2. Materials and Methods

2.1 Materials

2.1.1 EEG recording

EEG dataset “EEG Motor Movement/Imagery Dataset”[1] was used from PhysioNet databank[2]. In this dataset subjects performed different motor/imagery tasks (we use only motor tasks) while 64-channel EEG was recorded using the BCI2000 system [3]. In total data from 105 subjects was used, each subject has around 25 trials (variate from 22 to 26). Each subject performed two one-minute baseline runs (one with eyes open, one with eyes closed) before all trials. For authentication, the following tasks were used: a target appears on either the left or the right side of the screen. The subject opens and closes the corresponding fist until the target disappears. Then the subject relaxes.

2.1.2 EEG preprocessing

Data was imported into MATLAB for analysis using custom-written scripts. The duration of the experiment excluding electrode setup was around 30 min. In total, for each subject around 25 trials for each task were used.  Before each trial there were 3-4 sec recording of relaxation, which then was used as a baseline. The EEG signal then was filtered using zero phase delay with range 1-50 Hz.

2.2 Feature Extraction Methods

2.2.1 EEG Spectral Analysis

Power Spectral Density (PSD) was calculated with multitaper method on Chronux toolbox[4]. For each channel 20 frequency bands were chosen from 1 to 40 Hz in log scale, and PSD was extracted.

2.2.2 EEG signal univariate complexity analysis

One of the methods of a signal representation is Empirical Mode Decomposition (EMD). This method represents a signal as a sum of modulated components known as IMFs.

Signal = IMF1 + IMF2 + IMF3 + ··· + IMFn

Each IMF connected with width of a frequency band: width gradually decrease with increasing of IMF number. The first IMF is calculated as mean signal between two signals, which created from local minimums and local maximums of the signal (figure 1). The next IMF is created with the same method, but using previous IMF as input.

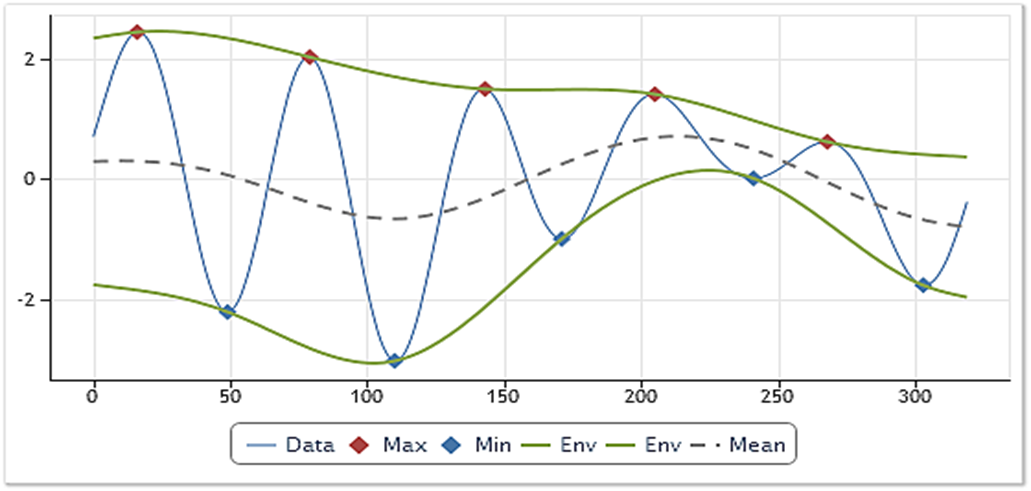


Figure 1: Empirical Mode Decomposition method visualization

For each trial EMD was made with 40 dB resolution and 60 dB residual energy, and only first 5 intrinsic mode functions (IMFs) were left, as the most informative.

For each channel and for each first 5 IMFs we have obtained the following entropies: Univariate Shannon entropy, log entropy, Sample entropy, Approximate entropy.(ссылки на коды у каждой энтории).(коды не мои просто)

Shannon entropy *H* is given by the formula:

Log energy *H* is given by the formula:

Where *p(x)* is the probability of character number *x* appearing in the stream of characters of the message.

Approximate entropy (ApEn) is given by the formula:

Sample entropy (SampEn) is given by the formula:

Bm –probability the similarity between 2 sequences with length m obeys r (tolerance level).

Am+1 –probability the similarity between the same 2 sequences with length m+1 obeys r.

The calculations ApEn also includes self matches, SampEn is not. In our work m was equal 2, r was set as 15% of standard deviation of time-series.

2.3 Multivariate analyses methods

After feature extraction we got 40 features for each of 64 channels, from which matrix 40\*64 for each trial was formed. An obvious challenge was to reduce the size of the matrix to be able to use such machine learning techniques as SVM, and, what is more important, to investigate which of extracted features had the greatest significance in the problem of subjects authentication.

For this purposes, simultaneous multichannel analyses was proposed. It was found that within the same features between the channels there can be a connection, and the orders of values coincide within the characteristics, however, they can differ significantly between different features.

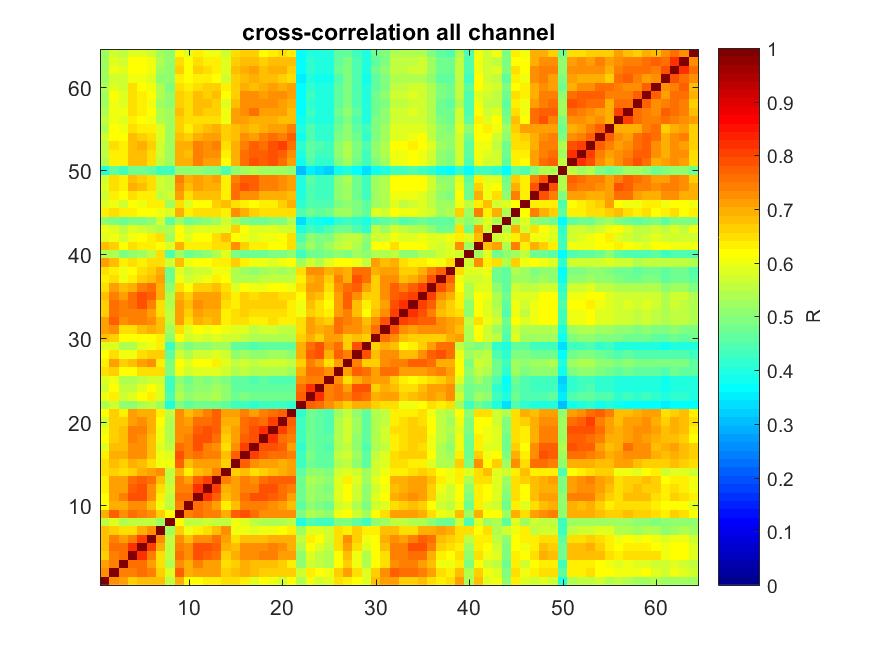
Based on cross-correlation method, we could distinguish 2 clusters of channels. Figure 2 represents mean cross-correlation of all characteristics. Therefore, the main idea was to do the convolution of the channels within each characteristic, after which we get 40\*2 matrix for each trial. Additionally, matrix 40\*1, but low accuracy of the system was performed in this case.

Figure 2: mean cross-correlation of all characteristics

2.3.1 Neural network horizontal convolution

Neural network is one of the artificial intelligence algorithms, which was modeled on the principle of biological neural networks functioning, and could be used for pattern recognition, discriminant analysis or clustering. Moreover, neural networks could be used to compress big-scaled data into the smaller set of features, which later can be used in other classification methods. Therefore, we used neural network as a first part of larger machine-learning system in order to do the convolution between channels.

The main idea was to use a set of convolutional layers to find dependencies only within the characteristics (within rows), while not affecting the dependencies between the various characteristics (between columns); that is why in the convolutional layers only 1\*n-type filters where used.

For this purposes, the neural network was built using Keras framework in Python. Its model was based on the part of  “InceptionV3” model, the main feature of which is that it has filters of several sizes working at the same level, and had the following structure (figure 3).

Subsequently, the matrix 40 \* 64 was converted by this neural network into a matrix 40 \* 2.

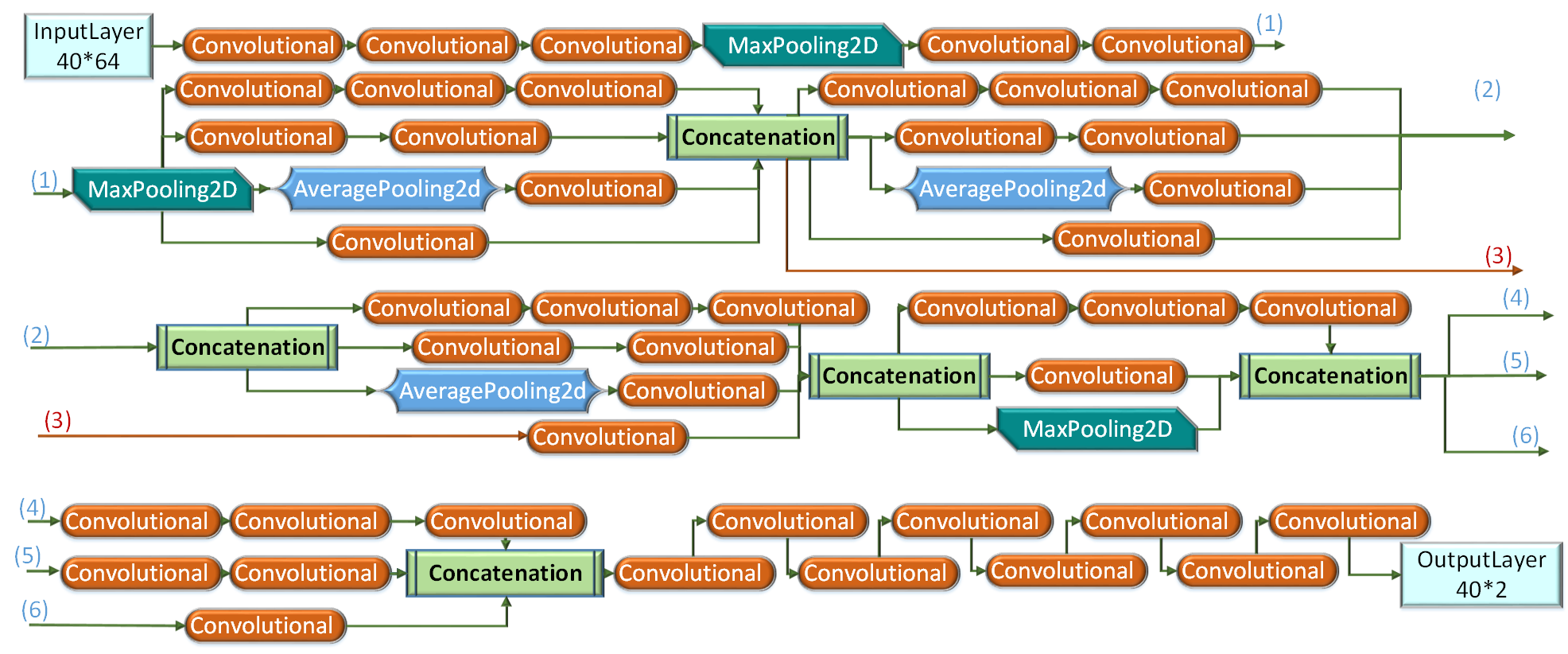
Then this vector was transformed into 80\*1 for SVM.

Figure 3: neural network structure

2.3.2 PCA

Principal component analysis (PCA) is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components (figure 4).

In our case we compared our convolution method with the common one, that is why matrix 40 \* 64 was converted by PCA into a matrix 40 \* 2, like NN does. Then this vector was transformed into 80\*1 for SVM.

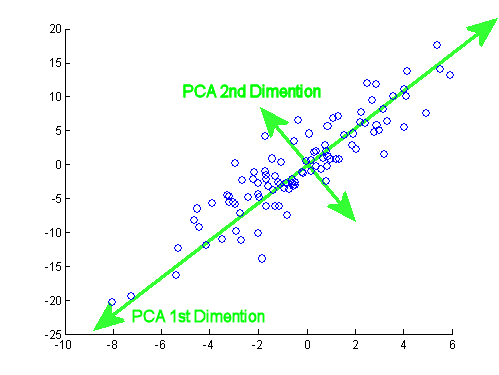


Figure 4: PCA method visualization

2.4 Classification method

2.4.1 Build classification model

Three authentication systems were built: in the first system, the input data was opening and closing of the left fist, in the second - opening and closing of the right fist, and input data for the third system was a sequence of two actions (compression of the left and right fist). In the third system, the subject passed authentication only in the case of correct prediction of each actions, which was done in order to reduce the type II error. For each system, 3 different models were applied.

In the first model (NN model), aforementioned NN, that was expanded with the set of Dense layers, was used for authentication.

In the second model (NN+SVM model), the combination of pre-trained neural network and SVM was used. Thus, at first horizontal convolution of matrix was done using NN, and then this data was given as input into SVM for final classification.

In the third model (PCA+SVM model), we performed horizontal convolution using PCA, and then this data was classified via SVM as in the NN+SVM system.

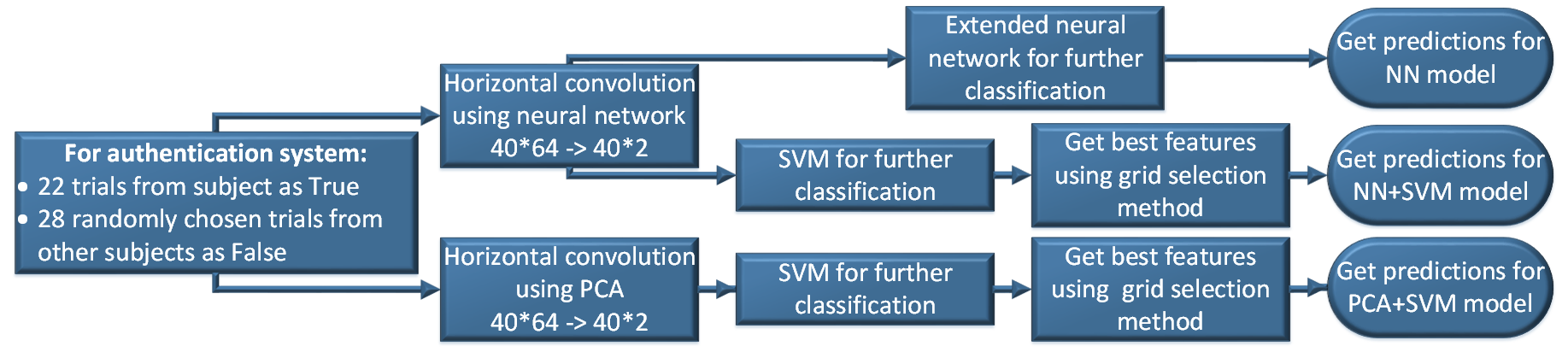
For training model for one user we used a sample, which included 22 recordings from a given subject, and 28 recordings from other users, selected as follows: for all registered in the system users, 28 users were randomly selected, and for each of them one EEG recording was randomly selected as well.

Figure 5: explanation of NN model, NN+SVM model, and PCA model

2.4.2 NN classification in end

For classification, as well as for training the layers of model of neural network, presented in the chapter 2.3.1, the aforementioned model was expanded with the set of Dense layers. Moreover, a Dropout layer was added in the middle of this set for reducing overfitting of network. The structure presented in the figure 6. For training, Adam optimization method was used with learning rate 10-4.

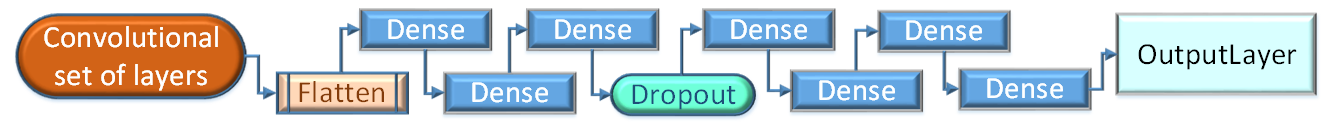


Figure 6: layers, by which NN was expanded

2.4.3 SVM

(explain SVM in 2 words)

Input for SVM could be from PCA or from NN. The main plus of SVM is the fast speed, that is why it is possible to choose the best group of features with method known as a grid selection.

2.4.3.1 Grid selection

We compared accuracy of classification which was obtained from 5 fold cross validation for each feature, then feature with the biggest accuracy was left. After that for all possible pairs (first best features and all other ones) accuracy was calculated as well, and best pair was chosen (figure ).

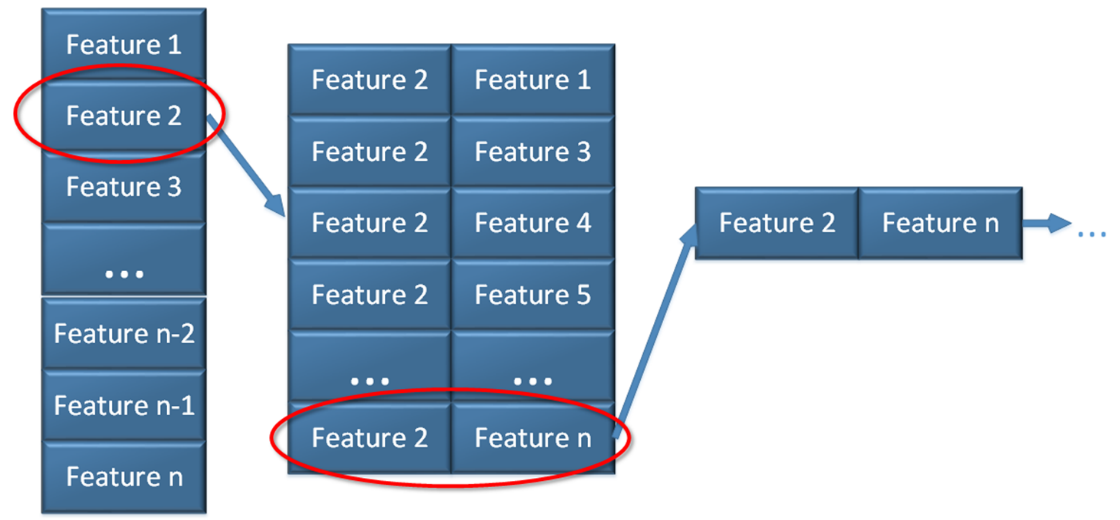
Then best groups of features was compared for different subject and for different decompositions.

Figure 7: Grid selection method visualisation

3. Results and discussions

3.1 Obtained accuracy of the system

To obtain the accuracy of the model, 5-fold cross-validation method was used. Inside each fold, the training data was normalized inside each characteristic, and then the model was trained in a training sample and tested on a test one, which was normalized using the minimum and maximum values of the training sample.

Roc-curves of all models are presented at Figure 8,9 and 10 for systems, based on right, left and both fists respectively.

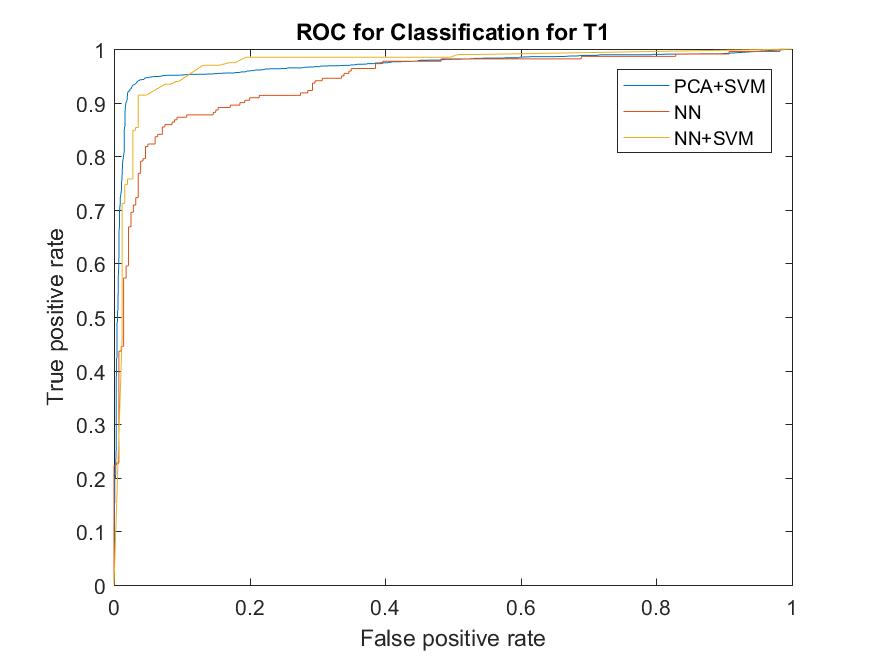


Figure 8: ROC for T1

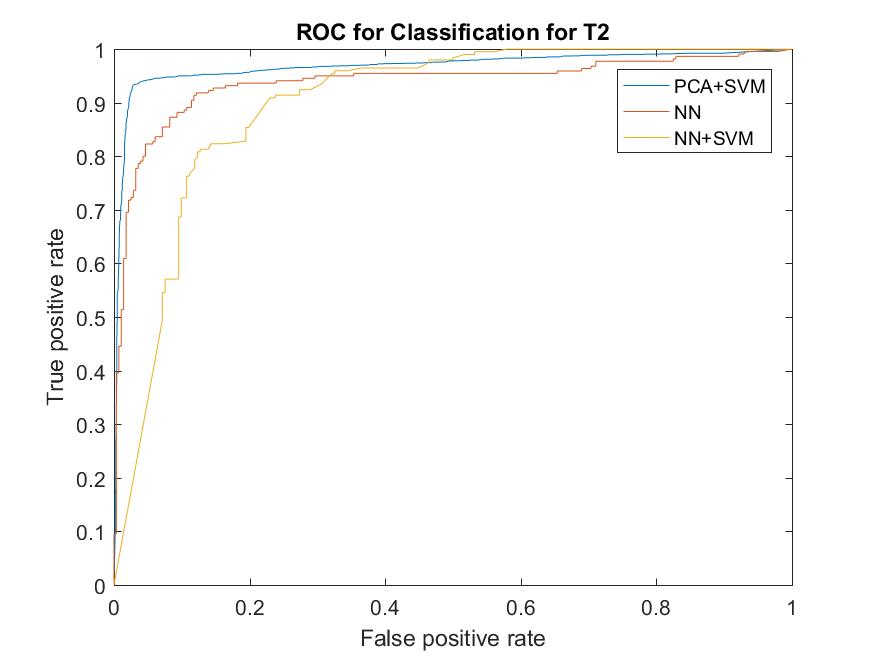
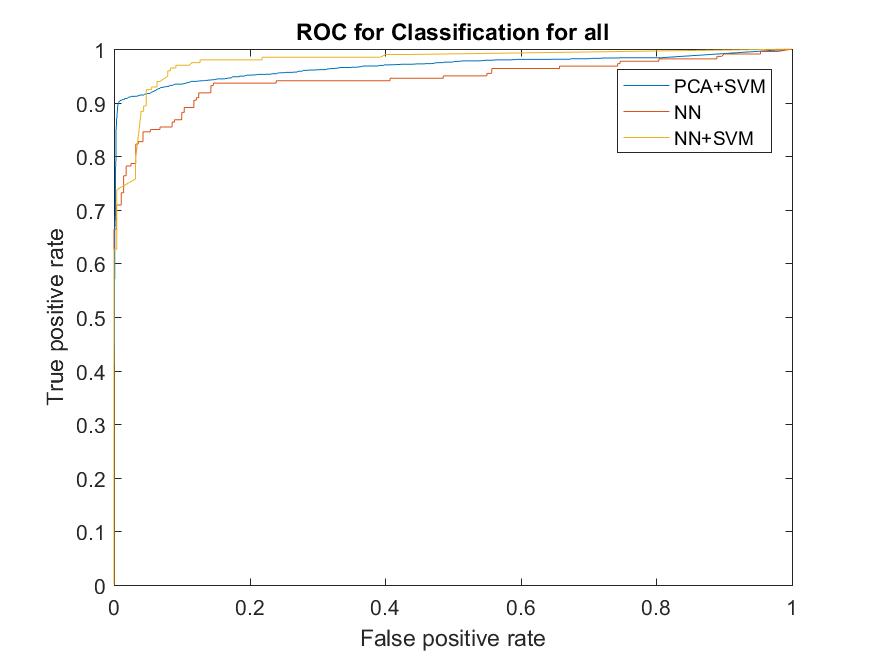


Figure 10: ROC for both

Figure 9: ROC for T2

Accuracy and error Type I, II were calculated for all models and all systems and presented at Table 1-3.

Type II error is the most important for authentication system, because the wrong person should never be authenticated.

Table 2: accuracy of PCA+SVM model

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Right fist** | **Left fist** | **Both fists** |
| Type I error | 0.0251 | 0.0265 | 0.0482 |
| Type II error | 0.0095 | 0.0097 | 0.0011 |
| Overall accuracy | 0.9653 | 0.9638 | 0.9507 |

Table 1: accuracy of NN model

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Right fist** | **Left fist** | **Both fists** |
| Type I error | 0.1272 | 0.1136 | 0.2272 |
| Type II error | 0.0999 | 0.1035 | 0.0178 |
| Overall accuracy | 0.888 | 0.892 | 0.89 |

Table 3: accuracy of NN+SVM model

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Right fist** | **Left fist** | **Both fists** |
| Type I error | 0.0200 | 0.0500 | 0.0700 |
| Type II error | 0.0120 | 0.0400 | 0.0040 |
| Overall accuracy | 0.9680 | 0.9100 | 0.9260 |

3.2 Best features

As a result of grid selection method, the following features were returned as the best for authentication in PCA+SVM system:

For right fist: Shannon entropy (1, 4 and 5 IMF), Approximate entropy (4 and 5 IMF)

For left fist: Shannon entropy (1 and 5 IMF), Approximate entropy (3 and 5 IMF), log energy (1 IMF),

T1- '5\_shannon' '5\_ApEn' '1\_shannon' '4\_ApEn' '4\_shannon'

T2- '1\_shannon' '5\_ApEn' '5\_shannon' '3\_ApEn' '1\_log\_energy'

NN+SVM

'from\_27.5423Hz\_to\_33.1131Hz' 'from\_1.2023Hz\_to\_1.4454Hz' '1\_SampEn' 'from\_33.1131Hz\_to\_39.8107Hz' 'from\_19.0546Hz\_to\_22.9087Hz'

'3\_log\_energy' 'from\_1.4454Hz\_to\_1.7378Hz' 'from\_33.1131Hz\_to\_39.8107Hz' 'from\_27.5423Hz\_to\_33.1131Hz' 'from\_15.8489Hz\_to\_19.0546Hz'

For T1 and for T2 – different features!

Compare best features from nn+svm and from pca+svm

python train1.py

python train2.py

python train3.py

python train4.py

python train5.py

python train6.py

python train7.py

python train8.py

python train9.py

python train10.py

python train1.py