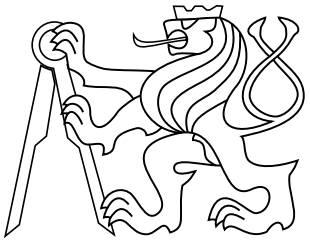
Master's thesis

Prediction of Epileptic Seizures from Intracranial EEG

Bc. Lenka Zoulová



2017

Ing. Martin Macaš, Ph.D.

Czech Technical University in Prague
Faculty of Electrical Engineering, Katedra kybernetiky

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

DIPLOMA THESIS ASSIGNMENT

Student: Bc. Lenka Z o u l o v á

Study programme: Biomedical Engineering and Informatics

Specialisation: Biomedical Informatics

Title of Diploma Thesis: Prediction of Epileptic Seizures from Intracranial EEG

Guidelines:

1. Propose a system for classification of preictal intracranial EEG records. Choose a proper preprocessing of signals, feature extraction procedure and pattern recognition methods of different types. Justify your choices.
2. Implement the system in Matlab, use data provided by the supervisor.
3. Test the system, compare different classifiers, evaluate and identify future challenges.

Bibliography/Sources:

[1] Gadhoumi, K., Lina, J. M., Mormann, F., & Gotman, J. (2016). Seizure prediction for therapeutic devices: A review. Journal of neuroscience methods, 260, 270-282.

[2] Cook MJ, O'Brien TJ, Berkovic SF, Murphy M, Morokoff A, Fabinyi G, D'Souza W, Yerra R, Archer J, Litewka L, Hosking S, Lightfoot P, Ruedebusch V, Sheffield WD, Snyder D, Leyde K, Himes D (2013) Prediction of seizure likelihood with a long-term, implanted seizure advisory system in patients with drug-resistant epilepsy: a first-in-man study. LANCET NEUROL 12:563-571.

Diploma Thesis Supervisor: Ing. Martin Macaš, Ph.D.

Valid until: the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 11, 2017

Acknowledgement

I would first like to thank my thesis advisor Ing. Martin Macaš, Ph.D. for his guidance.

I would also like to thank my sister for her help and great deal of patience.

Last but not the least, I would like to thank my husband for his support and his trust in me.

Author statement for undergraduate thesis:

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, date . . . .

signature

Abstrakt

Epilepsie je onemocnění mozku vyznačující se opakovanými záchvaty. Tyto záchvaty ztěžují pacientům život a v některých situacích mohou mít fatální následky. Stále neexistuje spolehlivá metoda, která by epilepsii vyléčila nebo potlačila její příznaky.

Nedávné studie potvrdily, že lze v mozkové aktivitě pozorovat změny ještě dříve, než dojde k záchvatu. Tyto změny zatím nebyly popsány. Zatím není ani zjištěné, jak dlouho před blížícím se záchvatem jsou změny pozorovatelné.

Cílem této práce je navrhnout postup klasifikace úseků předcházejících epileptickému záchvatu v lidském nitrolebečním záznamu EEG. Tento úsek byl definován jako jedna hodina před záchvatem.

Pro optimalizaci a validaci algoritmu byla použita data volně dostupná z portálu ieeg.org. Data od pěti pacientů byla použita pro optimalizaci a další dva datasety pro otestování. Pacienti byli muži i ženy různého věku v rozmezí od 3 do 62 let. Data byla rozdělena na desetiminutové preictal (před záchvatem) a interictal (běžná aktivita) úseky.

Bylo vyzkoušeno více přístupů, jak se vypořádat s chybějícími hodnotami v datech, ale nakonec byla pouze nahrazena nulou. Také byl řešen problém s různým počtem vzorků z jednotlivých tříd. Tato nevyváženost byla vyřešena namnožením vzorků z menší třídy pomocí vygenerování nejbližších sousedů. Bylo navrženo okolo padesáti charakteristik a hladovým algoritmem bylo vybráno osm, které byly použity jako příznaky. Z mnoha testovaných klasifikátorů byl vybrán Bagged Trees.

S tímto nastavením bylo na nezávislých testovacích datech dosaženo hodnoty plochy pod ROC křivkou rovné 0.8405. Tyto výsledky stále nejsou dostačující pro využití v praxi. A to hlavně z toho důvodu, že není možné zaručit správnou klasifikaci všech úseků předcházejících záchvatu. Je třeba algoritmus ještě optimalizovat a otestovat na větším množství dat, které nejsou zatím k dispozici.

Klíčová slova

epilepsie; predikce; iEEG; strojové učení; klasifikace; evoluční algoritmus

Abstract

Epilepsy is neurological disease which is characterized by repeated seizures. These seizures make patient life more difficult and in some situation seizure can have fatal consequences. There is still not reliable method to cure epilepsy or eliminate symptoms.

Last studies confirmed that there are detectable changes in brain activity before seizure coming. These changes was not describes yet. Even amount of time before seizure in which are changes obvious (preictal segment) is not know.

Target of this work is to propose procedure of patient specific classification of preictal segment in human intracranial EEG record. Preictal segments was defined as one hour before seizure.

Freely available data from ieeg.org portal were used for optimization and validation algorithm. Data from five patient were used for optimization and two datasets for test. Patients were men and women of different age from 3 to 62 years old. Data were segmented and divided into ten-minute preictal (before seizures) and interictal (normal activity) segments.

More approaches how to work with missing values in the data were tried but at the end they were only replaced by zero. Also some methods of solving problem with different amount of samples from each class were proposed. The samples of minority class by generating the nearest neighbors to solve imbalance were multiplied. About fifty characteristic of data were proposed and eight of them were selected as features by greedy algorithm. Bagged trees from many tested classifiers were chosen.

With this setting average of area under ROC curve equal 0.8405 on independent test sets was achieved. But this result is still not sufficient for the practice use. Especially because is not possible to assure correct classification of all preictal segments. It is needed to optimize and test the algorithm on bigger amount of data which is not available at this moment.

Keywords

epilepsy; prediction; iEEG; machine learning; classification; evolutionary algorithm

Contents

1 Introduction

1.1 Motivation

Epilepsy affected about 50 million people worldwide. It is difficult to live normal life with this serious disease. In many areas of the world driving for people which have experience with seizures is restricted or permitted. Much more activities are problematic for these people, for example swimming as well as crossing the street. It could end worse when seizure occur during these activities. It is possible to attenuate symptoms of epilepsy by dietary changes or medication but these method are not absolutely reliable. Even brain surgery does not always fix the problem. People should take precautions and keep calm if we could predict a seizure coming.

Longterm records of brain activity are available nowadays. Using these data and techniques of machine learning should help to find a way how to predict seizures. It is important for prediction to be reliable. There can be some false positive classification (notification even though there is no seizure coming) but there should not be false negative classification (seizure coming without warning). Patient will have implanted the device to record brain activity.The second portable device will be connected to the first. The second device should warn of seizure coming and patient should take his medication and keep calm.

1.2 Objectives

In this work I will try to propose a patient specific system for classification of preictal intracranial EEG records. It is needed to choose a proper preprocessing of signals, feature extraction procedure and pattern recognition methods. I will discuss chosen methods. I implement methods in Matlab and optimize them on a free available intracranial EEG data. The final system will be tested on an independent data. At the end I evaluate the accurancy and discuss future challenges.

1.3 State of the art

Not long ago only medication or surgery was used to treat epilepsy. But for many patients antiepileptic drugs does not work and moreover it can be toxic. Even

surgery does not work for all of them. New methods of treatment was developed after the year 2000. To improve efficiency and decrease a toxicity of medication new techniques of drug delivery were introduced. For example it can be done by slow-release form of medication [1] or drug-loaded nanoparticles [2]. Another new treatment method of epilepsy is focal cooling as an alternative to surgery. The brain can be focally cooled to the temperature between 20°C and 25°C which leads to terminate epileptic discharges without the neuronal damage [3]. Next technique for treating drug-resistant epilepsy is electrical stimulation [4, 5].

Most of these methods are based on continuous stimulation but there are some methods which stimulate only when the seizure is detected. It is called the closed-loop stimulation. The advantages of this approach is that less stimulation is needed which decreases side effects of the long term stimulation and leads to lower energy consumption of the device. To maximize the effect of the stimulation is better to use the stimulation before the seizure comes [6]. The most effective time to apply the stimulation is not known.

The seizure prediction have to be sensitive and specific for the medical use. It is necessary for maximizing the efficiency and minimizing the side effects of the stimulation. Most often EEG records for prediction are used. It was not certain if there are some detectable changes before the seizure but later studies approve it [7]. Sensitivity and specificity requirements are not standardized. Some studies define the maximum false prediction rate. [8, 9] There is an effort to create guidelines for evaluating quality of seizure prediction methods. [10, 11] The studies mostly agree that the seizure prediction algorithm have to be better than chance but they differ in the way of comparing these predictions. Algorithm should be tested by using long-lasting, continuous and unseen EEG recordings. This data have to cover all pathological and physiological states of patients. The methods can be evaluated in different ways so it is difficult to compare algorithms.

Algorithms mostly differs in features which they used and in the way the features are tracked. The next paragraphs describes four algorithms published in the last few years. The works that describe them are available on IEEE Xplore.

Seizure Prediction using Hilbert Huang Transform on Field Programmable Gate Array

In [12], micro-volt scalp EEG was used. Preictal period was defined as 5 minutes before the seizure. Interval was divided into 15s non-overlaping segments. Segments were decomposed by using Empirical Mode Decomposition. Hilbert Huang Transform was used as features. The Least Square SVM (LSSVM) with a Radial Basis Function kernel and a Logistic Regressor (LR) were used for classification. Patient specific classifier was used. Area under the curve (AUC) of the Receiver Operating Characteristic (ROC) curve was between 0.995 and 1 for LSSVM and

between 0.928 and 1 for LR.

SVM-Based System for Prediction of Epileptic Seizures From iEEG Signal

In [13], intracranial EEG data from dogs brain were used. Continuous stream of records was segmented and labeled. Preictal period was defined as one hour before the seizure. Features were extracted from 20 s window and one hour segments were used for prediction. Three spectral characteristics were used as features in this work. These were spectral power in six Berger frequency bands, signal in time from six bands and crosscorrelation matrix. Features from 20 s windows were classified by SVM based system and prediction for one hour system was calculated. False positive varied between 0 and 5 % and false negative is between 0 and 33 %.

Seizure Prediction Using Undulated Global and Local Features

In [14], the patient specific prediction from intracranial data was presented. 30 minutes of signals before seizure was used as preictal segments. Global features were extracted from relative change among signal-type epoch. Local features were calculated as customized fluctuation and deviation from 10 s long segments with 128 samples overlap. LSSVM was use for classification. 5 minutes long window was use for decision if seizure coming. Prediction accuracy was 95.4% and false positive rate corresponds to 0.36 hour a day.

Seizure prediction using long-term fragmented intracranial canine and human EEG recordings

In [15], the authors presented seizure prediction from intracranial EEG recorded on dogs and human. They tried two feature sets and three classifiers. Features were extracted from 4 seconds long segments with 50% overlap. First feature set contained relative spectral power and spectral power ratio. Second feature set contained cross correlation coefficients between electrodes. It was extracted from 10 s long segments with 50% overlap. Classification and Regression Tree was used to choose 50 most important features. AdaBoost, SVM and artificial neural network (ANN) were used for classification. They combined best features for each object and their best result was achieved by ANN with mean AUC equal to 0.8884.

2 Problem definition

Epilepsy

There are small cells called neurons in the brain. Neuron usually consists of cell body, one long axon and several short dendrites. It is shown in Figure 1. Signal from other neurons come to the cell body through dendrites. Information is processed in cell body and sent next to other cells by axon as an electrical signal. Axon ends by connections called synapses. Synapses transfer information using chemical transmitter to dendrite of another cell or directly to cell body [16].

Synapses can be excitatory or inhibitory. Excitatory synapses increase activity in target neuron. Inhibitory synapses decrease this activity. Both types of synapses change concentration of ions and polarization in neuron surroundings. This stimuli can lead to produce action potential. Then axon carry electrical signal to next neuron. Producing of action potentials called firing. Neuron can fire with different frequency but all impulse have same strength. It called all-or-none

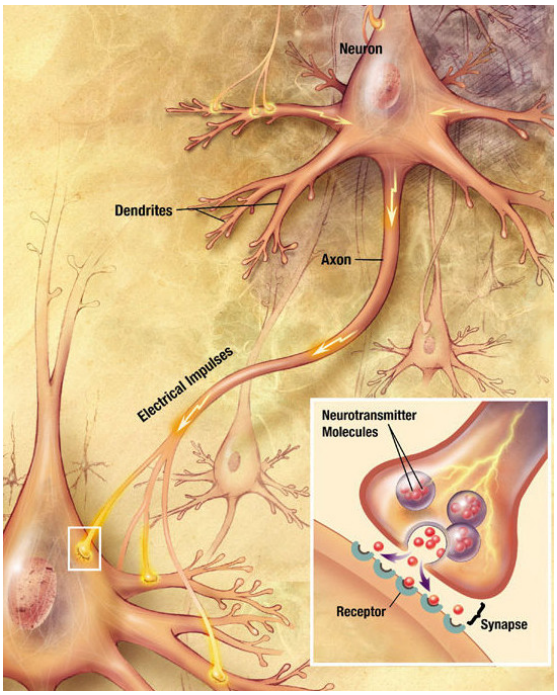


Fig. 1 Neuron (Source: US National Institutes of Health, National Institute on Aging [16])

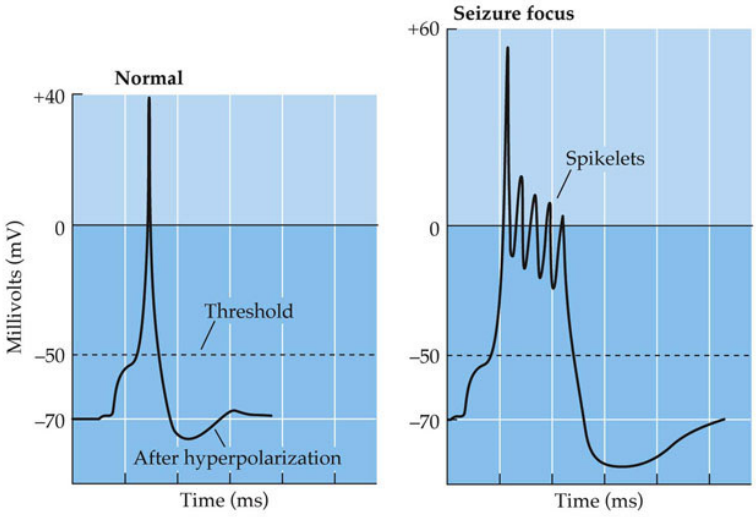


Fig. 2 Action potential in normal activity and during seizure (Source: Jerrold S. Meyer and Linda F. Quenzer [20])

principle. Intensity of stimulation not affect amplitude of signal but can affect its frequency.

There are some theories about information coding in brain and there are probably more principles of coding information in brain itself. One model describes information coding as firing rate. Average number of spikes per unit time depends on strength of stimulus. This behavior is typical for motor neurons [17]. On the one hand this method is inefficient but it is very robust on the other hand. Another model called temporal coding. According to this theory information is coded by precise timing of single spikes [18].

Neurons firing seems to be randomly at first time. Normally brain activity is non synchronous. After neuron firing it becomes more resistant to produce new spikes as seen in Figure 2. These mechanisms are broken during epilepsy seizures [19]. As a result of this abnormalities a group of neurons begin firing excessive and synchronized. It can be caused by head injury, infection, genetic or development condition but most often the cause is unknown.

Symptoms of disease must accomplish at least one of the following conditions to diagnose it as epilepsy [21].

First, there are two unprovoked (or reflex) seizures occurring greater than 24 hours. Seizures can be result of concussion, fever or alcohol-withdrawal. This cases are marked as provoked and would not lead to a diagnosis of epilepsy. The term unprovoked means that there is no temporary factor lowering the threshold for produce the seizure. It is quite misleading because we can never eliminate presence of this factor with certainty.

Second case, when epilepsy can be diagnosed, is one unprovoked (or reflex) seizure and a probability of further seizures similar to the general recurrence

risk (at least 60%) after two unprovoked seizures, occurring over the next 10 year. This condition accomplish for example patient with one seizure at least a month after a stroke.

Third option to diagnose epilepsy is diagnosis of an epilepsy syndrome. It can be apparent from the record of brain activity. There can be abnormalities even if patient have no seizure yet.

Brain electrical activity is more often measured by electrodes placed on the scalp as shown in Figure 3. This method is called electroencephalography (EEG). It is quite cheap but not much precise way of measuring neurons activity. Each electrode records summed signal from many neurons in the brain. Moreover, signal goes through skull to electrodes and therefore is weak. Particular result of these issues is the use of intracranial electroencephalography. Electrodes are placed directly on the brain and recorded signals are stronger.

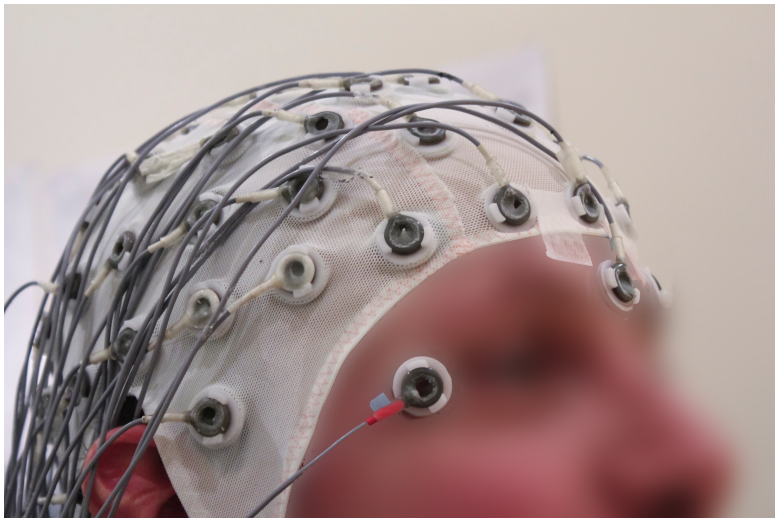


Fig. 3 Measuring of EEG (Source: Chris Hope, [22])

Machine learning

Machine learning is subfield of computer science. The main task is to create model which describes given samples. Such model is used for making decision, classification or prediction. Machine learning algorithms typically use learned rules instead of explicitly programmed instructions. Machine learning can be unsupervised learning or supervised learning. When we use supervised learning we need to know labels of samples. Computer tries to find model which maps input to output most precisely. In unsupervised learning we do not know samples labels or we purposely do not give them to learning algorithm. The goal in this case is to find a hidden pattern in the data.

In this thesis I try to solve problem of classification. It is a typical task in

machine learning. The data comes from two or more groups called classes and algorithm should find model which divides this data into correct classes. Usually, we know the target classes so we use the supervised learning.

Generally, the data are divided into two sets. First, called training is used to learn an algorithm and to find pattern in data. Second set is called test set and is used for accuracy evaluation. Training and test data should not be the same. If it is same it can lead to an optimistically biased evaluation of the classifier. A popular error estimate is cross-validation, which gives better error estimates than resubstitution or hold-out methods. This method divides data into k parts. After that, algorithm is used in k iteration. In each iteration, all parts except the k-th one are used for training and the kth part is used for testing, as you can see on Figure 4.

Fig. 4 5-fold crossvalidation

Usually input to the algorithm is not the data itself but values that describe the data. They form so called feature vector. It is array of measurable properties of original objects to be classified. Each object is described by one feature vector. Features can be categorical, ordinal, binary but in our case they are real-valued.

A model which is capable to classify the samples into classes is called classifier. The algorithm, which creates the model based on the data is called training algorithm. There are many types of classifiers and training methods.

Simple example are linear classifiers. These algorithms tries to divide the classes by a linear decision boundary. Examples are linear Bayes classifier or Perceptron. They are typically based on a function that gives score for each sample. Score is defined for each class. It is computed as dot product of feature vector of sample and the vector of weights corresponding to class. Sample is assigned to class with

highest score. In Bayes algorithms, such score is aposterior probability. Example of linear classifier is shown in Figure 5. Linear classifiers are often very fast but not as successfully as more complex algorithm. On the other hand, they typically do not suffer by overfitting.

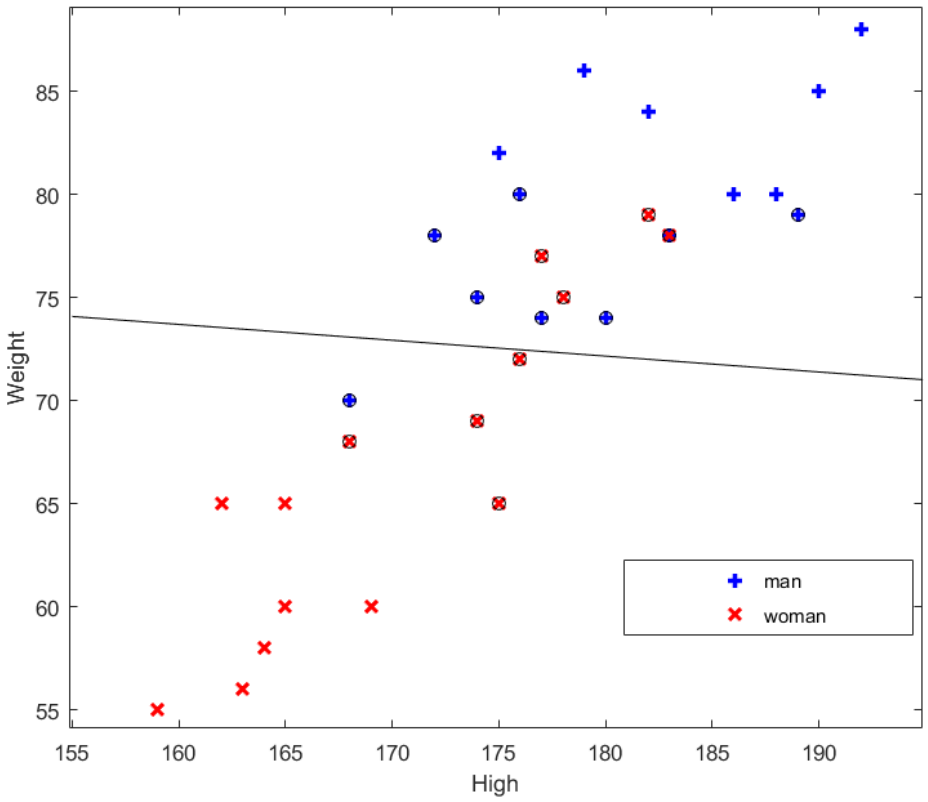


Fig. 5 Example of linear classifier

Another type of classifiers are decision trees. Algorithm is a tree in the meaning of graph theory. Usually, there is a threshold value assigned to each node. In binary trees, algorithm goes through the tree and continues in one branch if the value of a feature is greater than the threshold and in second branch otherwise. Each leave corresponds to a class the input sample is assigned to. The nodes may not be defined by number. They can correspond to a question with answer yes or no or another. Moreover there can be more branches then two from each node if the tree is not binary. Simple decision tree is depicted in the Figure 6.

Next type of classifier is algorithm called k-nearest neighbor. For each sample to be classified, the algorithm finds k-nearest training samples, where k is a parameter defined by the user. Sample is assigned to the same class as majority of its k neighbors.

Very popular algorithm in last few years are artificial neural networks. It is

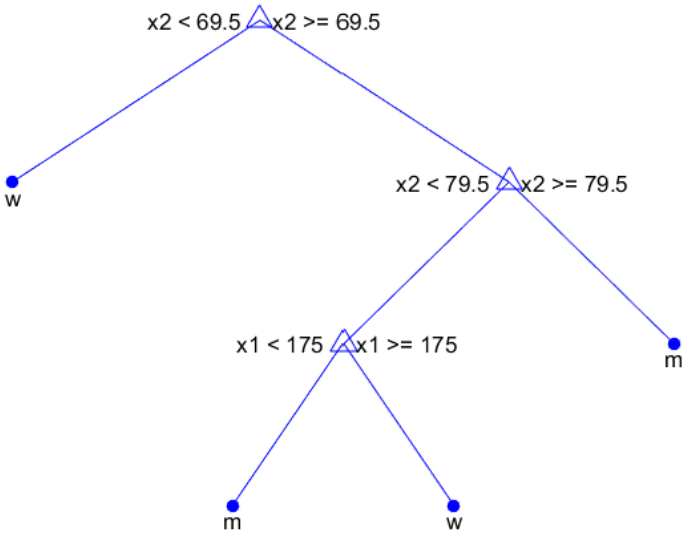


Fig. 6 Example of tree

a family of algorithms inspired by neurons in the brain. Typically, there are layers with units which are connected between themselves. On the input to each unit is weighted output from units from previous layer. Each unit has its own activation function which determines the output. Simple artificial network is depicted in Figure 7. Artificial neural networks are widely used not only for classification.

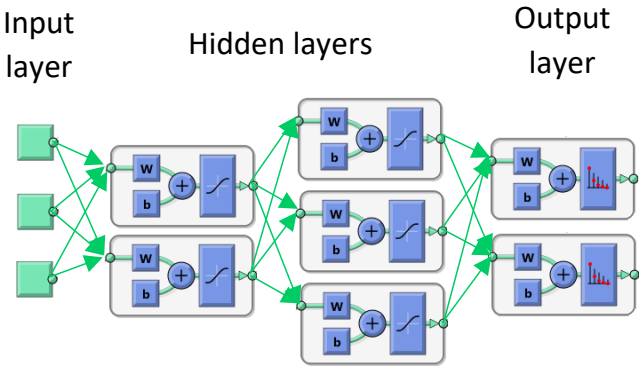


Fig. 7 Example of multilayer perceptron

Often, it is not possible to divide all samples successfully by one algorithm. For that reason, there are popular boosting procedures. Boosting method converts more weak classifiers to a strong one. There are many techniques how to do this but usually the output of boosting classifier is a weighted sum of response of weak

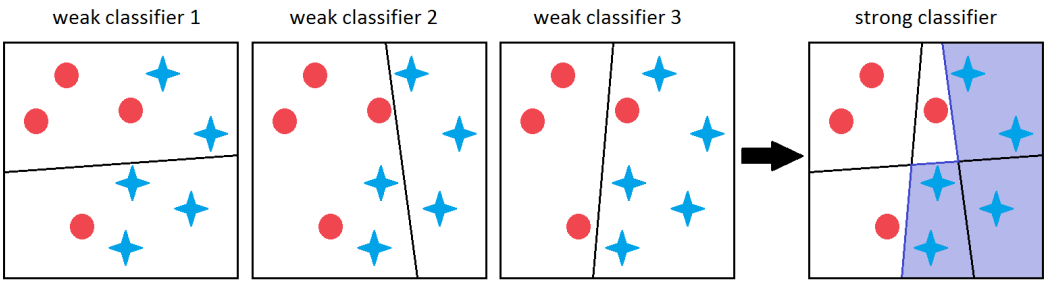


Fig. 8 Example of boosting method

classifiers. Example of boosting algorithm with linear weak classifier is shown in Figure 8.

2.1 Data

Electroencephalography (EEG) is highly used method for examine electrical activity of the brain. It is typically noninvasive. Action potential of neurons is recorded by electrodes from surface of the head. A disadvantage of this method is that summed signal from many neurons in brain are recorded. To obtain more precise information, it would be best to record the signals from each neuron separately. It is impossible because there are tens of billions of neurons in the brain. Moreover, the electrodes would need to be extremely small. In practice, a more precise intracranial EEG (iEEG) is used. Electrodes or stripes of electrodes are placed directly on the brain. This can be done mostly during a surgery operation. Consequently, there are only limit amounts of data.

The data which we used in this thesis comes from International Epilepsy Electrophysiology Portal [23]. Annotated intracranial EEG data is freely available on this website. The portal is developed by the University of Pennsylvania and the Mayo Clinic. All the recordings originate from Mayo Clinic in USA.

A special Matlab toolbox is available on the IEEG portal, which can be used for basic operations with signals. This toolbox was used here for creation of data sets for further processing. In such data sets there are two types of segments. Interictal segments are segments in normal iEEG signal. Neither three hours before nor three hours after them there wasn't seizure to avoid contamination by seizure activity. One hour long parts were took and divided to six ten-minutes long segments. Between parts there are 1.5 hours long spaces. Second types of segments are preictal segment. These segments starts one hour before seizures. Precisely first segment start one hour and five minutes before seizure and last end five minutes before seizure. The five minutes gap is there for two reason. Firstly, some epileptic activity before the beginning of the seizure can be missed

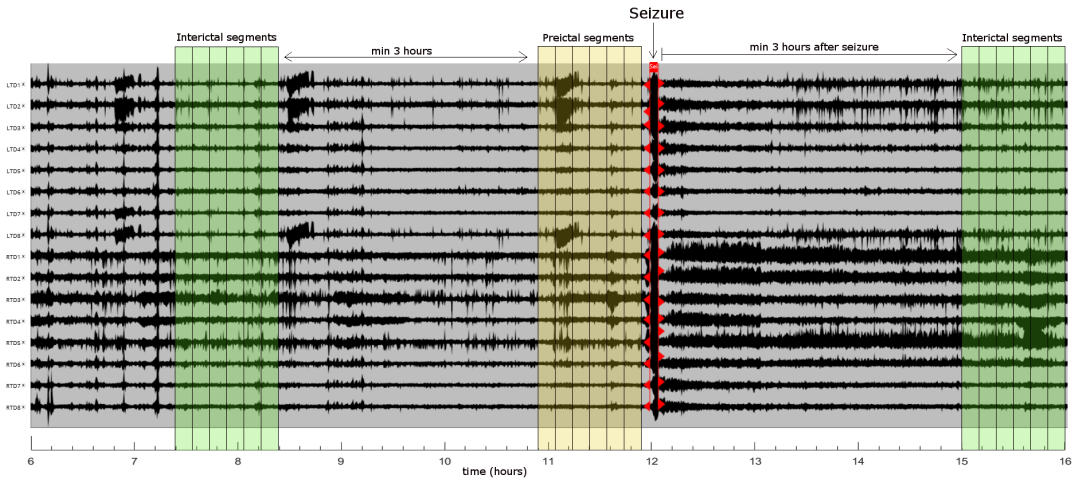


Fig. 9 Data segmentation.

by epileptologist (annotator) and can affect the prediction. Secondly, patient needs some time after seizure warning to take medication. As in the previous parts three hours before them was no seizures. Data segmentation is shown in Figure 9.

Each segment is saved as a Matlab structure containing channels, sampling rate and signal data. Data from different patients can have different number of channels and different sampling rate. Eight data sets from the portal were chosen. Six of them were used for training and optimization and the remaining two were used for testing of the proposed approach. Information about data is summarized in Table 1.

Tab. 1 Data summary

Data set	Study	Gender	Age	Length [dd:hh]	Number of seizures	Number of negative segments	Number of positive segments	Number of channels
1	012-2	Male	37	13:16	28	624	72	84
2	014	Female	33	06:00	57	102	48	104
3	019	Male	33	05:16	36	144	36	96
4	021	Male	16	06:11	13	270	36	108
5	024	Female	23	08:10	19	312	84	88
6	033	Male	3	06:17	17	246	36	128
7	017	Male	39	07:17	9	336	36	16
8	037	Female	62	08:23	9	426	30	80

2.1.1 Analysis of missing data

The data contain also segments with missing values. Most often it is caused by a poor contact between electrode and brain. In that case zero or "not a number" values are presented in the signal. There are two options. Missing values are completely random or there are some connection between occurrence of missing values and classification of segment. Therefor, the amount of missing values in positive and negative class was compared to accept or reject such dependency. First, histograms with probabilities of percentage of missing data were inspected. There are some parts where all data are missing. More often interictal segments are empty because there are more interictal segments then preictal. Segment with no data were excluded. Histograms are shown in Figure 10.

Mean value of distributions seems to be equally for both of segment types. Data from patient 1 and patient 4 seem to have some dependency between class and percentage of missing data. Assuming that different mean value corresponds to dependency, we can use t-test to compare means of two distribution. However, since an important assumption about equality of variances is not guaranteed, we used Welch's t-test, which does not assume equality of variances. Null hypothesis is that both distributions have approximately the same means. The level of significance 5% was used. In Table 2 one can see p-values for each patient.

Tab. 2 p-values for datasets

Patient	p
1	0.16
2	0.11
3	0.10
4	0.03
5	0.91
6	0.65

It can be observed that we can reject the null hypothesis only for patient 4 and only at the level of significance 5%. For other data we can not accept nor reject dependency between percentage of missing data and class.

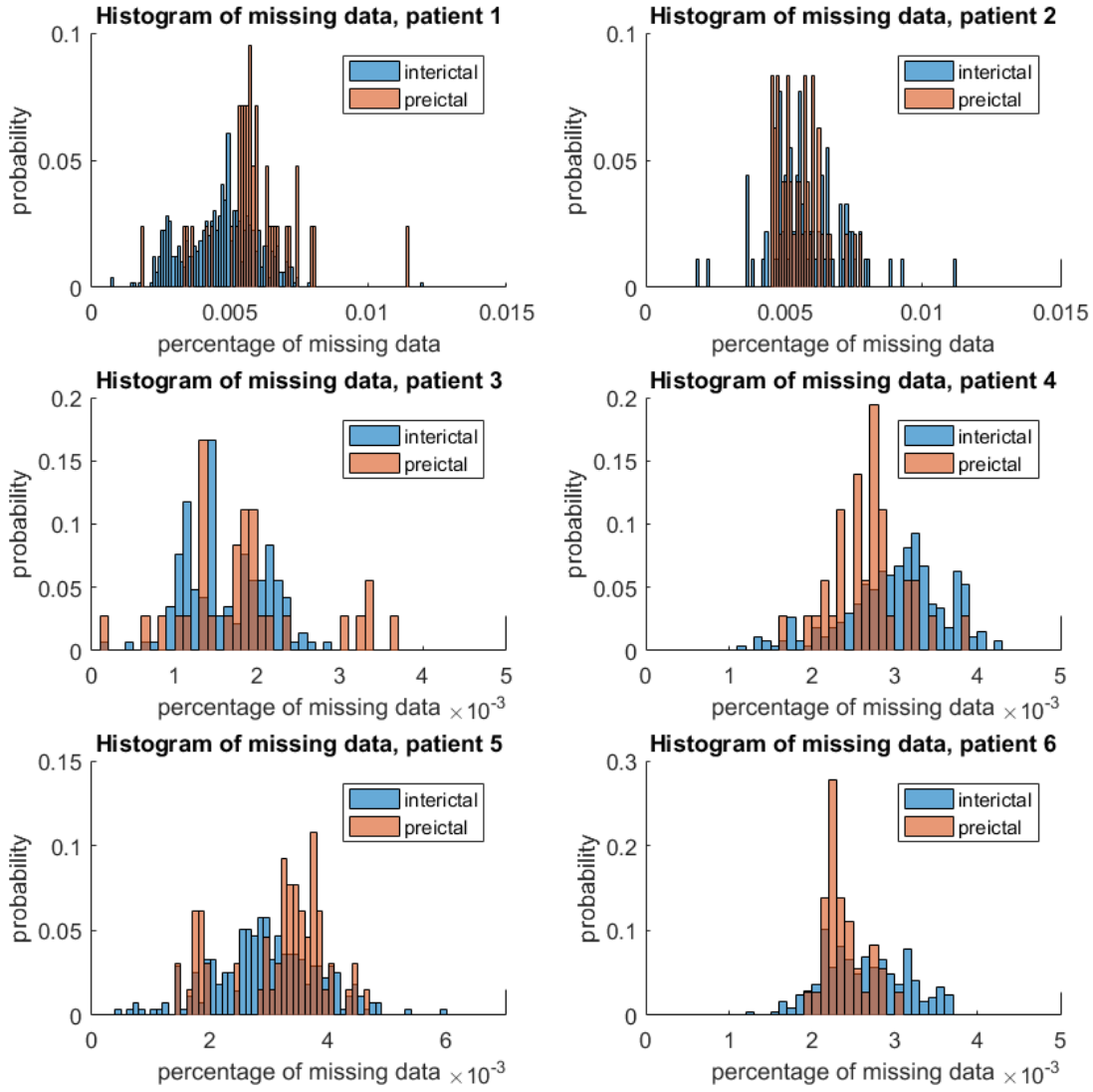


Fig. 10 Histograms of missing data

3 Methods

All proposed methods were implemented in Matlab [24]. For some calculation was used Matlab Pattern Recognition Toolbox for representation and generalization (PRTools) [25]. Figure 11 shows the entire prediction process.

First, we tried to find the best settings for data preprocessing. We used the sums of power spectral density in bands as features. Combinations of outlier removal procedure and class balancing procedure were evaluated. K-nearest neighbor classifier was used for the evaluation. The number of nearest neighbors was set to 40. The best combination of methods was chosen and different ways of handling of the missing data were tested.

Next, we tried to find the best feature set by the greedy algorithm and by the evolutionary algorithm. In both approach, the data preprocessing settings found before were used.

Finally, different classifiers with various parameter settings were tested. A script which can use the best data preprocessing method, extracts features and trains classifier (which was the best classifier) for further seizure prediction.

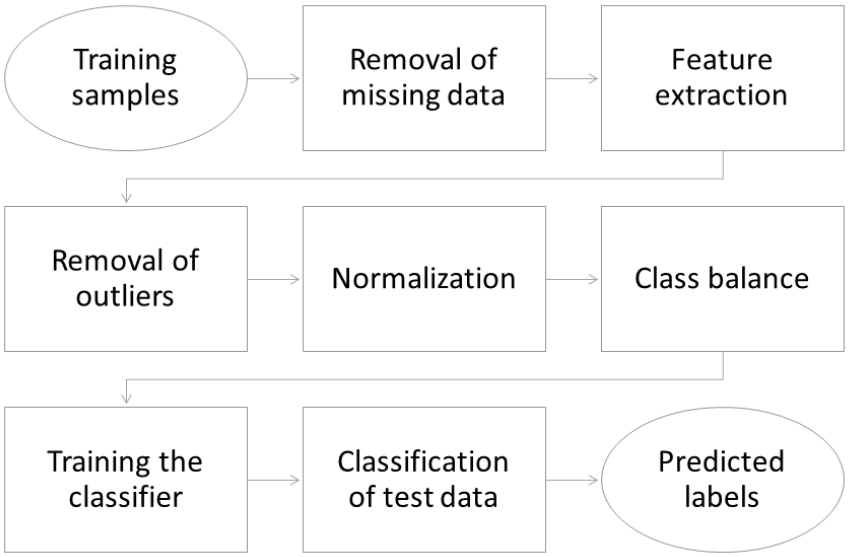


Fig. 11 The prediction process schema

3.1 Signal preprocessing

There is a lot of missing segments in the data. The values are sometimes absent in some channels but sometimes they are absent in all channels. The following solutions for this issue were proposed:

1. we replaced missing segments by zero.
2. we tried to remove all of the samples which have less than fifty percent of data from the training set
3. we tried the same method but this time the samples with more than ten percent of zero data were removed
4. we tried quite more sophisticated approach. At the beginning the training samples with less than fifty percent of data were removed. Then we found the most similar channel for each channel based on the cross-correlation. At the end the missing parts were replaced by corresponding parts from the most similar channel. In case the replacing part was empty we used the average value of the original channel.
5. the time segments in which the data were absent in some channel were removed.

3.2 Feature extraction

Several of features were tested. This section summarizes the features and provides a short description if needed. Most of features are inspired by the previous research in the area of prediction of epileptic seizure [26]. Features were extracted from the entire ten-minute segments.

Mean spectral power in six band

The spectral power is calculated from data and divided into six bands between limits - 0, 4, 8, 12, 30, 70, 180. The average of values from each band is added to the feature vector. It is done for each channel.

Covariance matrix characteristic

The covariance matrix is calculated from the spectrum. The feature vector consists of the average value, standard deviation and the three biggest values of the covariance matrix.

Fractal dimension

The feature vector is composed of three estimates of fractal index for each channel. The two first estimates are based on the second order discrete derivative, the

second one is wavelet based. The third estimate is based on the linear regression of the variance of detail versus level in logarithmic scale [27].

Sorted eigenvalues of correlation matrix of spectrum data

Correlation matrix between channel was calculated from spectrum. Infinite and NaN values were replaced by zero. Eigenvalues were calculated from correlation matrix and sorted by their values.

Upper triangle from correlation matrix of spectrum data

Spectrum was calculated from data and the amplitude of the lowest frequency was replaced by zero. Correlation matrix between channel was calculated. Feature vector consist of values from upper triangular part.

Frequency with maximal amplitude

Spectrum was calculated from data. Amplitudes for frequency under 0.1 Hz were set to zero. The frequency with the biggest amplitude was chosen from other values. It was done for each channel.

Higuchi fractal dimension

Features were calculated by Higuchi's technique for computing fractal dimension [28]. I used implemented function in MATLAB to computing Higuchi fractal dimension [29].

Hjorth parameters

Three indicators of statistical properties which is known as Hjorth parameters [30] were used. They are defined in time domain and they called Activity, Mobility, and Complexity.

Hurst exponent

Hurst exponent is used to measure influence of distance between two points on their statistical dependency. It relates to autocorrelation function of time series and the change of this function when the distance between points are changed. Matlab implementation of Hurst exponent was used to compute Hurst exponent [31].

Kurtosis

Kurtosis in each channel was calculated. It describes "tailedness" of the probability distribution. Feature vector consists of values from all channels.

Spectral magnitude divided into 18 bands

First, logarithm of spectral power is calculated. After that it is divided into 18 bands by its value.

Mean of spectral power with logarithmic scaling for bands up to 48 Hz

Spectral power is divided into ten frequency bands determined by logarithm - limits are (0.5, 2.25, 4, 5.5, 7, 9.5, 12, 21, 30, 39, 48). Then the average of values in each band is added to feature vector.

Median of lower 20 percent of spectrum divided into 24 bands

Lower twenty percent of frequencies is equally divided into twenty-four frequency bands. Feature vector consists of medians of values in bands.

Mean, standard deviation and maximal value in frequency and time domain in each channel and in their average

Mean, standard deviation and maximal value were calculated from data in each channel separately. The same procedure was done for channel which was calculated as average of all channels. And the same procedure was done once more for spectrum.

Percentage of missing data per channel

Percentage of missing values in each channel was calculated. These values were used as feature vector.

Shannon's entropy at dyadic frequency bands

Shannon's entropy is calculated with summed probabilities per bands. Limits of bands are calculated by interval bisection - the last band is from the half of maximal frequency to maximal frequency, the penultimate band is from the quarter of maximal frequency to the half of it, etc.

Skewness

Skewness of the data was calculated. It measured asymmetry of the probability distribution about its mean.

Sorted eigenvalues of correlation matrix of spectrum data summed into dyadic bands

First, data are divided into dyadic bands (limits are found by bisection). Then bands are replaced by sum of data which they contain. The correlation matrix of these values is created and its sorted eigenvalues form a feature vector.

Spectral edge frequency 50 up to 40 Hz

Spectral edge frequency 50 was calculated. It means frequency below which 50 percent of total power are located. Only the values below 40 Hz were used for cumulative sum.

Spectral edge frequency 75

Frequency below which 75 percent of total power are located was found in each channel. These values were used as features.

Spectral edge frequency 90

Same as in the previous paragraph but 90 percent was used instead of 75.

Eigenvalues of inter channel correlation of spectrum entropy

Entropy is calculated in each channel and in each of six frequency bands (limits - 0.1, 4, 8, 12, 30, 70, 180). Correlation inter channels is calculated. Feature vector is equal to sorted eigenvalues of mentioned correlation matrix.

Eigenvalues of inter channel correlation of spectrum probabilities

Probability is calculated in each channel and in each of six frequency bands (limits - 0.1, 4, 8, 12, 30, 70, 180). Correlation inter channels is calculated. Feature vector is equal to sorted eigenvalues of mentioned correlation matrix.

Eigenvalues of inter dyadic bands correlation of spectrum probabilities

Probability is calculated in each channel and each of frequency bands (dyadic). Correlation inter bands is calculated. Feature vector is equal to sorted eigenvalues of mentioned correlation matrix.

Mean spectral power in six channels

Probability is calculated in each channel and in each of six frequency bands (limits - 0.1, 4, 8, 12, 30, 70, 180). Correlation inter bands is calculated. Feature vector is equal to sorted eigenvalues of mentioned correlation matrix.

Entropy of frequency bands

Entropy is calculated with summed probabilities per bands. Limits of bands are 0.1, 4, 8, 12, 30, 70, 180.

Standard deviations of frequency amplitude in bands and channels

Standard deviation is calculated in each of six bands (0.1, 4, 8, 12, 30, 70, 180) in each of channels.

Sums of power spectral density in bands

Power spectral density was estimated by Welch's method. Amplitudes were summed in bands in each channel separately. Bands limits are powers of two (1, 4, 16, 32, 64, 128, 200).

Eigenvalues of correlation matrix inter channels

Spectrum was calculated from data. Amplitude for frequency 0 Hz was replaced by zero. After that probability of each frequency were calculated. Data were divided into bands and in each band were summed. Limits of bands are 0.1, 4, 8, 12, 30, 70, 180. It was done for each channel separately. Correlation matrix between channels was calculated. Feature vector consist of sorted eigenvalues of this matrix.

Correlation matrix of data

Correlation matrix between channel was calculated in time domain data. Whole matrix was used as feature vector.

Upper triangle matrix of correlation matrix

Same procedure as in previous paragraph was done. Only values from upper triangular part was used as feature vector.

Variance of values in channels

Variances of all segment in each channel were calculated and used as features.

3.2.1 Features processing

This section describes the preprocessing of data that was made prior to classifier training.

Removal of outliers

The outliers could negatively affect normalization so it is appropriate to remove them. There are many methods how the outliers can be detected (and subsequently removed). We tried two of them. In the first technique, the outliers are features which contain many values which are further from the median more than three times of the standard deviation.The second approach uses mean instead of the median.

Normalization

Some classifiers are sensitive to relative ranges of feature values. Features with bigger range can have greater impact on classification decision. Therefore, it is beneficial to scale the data. In this thesis, the features were shifted to their mean and divided by the variance. Finally, the scaled features have zero mean and unit variance. First we computed the scale mapping from the training data. Such mapping was used to normalize the training and testing datasets.

Class balance

As expected, the records of the positive class are less common than the negative class records. Many of classifiers minimize error during their learning and can prefer the negative class. To solve this issue, some methods can be used to balance samples of both classes. We tried five methods:

1. First we used the easiest way. The representatives from bigger class are randomly chosen and removed so both classes contain approximately the same number of the samples as smaller class.
2. Second we tested another method of under-sampling. This method uses distances between samples from different classes for finding the values to remove. For each sample from the majority class three furthermost samples from minority class are found. The samples whose mean of this distances is smaller are removed.
3. In the third approach we tried to make the minority group bigger by copying their samples with Gaussian noise added.
4. Next way is oversampling too. New samples are generated as the nearest neighbors of minority class samples.
5. At last, Synthetic Minority Over-sampling Technique (SMOTE) was used [32].

3.3 Feature selection

We used two approaches of the feature selection. First, we tried greedy algorithm which selects the best feature vector in each run and adds it to the feature set. Then we tried simple evolutionary algorithm.

3.3.1 Greedy algorithm

In this algorithm also called Sequential Forward Selection (see Algorithm 1), maximal number of feature vectors which will be selected is set. After that all of the feature vectors are evaluated and one with the best result is added to final feature vector. In the next round the combinations of the best vector from the first round and each of other vectors are evaluated. This process is repeated until the preset number of features is selected.

Algorithm 1 Greedy algorithm

```
1: procedure GreedySelection(maxNum, features)
2:     bestFeatures ← ∅
3:     for i < maxNum + 1 do
4:         AUC ← ∅
5:         j ← 0
6:         for all feature in features do
7:             AUC_j ← GetAUCforCombination(feature, bestFeatures)
8:             j++
9:         end for
10:        I ← GetMaxIndex(AUC)
11:        bestFeatures_i ← feature_I
12:    end for
13:    return bestFeatures
14: end procedure
```

3.3.2 Evolutionary algorithm

Evolutionary algorithms (EA) is set of techniques for optimization inspired by natural evolution. There have to be defined some representation of problem called individual. After that set of individuals are generated. These individuals form a population. A generation can be absolutely random or can have some restrictions. Value called fitness function is computed for each individuals in the population. Target of EA is to maximize the fitness function. Next, individuals are repeatedly chosen from population and cross-overed (combined) between themselves while new population is formed. Chosen individuals for cross-overing are called parents and their combinations are called children. A child can be randomly modified after cross-over during process called mutation (see Algorithm 2). There are different approaches for selection of parents, their crossing, mutation, population filling and some other details.

The combination of feature vectors is optimized in our implementation of EA. Features which will be used for classification is represented as binary string. The

Algorithm 2 Evolutionary algorithm

```
1: procedure EvolutionarySelection(features)
2:     X ← InitializePopulation()
3:     while not TerminationCondition() do
4:         X_new ← ∅
5:         while not CreatedNewPopulation() do
6:             Parents ← SelectParents(X, f)
7:             Children ← Crossover(Parents)
8:             Children ← Mutate(Children)
9:             X_new ← Combine(X_new, Children)
10:        end while
11:        X ← Join(X, X_new)
12:    end while
13:    return GetBestIndividual(X)
14: end procedure
```

value one or zero at some position means that feature corresponding to this position is or is not used for classification, respectively. As fitness function is used value discussed in validation methodology (cross-validated AUC). Each bit in the string is generated absolutely randomly. Children are produced by one-point crossover of binary string. Each bit is flipped with a probability during mutation. Best individual from old population is propagated to new population and all of other individuals are replaced by children.

3.4 Classification

We used KNN classifier during the feature selection. After that we tried the different classifiers. There is a list of them with their parameters. All classifiers which we used are implemented in Statistics and Machine Learning Toolbox in MATLAB [24].

3.4.1 Decision trees

Decision trees can be used for classification or for regression. We used decision trees for classification. Advantages of decision trees are that they are simple to understand and interpret because they make decision similarly as human do. They are independent on data normalization. Data can have different ranges. Moreover variables do not have to be numerical. Decision trees are fast even on big amount of data.

On the other hand, learning of decision tree is NP-complete task and thus learning algorithm do not guarantee globally-optimal tree. Moreover they are

very dependent on training data. Small change in the training data can make big change in the tree. Another problem is overfitting. If inappropriate number of nodes are set, a very complex tree can be formed which does not classify well an unseen data.

We tried three classification trees with different settings of parameters. Trees varied in maximum number of splits. It means maximal number of nodes which graph contains. Set values are listed in Table 3. Learning algorithm works from the root to leaves. At each step it finds a variable, which best splits classes. For all trees, Gini's diversity index was used as a split criterion.

Tab. 3 Parameters of decision trees

Parameter	Simple tree	Medium tree	Complex tree
Maximum number of splits	4	40	100

3.4.2 Discriminant analysis

This classifier tries to find a linear combination of features to characterize the classes. There are two types of discriminant analysis classifiers. Linear Discriminant Analysis (LDA) which assumes equality of class covariances. In Quadratic Discriminant Analysis (QDA), there is no assumption that the covariances of classes are identical. We tried both types of this classifier.

3.4.3 Support vector machine

A Support Vector Machine (SVM) classifier represents samples as points in the space. Target of learning SVM is find empty gap in space as big as possible which splits points correctly. First SVM was proposed as a linear classifier but a method to make them nonlinear was suggested later. The samples are projected into space with higher dimension. SVM compute kernel function. There is a risk of overfitting, when too high dimension is used for projection. We tried six SVM classifiers. They vary mostly in the kernel function and are listed in Table 4.

3.4.4 Nearest neighbor classifiers

As previously written, Nearest neighbor classifier finds k nearest samples in training data. Meaning of nearest can vary with chosen distance metric. Moreover, the number of nearest neighbors affects the result of classification. Classifiers with their parameters are listed in Table 5.

Tab. 4 Parameters of SVM

Parameter	Linear SVM	Quadratic SVM	Cubic SVM	Fine Gaussian SVM	Medium Gaussian SVM	Coarse Gaussian SVM
Kernel function	Linear	Quadratic	Cubic	Gaussian	Gaussian	Gaussian
Kernel scale	Auto	Auto	Auto	5.6	22	89

Tab. 5 Parameters of KNN

Parameter	Fine KNN	Medium KNN	Coarse KNN	Cosine KNN	Cubic KNN	Weighted KNN
Number of neighbors	3	40	100	10	10	10
Distance metric	Euclidean	Euclidean	Euclidean	Cosine	Minkowski	Euclidean
Distance weight	Equal	Equal	Equal	Equal	Equal	Squared inverse

3.4.5 Ensemble classifiers

Sometimes, only one classifier is not accurate enough. Ensemble classifiers are strong classifiers made from more weak classifiers. There are three main types of them.

1. Bagging methods learn each weak classifier on random chosen subset of the training data. The result of classification is a combination of results from each weak classifier.
2. Subspace methods are similar to bagged methods but only some features are chosen to train each weak classifier.
3. Boosting method are also similar to bagging methods. However,the subsets for training are not chosen randomly, but incorrectly classified samples are preferred.

Classifiers with parameter values are listed in Table 6.

Tab. 6 Parameters of Ensemble classifiers

Parameter	Boosted Trees	Bagged Trees	Subspace Discriminant	Subspace KNN	RUSBoosted Trees
Ensemble method	AdaBoost	Bag	Subspace	Subspace	RUSBoost
Learner type	Decision Tree	Decision Tree	Discriminant	Nearest neighbors	Decision Tree
Maximum number of splits	20	20	-	-	20
Number of learners	30	30	30	30	30
Learning rate	0.1	-	-	-	0.1
Subspace dimension	-	-	248	248	-

3.4.6 Multilayer perceptron

Multilayer Perceptron (MLP) consists of input layer, one or more hidden layers and one output layer. In input layer there is the same number of neurons as number of features. Number of neurons in hidden layers vary. Output layer consists of one neuron for each class.

It is necessary to find close to optimal weights for each neuron. It is done during learning phase. MLP is typically learned by technique called back-propagation, however there are many othe algorithms. We tried two of them.

Levenberg-Marquardt backpropagation (LM) is one of the fastest methods. This method combine advantages of two optimization methods: Gauss–Newton algorithm, which is very fast but not always finds a solution and gradient descent which always finds a local optimum but is not so fast. LM is quite slower than Gauss-Newton but always converges to local optimum. Disadvantages of LM method is that found local optimum is not necessarily the global one. Another limitation of this method is that mean or sum of squared errors have to be used as performance function.

Second used learning method called Scaled Conjugate Gradient Backpropagation (SCG). Common conjugate gradient adjusts the weights along conjugate direction instead of steepest descent direction as other methods like LM. SCG avoids the time-consuming line search which other conjugate gradient methods do.

Tab. 7 Parameters of MLP

Parameter	Fine MLP	Two hidden layer MLP	SCG MLP
Number of hidden layers	1	2	1
Number of neurons in layer	10	20, 20	30
Training function	LM	LM	SCG
Performance function	Mean squared error	Mean squared error	Cross entropy

Used MLP classifiers are listed in Table 7.

3.4.7 Self-organizing map

Self-organizing map (SOM) is another type of artificial neural network. It is typically two dimensional network of neurons. Each neuron is connected to its neighbors. Neighbors are defined by topology function and neighborhood size. Neighborhood size and distance between neurons are changed during learning process. Parameter values are listed in Table 8.

Usually, SOM is trained using unsupervised learning. Neurons made clusters with similarly properties during learning. Each neuron is assigned to one class according to majority class of its associated samples. New samples are classified according to the nearest neuron.

Tab. 8 Parameters of SOM

Parameter	Fine SOM	Medium SOM	Coarse SOM	Grid SOM	Manhattan SOM	Big neighborhood size SOM
Size	2, 2	7, 7	10, 10	7, 7	7, 7	7, 7
Initial neighborhood size	3	3	3	3	3	7
Layer topology function	Hexagonal	Hexagonal	Grid	Hexagonal	Hexagonal	Hexagonal
Neuron distance function	Link distance	Link distance	Link distance	Link distance	Manhattan distance	Link distance

4 Experiments

This chapter summarizes all the experiments performed on our data.

4.1 Validation methodology

Area under receiver operating characteristic (AUC) was used as main quality indicator. We calculated AUC in five-fold crossvalidation for each patient. This calculation was iterated ten times and all values were averaged. This average was used for final comparison.

It is important to divide data into test and training sets correctly because there are six samples from one hour. Data coming from the same hour can be very similar. An appearance of samples from one hour in both training and testing set would lead to an optimistically biased cross validation estimate. For this reason, we placed each whole sextuplet either into test or into training set.

4.2 Results

Feature preprocessing

First we found the best setting of feature preprocessing. Different combinations of outliers removing and class balancing were tested. Results are shown in Figure 12. Descriptions of combinations are in Table 9 with means and standard deviations. Three values are written in each cell. First number is ID of combination corresponding with Figure 12. Second number is mean of AUC and third value is standard deviation.

We can observe two phenomena in Figure 12. First, outliers removing method based on distance matrix seems to be better than the second method. Second, class balance method based on undersampling gets worse results than oversampling methods. It was expected because there is lack of data, which gets even worse when undersampling is done.

Combination with ID 5 and 7 perform similarly. We choose combination 7 for the remaining experiments, because it has bigger value of 25th and 75th percentile. In final algorithm, nearest neighbor method was used for balancing classes together with distance matrix based outlier removal.

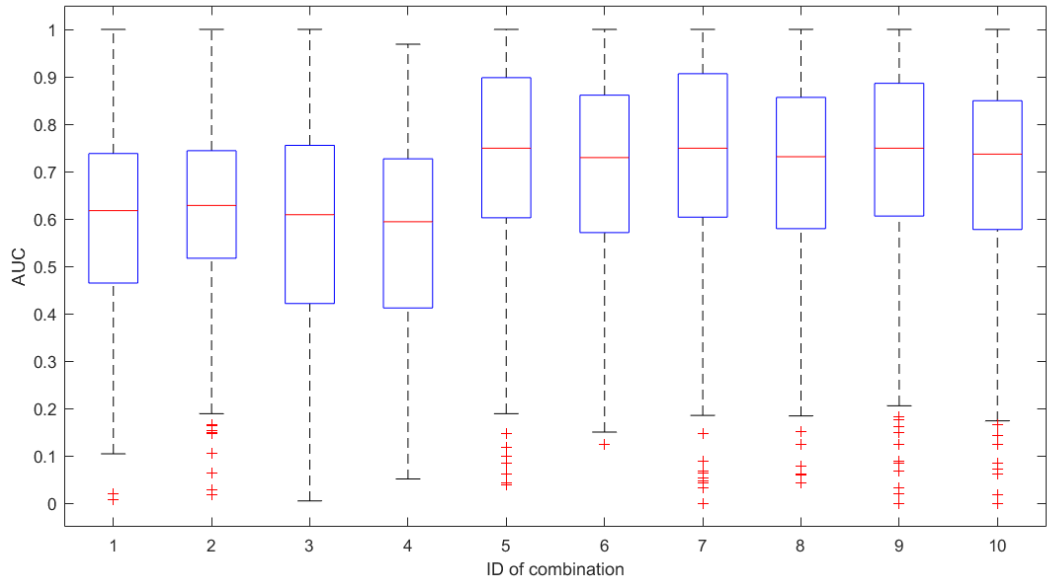


Fig. 12 AUC of different combination of preprocessing methods

Tab. 9 Combination ID and their means with standard deviations of AUC

	Distance matrix			Distance from median		
	ID	mean	std	ID	mean	std
Random undersampling	1	0.5992	0.1945	2	0.6178	0.1868
Near miss undersampling	3	0.5799	0.2175	4	0.5632	0.2149
Gauss noise oversampling	5	0.7127	0.2241	6	0.6919	0.2105
Generating NN	7	0.7103	0.2321	8	0.6907	0.2156
SMOTE	9	0.7090	0.2285	10	0.6916	0.2140

Missing values

Next we tried different approaches to handle missing values. The results are shown in Figure 13 and listed in Table 10.

We can see that proposed method does not work well. Third method does not work at all. It is because this method uses for training only samples with more than 90% measured data and there are not enough samples like that.

Only the last method gets better result than simple replacement of missing values by zero. But the difference is not satisfying. we choose replacing missing values by zero for this reason. Moreover analyze of missing data indicated that there sometimes can be dependency between class where sample belong and percentage of missing values. Moreover, when the algorithm should be used in practice the speed of methods is important. Replacing missing values by zero is definitely fastest from proposed methods.

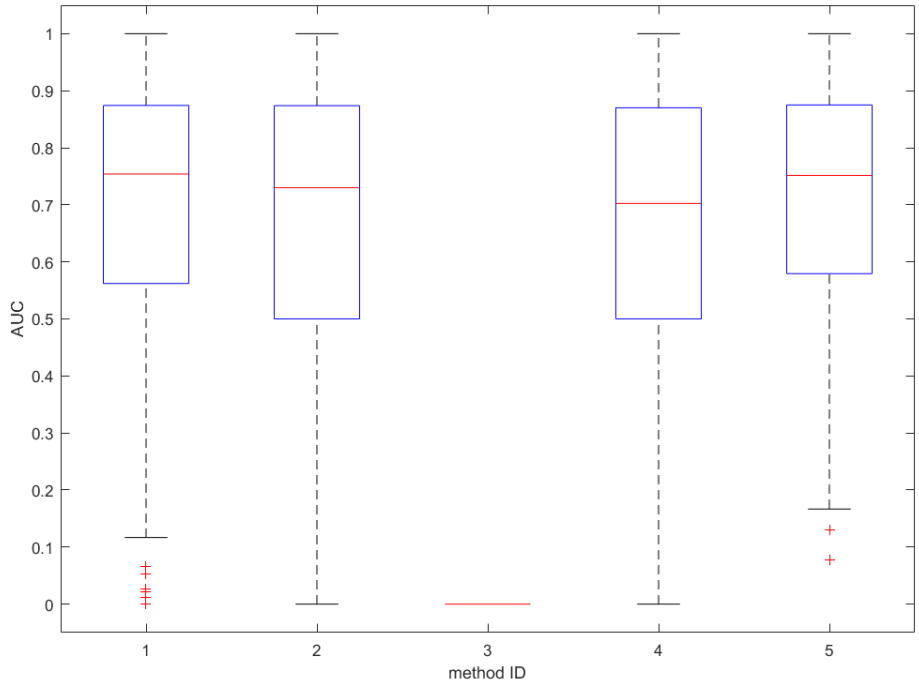


Fig. 13 AUC of different combination of missing data removing methods

Tab. 10 Missing value removing method ID and their means with standard deviations of AUC

Missing value removing method	ID	mean	std
Replaced by zero	1	0.6973	0.2358
More than 50%	2	0.6920	0.2145
More than 90%	3	0	0
Replaced by most similar	4	0.6976	0.2214
Delete missing parts	5	0.7114	0.2147

Feature selection

Greedy algorithm

Results of greedy algorithm for feature selection is shown in Figure 14 and listed in Table 11. AUC in table means AUC for combination of feature in the same row and all features above it. We can see that the best result is obtained for first eight features. Using more features is unnecessarily because we do not have enough data for learning and it leads to over-fitting.

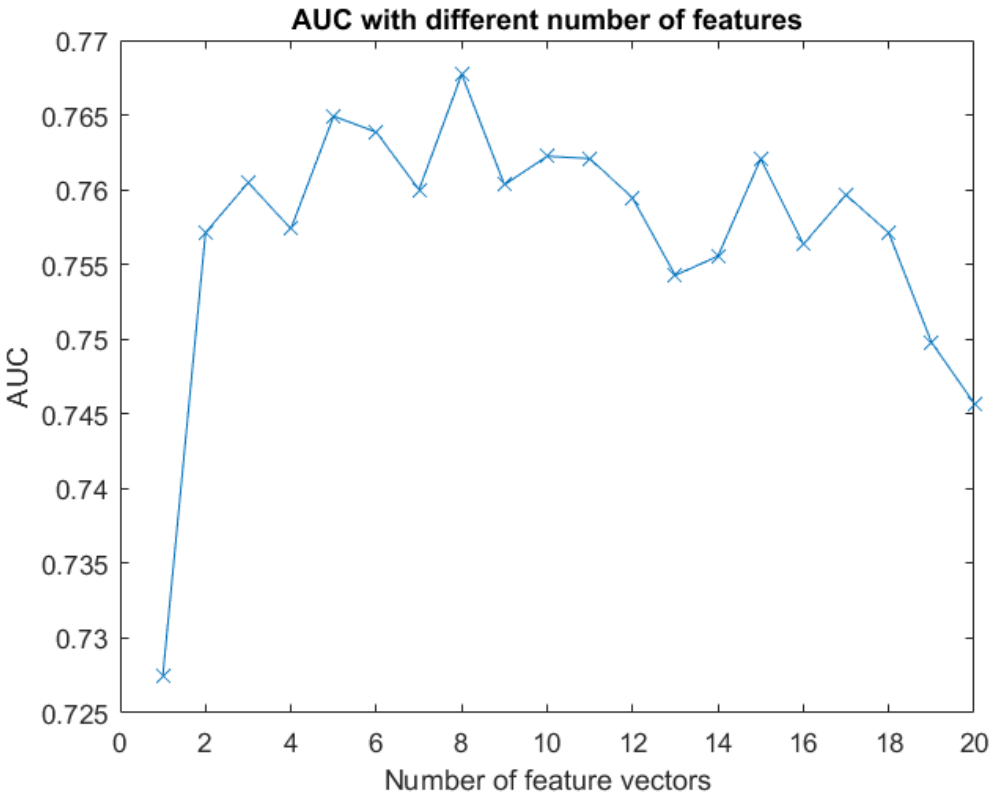


Fig. 14 AUC for greedy algorithm by different number of features

Tab. 11 AUC for greedy algorithm by different number of features

	Feature	AUC
1	mean amplitude in time domain	0.7275
2	hjorth mobility	0.7571
3	percentage of missing data on channel	0.7605
4	higuci fractal dimension	0.7574
5	spectral edge 50 under 40 Hz	0.7649
6	time series correlation eigenvalues	0.7639
7	maximal amplitude in frequency domain	0.7600
8	spectral eigenvalues per frequencies	0.7677
9	maximal amplitude in frequency domain in average channel	0.7603
10	minimal amplitude in time domain	0.7622
11	maximal amplitude in time domain in average channel	0.7621
12	spectral edge 90	0.7595
13	correlation eigenvalues in spectral domain	0.7543
14	kurtosis	0.7556
15	spectral entropy	0.7621
16	mean amplitude in frequency domain in average channel	0.7564
17	spectral edge 75	0.7597
18	standard deviation of amplitude in frequency domain in average channel	0.7571
19	correlation matrix triangle in time domain	0.7498
20	standard deviation in time domain	0.7457

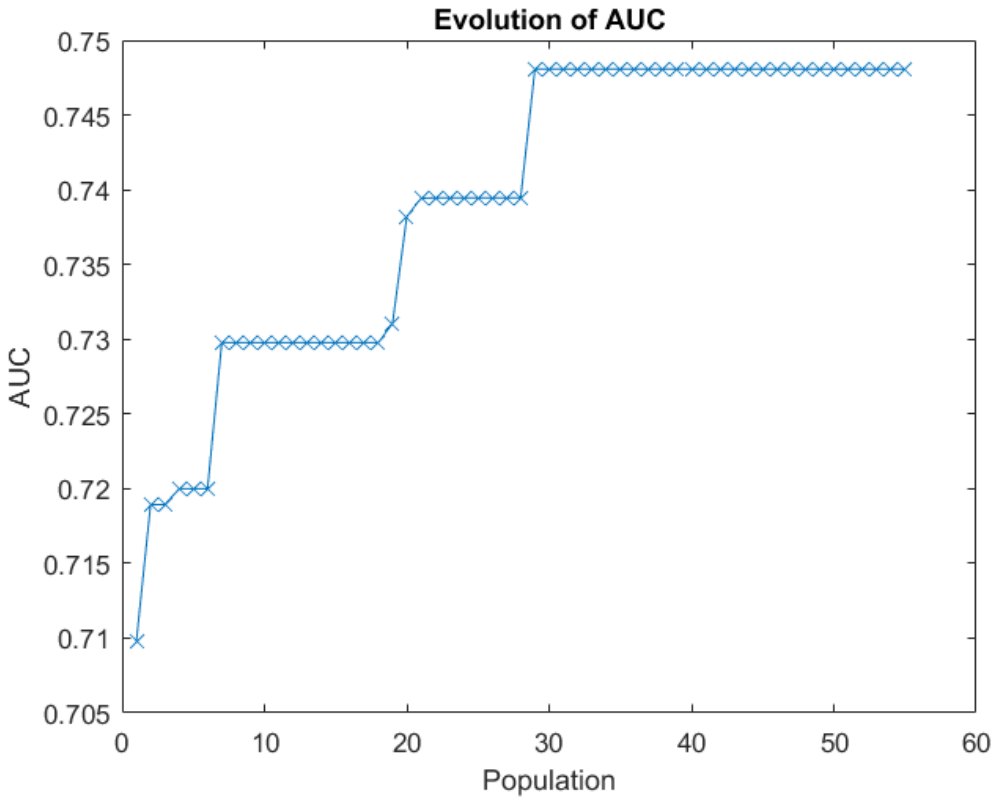


Fig. 15 AUC for evolutionary algorithm of different population

Evolutionary algorithm

We had great expectation from evolutionary algorithm (EOA) and assumed good results. Other works use EOA and their results are improved [33, 34, 35]. Unfortunately those expectations were not confirmed. Power of evolutionary algorithm is in large number of population. But calculating of fitness function as AUC from 5-fold cross-validation is too slow. We used population with 10 individuals but did not get better result than with greedy algorithm in a reasonable amount of time. Evolution of AUC is shown in Figure 15 and list of used features in last population is listed in Table 12.

It would be appropriate to propose faster fitness function in terms of a filter approach. For example, maximization of distance between samples of different classes ad minimization of distance between samples of same classes could work.

It could be beneficial to set some restriction about amount of initially selected features. Now each feature is used with probability 0.5. There are 48 possible selected features so about 24 features are selected in average. Optimal number of features is according to greedy algorithm about eight (considering our particular amount of available training data).

Unfortunately, there was no more space for research about improvement of

Tab. 12 List of features selected by evolutionary algorithm

1	characteristic of covariance matrix in frequency domain
2	correlation eigenvalues in frequency domain
3	hjorth activity
4	hjorth complexity
5	hjorth mobility
6	hurst exponent
7	kurtosis
8	mean of spectral power with logarithmic scaling for bands up to 48 Hz
9	maximal amplitude in time domain
10	maximal amplitude in time domain in average channel
11	mean amplitude in frequency domain in average channel
12	mean amplitude in time domain in average channel
13	mean amplitude in time domain
14	minimal amplitude in time domain
15	percentage of missing data on channel
16	spectral edge 50 under 40 Hz
17	spectral edge 75
18	spectral eigenvalues per frequencies in dyadic
19	spectral eigenvalues per frequencies
20	spectral entropy
21	standard deviation of amplitude in frequency domain
22	standard deviation of amplitude in time domain in average channel
23	standard deviation of amplitude in time domain
24	correlation matrix in time domain
25	correlation matrix triangle in time domain

evolutionary algorithm in this work because it is not main area of interest of this work. We used features found by greedy algorithm in the final algorithm due to the better results.

Classifier

We tried to classify data with 31 different classifiers. Results are shown in Figure 16 and classifiers are listed in Table 13.

Only two classifier did not work. It was Boosted and RUSBoosted Tress. All of other classifier had median over 0.5. This was probably because an incorrect usage of the available implementation.

We can see in results that often smaller classifiers obtain better results than more complex ones. For example decision tree with 4 splits gets better result than one with 40 and one with 100 splits. Similar situation can be seen with

Tab. 13 List of classifiers, their AUC and standard deviation

ID	Classifier	mean AUC	std AUC
1	Simple tree	0.6581	0.2338
2	Medium tree	0.6158	0.2172
3	Complex tree	0.6158	0.2172
4	Linear discriminant	0.6892	0.2136
5	Quadratic discriminant	0.6642	0.1814
6	Linear SVM	0.6865	0.2638
7	Quadratic SVM	0.6923	0.2471
8	Cubic SVM	0.6599	0.2740
9	Fine Gaussian SVM	0.7029	0.2345
10	Medium Gaussian SVM	0.7493	0.2260
11	Coarse Gaussian SVM	0.7339	0.2439
12	Fine KNN	0.6726	0.2037
13	Medium KNN	0.7413	0.2357
14	Coarse KNN	0.7397	0.2204
15	Cosine KNN	0.6877	0.2199
16	Cubic KNN	0.7168	0.2169
17	Weighted KNN	0.7213	0.2147
18	Boosted Trees	0.0978	0.2407
19	Bagged Trees	0.7556	0.2224
20	Subspace Discriminant	0.6660	0.2625
21	Subspace KNN	0.6831	0.2036
22	RUSBoosted Trees	0.0856	0.2111
23	Fine MLP	0.6530	0.2719
24	Two hidden layer MLP	0.6492	0.2707
25	SCG MLP	0.6275	0.2721
26	Fine SOM	0.6521	0.2094
27	Medium SOM	0.6773	0.2002
28	Coarse SOM	0.6389	0.2028
29	Grid SOM	0.6829	0.2032
30	Manhattan SOM	0.6776	0.2030
31	Big neighborhood size SOM	0.6795	0.2055

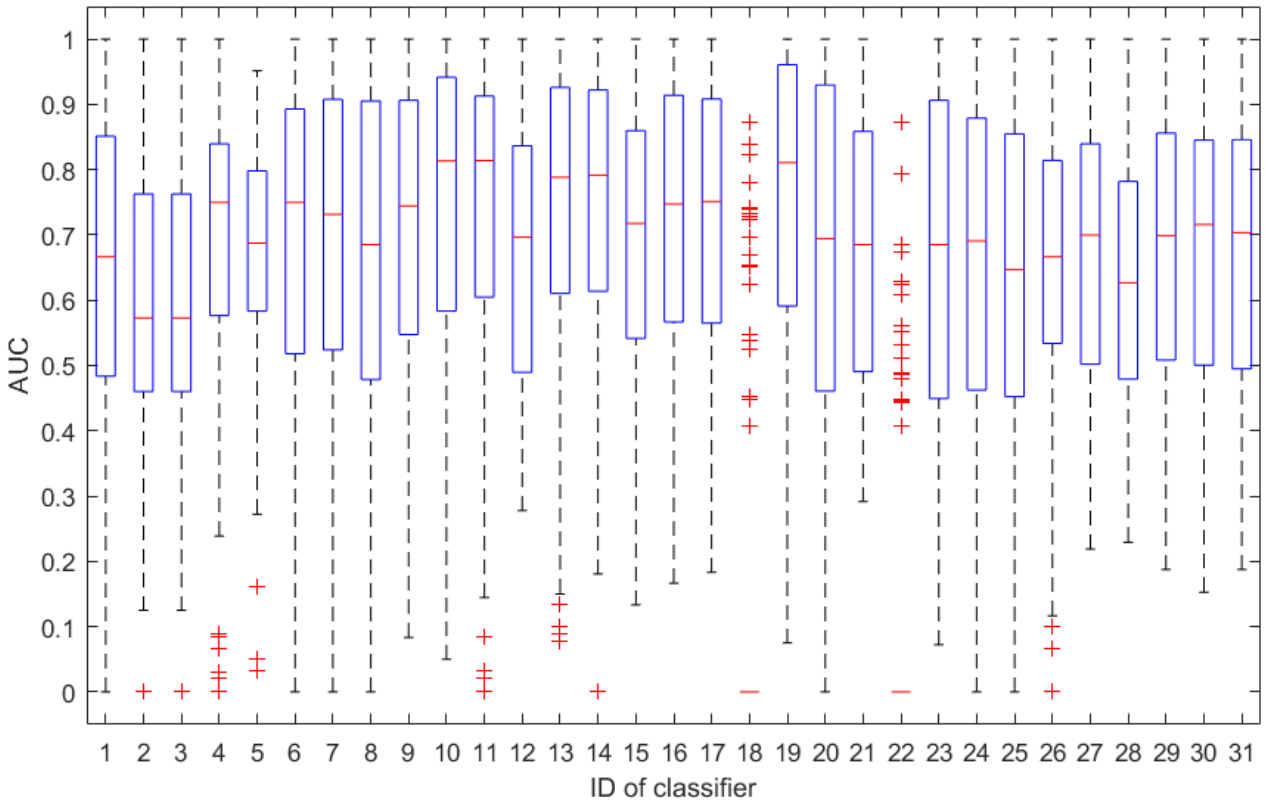


Fig. 16 AUC for different classifiers

multilayer perceptrons. The best results of MLP gets fine MLP with only one hidden layer which contains ten neurons. One the other hand the simplest 4NN classifier with four neighbors gets worse results than others KNN classifiers. KNN classifiers with 40 and 100 neighbors achieve similar results but had a bit more stable results with less number of extra poor results.

We choose Bagged trees classifier as the final algorithm because its results are the best from all results. It is quite interesting that classifier which is composed from trees are the best while decision trees themselves are not much good in this context. On the other hand weak decision classifiers are converted to the strong one which is exactly what this algorithm should do.

Results on independent data

Finally, we test the algorithm on two patients which we did not use during optimization. We used the same way of validation with cross-validation as during optimization to obtain more precise results. Results are shown in Figure 17 and listed in Table 14.

Apart of some outstanding runs, algorithm achieves AUC higher than 0.5 on both patients. Moreover averages are over 0.8 for both datasets. Median of results

Tab. 14 Results for independent data

Patient ID	mean AUC	std AUC
7	0.8282	0.1401
8	0.8527	0.1939

is worse for the first patient but on the other hand minimum of results are higher for the first patient. It means the algorithm works more reliable on data from first patient and results are more stable. But the algorithm also achieves very good results for quite big amount of iteration on data from the second patient although in some few cases results are quite bad for them. It depends on subjective opinion, on which data the results are better.

In Figure 18 we can see five examples of ROC for each one of test datasets. One can see an imperfect classification in Figure 18a but for four of five ROC curves we can set threshold for which all of preictal segments will be classified correctly. It is most important property for use in practice. When set threshold like true positive rate (TPR) will be one, false positive rate (FPR) will be under 0.2. It means that maximal 20% of interictal segments will be classified as preictal. It is quite a lot but there is some prediction power. Unfortunately not for all iteration we can set threshold optimal. As we can see on green ROC curve in Figure 18a. There is no possible way to set threshold which leads to classification of all preictal segments correctly and gets good FPR at the same time. Similar situation is in Figure 18b. There are two absolutely correct classification (second ROC is hide behind the green one). Moreover for another two figures, false positive rate is be about 0.1, while true positive rate is preserved equal to one. Unfortunately, even there, for some iterations, it is not possible to avoid wrong classification of preictal segments.

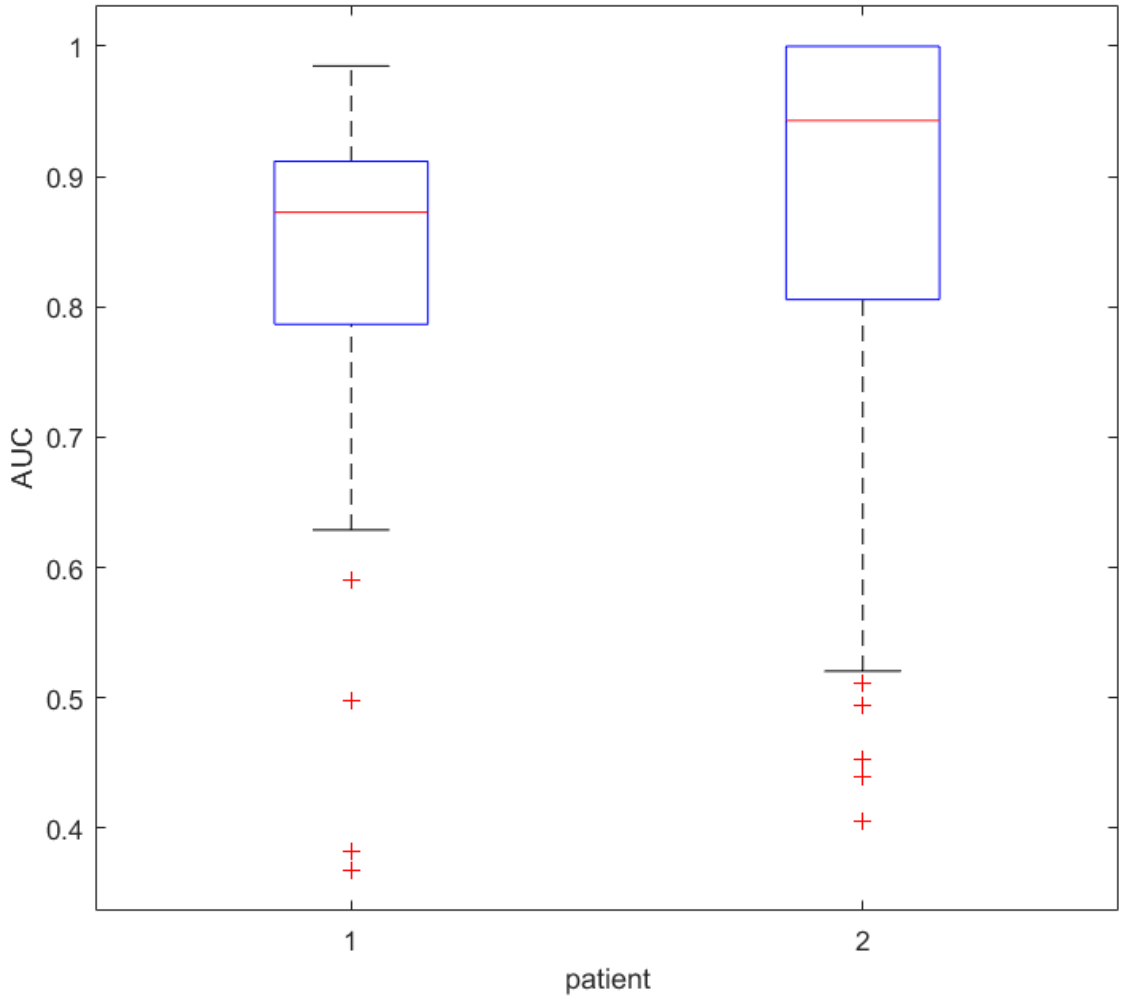
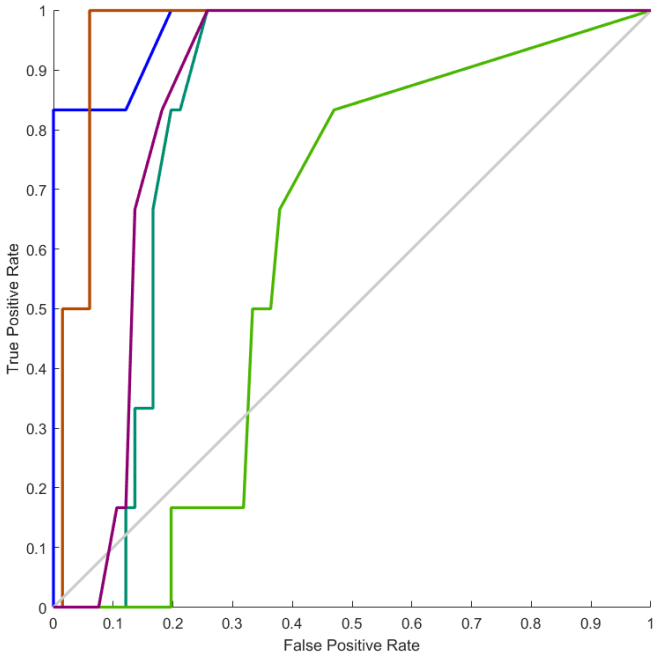
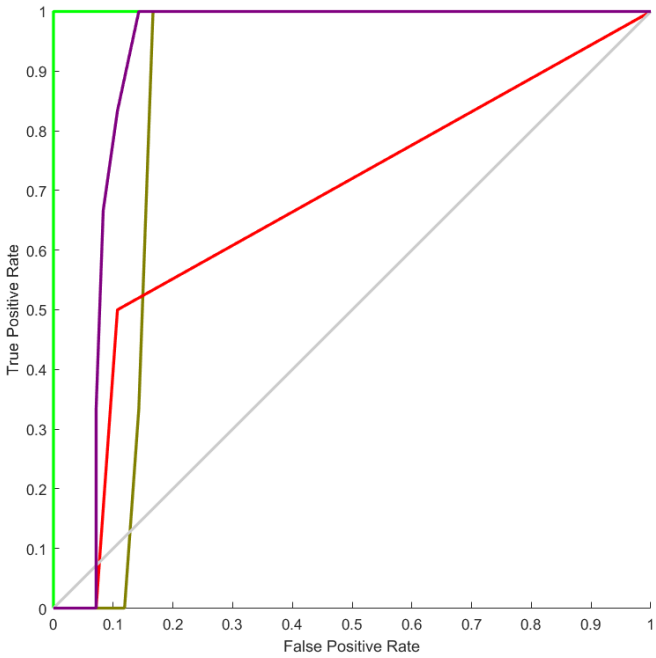


Fig. 17 Results for independent data



a) Patient 7



b) Patient 8

Fig. 18 Examples of ROC for independent data

5 Conclusions

All goals from assignment were accomplished. Missing values in given data were replaced by zero because it was fastest from all of the proposed methods and other proposed methods did not obtain better result.

Oversampling method was used to achieve balanced amount of samples from each class. Classes were ballanced by using nearest neighbors based method. This method lead to best results. Both proposed undersampling methods reduced the accuracy of classification because they probably reduced the number of samples too much.

Eight features were selected by greedy algorithm containing characteristics both from time and spectral domain. Evolutionary algorithm was too slow with used fitness function and did not provide sufficient results in reasonable time.

Support vector machine classifiers with Gaussian kernel function and K-nearest neighbors classifiers obtained better result than other proposed classifiers. Bagged trees classifier with 30 trees obtain best results from all proposed classifiers. Simple trees had lower accuracy.

Proposed procedure achieved average value 0.8405 of area under the ROC curve on the independent test data. This result is not still sufficient for use in practice. Moreover results are not stable enough which is main reason why the algorithm is not applicable, yet.

On the other hand results confirm that there are detectable changes in preictal segment and it is possible to predict seizure coming.

There are not enough available intracranial EEG data at this moment. So it is appropriate to record as much data as possible and continue with optimization in bigger amount of records to achieve better and more stable results. Moreover data should cover all physiology and psychology states of the patient.

After the result will be sufficient enough algorithm could be converted to prediction of seizure in real time instead of classification of single segments. Finally, an implantable device for seizure coming alert can be developed.

Appendix A

Appendix

A.1 Content of the CD

Folders and most important files are listed in the Table 15. CD contains also examples of data. It is only few samples due to size of each of them. Implemented algorithm is in main.mat. ROC curve generated by algorithm has no information value due to amount of training data.

Tab. 15 List of folders and files on enclosed CD

Path	Description
`data`	folder with data
`data\lists`	folder with lists of data used for optimization
`data\test`	example test data
`data\training`	example training data
`matlab`	folder with matlab files
`matlab\3rd_party`	third party functions
`matlab\classifiers`	functions for training classifiers and computing AUC
`matlab\data_preparing`	functions for samples creation
`matlab\data_preprocessing`	function for removal of missing data
`matlab\feature_extraction`	functions for features extractions
`matlab\feature_preprocessing`	functions for removal of outliers, normalization and class balance
`matlab\feature_selection`	greedy algorithm and evolutionary algorithm implementation
`matlab\other_functions`	other useful functions
`matlab\main.mat`	implemented algorithm
`lenkazoulova_MT.pdf`	this thesis

Bibliography

[1] R. S. Fisher and H. Jet. "Potential New Methods for Antiepileptic Drug Delivery". In: *CNS Drugs* 16.9 (Feb. 2002), pp. 579–93. DOI: 10.2165/00023210-200216090-00001.

[2] M. F. Bennewitz and W. M. Saltzman. "Nanotechnology for delivery of drugs to the brain for epilepsy". In: *Neurotherapeutics* 6.2 (Apr. 2009), pp. 323–36. DOI: doi:10.1016/j.nurt.2009.01.018.

[3] M. Fujii et al. "Application of Focal Cerebral Cooling for the Treatment of Intractable Epilepsy". In: *Neurologia medico-chirurgica* 50.9 (2010), pp. 839–844. DOI: 10.2176/nmc.50.839.

[4] R. S. Fisher. "Therapeutic devices for epilepsy". In: *Annals of Neurology* 71.2 (2012), pp. 157–168. ISSN: 1531-8249. DOI: 10.1002/ana.22621. URL: http://dx.doi.org/10.1002/ana.22621.

[5] R. Fisher et al. "Electrical stimulation of the anterior nucleus of thalamus for treatment of refractory epilepsy". In: *Epilepsia* 51.5 (2010), pp. 899–908. ISSN: 1528-1167. DOI: 10.1111/j.1528-1167.2010.02536.x. URL: http://dx.doi.org/10.1111/j.1528-1167.2010.02536.x.

[6] G. K. Motamedi et al. "Optimizing Parameters for Terminating Cortical Afterdischarges with Pulse Stimulation". In: *Epilepsia* 43.8 (2002), pp. 836–846. ISSN: 1528-1167. DOI: 10.1046/j.1528-1157.2002.24901.x. URL: http://dx.doi.org/10.1046/j.1528-1157.2002.24901.x.

[7] K. Lehnertz. "Seizure anticipation techniques: state of the art and future requirements". In: *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* Vol. 4. 2001, 4121–4123 vol.4. DOI: 10.1109/IEMBS.2001.1019763.

[8] S. R. Haut et al. "Seizure Clustering during Epilepsy Monitoring". In: *Epilepsia* 43.7 (2002), pp. 711–715. ISSN: 1528-1167. DOI: 10.1046/j.1528-1157.2002.26401.x. URL: http://dx.doi.org/10.1046/j.1528-1157.2002.26401.x.

[9] R. Aschenbrenner-Scheibe et al. "How well can epileptic seizures be predicted? An evaluation of a nonlinear method". In: *Brain* 126.12 (2003), p. 2616. DOI: 10.1093/brain/awg265. eprint: /oup/backfile/content_public/journal/brain/126/12/10.1093/brain/awg265/2/awg265.pdf. URL: +%20http://dx.doi.org/10.1093/brain/awg265.

[10] D. E. Snyder et al. "The statistics of a practical seizure warning system". In: *Journal of Neural Engineering* 5.4 (2008), p. 392. URL: http://stacks.iop.org/1741-2552/5/i=4/a=004.

[11] R. G. Andrzejak et al. "Seizure prediction: Any better than chance?" In: *Clinical Neurophysiology* 120.8 (Aug. 2009), pp. 1465–1478.

[12] D. S. Wickramasuriya, L. P. Wijesinghe, and S. Mallawaarachchi. "Seizure prediction using Hilbert Huang Transform on field programmable gate array". In: *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. Dec. 2015, pp. 933–937. DOI: 10.1109/GlobalSIP.2015.7418334.

[13] H. T. Shiao et al. "SVM-Based System for Prediction of Epileptic Seizures From iEEG Signal". In: *IEEE Transactions on Biomedical Engineering* 64.5 (May 2017), pp. 1011–1022. ISSN: 0018-9294. DOI: 10.1109/TBME.2016.2586475.

[14] M. Z. Parvez and M. Paul. "Seizure Prediction Using Undulated Global and Local Features". In: *IEEE Transactions on Biomedical Engineering* 64.1 (Jan. 2017), pp. 208–217. ISSN: 0018-9294. DOI: 10.1109/TBME.2016.2553131.

[15] Z. Zhang and K. K. Parhi. "Seizure prediction using long-term fragmented intracranial canine and human EEG recordings". In: *2016 50th Asilomar Conference on Signals, Systems and Computers*. Nov. 2016, pp. 361–365. DOI: 10.1109/ACSSC.2016.7869060.

[16] F. E. A. Martini. *Anatomy and Physiology' 2007 Ed.2007 Edition*. Rex Bookstore, Inc. ISBN: 9789712348075.

[17] T. M. Jessel E. Kandel J. Schwartz. *Principles of Neural Science*. Appleton & Lange, 1991. ISBN: 0444015620.

[18] S. J. Thorpe. *Parallel Processing in Neural Systems and Computers*. North-Holland, 1990. ISBN: 0444883908.

[19] S. J. McPhee and G. D. Hammer. *Pathophysiology of Disease An Introduction to Clinical Medicine, Sixth Edition (Lange Medical Books)*. McGraw-Hill Medical, 2009. ISBN: 9780071621670.

[20] J. S. Meyer and L. F. Quenzer. *Psychopharmacology: Drugs, the Brain, and Behavior*. Sinauer Associates is an imprint of Oxford University Press, 2013. ISBN: 087893510X.

[21] R. S Fisher et al. "ILAE Official Report: A practical clinical definition of epilepsy". In: *Epilepsia* 55.4 (2014), pp. 475–82.

[22] Ch. Hope. *A cap holds electrodes in place while recording an EEG*. [Online; accessed May 17, 2017]. 2012. URL: http://www.flickr.com/photos/tim_uk/8135755109/.

[23] University of Pennsylvania and Mayo Clinic. *International Epilepsy Electrophysiology Portal*. https://www.ieeg.org/.

[24] MATLAB. *version 9.1.0 (R2016b)*. Natick, Massachusetts: The MathWorks Inc., 2016.

[25] R. P. W. Duin et al. *PRTools, a Matlab toolbox for pattern recognition*. 2004. URL: http://www.prtools.org.

[26] B. H. Brinkmann et al. "Crowdsourcing reproducible seizure forecasting in human and canine epilepsy". In: *Brain* 139.Pt 6 (June 2016), pp. 1713–1722.

[27] M. Misiti, Y. Misiti, and J.M. Poggi G. Oppenheim. *wfbmesti*. May 2003. URL: https://www.mathworks.com/help/wavelet/ref/wfbmesti.html.

[28] T. Higuchi. "Approach to an irregular time series on the basis of the fractal theory". In: *Physica D: Nonlinear Phenomena* 31.2 (1988), pp. 277–283.

[29] S. Popinet. *COMPLETE HIGUCHI FRACTAL DIMENSION ALGORITHM*. https://www.mathworks.com/matlabcentral/fileexchange/30119-complete-higuchi-fractal-dimension-algorithm. 2011.

[30] A. B. Elema-Schönander B. Hjorth. "EEG analysis based on time domain properties". In: *Electroencephalography and Clinical Neurophysiology* 2.2 (Apr. 1970), pp. 306–310.

[31] B. Davidson. *The Hurst exponent MATALB function*. https://www.mathworks.com/matlabcentral/fileexchange/9842-hurst-exponent. 2005.

[32] N. V. Chawla et al. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[33] K. Amarasinghe, P. Sivils, and M. Manic. "EEG feature selection for thought driven robots using evolutionary Algorithms". In: *2016 9th International Conference on Human System Interactions (HSI)*. July 2016, pp. 355–361. DOI: 10.1109/HSI.2016.7529657.

[34] K. Chen and H. Liu. "Towards an evolutionary algorithm: a comparison of two feature selection algorithms". In: *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*. Vol. 2. 1999, 1313 Vol. 2. DOI: 10.1109/CEC.1999.782597.

[35] A. Zagorecki. "Feature selection for naive Bayesian network ensemble using evolutionary algorithms". In: *2014 Federated Conference on Computer Science and Information Systems*. Sept. 2014, pp. 381–385. DOI: 10.15439/2014F498.