for clean signals and equal to 0 for signals with artifacts, according to:

$$C = \frac{1}{\sum_{i=1}^{N} a_i} \sum_{i=1}^{N} a_i P_j,$$  \hspace{1cm} (1)
where $P_j$ is the normalized power spectral density (also norm. PSD) of signal segment $X_j$ of fixed length of 1s: signal PSD is computed using Welch’s method and is divided by its sum. Therefore, the sum of all spectral bins in the resulting $P$ spectrum is equal to one, and is thus independent of the total signal power. The length of the discrete Fourier spectrum was set to $M = 2048$, equal to the window length (Hamming type)$^1$. The window overlap was set to 50%.

For an unseen signal segment with normalized spectrum $P_j$, the maximum absolute difference from the mean spectrum of clean segments $C$ can be calculated by finding the maximum value over all $M$ spectral bins:

$$d = \max_{m=1...M} |P_j(m) - C(m)| \quad (2)$$

The optimal detection threshold for $d$ was set to the value that maximized performance (the $J$-statistic – see Equation 3 below) on the training dataset. Once fixed, the threshold was used for classification. A clean spectrum $C$, and also the spectrum for the three major artifact types, can be found in Figure 2. A Matlab implementation of the method is available in the online supplementary material. Throughout the performance comparison below, all spectra for the spectral method (abbreviated maxAbsDiffPSD) were computed from one-second signal segments.

![Figure 2: Normalized power spectral density for different artifact types](image)

Figure 2: Normalized power spectral density for different artifact types, each computed from 1000 randomly selected signals from the cross-validation dataset. The mean value for each type of artifact (orange) with the 5th and 95th percentile for each spectral bin is compared with the normalized PSD ($C$) of clean signals (black).

1. See Appendix C for a discussion of spectrum length and the PSD estimation method.

### 2.3.3. Multi-feature classifiers

In addition to the simple detection methods mentioned above, we have implemented a range of classification methods based on multiple features derived from the raw MER signal and its normalized power spectrum. The features were designed to describe the most prominent properties of various artifact types, in comparison with a clean MER signal. After initial classification tests on a sample from the cross-validation database, misclassified samples were identified, and were used to devise additional features ($\text{maxCorr}$ and $\text{ksnorm}$). The initial tests were also used to specify feature properties – such as the length of the Fast Fourier Transform (FFT), and to identify suitable classifier parameter ranges for further testing.

For all features based on the frequency spectrum, the normalized power spectral density was computed according to the definition in section 2.3.2 above, using Welch’s method with FFT of length 2048 (equal to the window length - Hamming type) and 50% window overlap. Features were calculated from one-second signal segments to match the temporal resolution of the annotation. All 19 features in the feature set are summarized in Table 1.

Due to the similarity in the definition of some features, high inter-feature correlation is to be expected – e.g. $\text{psdMaxStep}$ and $\text{psdMax}$, due to the very sharp character of the spectral peaks in signals with electromagnetic interference, or $\text{sigP90}$ and $\text{sigP95}$, due to the smooth character of the signal amplitude distribution in the lower percentiles. Therefore, all selected classification methods have to perform some sort of feature selection, allowing for correlated features. The multi-feature classifiers that have been implemented include:

- The **decision tree** classifier, with limits on minimum parent node and leaf size, and different splitting criteria.
- The **Support Vector Machine** (SVM) classifier, with a linear or radial-basis kernel using different optimization methods and kernel properties. The SVM classifier was preceded by the feature selection step – see the description below.
- **Boosting** classifiers, using various algorithms (AdaBoostM1, LogitBoost, GentleBoost, RobustBoost, Bagging) and varying the learning rate. The weak learner that was used was a decision stump (boosting) or a decision tree (bagging).

All classifier parameters and the ranges across which they were optimized can be found in Table E.1. An exhaustive search of all possible combinations of parameters was performed for each classifier type.
The SVM classifier was preceded by a feature selection step in order to reduce the number of features, and also to reduce their redundancy. We used forward wrapper feature selection with a discriminant analysis classifier. The algorithm starts with an empty feature set and adds a single feature that provides the best accuracy using the selected classifier. Then, all possible sets of two features, including the already selected feature and a candidate feature from the remaining part of the feature set are tested. The best-performing two-feature set is fixed, and the process continues with three-feature sets and so on, until the stopping criterion (minimum improvement in classification accuracy) is achieved. The classifier used for feature selection was discriminant analysis (linear, quadratic or Mahalanobis distance-based), and we used internal 5-fold crossvalidation to estimate out-of-sample performance. The classifier type used for feature selection, as well as the value of the stopping criterion, was subject to parameter optimization, and all possible combinations were tested (see Table E.1 in the Appendix).

2.4. Crossvalidation scheme

In order to evaluate the out-of-sample classifier performance (i.e. performance on unseen data), we divided the training data from our database into two datasets: data from three DBS patients (6 MER trajectories, each consisting of 5 parallel micro-electrodes) were kept aside for final classifier testing as the test set, while the remainder, denoted cross-validation (or CV) set, were used for feature evaluation, classifier training and parameter optimization.

The main crossvalidation procedure was as follows:

1. **Ten-fold crossvalidation**: The cross-validation dataset was divided randomly into 10 subsets. In each iteration, all parameter combinations were trained on 9 subsets and were validated on the remaining subset. The confusion matrix on the validation sample was stored for each combination of parameter values and the process continued with the next iteration. Data from individual patients were kept together, never using data from the same patient for training and for validation in one iteration.

2. **Parameter optimization**: Once the cross-validation was completed, the parameter set which optimized the validation performance of each classifier type was selected.

3. **Final classifier training**: Each classifier was trained on the whole cross-validation dataset, using the optimal parameters obtained in the previous step.

<table>
<thead>
<tr>
<th>feature</th>
<th>definition</th>
<th>rationale</th>
</tr>
</thead>
<tbody>
<tr>
<td>pow</td>
<td>signal power in the entire window</td>
<td>higher overall power in signal windows with artifacts</td>
</tr>
<tr>
<td>powDiff</td>
<td>maximum power difference between adjacent 0.05s signal segments</td>
<td>power artifacts abrupt in time</td>
</tr>
<tr>
<td>sigP90, sigP95, sigP99</td>
<td>raw signal amplitude percentile (90th, 95th, 99th)</td>
<td>artifacts commonly include peaks of very large amplitude</td>
</tr>
<tr>
<td>knorm</td>
<td>value of the KS statistic of the one-sample Kolmogorov-Smirnov normality test in the entire window</td>
<td>pronounced non-normality in signals with artifacts</td>
</tr>
<tr>
<td>maxCorr</td>
<td>maximum correlation coefficient among multiple signal channels in 0.05 signal segments</td>
<td>mechanical artifacts often spread across channels and cause high outlier values and thus high correlation</td>
</tr>
<tr>
<td>psdP75, psdP90, psdP95, psdP99</td>
<td>percentile of norm. PSD</td>
<td>artifacts localized in spectrum - high spectral peaks</td>
</tr>
<tr>
<td>psdMax, psdStd</td>
<td>maximum and standard deviation of norm. PSD</td>
<td>global PSD description</td>
</tr>
<tr>
<td>psdMaxStep</td>
<td>maximum difference between adjacent bins of the norm. PSD</td>
<td>artifacts localized in spectrum - sharp spectral peaks</td>
</tr>
<tr>
<td>psdF100</td>
<td>maximum of the norm. PSD below 100 Hz</td>
<td>baseline artifacts at a well localized frequency</td>
</tr>
<tr>
<td>psdFreq</td>
<td>maximum of the norm. PSD, divided by median value below 5 kHz</td>
<td>additional normalization of the norm. PSD</td>
</tr>
<tr>
<td>psdPow</td>
<td>maximum of the norm. PSD in range 60-600 Hz, divided by mean norm. PSD between 1 and 3 kHz</td>
<td>power artifacts very common in this range</td>
</tr>
<tr>
<td>psdBase</td>
<td>maximum of the norm. PSD in range 1-60Hz, divided by mean norm. PSD between 1 and 3 kHz</td>
<td>baseline artifacts, normalized</td>
</tr>
<tr>
<td>maxAbsDiffPSD</td>
<td>maximum absolute distance between norm. PSD and mean norm. PSD of clean sample signal segments</td>
<td>high artifact peaks in PSD</td>
</tr>
</tbody>
</table>
4. **Out-of-sample performance estimation**: The final classifiers were used to classify data from the test dataset, and the final performance evaluation was stored.

![Figure 3: Overview of the crossvalidation (CV) scheme](image)

An overview of the crossvalidation procedure is also available in Figure 3. The presented validation scheme was chosen in order to provide as unbiased as possible an estimate of the classification performance on unseen data (i.e. data from unseen subjects).

As the dataset contained a notable class imbalance (about 75% of clean signals; see section 3.2 for detailed dataset statistics), we considered the classification error to be an inappropriate performance measure that should be optimized – 75% accuracy could be achieved by labeling all signals as clean. We therefore chose Youden’s J-statistic, computed as

\[ J = \text{sensitivity} + \text{specificity} - 1. \]  

(3)

In this paper, we use the convention denoting artifact-contaminated signal samples as a positive class. Thus, the sensitivity, also called the true positive rate, is computed as the ratio of correctly classified artifact samples (true-positives, TP), divided by the count of all artifact samples \( \text{TP} / P = \text{TP} / (\text{TP} + \text{FN}) \), where FN is the false negative count, i.e. the number of artifact samples incorrectly classified as clean. Specificity is then the false negative rate, computed as the ratio of correctly classified negative examples (clean signals, true negatives, TN) to all negative examples \( \text{TN} / N = \text{TN} / (\text{TN} + \text{FP}) \), where FP are false positives (clean signals incorrectly marked as artifacts).

3. **Experimental results**

3.1. **Overview of the collected data**

The collected and annotated data set contained a total of 1676 multi-channel recordings (5735 single-channel signals) from 99 MER exploration trajectories of 58 patients. The typical recording length (and also the median recording length) was 10 seconds, with a small proportion of shorter recordings. The total length of all recordings was 56814 signal seconds, which is 15h 47min. Prior to all analyses, the data were divided into two sets: *cross-validation* – for classifier training and parameter optimization, and *test* – for final performance evaluation. Detailed information about both sets can be found in Table 2.

<table>
<thead>
<tr>
<th>dataset</th>
<th>No. pat.</th>
<th>No. traj.</th>
<th>No. pos.</th>
<th>No. signals</th>
<th>total length [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>cross-validation set</td>
<td>55</td>
<td>93</td>
<td>1532</td>
<td>5015</td>
<td>49684</td>
</tr>
<tr>
<td>test set</td>
<td>3</td>
<td>6</td>
<td>144</td>
<td>720</td>
<td>7130</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>58</strong></td>
<td><strong>99</strong></td>
<td><strong>1676</strong></td>
<td><strong>5735</strong></td>
<td><strong>56814</strong></td>
</tr>
</tbody>
</table>

3.2. **Data annotation evaluation**

After the training procedure described above, the team of five raters annotated all MER data including a common sample of about 5% (95 ten-second MER signals). The agreement of the five raters with consensus annotation computed as a majority vote of all members was 93.5%, the Fleiss’ kappa 0.722, the consensus annotation contained 73.2% seconds of clean signal seconds and 26.8% seconds with artifacts.

Unlike the determination of a clean signal/an artifact on the initial database, the level of agreement on the exact artifact type was evaluated as poor, mainly due to the presence of many borderline cases that were difficult to assign to a particular category. The classifiers were therefore trained to distinguish between two classes (clean signals/contaminated signals), and the presented
proportion of the assigned artifact type on each database is provided for informative purposes only.

3.3. Feature evaluation

The classification power of each feature in the final feature set (the feature list in Table 1) was evaluated in terms of the area under the receiver-operator characteristic (AUC); histograms on the cross-validation set and also mean AUC values on the ten validation subsets can be found in Figure 4. It can be noted that the features based on steps and differences in PSD (maxAbsDiffPSD, psdMaxStep, psdStd and psdMax) showed the best discriminative properties with AUC values reaching 0.91, which may be considered very good. However, the additional features maxCorr and ksnorm showed relatively low detection capability, with AUC values around 0.63 and 0.58, respectively. However, it has to be noted that the AUC values reflect the detection capability in a single-feature classification scenario and are therefore biased towards features designed for detecting the most prevalent artifact types (such as FREQ in the case of the spectral features). In addition, the histograms presented here suggest that some of the features are strongly correlated and overlapping (which was further confirmed by correlation analysis). It can be assumed that an appropriate feature selection method should be capable of selecting a combination of features, consisting of complementary features, including features with high additional benefit to the feature set despite their low overall performance on the whole unbalanced dataset. All 19 features were therefore computed, and were used as an input into the classifiers.

3.4. Classification results

The classification performance evaluation was done according to the procedure described above and depicted in Figure 3. After parameter selection, carried out on the cross-validation validation sets, the performance of the final classifiers was tested on the test dataset. The results of the performance evaluation are presented in Table 3, and the best performing classifiers in each category (stationary segmentation, maxDiffPSD, tree and boosting) are shown in grey. An additional point of view on the classification performance of the different methods across the 10 cross-validation folds can be found in Figure 5. The results show that the best performing classifiers: Bagging with 75 learners and the decision tree, achieved classification accuracy close to 90% on the cross-validation set and accuracy higher than 88% on the unseen validation set.

The best-performing multi-feature classifiers were closely followed by SVM (with a drop in accuracy of about 0.5%) and by the simple maxDiffPSD method, with 87.7% cross-validation accuracy and 86.2% test set accuracy. Considering the performance evaluation in Figure 6 a), it is apparent that slightly higher accuracy could be achieved by using a higher threshold, at the cost of slightly reduced sensitivity.

The results for the segmentation approaches COV and SWT, based on original research reported on in [22, 23] and [24], were inferior by approximately 5-10% to the best-performing methods mentioned above. This may be attributed to the fact that the methods only segment the signal at substantial change points, and use no information about the properties of clean signals or of artifacts. In the cases of stationary long-term artifacts, e.g. some cases of the FREQ type, a long and contaminated signal section may be selected as the longest stationary component. This property is inherent to the unsupervised nature of both the original methods and their extended versions, which are implemented in this paper.

It may be noted from the parameter values in Table 3 that the optimal parameters for both the COV technique and the SWT technique included the lower bound of available time-window lengths (0.25 s) and also the lowest available aggregation threshold: each second was divided into four segments, and the presence of a single segment labeled by the classifier as an artifact was sufficient for the whole second to be labeled as an artifact. This property was in all cases also chosen for longer windows (0.33 s and 0.5 s), and it apparently provides the classifier with better ability to detect short-term events appearing within the one-second signal. The dependency of cross-validation performance on the detection threshold and on the window length for both methods can be found in Figure 6 b) and c). It can be noted that the performance was very close for all short windows, especially 0.25 s and 0.33 s. The use of even shorter windows would most likely lead to only very minor, if any, performance improvement.

4. Discussion

All supervised classifiers presented above (i.e. all except COV and SWT) achieved test-set accuracy between 85.5% (RobustBoost) and 89.0% (bagging) – see Table 3. A good classification performance with test-set accuracy of 86.2% was observed even with the simplest spectral method (maxDiffPSD), which relies on smoothness of the PSD spectrum in clean signals and on sharp peaks in the PSD spectrum of signals with artifacts. This result is in line with performance evaluation of the individual features, where the 10 best-performing features (out of the total of 19) were PSD-based, and the maxDiffPSD
Figure 4: Histograms of all feature values on the cross-validation database, sorted by area under the receiver-operator characteristic (AUC), in descending order with artifacts in blue and clean signals in red. In each subplot, the x-axis represents the value of the respective feature, and the y-axis represents the relative frequency of signal segments with a given feature value in the cross-validation set. The AUC values (mean ± std) were calculated on the ten validation subsets. The similarity between the shapes of the histograms (and also between the feature definitions) suggests high inter-feature correlation. The classification methods need to be chosen in order to handle high correlation within the feature set.

Despite the extremely high detection accuracies of change-point detection, presented by the authors of segmentation approaches COV [22, 23] and SWT [24], which were almost 100% on simulated data, we show that the real-world performance of these methods in selecting clean signal segments may not be as outstanding, and has been superseded by a relatively large margin in all settings by all the other newly proposed classifiers. It has to be noted that the scenario used in this paper was different – detecting artifacts rather than change points – and we used a modified version of both algorithms. Nevertheless, we believe that the scenario presented here is closer to the typical use-case of identifying clean MER signal sections for further processing, and the modifications are therefore well justified.

Thanks to the extensive procedures for identifying appropriate artifact types for annotation, and also for harmonizing the team of raters, the annotation reliability was satisfactory (achieving 93.5% accuracy on the proofing sample). However, there still remained a significant zone with unclear annotation. An inspection of signal examples with low inter-rater agreement showed mostly unclear cases where the artifact was either very weak and therefore questionable, or very short in time and easy to mistake for a physiological spike. Both of these cases are very hard to distinguish objectively with no ground-truth data, which is achievable only in laboratory conditions or in computer simulations.
Table 3: **Classifier evaluation results** on the ten *cross-validation* sets and on the independent *test set*. The classifier parameters resulting from optimization on the *cross-validation* set are shown together in the second column. The best-performing classifier of each type is shown in grey.

<table>
<thead>
<tr>
<th>classifier type</th>
<th>parameters</th>
<th>cross-validation</th>
<th>test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc</td>
<td>Sens</td>
</tr>
<tr>
<td>COV</td>
<td>win. .25s, perc. 25%,. thr 1.20</td>
<td>77.7</td>
<td>51.1</td>
</tr>
<tr>
<td>SWT</td>
<td>win. .25s, perc. 25%,. thr 10.0</td>
<td>73.2</td>
<td>66.4</td>
</tr>
<tr>
<td>maxDiffPSD</td>
<td>threshold 0.0085</td>
<td>87.7</td>
<td>81.7</td>
</tr>
<tr>
<td>Tree</td>
<td>parent: 1000. leaf 250. deviance</td>
<td>89.5</td>
<td>76.3</td>
</tr>
<tr>
<td>SVM</td>
<td>feat.sel.: linear. thr 0.001. linear</td>
<td>88.8</td>
<td>83.3</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>100 learners. learn. Rate 0.7</td>
<td>89.8</td>
<td>75.7</td>
</tr>
<tr>
<td>Bagging</td>
<td>75 learners</td>
<td>90.0</td>
<td>75.9</td>
</tr>
<tr>
<td>GentleBoost</td>
<td>250 learners. lrn. rate 0.1</td>
<td>89.8</td>
<td>75.8</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>150 learners. lrn. rate 1</td>
<td>90.0</td>
<td>75.6</td>
</tr>
<tr>
<td>RobustBoost</td>
<td>10 learners. e. goal 0.2. e. marg. .1</td>
<td>86.7</td>
<td>83.2</td>
</tr>
</tbody>
</table>

**Figure 5:** **Classification results on the cross-validation set.** The boxes represent the crossvalidation performance across the ten folds (boxes extend from the 25th to the 75th percentile, median in red), the green markers denote the test performance on the test dataset. The proposed (supervised) techniques provide a clear advantage over the unsupervised SWT and COV methods.

In our experience, the spectrogram was very helpful for revealing artifacts not easily visible in the time series plot (especially the *FREQ* type). Auditory inspection was also very useful. Even with the use of these tools, our early experiments showed that accurate identification of the exact artifact start and end time is a very challenging task, mainly due to the gradual nature of many artifacts. Our annotation procedure therefore used one-second segments, as did the classifiers presented above. We believe that a similar technique could also be used for shorter signal segments. Nevertheless, we find one-second windows convenient for manual annotation.

All the supervised techniques presented above may also be applied to a sliding window, and may thus be used as online algorithms, which may be useful e.g. in animal MER studies or in closed-loop scenarios. However, the presented versions of both the COV and SWT segmentation techniques search for the largest stationary component in the whole signal, and therefore cannot be used in real time. Note that this is in contrast with the original versions of the segmentation techniques aimed at change-point detection, which could be applied to subsequent window-pairs upon recording in real time.

### 4.1. Limitations

The main limitation stems from the composition of the training and validation set: all data were from DBS microelectrode trajectories, targeting the subthalamic nucleus in a single DBS center using one recording system. However, after performing pilot testing on additional data from three other DBS centers, we believe that the methodology presented here can be applied in general.

All classifiers presented in this paper ignore the specific types of annotated artifacts, and thus distinguish
Figure 6: Impact of detection threshold on classification performance. An evaluation of artifact detection using a) the maximum difference from a clean sample spectrum $\text{maxDiffPSD}$, b) the extended COV method, and c) the extended SWT method. The performance on the crossvalidation set is shown versus the detection threshold for each method. For all methods, the threshold which optimizes the J-statistic is shown as a vertical dotted line, and it achieves sub-optimal accuracy; slightly greater accuracy could be achieved at the cost of decreased sensitivity – i.e. more overlooked artifacts. (COV: autocorrelation-based stationary segmentation, adapted from [23], SWT: Wavelet-transform-based stationary segmentation, adapted from [24])

only a clean MER signal from a signal contaminated with an artifact of any type. This was due to the rather low level of agreement on the specific artifact type (unlike the good agreement on clean signal/artifacts), and due to the fact that a high proportion of segments were contaminated with artifacts of multiple types at the same time (see Table A.1 in the Appendix). However, the ability to classify only artifacts of a particular type might be useful, e.g. for detecting only artifacts having a negative impact in one’s specific data processing pipeline. In such cases, multi-feature classifiers such as decision trees or SVM can be used.

This paper focuses on heavily-used single-channel MER data (i.e. one channel or multiple electrodes spaced away in the order of mm or cm), e.g. data obtained during DBS microexploration, as opposed to artifacts from concurrent electrical stimulation, which can be well described and have been sufficiently studied in the litera-
ture [27, 28, 29, 3], and also artifact detection methods based on blind source separation or inter-electrode correlation, which can be applied to microelectrode arrays [30, 26].

5. Conclusion

In this study, we have shown that external noise poses a serious issue in human subthalamic DBS MER recordings: on our extensive database containing over 15 hours of manually-annotated MER, more than 25% of the recording time was affected by clearly nonphysiological artifacts. The proposed supervised classification methods showed good classification performance (88.2% test-set accuracy for the decision-tree classifier), which is close to the accuracy of the annotation itself (93.5% inter-rater agreement). By contrast, existing unsupervised methods (COV and SWT) showed classification accuracy 5-10 percentage points inferior, and proved to be not well suited for the artifact vs. clean signal classification task. On the basis of the the results presented here, it can be stated that the proposed supervised methods provide an efficient way to identify artifact-bearing MER segments – especially the very simple maxDiff-PSD, which is very easy to implement and deploy into any MER pre-processing pipeline.

The supplementary material includes source codes for the maxDiffPSD and COV methods, as well as an example of MER signals. We are currently preparing a software package for MER data annotation that will contain additional classification methods and will be presented in an upcoming paper.

Acknowledgement

We gratefully acknowledge the help of Michal Olbrich and Tomas Grubhoffer during data annotation.

The work presented in this paper was supported by the Czech Science Foundation (GACR), under grant no. 16-13323S, by the the Ministry of Education Youth and Sports, under NPU I program Nr. LO1611, and by grant no. SGS16/231/OHK3/3T/13 of the students’ grant agency of the CTU in Prague.
Appendix A. Artifact content in the database

Table A.1 shows the percentage of signal seconds in both databases containing a given combination of artifacts.

Table A.1: Percentage of assigned artifact type combinations in each dataset, CLN represents clean signal seconds.

<table>
<thead>
<tr>
<th></th>
<th>CLN</th>
<th>POW</th>
<th>BASE</th>
<th>FREQ</th>
<th>POW</th>
<th>BASE</th>
<th>FREQ</th>
<th>POW</th>
<th>BASE</th>
<th>FREQ</th>
<th>BASE</th>
<th>FREQ</th>
<th>BASE</th>
<th>FREQ</th>
<th>BASE</th>
<th>FREQ</th>
<th>BASE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>64.7</td>
<td>6.6</td>
<td>3.0</td>
<td>0.5</td>
<td>17.3</td>
<td>9.6</td>
<td>1.3</td>
<td>0.6</td>
<td>0.3</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Appendix B. Extension of the stationary segmentation techniques

This section describes the extension of stationary segmentation methods by Aboy, Falkenberg and Guarnizo [22, 23, 24], presented in our previous paper [25].

In their original version, both methods divide the signal $X$ into $m$ non-overlapping segments $(X_1, X_2, ..., X_m)$ and compute statistics $γ(X_i)$ for each segment, where $γ(·)$ is the autocorrelation function of the segment (COV) or the stationary wavelet transform (SWT). In the next step, the variance of each transformed segment is calculated according to

$$\nu_i = \text{var}(γ(X_i)), i \in (1, m), \quad \text{ (B.1)}$$

Variances of neighboring segments are then compared according to:

$$d_{ij} = \frac{\text{max}(\nu_i, \nu_j)}{\text{min}(\nu_i, \nu_j)}, i \in (1, m - 1), j = i + 1 \quad \text{ (B.2)}$$

Divisions between segments with distance statistic $d_{ij}$ exceeding a manually pre-chosen threshold $Θ$ then determine breakpoints between stationary segments. The longest stationary segment can then be found and returned.

We further extend this method by computing the distances between all possible segment pairs, forming a distance matrix

$$D = \begin{pmatrix} 0 & d_{1,2} & \cdots & d_{1,m} \\ d_{2,1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ d_{m,1} & d_{m,2} & \cdots & 0 \end{pmatrix}. \quad \text{ (B.3)}$$

Note that due to the properties of the distance measure from Equation B.2 the matrix is symmetric with $d_{ij} = d_{ji}$.

In the next step, all values $d_{ij}$ exceeding the classification threshold $Θ$ are replaced by zeros, and the remainder are replaced by one, leading to a graph defined by the following adjacency matrix:

$$E = \begin{pmatrix} 0 & e_{1,2} & \cdots & e_{1,m} \\ e_{2,1} & 0 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ e_{m,1} & e_{m,2} & \cdots & 0 \end{pmatrix}, e_{ij} = \begin{cases} 1, & \text{if } d_{ij} < Θ \\ 0, & \text{otherwise} \end{cases} \quad \text{ (B.4)}$$

The graph represented by adjacency matrix $E$ is then scanned for the maximum component of transitively adjacent segments. With this modification, the algorithm returns the largest component of similarity in the original signal, which may even be a non-contiguous signal subset. The procedure used to search the maximum signal component from the adjacency matrix is outlined in Matlab-style pseudocode in Algorithm 1. Note that the method only requires all segments to be connected by a non-interrupted path – sub-threshold similarity between all possible segment pairs within the component is not required. The value of the optimal detection threshold will therefore also differ from the originally published methods.

Assuming higher stationarity in artifact-free segments, this method allows comparison with manual signal annotation for the whole – typically ten-second – signal. In addition, in analyses where signal continuity is not required (such as in background activity feature calculation), this approach may minimize the amount of unnecessarily removed data. In cases where signal continuity is necessary (e.g. spike-train analyses), the largest uninterrupt signal segment is the only option.

Appendix C. Power spectral density estimation

In order to estimate the PSD spectrum, we used Welch’s method with the window length equal to the number of spectral bins in the discrete Fourier transform. The number of bins used throughout the paper was $M = 2048$, which achieved good artifact detection sensitivity, as shown in Fig. C.1. In this figure, we demonstrate the impact of a decreasing number of spectral bins on the sensitivity of the resulting PSD spectrum to narrow-band MER signal components – which are typical for the prevailing FREQ artifacts. Alternatively, other PSD estimation techniques (such as parametric estimation or other methods), or time-frequency analysis, may be utilized in future studies in order to achieve higher sensitivity to artifacts, while preserving good temporal resolution.
Appendix D. A demonstration of artifact impact on spike detection and sorting

In order to demonstrate the impact of artifacts on further processing of MER signals, we show how artifacts affect spike sorting. A five-second segment of a MER signal (signal No. 21 in the supplementary data) was processed using the WaveClus spike sorting toolbox [9] in two ways: i) as a whole, and ii) with the artifact (the 2nd second of the signal) removed - see Figure D.1.

Owing to the robust nature of the spike detection threshold, artifact removal led only to a small threshold reduction. However, even this small change led to the detection of 22 additional spikes when the artifact was removed. Evaluating the quality of the spike sorting results, processing the whole signal led to the detection of 233 spikes, out of which 6 were rejected as outliers and the remainder was divided into two clusters with 204 spikes and 23 spikes. Out of the 204 spikes assigned to the first cluster, 41 spikes (20%) were within the 3 ms refractory period, indicating heavy noise in the data. The second cluster contained 23 spikes, 20 of which were within the 3 ms refractory period (87%) and contained almost exclusively noise.

When processing the signal with the artifact removed, the method detected 154 spikes, two of which were marked as outliers and the remainder was determined to come from a single putative neuron. Only 9 spikes (6%) were within the 3 ms refractory period, and the variability of the spike waveforms was significantly reduced. An additional inspection of the inter-spike interval histogram also revealed a much more realistic shape of the mean spike waveform. We therefore conclude that artifact removal can increase the sensitivity and the precision of spike detection and sorting, at the cost of rejecting noisy signal segments.

Appendix E. List of optimized classifier parameters

An overview and the ranges of all parameters optimized during the cross-validation are presented in Table E.1
Figure D.1: **Impact of artifacts on spike detection and sorting** using the WaveClus toolbox [9].

A) A raw MER signal with a detected one-second artifact marked in light grey. The blue/red marks (top) denote 227 spikes detected using the blue threshold on the whole signal and sorted automatically into two clusters. Longer marks denote 97 spikes undetected in the artifact-free scenario. The orange marks (bottom) denote 152 spikes, detected on the artifact-free signal using the orange dashed threshold. In this case, longer marks denote 22 spikes undetected on the whole signal.

B) Clustering result on the whole signal: all detected spike shapes in grey, overlaid by the cluster mean. Out of the 204 spikes assigned to the first cluster, 41 spikes (20%) were within the 3ms refractory period, indicating heavy noise in the data. Clear amplitude clipping and technical character can be seen in cluster 2, which contains exclusively noise spikes.

C) Spike sorting after artifact removal shows much more consistent spike shapes, with lower variability around the mean spike waveform (orange).
Figure D.2: Example of a single DBS MER exploration annotated using the presented extension of the COV method (window length 0.25 s, threshold 1.2).

Table E.1: Overview of optimized classifier parameters and values

<table>
<thead>
<tr>
<th>classifier</th>
<th>optimized parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>stationary</td>
<td>method: i) COV (covariance, Aboy) or ii) SWT (wavelets, Guarnizo)</td>
</tr>
<tr>
<td>segmentation</td>
<td>segment length: [.25,.33,.5,1]</td>
</tr>
<tr>
<td></td>
<td>aggregation threshold: for .25 s window: [1, 2, 3, 4], for .33 s [1, 2, 3], for .5 s [1,2]</td>
</tr>
<tr>
<td></td>
<td>threshold for COV: &lt;.8, 3.5&gt;in .1 steps</td>
</tr>
<tr>
<td></td>
<td>threshold for SWT: &lt;9.5, 13&gt;in .1 steps</td>
</tr>
<tr>
<td>diffPSD</td>
<td>threshold: &lt;0,0.025&gt;in 0.0005 steps</td>
</tr>
<tr>
<td>decision tree</td>
<td>split criterion: i) Gini’s diversity index ii) max. deviance reduction</td>
</tr>
<tr>
<td></td>
<td>min size of parent node: {1, 100, 200, ...,500, 100, 1500, 5000}</td>
</tr>
<tr>
<td></td>
<td>min. leaf size: {1, 250, 500, 750, ...,2500} maximum up to half of current parent node min. size</td>
</tr>
<tr>
<td>SVM</td>
<td>feature selection criterion: quadratic, linear, mahalanobis</td>
</tr>
<tr>
<td></td>
<td>feature selection stopping tolerance: {.001, .005, .01}</td>
</tr>
<tr>
<td></td>
<td>SVM method: i) Sequential minimal optimization or ii) least squares</td>
</tr>
<tr>
<td></td>
<td>SVM kernel: i) linear ii) radial basis function (RBF)</td>
</tr>
<tr>
<td></td>
<td>SVM kernel sigma (only for RBF): [.5,1,2]</td>
</tr>
<tr>
<td>Boosting</td>
<td>algorithm: AdaBoostM1, LogitBoost, GentleBoost, RobustBoost, Bag</td>
</tr>
<tr>
<td></td>
<td>number of learners: {10, 20,...,50,75,150,200,250}</td>
</tr>
<tr>
<td></td>
<td>learning rate: {.1, .4, .7, 1}</td>
</tr>
<tr>
<td></td>
<td>RobustBoost error goal: {.05, .1, .15, .2}</td>
</tr>
<tr>
<td></td>
<td>RobustBoost error max margin: {.01, .05,.1}</td>
</tr>
</tbody>
</table>
Algorithm 1 Identification of the maximum component from the adjacency matrix

input: $E$: $m \times m$ adjacency matrix, $m$ is number of segments
output: $\text{max\_comp}$; indices of maximum component

$\text{comp} = \text{zeros}(1, m)$; % denotes which segment belongs to which component
$\text{act\_comp} = 0$; % Actual component

while any($\text{comp} == 0$) do % loop as long as there are unassigned segments
    $\text{open} =$ first zero in $\text{comp}$
    $\text{closed} = []$
    $\text{act\_comp} = \text{act\_comp} + 1$
    while NOT isempty($\text{open}$) do
        % assign actual comp. to actual segment
        $\text{comp}(\text{open}(1)) = \text{act\_comp}$
        % expand current state (all segments adjacent to current segment)
        $\text{children} = \text{find}(E(\text{open}(1), :))$
        % take the first element from open, find to which segments exists a direct path
        for $ch$ in $\text{children}$ do
            if $ch$ not in $\text{open}$ OR $\text{closed}$ then
                $\text{open} = [\text{open} \ ch]$ % add ch to open
            end if
        end for
        % move current node from open to closed
        $\text{closed} = [\text{closed} \ ch]$ % add ch to closed
        $\text{open} = \text{open}(2:end)$
    end while
    end while

% find the largest component
$\text{comp\_len} = \text{zeros}(1, \text{act\_comp})$
for $\text{cur}$ in $1: \text{act\_comp}$ do
    $\text{comp\_len}(:, \text{cur}) = \text{sum}(\text{comp} == \text{cur})$
end for
$[\sim, \text{max\_comp}] = \text{max}(\text{comp\_len})$
return $\text{max\_comp}$
References


