Neural Network Model for Artifacts Marking in EEG Signals

Olga Komisaruk, Evgeny Nikulchev
MIREA — Russian Technological University
Moscow 119454, Russia

***Abstract*—One of the main methods for research of the holistic activity system of human brain is the method of electroencephalography (EEG). For example, eye movements, blink, hearth activity, muscle activity that affects EEG signal interfere with cerebral activity. The paper describes the development of an intelligent neural network model aimed at detecting the artifacts in EEG signals. The series of experiments were conducted to investigate the performance of different neural networks architectures for the task of artifact detection. As a result, the performance rates for different ML methods were obtained. The neural network model based on U-net architecture with recurrent networks elements was developed. The system detects the artifacts in EEG signals using the model with 128 channels and 70% accuracy. The system can be used as an auxiliary instrument for EEG signal analysis.**

Keywords—Artifacts in EEG signal; neural network model; recurrent neural network; U-net architecture

I. INTRODUCTION

Electroencephalography provides quantitative and qualitative analysis of human brain functionality and its reactions to stimulants. Electroencephalogram (EEG) is important for brain activity and behavior recognition, but there are always artifacts in electrical activity records that have influence on EEG signal analysis.

Measuring instruments, including defective electrodes, disturbances and high electrode resistance can be the reason of artifact occurrence. These artifacts can be recognized by more accurate recording system, but physiological artifacts are more complex. The eye movements, blink, hearth activity, muscle activity that affects EEG signal interfere with neural activity and can be used as normal phenomenon [1].

Artifact is a signal, caused by an extracerebral source, observed during EEG recording. They identify physical and physiological causes of artifacts [2]. Artifacts obtained during an electroencephalographic investigation represent a recording defect [3]. Modern electroencephalographic equipment records extremely small values of changes in bioelectric potentials, and therefore the true EEG recording can be distorted due to the influence of a variety of physical (technical) or physiological artifacts [4]. In some cases, such artifacts can be removed using analog-digital converters and various filters, but if the artifact effect coincides in characteristics of wave frequency with a real EEG recording, then these methods become ineffective.

The most common physical artifacts are mains frequency, phone artifact, wire breakage, poor electrode contact, high resistance artifact.

The following physiological artifacts are often recorded: ECG artifact, vascular artifact, galvanic skin artifact, oculomotor artifact, electrooculogram, myographic artifact - electromyogam [2]. The appearance of such artifacts is due to various biological processes occurring in the patient's body.

An ECG artifact most often occurs in the examined patients suffering from increase in arterial pressure, mainly in monopolar and transverse biopolar leads [5]. Usually, its occurrence is associated with an increase in the activity of the sympathetic nervous system, which facilitates the conduction of an ECG signal to peripheral tissues. Galvanic skin artifact occurs due to the activation of the patient's parasympathetic nervous system and increased sweating. As a result, there is a general cyclical change in the resistance of the skin and the skin-electrode system [1]. An oculomotor artifact, an electrooculogram (EOG), appears as slow-wave oscillations in the frontopolar leads with a frequency of 0.3–2 Hz. The appearance of EOG is associated with a change in the position of the eyeball (retina). Myographic artifact occurs when the frontal, chewing and occipital muscles are strained. The appearance of such an artifact can be both a spontaneous stress of the patient and involuntary reaction to an overly tightly put on fixing electrodes system [6].

The use of machine learning methods and neural networks determines promising research in the field of automatic artifact detection. In neurocomputer technologies, there is a general training scheme [7], which is divided into a training set, in which optimization of parameters is carried out, and a test set, according to which the quality of the resulting model is assessed. At the stage of training, it is necessary to understand the signs by which the classifier will be trained [8].

The paper contains six sections. The second section presents the overview of the approaches used. The third section describes the source data for current study. The fourth section presents methods used in the study including the description of software and the types of the neural network architectures. In the fifth section, conclusions of the study are given. This section presents the result of the searching for effective architecture for a qualitative solution to the problem of searching for artifacts. The sixth section contains general conclusion.

II. Related Work

EEG is a tool for psychophysiologic researches. However, the record filtering is often accomplished by high qualified professionals and takes a lot of resources and special filtering techniques [9]. Under these conditions development of effective EEG data filtering methods is an urgent task.

Fast development of cheap high parallel computation infrastructure, powerful machine learning algorithms and big data caused a huge progress in deep learning. The modern approaches of automatic interpretation of EEG use modern techniques such as neural networks and support vector machine.

Machine learning and neural network techniques in particular [2] determine perspective in researches in the automatic artifact recognition domain.

There are five basic algorithms [10] that are widely used in classifiers:

- Linear classifier [11]. It is more popular in online applications including real-time applications. One of the most effective method is support vector machine that usually better than other classifiers.
- Neural networks [12]. The most frequent methods for time series analysis are such architectures as convolutional neural network and recurrent neural network.
- Non-linear classifier [13]. Common methods are hidden Markov models and Bayesian classifiers.
- K-means [14]. These classifiers are based on neighbor distance values.
- Classifier combinations [15]. This method combines different classifiers and demonstrates good efficiency for autonomous applications.

Due to real-time classification, described classifier methods are more optimal for EEG signal analysis.

The task of the machine in unsupervised learning is to find relationship between individual data, to identify patterns, to select patterns, to organize data or describe their structure, and to classify data.

One of the most known drawbacks of machine learning methods is that the source data for training and data for test belong to the same feature space and follow the same probability distribution.

The aim of the research is development of intelligent tools based on neural network technologies that can recognize artifacts in EEG obtained via 64-channel electroencephalograph.

III. Data

EEG data is recorded using electroencephalograph Brain Products, containing 128 channels, 64 sensors placed on the international system "10-10%".

The aim of the experiment was analyzing brain activity zones in resting state and nonverbal intelligence dependencies.

The study was conducted in a sound-attenuated and electrically shielded dimly lit room. Impedance was kept under 25 kOhm with high conductive chloride gel. The time of EEG settling was approximately 15 minutes.

The BrainProducts PyCorder system was used as a data collection system. This system allows continuous recording without any filtering and continuous sampling at 500 Hz. The reference electrode was located at Cz. The data was re-referenced to the common reference after the recording and downsampled to 256 Hz. The data were filtered from 0.1 Hz to 30 Hz and then re-referenced to an averaged reference and manually cleaned from artifacts, with noisy channels excluded.

To remove blink and vertical eye-movement artifacts, independent component analysis (ICA) was performed on the following electrodes: VEOG — Fp1, HEOG — FT9 and FT10. After ICA, the excluded channels were topographically interpolated, and semiautomatic artifact rejection was conducted.

Dataset contains two types of files:

1) Edf files are source data of EEG recording process (see example in Fig. 1)

2) "Markers" files contain description of artifacts (see example in Fig. 2):

- type - type of interval;
- description - artifact description (for example, "Blink");
- position - time of artifact appearance (unit of time represented in "SamplingInterval" field, that equals to 3.90625 ms);
- length - artifact duration;
- channel - channel name, representing the location of artifact (Fp1, Fp2 – "Blink", All - artifact that appeared in all channels).

There are only two types of artifacts. Thereby, neural network will classify three classes: "Blink", "Global artifact" and "Resting state" (when there are no artifacts).

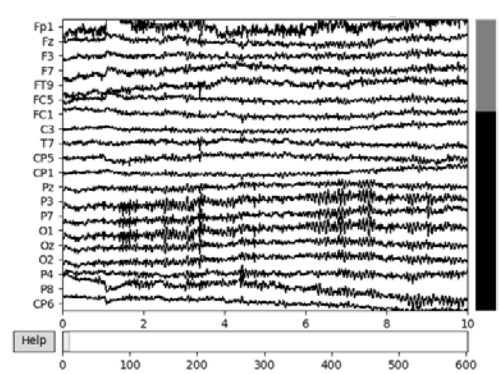
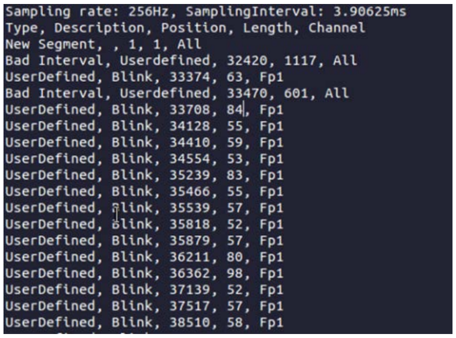


Fig. 1. Example of Edf File Format Content.



```
Sampling rate: 256Hz, SamplingInterval: 3.90625ms
Type, Description, Position, Length, Channel
New Segment, , 1, 1, All
Bad Interval, Userdefined, 32420, 1117, All
UserDefined, Blink, 33374, 63, Fp1
Bad Interval, Userdefined, 33470, 601, All
UserDefined, Blink, 33708, 84, Fp1
UserDefined, Blink, 34128, 55, Fp1
UserDefined, Blink, 34410, 59, Fp1
UserDefined, Blink, 34554, 53, Fp1
UserDefined, Blink, 35239, 83, Fp1
UserDefined, Blink, 35466, 55, Fp1
UserDefined, Blink, 35539, 57, Fp1
UserDefined, Blink, 35818, 52, Fp1
UserDefined, Blink, 35879, 57, Fp1
UserDefined, Blink, 36211, 80, Fp1
UserDefined, Blink, 36362, 98, Fp1
UserDefined, Blink, 37139, 52, Fp1
UserDefined, Blink, 37517, 57, Fp1
UserDefined, Blink, 38510, 58, Fp1
```

Fig. 2. Example of «Markers» File Format Content.

IV. METHODS

To select a neural network model, it is necessary to conduct experimental studies of various architectures. An intelligent EEG signal analysis circuit has been developed (Fig. 3). Intelligent analysis of EEG signals consists of the process of recording and forming a database, processing signals and training a neural network model.

Recording process consists of taking readings using an electroencephalograph, the data of the electrodes located on the surface of the head are sent to the BrainProductsPyCorder data acquisition system. Next, expert analysis and processing of the generated database is carried out, in which different types of artifacts are marked, and then a new database is formed containing information about artifacts in each .edf file.

Based on the Database analysis, the size of input and output of neural network was determined. Pre-processing block reads Markers Database. Then, Data analyze block analyzes it. After that, train and test samples formed.

It is necessary to determine input and output. To find the solution, data was analyzed where distance between artifacts and maximum duration of every type of artifact were found. Also, quantity for every type of artifact was analyzed for data balance. For that, Data_analyzer.py library was created. The library consists of the following methods:

- max_artifact_length - returns maximum length of the artifacts;
- max_type_length - returns maximum length of the artifact of the specific type;
- channel_stats - based on markers data, it returns quantity of artifacts for every channel;
- normal-state-lengths - returns distances between artifacts (lengths of «resting state»);
- getMaxMin_by_edf - returns maximum and minimum values of frequency in edf file;
- getMaxMin_by_train - returns maximum and minimum values of frequency in input samples.

Using the described methods, the most optimal time window was selected for determining artifacts, based on the maximum length of the artifact Blinking (1.8 seconds) [16].

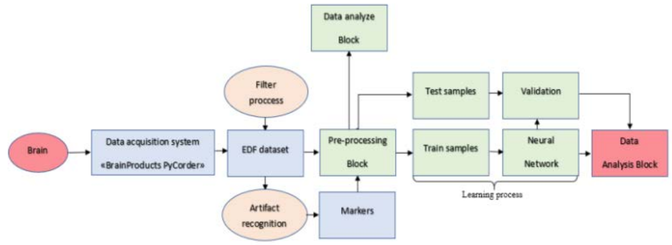


Fig. 3. EEG Signal Intelligent Analysis Scheme.

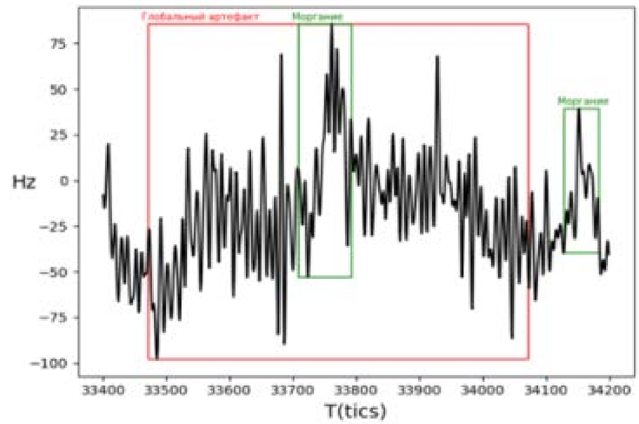


Fig. 4. Signal Graph with Artifact Marking.

It is necessary to determine number of channels that will be included in classification. Based on fact that most «Blinks» appear in Fp1 sensor, neural network can be trained only on one sensor. There are two artifacts: Blink and global artifact. Output of neural network consists of three classes: «Blink», «Global artifact» and «Resting state».

Samples were formed based on Markers database. Blink artifacts are put randomly in samples (Fig. 4).

Raw data in dataset still has noise. To filter the signal Fast Fourier Transform was implemented. The result is showed in Fig. 5.

Based on the developed mining analysis scheme presented in Fig. 4, it is necessary to develop an environment for conducting experiments. The interaction of software is shown in Fig. 6. Number of neural network models were trained.

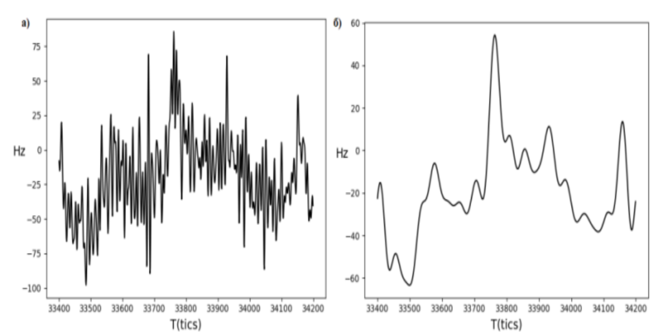


Fig. 5. Result of FFT: a) EEG Signal; b) EEG Signal with FFT.

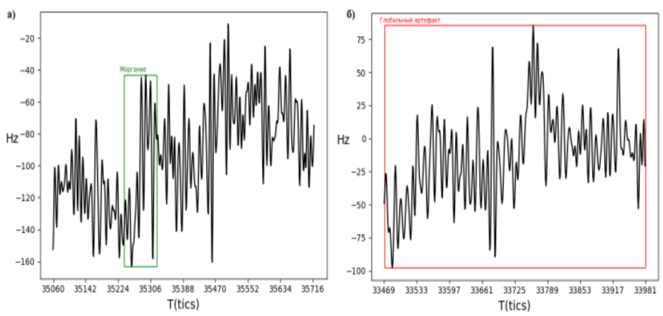


Fig. 6. Train Sample.

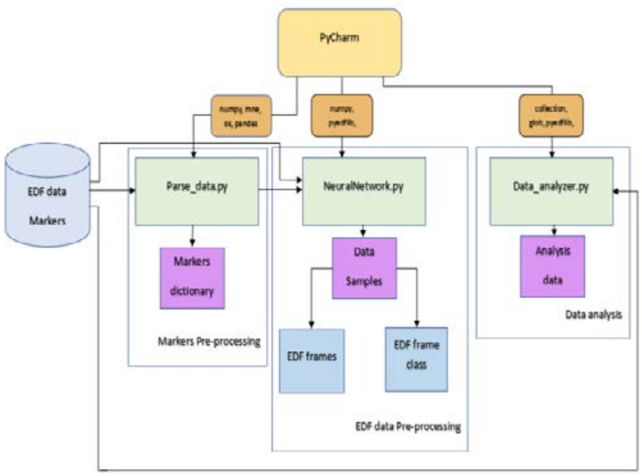


Fig. 7. Software Structure for the Analysis and Formation of Training Samples for a Neural Network Model.

In the PyCharm development environment, the main source code was developed to analyze and process the input values of the neural network model. The software tools interaction is shown in Fig. 7. The Parse_data.py library is used to convert Markers files into an associative array containing all the artifact information for each record in .edf files. The library contains the artifacts_supression method, which is used to translate the Position and Length format in seconds. In the read_markers_from_dir method, an associative array is formed from the specified directory using the pandas data analysis library containing the file name and its information about artifacts: the position of the artifacts, their description and length. This approach is used to obtain data in the function of generating training samples for a neural network.

The NeuralNetwork.py library allows creating samples for training a neural network based on arrays that are generated using the Parse_data.py library. The main method is prepare_data, which is based on information about artifacts, a database of EEG signals, used channels, and the size of the input window (in seconds) and a given ratio of samples with a normal state to samples with artifacts forms training samples for a neural network. Since the window size is larger than the maximum length of the Blinking artifact, this class is added to the selection completely. This takes into account the random shift of the artifact relative to the start of the sample. The Global artifact class is divided into several samples, from the beginning of the artifact to the sample that captures the end of the artifact and part of the signal without artifacts. In the process of recording samples with artifacts, the distance between them is calculated, and samples with the "normal state" class are taken, located between the artifacts. The Data_analyzer.py library contains the methods for analyzing the database described previously. An executable file "main.ipynb" was created in the Colaboratory environment, which contains the interactions of the libraries shown in Fig. 7, and also contains the architecture and process of training a neural network. The implementation scheme of an intelligent system for determining artifacts in an EEG signal is described in Fig. 8.

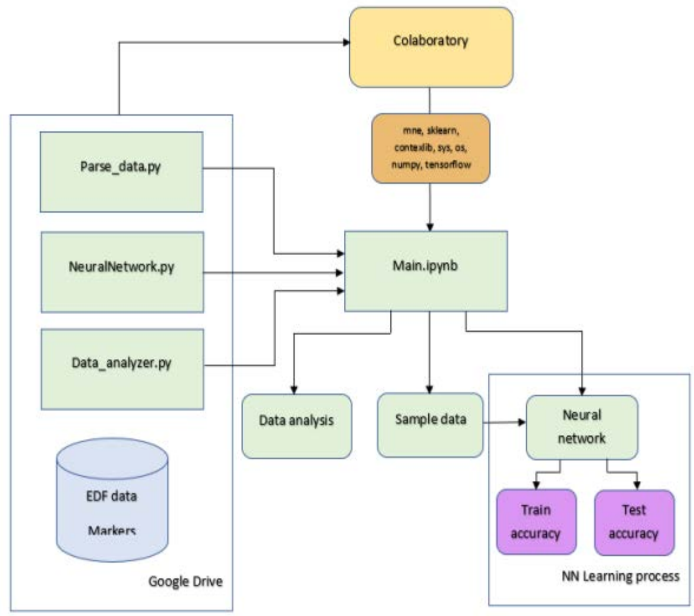


Fig. 8. Neural Network Training Scheme.

The neural network training scheme is an interaction of the libraries described earlier in the executing part of the Main.ipynb program. The executable file contains methods from the libraries for processing the database, conclusions of analytical data, sampling, the architecture of the neural network and its learning process located in the GoogleDrive cloud storage. The training process was conducted on the Colaboratory platform.

V. Experimental Selection of Neural Network Architecture

After analyzing the results of studies related to signal processing using neural networks, the architectures were selected based on convolutional and recurrent neural networks. Thus, 4 architectures were obtained:

- Batch_normalization + CNN + Dense using spectrograms;
- RNN (LSTM) + CNN + RNN (LSTM) + Dense;
- Batch_normalization + CNN + Dense;
- LSTM + NN based on "U-net"

1) Batch_normalization + CNN + Dense using spectrograms: The signal was converted to a spectrogram, a corresponding function was created using the fast Fourier transform (performed using the spectrogram method of the Scipy library) (Fig. 9).

A neural network model was applied to this type of data (Fig. 10), which is based on the convolutional neural network (CNN) [17]. The architecture was selected experimentally. It is the input data that comes to the normalization layer (BatchNormalization) with the aim of uniform learning.

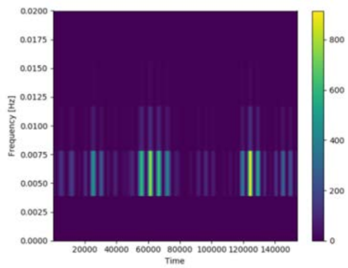


Fig. 9. Spectrogram of the EEG Sensor Signal.

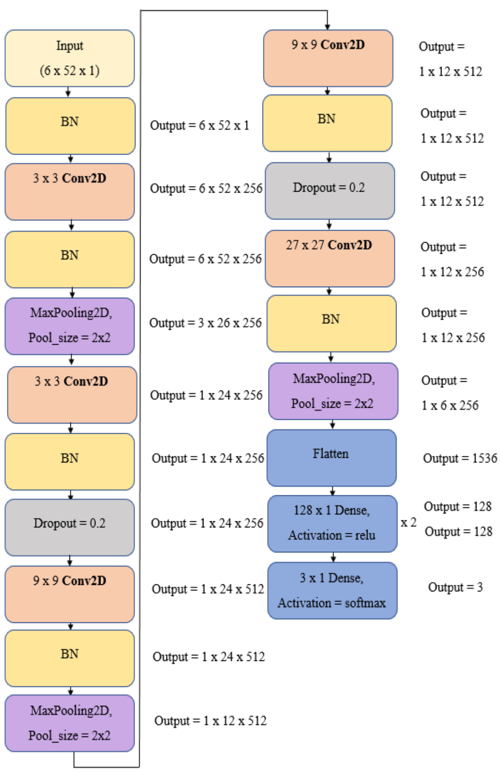


Fig. 10. Architecture of CNN.

BatchNormalization is a method for deep learning accelerating that solves a problem of learning efficiency. Normalization is implemented before every neural network layer [18]. Further, the convolutional neural network [19] receives normalized data at the input, and convolutional layers form 3x3 feature maps from it. During the experiment, it was revealed that a gradual twofold increase in the convolution core is two times more optimal for this architecture. With pooling (MaxPooling), the sample of the input space is being reduced by half (2x2), after that the "Dropout" layer is used to exclude a certain percentage of random neurons, since the neural network was overtrained during training. Then data is being converted to a one-dimensional vector using the Flatten layer. Classification is performed by the fully connected layer.

Based on the results obtained from experimental studies of the first model, several neural network models have been developed. The difference between the second and the third models (Fig. 11, 12) are that the data of the neural network model were presented in the form of a sequence, which were also converted using the fast Fourier transform.

2) RNN (LSTM) + CNN + RNN (LSTM) + Dense: A neural network model is shown on Fig. 11, the basis of which is a convolutional neural network (CNN) [20], that uses time convolutional layers (Conv1D). This layer creates a convolution core, which is convoluted with the input layer in one time dimension [21]. The architecture is as follows: in the second experiment, the input data comes to the recurrence layer (LSTM) with the maximum number of neurons, depending on the GPU capability, then the data goes to a time convolution layer in which a window of size 3 was specified empirically, after that the data is normalized by normalization layer (BatchNormalization). Based on the first experiment, the "Dropout" layer was applied, in which 20% of neurons are randomly turned off to exclude overtraining of the neural network. Then the data is transferred to the recurrent neural layer and converted into a one-dimensional vector using the "Flatten" layer. Classification is performed by the fully connected layer (Fig. 11).

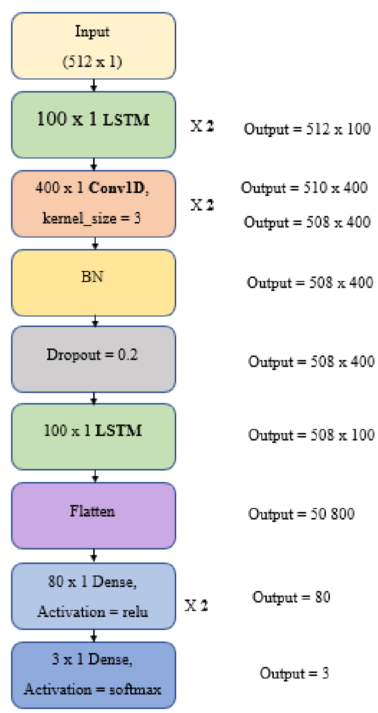


Fig. 11. Neural Network Architecture of the RNN (LSTM) + CNN + RNN (LSTM) + Dense Type using the Fourier Transform.

3) Batch_normalization + CNN + Dense: The difference between the architectures of the third model and the second model is that before the data is going to be transferred to the convolutional layer, it is being normalized. A normalization layer (BatchNormalization) was applied, before each time convolutional layer, then similarly, the data was converted into a one-dimensional vector, using the Flatten layer for fully connected layer and classification (Fig. 12).

4) LSTM + NN based on "U-net": fourth model was developed based on “Unet” (Fig. 13), using a time convolution, due to fact that database is small. The architecture of model 4 is shown in Fig. 14.

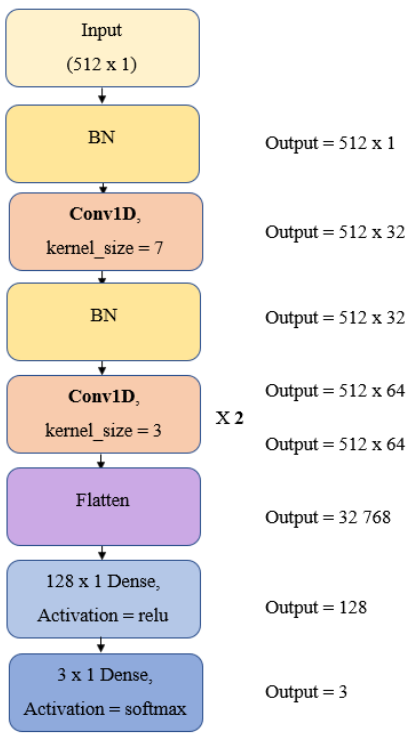


Fig. 12. Neural Network Architecture of Type Batch_Normalization + CNN + Dense using Fourier Transform.

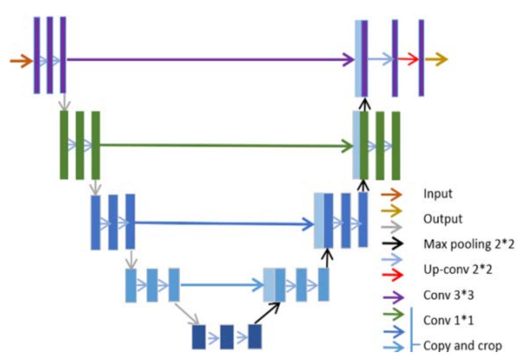


Fig. 13. The Architecture of the Neural Network “U-net”.

For signal preprocessing, the fast Fourier transform method was used. Also, to improve the quality of training, a function was created (blink_augmentation), where several positions are generated for each "Blink” artifact, where this artifact will be recorded. The result is number of "Blink" samples with the same artifact but the different location in samples.

During the study, the U-net architecture was used (Fig. 13), which consists of an encoder (narrowing part), a bottleneck and a decoder (expanding part). This architecture is used for the analysis of R-grams, MRI and other medical images.

The first part of U-net is the classical architecture of a classification convolutional neural network [22]. It consists of repeated applications of two convolutional layers, with a 3–3 kernel, followed by the ReLU activation function and the MaxPooling operation, which reduces the input representation by the maximum value in the window (poolsize, in this case the value is 2).

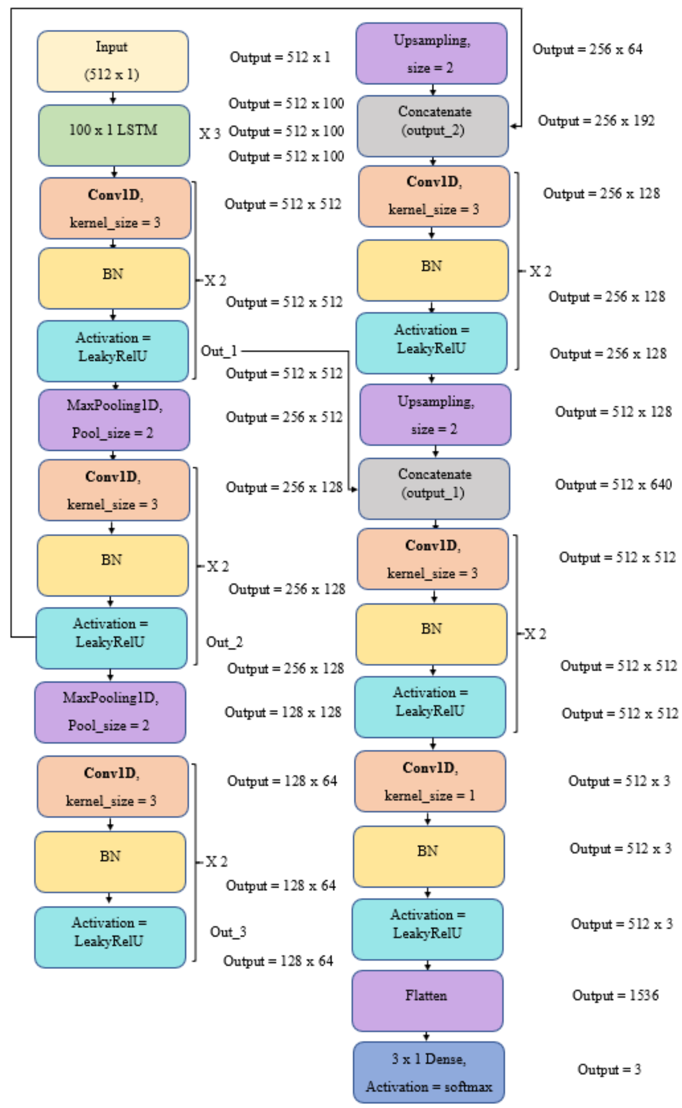


Fig. 14. The Architecture of the Neural Network Type LSTM + NN based on "U-net" using the Fourier Transform.

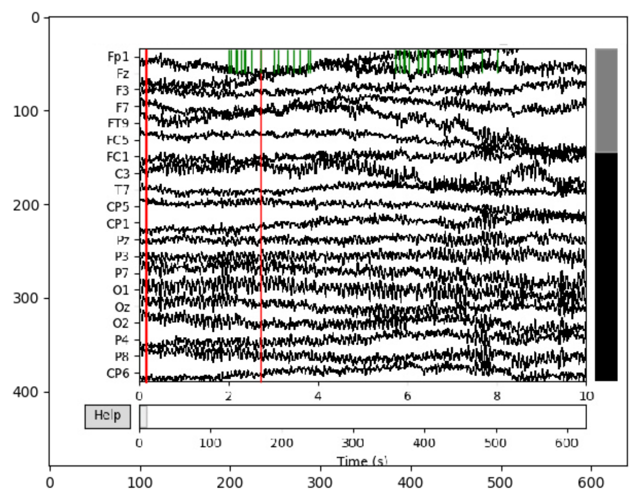
"Bottleneck" is a part of the network located between the contracting and expanding parts [23]. The second part consists of reverse convolution (deconvolution), which contains two convolutional layers with a 3–3 kernel and the Relu activation function, then concatenation is performed. At the last level, convolution 1x1 is used to match each vector with class attributes. Then the data is converted into a one-dimensional vector using the Flatten layer and the classification is performed by a fully connected layer [23].

In all experiments, the "fit" method was used for training. The number of samples, the gradient and the number of epochs for the model as well as compile method "Adam optimization function" and the error calculation function "categorical_crossentropy" were determined for each neural network model. It was revealed empirically that categorical_crossentropy is the most suitable error calculation function to optimize Adam parameters. To assess the quality of training, the Accuracy metric was chosen.

TABLE I. LEARNING OUTCOMES OF CLASSIFICATION MODELS WITH VARIOUS PARAMETERS

No.	Neural network architecture	Epochs	Batch_size	Accuracy	
				train	test
1	Batch_normalization + CNN + Dense with spectrogram	20	128	0.68	0.67
2	RNN (LSTM) + CNN + RNN (LSTM) + Dense	20	256	0.81	0.60
3	Batch_normalization + CNN + Dense	50	16	0.94	0.49
4	LSTM + NN based on U-Net model	10	300	0.70	0.70

A comparison of the results of an experimental study of four models is given in Table I. The training graph of neural network models was analyzed. The developed neural network based on the U-net architecture with recurrent layers demonstrates the best result of artifact recognition. 70% accuracy were acquired on test samples. Fig. 15 shows the results of automatic search for artifacts.



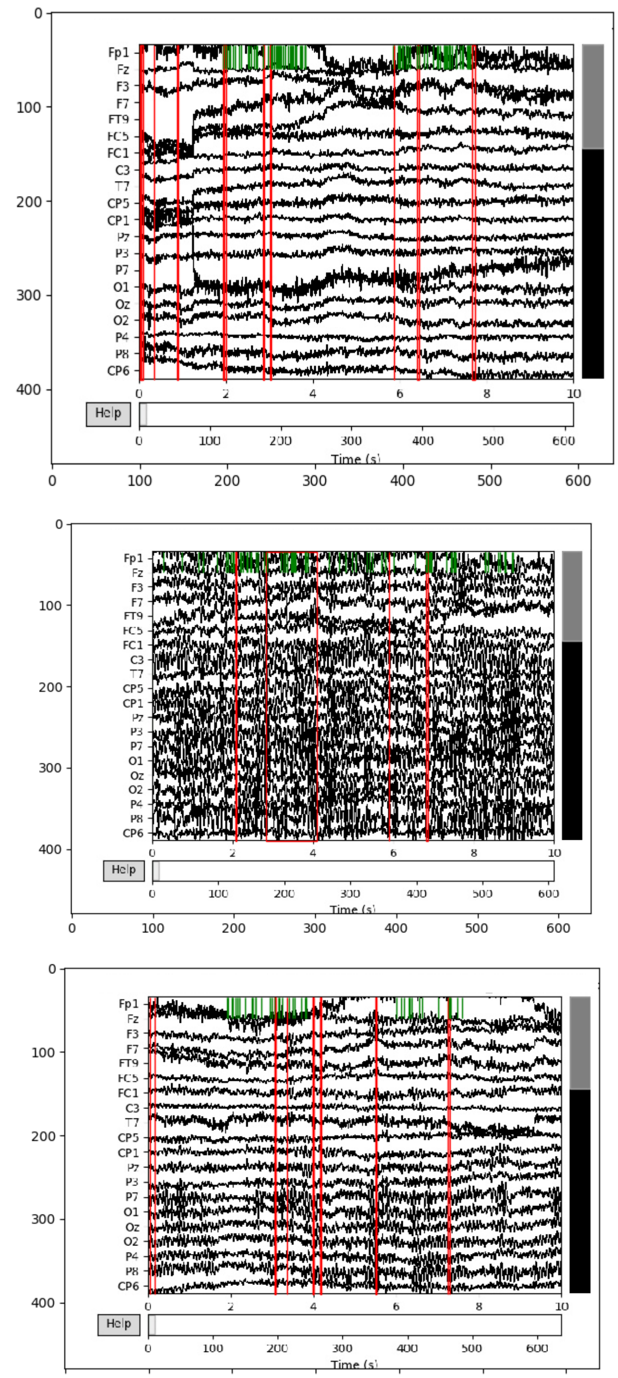


Fig. 15. The Result of Neural Network Activity. Examples of Highlighted Neural Network Artifacts: Blink Artifact is Green, Global Artifact Artifact is Red.

VI. CONCLUSION

A neural network model capable of recognizing artifacts in the process of recording EEG has been developed. Experimentally LSTM + U-net architecture was formed. To solve the problem, the U-net architecture, which is a two-dimensional convolution, was modified - a one-dimensional temporary convolution was used, the input of which received data from LSTM layers. Ensuring the required accuracy (70%)

is achieved due to the properties of the LSTM layers (trained to determine the signal state) and qualitative symmetric analysis (tension / compression) of the modified U-net layer.

Data analysis was carried out, in which the distance between artifacts in the signals, the maximum duration of each type of artifact was found. Using analytical functions, an optimal time window was allocated for artifact recognition, based on the maximum length of the "Blink" artifact. Since the data was manually filtered from the artifacts, and the database was small (9574 samples with artifacts), there was a problem with the quality of training of the neural network. The database was expanded using augmentation method, which partially influenced the learning process (28,722 samples with augmentation).

An analysis of existing architectures of neural networks, as well as an experiment with a training set was conducted. As a result, a neural network was developed based on recurrent neural network and U-net. The resulting neural network model is capable of detecting artifacts in the converted signal with an accuracy of 70%. The developed intelligent system can be used as an auxiliary tool for the analysis of the EEG signal.

During the study, the methods using libraries of applied software packages were developed for selecting an artificial neural network model of defects in digital signals, such as blinking artifacts in an EEG signal. Selected tools are able to create an environment for research and modeling of various signals.

The study showed the prospects for using the identified types of neural network architectures for analyzing EEG signals. The architectures selected during the study can be used in future studies aimed at modeling and clustering EEG signals.

REFERENCES

[1] J. N.Acharya, A. J. Hani, P. Thirumala, and T. N. Tsuchida, “American clinical neurophysiology society guideline 3: a proposal for standard montages to be used in clinical EEG,” The Neurodiagnostic Journal vol. 56, no. 4, pp. 253-260, 2016.

[2] A. S. Lundervold, A. Lundervold, “An overview of deep learning in medical imaging focusing on MRI,” Zeitschrift für Medizinische Physik, vol. 29, no. 2, pp. 102–127, 2019.

[3] K. Nathan, and J. L. Contreras-Vidal, “Negligible motion artifacts in scalp electroencephalography (EEG) during treadmill walking,” Frontiers in Human Neuroscience, vol. 9, p. 708, 2016.

[4] B. Singh, and H. Wagatsuma, “A removal of eye movement and blink artifacts from EEG data using morphological component analysis,” Computational and Mathematical Methods in Medicine, vol. 2017, p. 1861645, 2017.

[5] N. Gebodh, Z. Esmaeilpour, D. Adair, …, and M. Bikson, “Inherent physiological artifacts in EEG during tDCS,” Neuroimage, vol. 185, pp. 408–424, 2019.

[6] I. Obeid, and J. Picone, “The temple university hospital EEG data corpus,” Frontiers in Neuroscience, vol. 10, p. 196, 2016.

[7] M. Golmohammadi, … and J. Picone, “Automatic analysis of EEGs using big data and hybrid deep learning architectures,” Frontiers in human neuroscience, vol. 13, p. 76, 2019.

[8] X. Jiang, G. B. Bian, and Z. Tian, “Removal of artifacts from EEG signals: a review,” Sensors, vol. 19, no. 5, p. 987, 2019.

[9] B.A. Zaikin, anf A.F. Kotov, “An estimation of efficiency of filtering algorithms of state vector of small-sized observed object with non-Markovian approximation of trajectory,” Russian Technological Journal, vol. 9, no. 4, pp. 38-48, 2021. https://doi.org/10.32362/2500-316X-2021-9-4-38-48

[10] F. Lotte, L. Bougrain, …, and F. Yger, “A review of classification algorithms for EEG-based brain–computer interfaces: a 10-year update,” Journal of Neural Engineering, vol. 15, no. 3, p. 031005, 2018.

[11] A. Kilicarslan, and J. L. C. Vidal, “Characterization and real-time removal of motion artifacts from EEG signals,” Journal of Neural Engineering, vol. 16, no. 5, p. 056027, 2019.

[12] Z. Tayeb, J. Fedjaev, …, and J. Conradt, “Validating deep neural networks for online decoding of motor imagery movements from EEG signals,” Sensors, vol. 19, no. 1, p. 210, 2019.

[13] R. Hussein, H. Palangi, R. K. Ward, and Z. J. Wang, “Optimized deep neural network architecture for robust detection of epileptic seizures using EEG signals,” Clinical Neurophysiology, vol. 130, no. 1, pp. 25–37, 2019.

[14] N. N. Astakhova, L. A. Demidova, and E. V. Nikulchev, “Forecasting method for grouped time series with the use of k-means algorithm,” Applied Mathematical Sciences, vol. 9, no. 97, pp. 4813-4830, 2015.

[15] W. Chen, Y. You, Y. Jiang, M. Li, and T. Zhang, “Ensemble deep learning for automated visual classification using EEG signals,” Pattern Recognition, vol. 102, p. 107147, 2020.

[16] C. Holdgraf, S. Appelhoff, and S. Bickel, “iEEG-BIDS, extending the Brain Imaging Data Structure specification to human intracranial electrophysiology,” Scientific Data, vol. 6, p. 102, 2019.

[17] W. Liu, Y. Wen, Z. Yu, and M. Yang, “Large-margin softmax loss for convolutional neural networks,” In Proceedings of The 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research (PMLR, vol. 48), pp. 507–516, 2016.

[18] K. Ramasubramanian, and A. Singh, “Deep learning using keras and tensorflow. In: Machine Learning Using R. Berkeley: Apress, pp. 667-688, 2019.

[19] R. T. Schirrmeister, J. T. Springenberg,. …, and T. Ball, “Deep learning with convolutional neural networks for EEG decoding and visualization,” Human Brain Mapping, vol. 38, no. 11, pp. 5391–5420, 2017.

[20] A. Saxena, “Convolutional neural networks: an illustration in TensorFlow. XRDS: Crossroads,” The ACM Magazine for Students, vol. 22, no. 4, pp. 56–58, 2016.

[21] M. Wang, J. Hu, and H. Abbass, “Stable EEG Biometrics Using Convolutional Neural Networks and Functional Connectivity,” Australian Journal of Intelligent Information Processing Systems, vol. 15, no. 3, pp. 19–26, 2019.

[22] G. Xu,; X. Shen, S. Chen, “A Deep Transfer Convolutional Neural Network Framework for EEG Signal Classification,” IEEE Access, vol. 7, pp. 112767–112776, 2019.

[23] Y. Weng, T. Zhou, Y. Li, and X. Qiu, “NAS-Unet: Neural architecture search for medical image segmentation,” IEEE Access, vol. 7, pp. 44247–44257, 2019.