# Introduction

The authentication system is a system that involves verifying the identity that the one is claiming. The main difference between a system for authentication and identification is that the first require confirming, that given credentials of the subject corresponds to the template of proposed identity, and therefore a task of binary classification in such system is presumed. At the same time, in identification system the goal is to recognize the person from some dataset of subjects, so multiclass classification in such system is presumed. It is obvious that the identification approach would be too complex to use in a real-world system with many users. First of all, usage of machine learning is a popular approach for users’ classification. Therefore, it narrows the opportunities of using such technologies of machine learning as neural networks, since multiclass classification increases the computational cost, while performing the worse results than binary classification. It was proven in [1] that identification system performs lower accuracy than authentication system, even though the same input data was used. Moreover, usually identification systems are not intended for adding new users easily to the system, since it would require to change an output of the whole system and retrain it from scratch. Authentication systems, in which different biometric, i.e. person’s anatomical features (fingerprints, face, palm veins, hand geometry, iris, voice) or behavioral traits (signatures, gaits, etc.), are used as credentials, are called biometric authentication systems. Because such type of data is physically user-related and is hard to steal, biometric authentication systems are claimed to be one of the most reliable ones. However, all aforementioned biometrics are could be faked. One of the perspective types of biometrics recently has proved to be electroencephalogram (EEG) of brain activity. As long as it is recorded from a human head, it is hard to reproduce such signal.The reliability of biometric authentication system could be judged by False Accepted Rate (FAR) - the rate of successful authentications of impostors, and False Rejected Rate (FRR) - the rate of denied accesses for registered users. As long as the wrong person should never be authenticated, FAR is the most important for authentication system, and should be as low as possible.

In the last 20 years, in numerous studies different approaches of using EEG data as biometrics for authentication of the subjects were proposed. Poulos et al. [2] proposed the first authentication system in 1999. They collected a 3-minutes recording of a resting state with closed eyes (REC) from 4 subjects with 45 recording for each, and 75 imposters with 1 recording for each. Auto-Regression and Moving Average model (ARMA) was used for feature extraction, and the Kohonen’s Linear Vector Quantizer model of neural network was trained for binary classification. However, they obtained pretty high average FRR of 0.22 and extremely high for authentication FAR of 0.2, and therefore this system is inappropriate for a real-world authentication system. Marcel et al. [3] collected EEG data from 9 subjects during 12 sessions over 3 days. During the recording, subjects performed different mental tasks: imagining left and right hand movement, and generating the word that starts with same random letter. For forming a feature vector, PSD was extracted from 12 frequency bands for 8 chosen channels. For classification, Maximum A Posteriori model (MAP) was used. They obtained a lot of results from different experimental protocols, with the approximate average FRR and FAR of the best protocols at the level of 0.08 and 0.1 respectively. Interestingly, the task of imagining left hand moving performed the best results. Hu et al. [1] also used recordings of different imaginary tasks, but collected data only from 3 subjects. Feature vector was extracted from an ARMA linear model, as in [2], and neural network with 5 hidden layers was used to classify the subjects. The FRR varied from 0.15 to 0.25, however, the results FAR were not presented in this paper, so it is hard to judge on reliability of the built systems. Yeom et al. [4] collected data from 10 users, where self-face and non-self-face images were used as a stimulus. Interestingly, that 2 twins were among these subjects. For input vector, they extracted so-called ‘temporal’ and ‘dynamic’ features from 18 selected channels, and used Support Vector Machine for classification. The average FAR and FRR of 13.9 was obtained. The most recent work was presented by Wu et al. [5]. They collected EEG and EOG signals from 40 persons (15 as users and 30 as impostors), and used rapid serial visual presentation of faces (face-RSVP) as a stimulus. They then compared systems based on EEG only and the combination of EEG and EOG. However, we are interested in using the EEG signal only, and therefore the above explanation applies only for this system. In the Wu et al. system, unlike aforementioned systems, feature matrix, not a vector, was former for classification. As features, the average ERPs were used for specifically chosen channels. For classification, the convolution neural network was used. However, for this purposed feature matrix was expanded into a square matrix by the sparse method, which is an arguable method to use when forming an input for neural network. The average FAR was around 0.062 and FRR around 8.49.

It is well known that authentication require the more consistent data, and in works [citations] it has been proven that one of the most consistent types of recorded EEG data is recordings during motor activity. Therefore, we proposed to use motor activity stimulus in order to achieve better accuracy and create more reliable system in terms of consistency of the input data. We did not do any channels’ selection for information safety reasons. One of the methods of avoiding leaving the information that could be important is to use all channels and as much as possible features. However, we got a high dimensional matrix of features. As long as neural network has proven to deal with classifying a high dimensional data with a high accuracy and low computational cost (comparing to other machine learning algorithms) [1][5], in this work it was proposed to use it as a classifier. Moreover, in a lot of studies (for example [6], [7]), a method of improving the accuracy of classification by using the combination of neural network and Support Vector Machine (SVM) was proposed. Thus, this method was used in our work as well. Furthermore, a Principal Component Analysis (PCA), just as neural network, could be used for convoluting correlated set of variables into the new, smaller set of variables [8]. Therefore, in this work a PCA was used in a combination with SVM for classification as well.

# 2. Materials and Methods

## 2.1 Materials

### 2.1.1 EEG recording

EEG dataset “EEG Motor Movement/Imagery Dataset”[9] was used from PhysioNet databank[10]. In this dataset subjects performed different motor/imagery tasks (in our work, only motor tasks were used) while 64-channel EEG was recorded using the BCI2000 system [11]. 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 obtained.  Before each trial there were 3-4 sec recording of relaxation, which then was used as a baseline. For increasing number of trials, data were cut for sliding windows with length of 2 sec and 75% of overlap. It total there were gotten around 110 trials per subject for each task. 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[12]. For each channel, PSD was extracted for mu (7.5-12.5 Hz) and betta (16-31 Hz) frequency bands.

### 2.2.2 EEG signal univariate complexity analysis

In order to distinguish complexity of high frequency bands, one of the methods of a signal representation - Empirical Mode Decomposition (EMD)[13] – was used. This method represents a signal as a sum of modulated components known as IMFs.

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

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

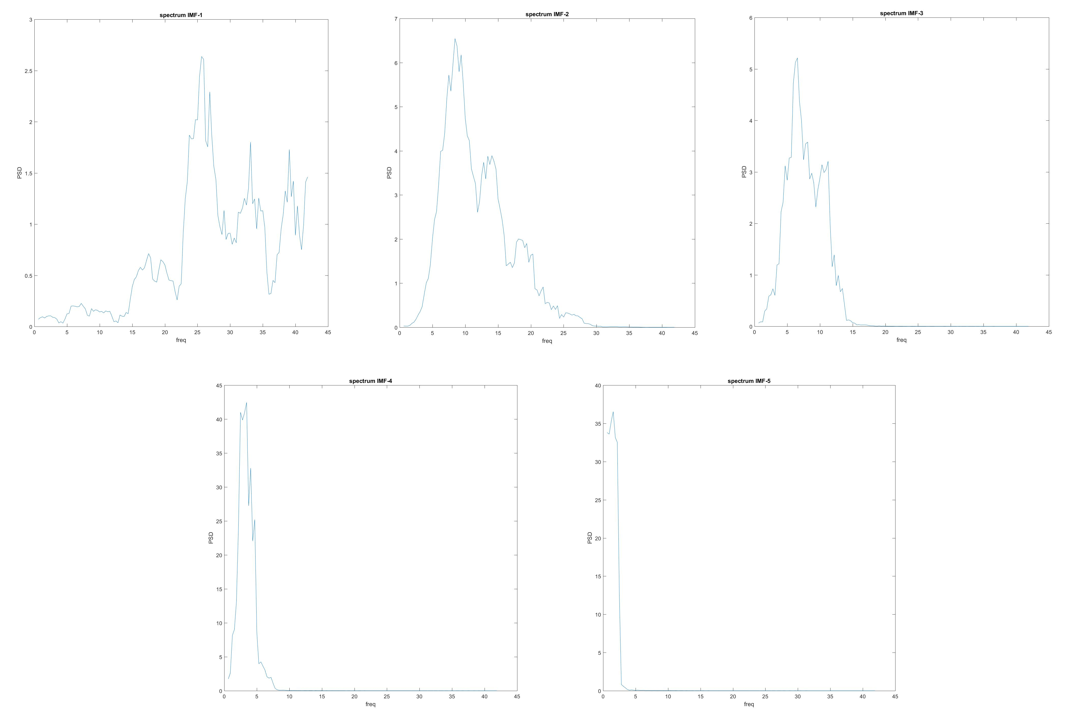
For each trial EMD was calculated with 40 dB resolution and 60 dB residual energy. The PSD of first 4 intrinsic mode functions (IMFs) (figure 1) shows that starting from 5th IMF more noise is contained than biological information, therefore only first 3 IMFs were left, as the most informative.

Figure 1PSD of first 5IMF

For each channel and for each first 4 IMFs the following entropies was obtained: Univariate Shannon entropy, log entropy, Sample entropy, Approximate entropy.

Shannon entropy *H*  [14] is given by the formula:

Log energy *H* [15] 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)[16] is given by the formula:

Sample entropy (SampEn) [17]is given by the formula:

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

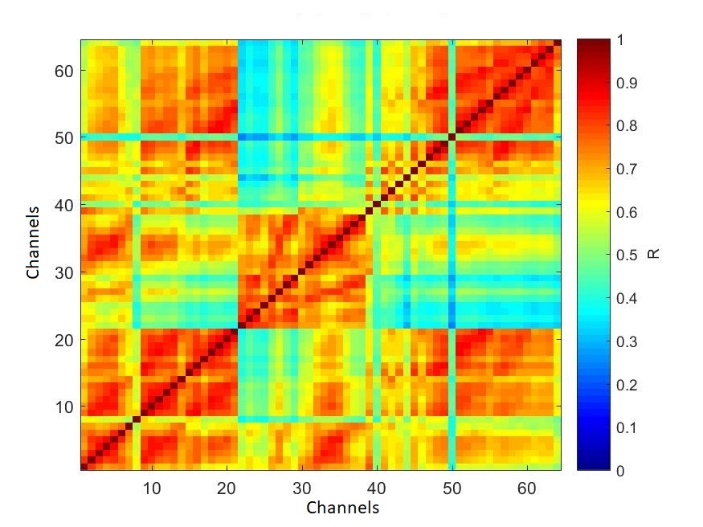
Am+1 –probability the similarity between the same 2 sequences with length m+1 obeys r.

The calculation of ApEn also includes self matches, while SampEn does not. In our work, m=2 and r was set as 15% of standard deviation of time-series.

## 2.3 Multivariate analyses methods

After feature extraction, 24 features for each of 64 channels were gotten, from which matrix 24\*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 features, however, they can differ significantly between different features.

Based on cross-correlation method[18], 2 clusters of channels could be distinguished. For each feature a cross-correlation between all pairs of channels was calculated, then the mean of cross-correlation of all features and standard deviation of the mean were calculated (Figure 2). Therefore, the main idea was to do the convolution of the channels within each features, after which 24\*2 matrix was formed for each trial. Additionally, matrix 24\*1 was checked, but low accuracy of the system was performed in this case.

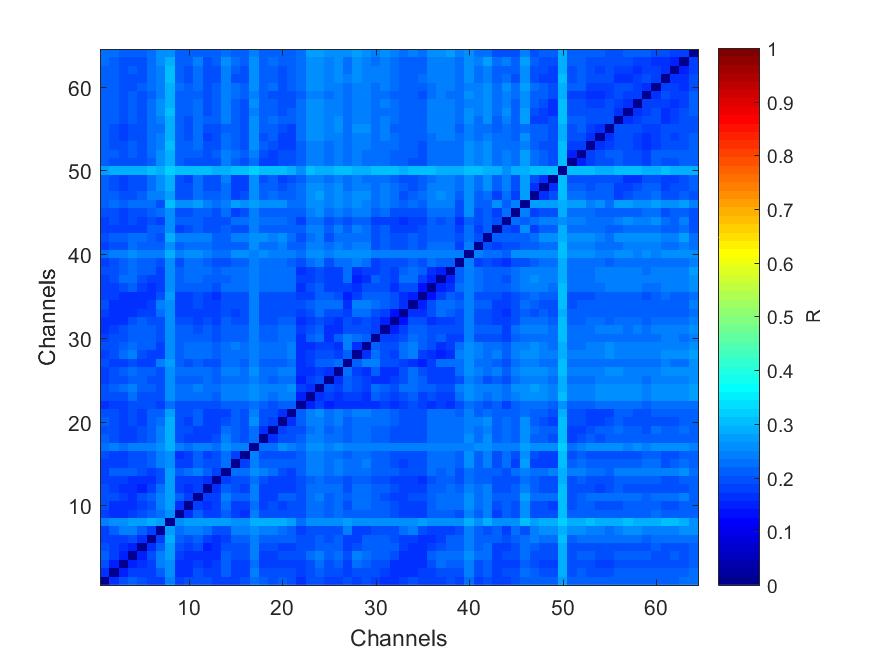


Figure 2: Mean cross-correlation of all features and standard deviation of the mean

2.3.1 Neural network horizontal convolution

Neural network[19] 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, in proposed systems neural network was used as a first part of larger machine-learning system in order to do the convolution between channels. For this purposes, the neural network was built using Keras framework in Python.

The main idea was to use a set of convolutions to find dependencies only within the features (within rows), while not affecting the dependencies between the various features (between columns); that is why in the convolutional layers with only 1\*n-type kernels where used, and only 1\*m strides were used across the network where needed. The convolution is consist of convolutional layer with linear activation, followed by batch normalization, to which the ReLU activation is applied. The Batch Normalization is added in order to avoid the overfitting.

The network architecture was based on the idea of using Inception-like modules, that combines convolutions with different kernels on the same level and act as a small networks within the bigger one. One of the distinctive features of such modules is the usage of convolutions with 1\*1 kernels for dimension reductionality within the module, as well as for adding more non-linearity to the network by applying ReLU activation right after them.

The usage of such modules allows building deep network while avoiding the vanishing gradient problem by keeping balance in both width and depth of the network. However, as it was mentioned in the [20], it is more efficient to use Inception in the middle of the network. Therefore, at the beginning of the network, traditional set of convolutions, followed by pooling layers for grid reduction, was used.

As it was noted by the authors of Inception architecture, the choice of the set of convolutions and pooling layers in these modules can be varied, therefore the optimal configuration of the main modules was selected experimentally. The module A and C are intended for extending the number feature maps, so that network can go deeper, and therefore increase the performance. For first two modules A, additional convolutions were concatenated to keep the balance between width and depth of the network. Moreover, additional convolution was added to concatenate the output of the first and the third module A in order to prevent vanishing gradient problem. The module B is intended for reduction of the grid size and is applied between set of modules A and module C.

To avoid making structure of the network too complex, more traditional set of 2 convolutional layers, followed by pooling layers was applied after the module C for grid reduction. In the tails of the network, sequence of convolutions with 1\*1 kernels were used in order to combine all feature maps into one.

Thereby, the input matrix 18 \* 64 was converted by the above set of layers into a matrix 18 \* 2. Then this vector was transformed into 36\*1 by Flatten layer.

### 2.3.2 PCA

Principal component analysis (PCA)[22] is a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components[8].

In this work, our convolution method was compared with the common one, that is why matrix 24 \* 64 was converted by PCA into a matrix 24 \* 2, like NN does. Then this vector was transformed into 48\*1 for SVM.

## 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), horizontal convolution was performed using PCA, and then this data was classified via SVM as in the NN+SVM system.

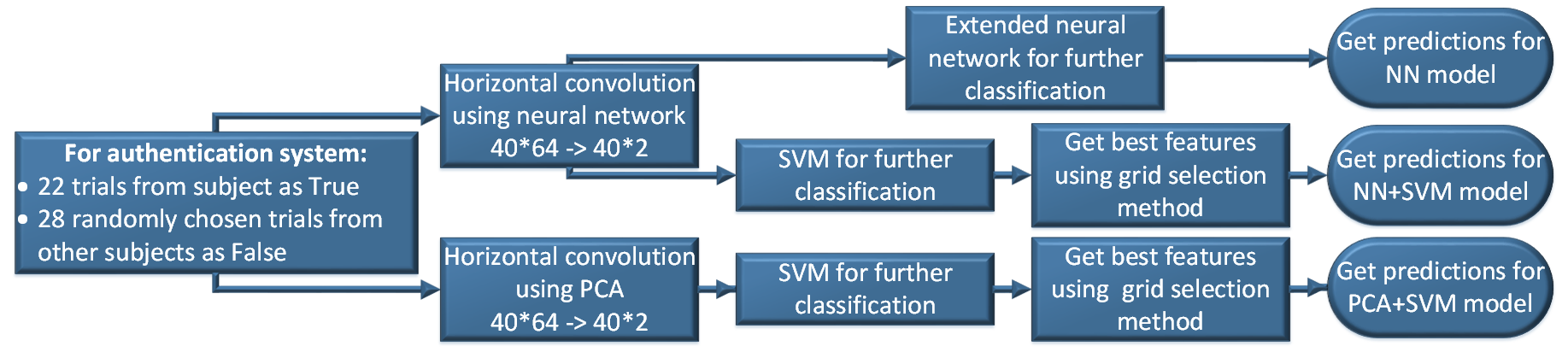
For training model for one user was used a sample, which included 22 recordings from a given subject, and 28 recordings from other users, selected as follows: for all registered in the system users, 28 users were randomly selected, and for each of them one EEG recording was randomly selected as well.

Figure 4: explanation of NN model, NN+SVM model, and PCA model

### 2.4.2 Training neural network for NN and NN+SVM models

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 avoiding overfitting of the network. The structure is presented in the figure 6. The network was subsequently trained with Adam optimization method[23] for 100 epochs with learning rate 10-3 to speed up convergence, and then for another 100 epochs with learning rate 10-4 to increase the accuracy of the network.

### 2.4.3 SVM

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

Accuracy of classification was compared, which was obtained from 5 fold cross validation for each feature, and 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 7).

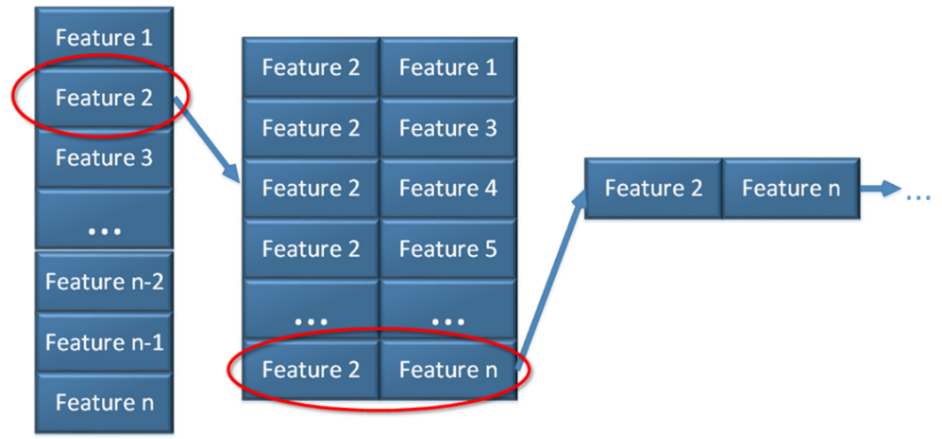
Then best groups of features was compared for different subjects and for different convolutions types.

Figure 6: Grid selection method visualisation

## 2.5 Grading methods

To compare the reliability of the built models between each other, as well as between previous work, the mean accuracy was calculated. To obtain the accuracy of the model, 5-fold cross-validation method was used. Inside each fold, the training data was normalized inside each feature, and then the model was trained in a training sample and tested on a test sample, which was normalized using the minimum and maximum values of the training sample.

For representation of the authentication system performance, mostly False Accepted Rate (FAR) and False Rejected Rate (FRR) are used.

FAR and FRR of the system was calculated between all users as follows:

where *i* – number of the user, j – number of the fold.

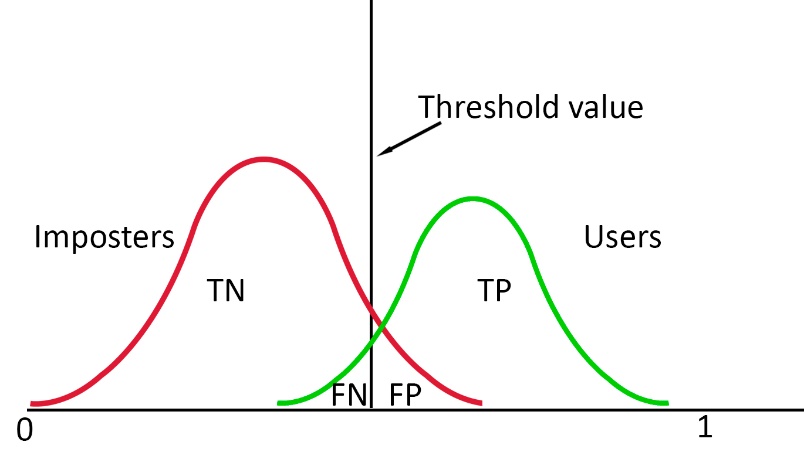
As it can be seen from figure 7, these parameters depend on the chosen threshold. Thus, the Receiver Operating Characteristic (ROC) curves analysis [24] was used to show the most complete picture of the authentication systems performance.

Figure 7: The distribution of the users (green line) and imposters (red line). The TP – True Positive - is the number of correctly granted authentication attempts of the users, TN – True Negative – is the number of correctly denied authentication attempts of the imposters, FN – False Negative - is the number of incorrectly denied authentication attempts of the users, and FP – False Positive – is the number of incorrectly granted authentication attempts of the imposters.

# List of reference:

[1] J. F. Hu, “New biometric approach based on motor imagery EEG signals,” *FBIE 2009 - 2009 Int. Conf. Futur. Biomed. Inf. Eng.*, pp. 94–97, 2009.

[2] M. Poulos, M. Rangoussi, V. Chrissikopoulos, and A. Evangelou, “Parametric person identification from the EEG using computational geometry,” *Proc. IEEE Int. Conf. Electron. Circuits, Syst.*, vol. 2, no. 1, pp. 1005–1008, 1999.

[3] S. Marcel and J. R. Millan, “Person Authentication Using Brainwaves (EEG) and Maximum A Posteriori Model Adaptation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 743–752, Apr. 2007.

[4] S. K. Yeom, H. Il Suk, and S. W. Lee, “Person authentication from neural activity of face-specific visual self-representation,” *Pattern Recognit.*, vol. 46, no. 4, pp. 1159–1169, 2013.

[5] Q. Wu, Y. Zeng, C. Zhang, L. Tong, and B. Yan, “An EEG-based person authentication system with open-set capability combining eye blinking signals,” *Sensors (Switzerland)*, vol. 18, no. 2, pp. 1–18, 2018.

[6] S. Kaur and E. D. Kaur, “HYBRID MODEL USING COMBINATION OF NEURAL NETWORK AND SUPPORT VECTOR MACHINE FOR DETECTION OF LUNG CANCER,” vol. 2, no. 2, pp. 39–42, 2015.

[7] Y. Cao, R. Xu, and T. Chen, “Combining Convolutional Neural Network and Support Vector Machine for Sentiment Classification,” Springer, Singapore, 2015, pp. 144–155.

[8] “Supervised Machine Learning — Dimensional Reduction and Principal Component Analysis.” [Online]. Available: https://hackernoon.com/supervised-machine-learning-dimensional-reduction-and-principal-component-analysis-614dec1f6b4c. [Accessed: 07-May-2019].

[9] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, “BCI2000: A General-Purpose Brain-Computer Interface (BCI) System,” *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 1034–1043, Jun. 2004.

[10] A. L. Goldberger *et al.*, “PhysioBank, PhysioToolkit, and PhysioNet,” *Circulation*, vol. 101, no. 23, Jun. 2000.

[11] “BCI2000 | Schalk Lab.” [Online]. Available: http://www.schalklab.org/research/bci2000. [Accessed: 07-May-2019].

[12] “Chronux Home.” [Online]. Available: http://chronux.org/. [Accessed: 07-May-2019].

[13] “Empirical Mode Decomposition: Riding the Waves.” [Online]. Available: http://brocabrain.blogspot.be/2011/04/empirical-mode-decomposition-riding.html. [Accessed: 10-May-2018].

[14] C. E. Shannon and W. Weaver, “THE MATHEMATICAL THEORY OF COMMUNICATION,” 1949.

[15] S. Aydın, H. M. Saraoğlu, and S. Kara, “Log Energy Entropy-Based EEG Classification with Multilayer Neural Networks in Seizure,” *Ann. Biomed. Eng.*, vol. 37, no. 12, pp. 2626–2630, Dec. 2009.

[16] S. M. Pincus, “Approximate entropy as a measure of system complexity.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 88, no. 6, pp. 2297–301, Mar. 1991.

[17] J. S. Richman and J. R. Moorman, “Physiological time-series analysis using approximate entropy and sample entropy,” *Am. J. Physiol. Circ. Physiol.*, vol. 278, no. 6, pp. H2039–H2049, Jun. 2000.

[18] R. N. Bracewell, “The Fourier Transform and applications,” *McGraw Hill*, pp. 1–10, 2000.

[19] M. H. Hassoun and M. H., *Fundamentals of artificial neural networks*. MIT Press, 1995.

[20] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” *Cvpr*, vol. 91, no. 8, pp. 2322–2330, 2016.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” Dec. 2015.

[22] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, Jul. 2010.

[23] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” Dec. 2014.

[24] C. E. Metz, “Basic principles of ROC analysis.,” *Semin. Nucl. Med.*, vol. 8, no. 4, pp. 283–98, Oct. 1978.