

DOI: 10.35784/IAPGOS.116

ATRIAL FIBRILLATION DETECTION ON ELECTROCARDIOGRAMS WITH CONVOLUTIONAL NEURAL NETWORKS

Viktor Kifer¹, Natalia Zagorodna¹, Olena Hevko²¹Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Ternopil, Ukraine²Ternopil Ivan Puluj National Technical University, Faculty of Applied Information Technologies and Electrical Engineering, Ternopil, Ukraine

Abstract. In this paper, we present our research which confirms the suitability of the convolutional neural network usage for the classification of single-lead ECG recordings. The proposed method was designed for classifying normal sinus rhythm, atrial fibrillation (AF), non-AF related other abnormal heart rhythms and noisy signals. The method combines manually selected features with the features learned by the deep neural network. The Physionet Challenge 2017 dataset of over 8500 ECG recordings was used for the model training and validation. The trained model reaches an average F1-score 0.71 in classifying normal sinus rhythm, AF and other rhythms respectively.

Keywords: electrocardiography, machine learning, neural networks

DETEKCJA MIGOTANIA PRZEDSIONKÓW NA ELEKTROKARDIOGRAMACH Z WYKORZYSTANIEM KONWOLUCYJNEJ SIECI NEURONOWEJ

Streszczenie. W tej pracy, przedstawiamy nasze badania, które potwierdzają przydatność zastosowania konwulucyjnej sieci neuronowej dla klasyfikacji zapisów jedno-odprowadzeniowego EKG. (tak brzmi ta nazwa). Proponowana metoda została zaprojektowana dla klasyfikowania prawidłowego rytmu zatokowego, migotania przedsionków (AF), poza-AF powiązanych z innymi nieprawidłowymi rytмами serca i zaszumionymi (głośnymi?) sygnałami. Ta metoda łączy cechy wyselekcjonowane ręcznie z cechami wyuczonymi przez głębokie sieci neuronowe. Zbiór danych Physionet Challenge 2017 zawierający ponad 8500 zapisów EKG został zastosowany dla modelu szkolenia oraz walidacji. Model nauczony (wyszkolony?) osiąga odpowiednio średni F1-wynik 0.71 w klasyfikowaniu prawidłowego rytmu zatokowego, rytmu AF oraz innych rytów.

Słowa kluczowe: elektrokardiografia, nauczanie maszynowe, sieci neuronowe

Introduction

Heart diseases are the leading cause of death in the EU and the US for both men and women. The fact that around 47% of sudden cardiac deaths occur outside a hospital suggests that many people with heart disease do not react on early warning signs.

An electrocardiogram (ECG) can be used for diagnostics of many heart malfunctions like the presence of any damage to the heart's muscle cells or conduction system. Many handheld ECG recorders were developed to produce a single-lead ECG. WIWE [23], AliveCor [1], OMRON HCG801 HeartScan [22] are the examples.

Automated ECG analysis is a complex problem, which includes a number of basic tasks such as noise reduction, QRS complex detection, P and T wave detection, analysis of the shapes of the waves and their positions and lengths. Each of those tasks is a separate topic of scientific research. For example, noise reduction is a well-explored task with a lot of developed techniques and methods, including band-pass filters [5], spectrograms [6], Fourier [15] and wavelet [8] transformations. QRS detection is also represented with a large list of methods reaching sensitivities and specificities of about 99.5. The detailed comparison of the QRS detection methods is done in [3].

Atrial Fibrillation (AF) is a common type of heart disease that leads to stroke, heart failure or other complications. Every year millions of people get affected by AF, and this number tends to increase. The percentage of people with AF also increases with age: from 0.14% under 50 years old to over 14% for those over 80 years old. And AF is often an episodic event, which makes early diagnostics even more complicated.

AF detection methods can belong to one of two categories: atrial activity analysis-based methods, which search for the absence of P waves or the presence of F-waves in the TQ interval; and ventricular response analysis based on the predictability of the inter-beat timing ('RR intervals').

Park *et al* [9] analyzed heartbeat variability from inter-beat intervals obtained by a wavelet-detector. The features are extracted from the Poincare plot of the intervals. The method performs a binary classification of the signal into AF/non-AF.

Rodenas-Garcia *et al* [11] incorporated wavelet entropy in ECG analysis. The signal is analyzed in chunks of 10 consequent

noise-free heartbeats, for which a median of TQ segment is extracted and the wavelet entropy is computed. The method can be used to detect common cardiac arrhythmias and atrial fibrillation.

Petrenas *et al* [10] proposed a simple method for long-term AF monitoring. The signal is processing with the 8-beat sliding window in order to reduce computational power. The building blocks of the algorithm are the estimation of RR trend and irregularity, ectopic beats filtering, bigeminy suppression and fusion of the signals. The model uses binary AF vs non-AF classification.

Tziakouri *et al* [13] built a 3-stage model for multi-label ECG classification. The first stage analyzes the quality of the signal and rejects noisy signals. In the second stage, the signal is classified into normal sinus rhythm and abnormality. And on the third stage, atrial fibrillation is separated from the other abnormal rhythms.

The previously developed methods are usually limited in applicability due to following reasons: 1) only normal and AF recording were considered, while all the other diseases were discarded; 2) good performance was shown of carefully-selected often clean data; 3) test dataset was not used or was too small for making a conclusion; 4) a small number of patients was used.

The aim of this research is to develop an atrial fibrillation detection model, which would not require complex and long-running computations for feature extraction. The input of our model is a raw ECG signal. The developed algorithm runs initially basic signal filtering and splits the signal into separate heartbeats. Then the sequence of heartbeats is used for heart disease prediction. Our model is also applicable for noisy signals with significant signal-to-noise ratio, because it involves filtering preprocessing algorithms. The model can also distinguish between atrial fibrillation and other abnormal rhythms.

1. Methodology

Our method is based on the usage of state-of-the-art ECG processing techniques in noise reduction and R-peaks combined with deep neural networks. Convolutional neural network layers are used as feature extractors over a 2D matrix of the signal heartbeats. The classification decision is done using fully connected network layers.

1.1. Dataset overview and preprocessing

The dataset used for this project was taken from PhysioNet/Computing in Cardiology Challenge 2017 [2]. It contains 8528 single-lead ECG recordings, sampled at 300Hz and varying lengths from 2700 to 18300 points. The records are labeled with 4 classes: normal sinus rhythm (N, 5154 records), atrial fibrillation (A, 771 records), other abnormal rhythms (O, 2557 records) and noise (originally marked with tilde in the dataset, replaced with P, 46 records).

The first and the most important step of the analysis is the visualization of the data. We plotted some samples from each class and analyze them visually. This helps to identify potential problems with the dataset as well as gives insights on how to create a set of relevant features to represent the data.

We paid more attention to the analysis of the class “Other” as it includes a wide range of heart diseases. We tried to identify those diseases and find out the most common ones. Some of them are 1) atrial flutter; 2) ventricular fibrillation; 3) tachycardia (more than 100 bpm); 4) bradycardia (less than 60 bpm); 5) wide QRS complex (more than 0.1 s – prolonged, more than 0.12 s – abnormal); 6) presence of extrasystole; 7) long PR interval (more than 0.2 s).

The review of the dataset revealed underlying problems, including the following: 1) some of the records were inverted, probably due to electrode misplacement; 2) values of the records are absolute and not restricted to any range; 3) imbalanced representation of the classes.

Our preprocessing steps include:

Noise reduction

The ECG signals are recorded using single-lead, non-medical equipment and usually by patients themselves. It means they are often noisy. The most common sources of noise are:

- electrode contact noise (frequency is about 1 Hz) caused by improper contact between the body and electrodes;
- motion artifacts produced by patient’s movements which affect electrode-skin impedance, resulting in usually short-term distortions;
- muscle contractions - noise with 10% of regular peak-to-peak ECG amplitude and frequency up to 10 kHz;
- baseline wander caused by a respiratory activity with a 0–0.5 Hz frequency.

The goal of this process is to remove noises while keeping as much of the signal as possible.

The most important cardiac information in ECG is stored within a frequency range of 0–20 Hz [14]. We apply both high pass (HPF) and low pass (LPF) FIR filters with the order of 100 (1/3 of the sampling frequency). For HPF cut-off frequency of 0.5 Hz was used to remove the baseline movement of the signal. And LPF was set with a cut-off frequency of 40 Hz to reduce the impact of other noise sources on the ECG.

Signal normalization

The ECG recorder might generate a wide range of values depending on different conditions when the recording was performed (Fig. 1). Also, manufacturers might provide their own value ranges for different recorders.

Neural networks can be considered as a function (usually non-linear) of input values multiplied by weights and combined with a bias. The weights and the bias are estimated during the model training. So, if the range of the input values changes, then the output of the function might also change. To avoid such cases, we applied a standard data normalization technique in the field of neural networks: to scale signal values to a specific range. The values of every record were converted to the range of [-1;1] by dividing original values by the absolute value of the signal.

Lead inversion detection

Inverted ECG signals happen in around 4% of the recordings even during a professional medical examination. Inverted records (for example, A06667, A08477) are more likely to be classified as abnormal rhythm due to changes in morphologies of the QRS and P and T waves.

We used a simple principle of comparing the absolute values of the minimum (abs_min) and the maximum (abs_max) in the QRS complex of the filtered signal.

We applied the lead inversion detection after running low pass and high pass filter. It is important as it removes most of the noise in the ECG signal and reduces misleading high peaks in the data, leaving primarily clear QRS complexes. For those cases, when the abs_max is at least less than 0.6 of abs_min, we considered such a record as inverted and performed the sign change for the values of the record. The algorithm detected and worked out the 734 inverted records in our dataset.

Handling of the imbalanced dataset problem

The dataset is highly unbalanced, which is common for natural signals, but it complicates the training process. While the model should learn the natural distribution of the classes (eg, most people would have normal sinus rhythm, and less would have some disease), the actual goal of the model is disease detection.

Normal sinus rhythm is more than 100 times more represented in the dataset, than noise and about 18 times more – than AF. The model might easily overfit and predict the most represented class. Severe overfitting was observed in early experiments.

There are several known approaches to handle the problem, including the usage of abnormalities detection algorithms. These algorithms apply class weights (penalties) on classifiers and perform data balancing. Another approach is to use the models which are better at handling unbalanced data. The experiments with class weights did not show any significant improvement in classification results. Therefore, data balancing techniques were applied to the dataset, including oversampling and undersampling [4].

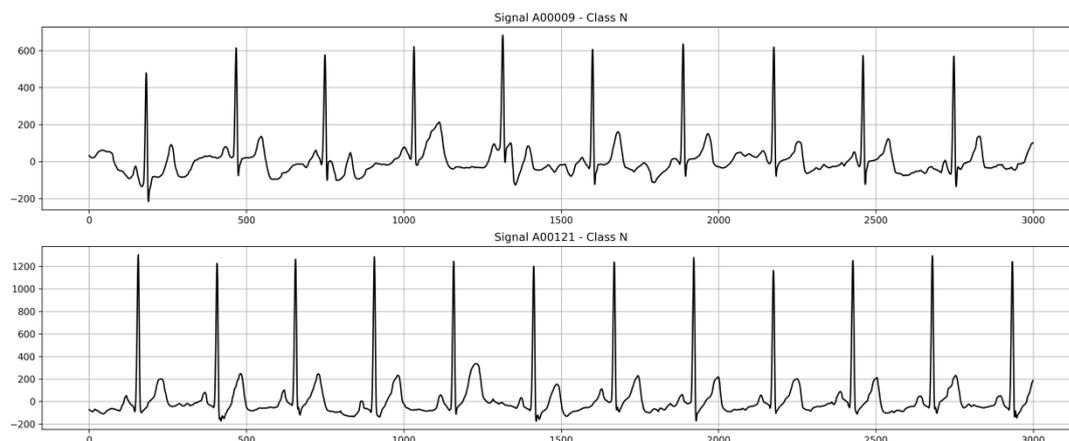


Fig. 1. The difference in value ranges for normal ECG signals (records A00009 and A00121, only 10 seconds of each record)

Data oversampling is used when the amount of collected data is insufficient. The technique is applied to the underrepresented minority classes in the dataset. The main idea is to create synthetic samples of the data based on the characteristics of existing samples. The signals were interpolated in time (either making them shorter or longer) by 10 to 25% (chosen randomly). The generated signals might have a very low or very high heart rate, as we do not take into account the heart rate of the original signal.

If a class of data is the overrepresented majority class, undersampling may be used to balance it with the minority class. This operation took the instances of the majority class and compared the time-shifted filtered signals by the square root of differences in the developed method. About 10% of the records with the lowest differences were removed from the training samples.

1.2. Heartbeat detection

The detection of the heartbeat starts with the search of R-picks, the highest and the most resistant to the noise component of a single heartbeat. This is a well-known problem, and one of the most commonly used methods is the Pan-Tompkins algorithm [7]. The original paper states the detection rate of 99.3%. Our algorithm is a modification of the original one. It runs the Pan-Tompkins algorithm. Then it runs backward and looks for intervals where the RR interval length is larger than 1.5 of the mean RR interval length and reruns R-peak detection with the lower threshold.

After R-picks detection, we started the lookup of the Q and S positions, which are the local minimums on both sides of the R pick. After that, the algorithm also looks for the local maximums

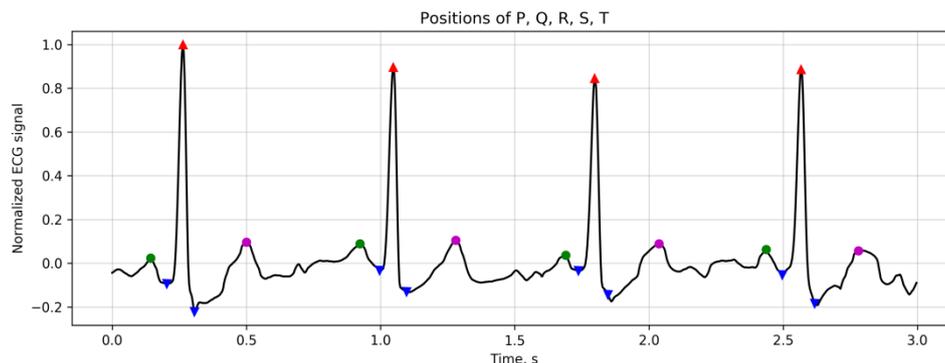


Fig. 2. Detection of heartbeat components: QRS points marked with triangles, maximums of P and T waves with circles

2. Neural network topology

We propose to use a convolutional neural network for ECG classification. Our model was particularly inspired by the VGG-16 neural network [12], which is commonly used for image recognition problems. We reduced the number of layers to 6, which includes two groups of 2 convolutional layers and a group of dense layers.

The input dimension for the neural network was set to 160 by 180, where 160 represents the heartbeats, and 180 is the number of data points per each heartbeat (0.6 s).

The input is supplied into a convolutional layer with 32 filters and a convolution window length 3. It is followed by the rectified linear unit activation (ReLU), a simple non-linear function that replaces all negative values in the activation map by zero. Then it goes the same convolutional layer followed by a max-pooling layer with pool size set to 2 and the downscaling factor of 2. Pooling helps resolve multiple problems related to neural networks: makes input representations smaller, reduces the number of parameters (which makes computations faster and controls overfitting), makes network resistant to small transformations of the input, helps to detect features in data no matter where they are located. Then another ReLU layer follows.

in the interval of 0.2 s before R pick and 0.4s after R pick. Those points are assumed to be positions of the maximum of P and T waves respectively. These important points are marked in Fig. 2.

Based on the positions of R-peaks, we extracted the heartbeats from the filtered signal, using the static window of 0.6 s (Fig. 3). We ignored the heartbeats, where the window cannot be applied (heartbeats on start or end of the signal).

The extracted heartbeat had 180 points. We constructed a matrix filled with all detected heartbeats. Input for the neural network should have the same number of templates for every record, therefore an upper limit of 160 heartbeats was set. In case, there were no enough heartbeats, we added empty heartbeats (zero-values) to the end of the matrix.

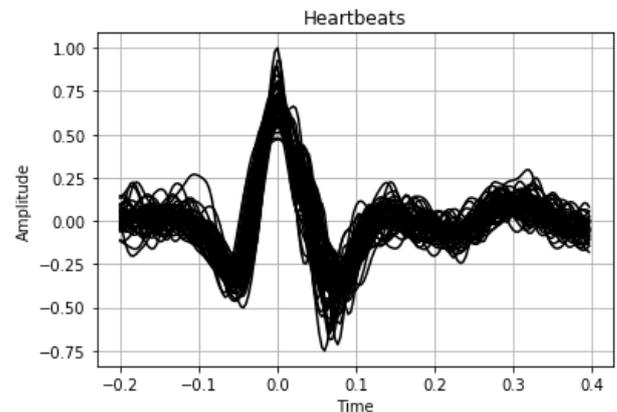


Fig. 3. Extracted normalized heartbeats from the ECG signal

MaxPooling was performed before ReLU activation to reduce the number of operations and fasten the training process. These layers create the first block of our neural network. It is finalized with the drop out layer with a drop rate of 0.25, which has a slight regularization effect and reduces overfitting. The second block is built using the same layers structure as the previous one, but we increased the number of filters in convolutional layers to 64. The architecture of the proposed neural network is given in Fig.4.

We transform the 2D output of the second block into 1D and supply it into a fully connected neural network (third block). This block contains a fully connected layer of 512 units and ReLU activation. It is followed by the drop out layer with a rate of 0.5.

The last layer of the neural network produces the model output. It should have the same number of neurons as the number of possible output labels. Therefore, the neural network ends with a fully connected layer of 4 units. The layer uses softmax activation, which normalizes the layer outputs sum to 1 (eg, each value is a probability of the input to be an instance of a given class).

The network was optimized using Stochastic Gradient Descent with Nesterov momentum of 0.9 and the initial learning rate set to 0.01.

We used categorical cross-entropy as a loss function for the neural network.

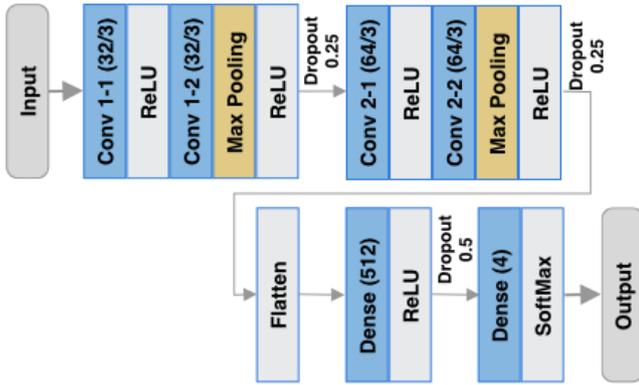


Fig. 4. The architecture of the neural network for ECG classification

1.3. Implementation notes and training procedure

We used Python [19] with Numpy [17] and SciPy [20] for signal processing, Scikit-Learn [18] and Keras [16] with Tensorflow CPU [21] backend for neural network construction and training.

We split the original dataset into training and validation sets, 2/3 and 1/3 respectively. Then we applied over- and undersampling on the training dataset. We used SciPy to resample the signals.

The network was implemented using Keras functional API. Convolutional layers were implemented using *Conv1D* and max-pooling was implemented using *MaxPooling1D* layers.

For neural networks, the categorical value of the output should be encoded with the one-hot encoding. This means that class 0 becomes a vector [1 0 0 0], class 1 becomes [0 1 0 0] and so on. We did that using Keras *to_categorical* utility function.

The neural network was trained in batches with 32 batch size. The training process was initialized with 30 training epochs and a controlling algorithm to stop training after reaching a minimum of the validation loss function. The training stop function was implemented using Keras *EarlyStopping* callback, with the patience parameter set to 7.

We also run a dynamic learning rate algorithm, which reduced the initial learning rate by a factor of 10, when the validation loss function did not improve during the last 3 epochs. The improvement delta was set to 10^{-4} , and the learning rate can be reduced from the initial value of 0.01 to 10^{-6} . It was implemented using *ReduceLROnPlateau* Keras callback.

3. Evaluation

Evaluation of the model is performed using F1 score which is a combination of precision (PPV) and recall (TPR).

$$F1_{class} = \frac{2 * PPV_{class} * TPR_{class}}{PPV_{class} + TPR_{class}} \quad (1)$$

The class detection accuracy of the model is a proportion of the number of true positive (TP) cases to the sum of true positive and false positive (FP) cases.

$$PPV_{class} = \frac{TP}{TP + FP} \quad (2)$$

The class recall of the model is a ratio of the number of true positive to the sum of true positive and false negative.

$$TPR_{class} = \frac{TP}{TP + FN} \quad (3)$$

For multi-class classification, we compute the F1 score for each class and the final score of the model is an average of all scores. But the distribution of normal, atrial fibrillation, other rhythms, and noisy data is highly imbalanced, and the amount of noisy data is significantly less compared to other classes, so the noise class evaluation was excluded from the final score. The F1 score of the model is an average of the scores for AF, N and O classes.

$$F1 = \frac{F1_{AF} + F1_N + F1_O}{3} \quad (4)$$

F1 score helps us to prevent an accuracy paradox, which is very common for the imbalanced datasets.

We also incorporated cross-validation technique, which helps to estimate how accurately a predictive model will perform in practice. It runs in rounds and each round involves partitioning of a dataset into training and test sets, fitting the training data into the model, and validating the classifier on the test data. 5-fold cross-validation was used for this project. This method splits the dataset into 5 subsets, then one subset is used for testing and 4 subsets for training. The parameter 5 not only defines the number of subsets, but also the number of rounds. A new subset is used for testing in each round. This means that during 5 rounds each of the generated subsets was used only once as a testing dataset.

4. Results

The model was validated both locally on the piece of the original dataset and remotely on the PhysioNet challenge hidden dataset.

Table 1 shows the detailed model performance report we generated based on the local test dataset (33% of the original dataset).

Table 1. Results of ECG classification using Convolutional Neural Network

| Class | Precision | Recall | F1 | Support |
|--------------|-----------|--------|------|---------|
| AF | 0.62 | 0.64 | 0.63 | 183 |
| Normal | 0.78 | 0.92 | 0.84 | 1273 |
| Other rhythm | 0.70 | 0.45 | 0.55 | 608 |
| Noise | 0.61 | 0.59 | 0.60 | 68 |
| Total | 0.68 | 0.65 | 0.66 | 2132 |

As it was mentioned earlier, we also validated the model using 5-fold cross-validation. The model got an average mean F1 score of 0.68 and a standard deviation of 0.016.

Next, we sent the model for evaluation on the hidden dataset. PhysioNet runs the model in the virtual environment and computes the F1 score of the model for classes Normal, AF and Other. The results of the model on the hidden dataset are shown in Table 2.

The final score of the model on the hidden dataset is 0.71.

Table 2. PhysioNet report on the model performance

| Class | Normal | AF | Other | Overall |
|-------|--------|------|-------|---------|
| F1 | 0.88 | 0.69 | 0.56 | 0.71 |

5. Conclusions

In this paper, we proposed an approach for the automatic detection of atrial fibrillation in single-lead ECGs using a convolutional neural network. Our model is designed to handle noisy signals and distinguish atrial fibrillation from other abnormal heart rhythms. The computational efficiency of the model is reached by using only basic signal preprocessing techniques like filtering and heartbeats detection.

The performance of the model was validated using 5-fold cross-validation, which showed an average F1 score on the test dataset equals 0.68 with a little variance of 0.016. The model accuracy was additionally confirmed by running it on a closed dataset of ECG signals from the PhysioNet website, where the model showed the overall result of 0.71.

Our result is lower compared to the results of the models which include manual feature extraction or a combination of multiple estimators. However, our model uses less computational power and it can be used on devices with limited resources.

There are several directions for future research. In particular, the signal resampling algorithm, used during model training, can be improved by taking the actual heart rate into account.

We also plan to improve the model performance (without reducing efficiency) by adding some of the computationally lightweight expert features like heart rate variability.

References

- [1] AliveCor ECG recording device. <https://www.alivecor.com>
- [2] Clifford G, Liu C, Moody B, Silva I, Li Q, Johnson A, Mark R.: AF classification from a short single lead ECG recording: the PhysioNet Computing in Cardiology challenge 2017. *Computing in Cardiology* 44, 2017, [DOI: 10.22489/CinC.2017.065-469].
- [3] Dilaveris P. E., Kennedy H. L.: Silent atrial fibrillation: epidemiology, diagnosis, and clinical impact. *Clinical Cardiology* 40(6), 413–418, 2017.
- [4] Hernandez J., Carrasco-Ochoa J. A., Martínez-Trinidad J. F.: An Empirical Study of Oversampling and Undersampling for Instance Selection Methods on Imbalance Datasets. In: Ruiz-Shulcloper J., Sanniti di Baja G. (eds): *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2013. Lecture Notes in Computer Science*, vol 8258. Springer, Berlin, Heidelberg 2013, [DOI: 10.1007/978-3-642-41822-8_33]
- [5] Himanshu S., Kumar J. S. J, Ashok V., Juliet A. V.: Advanced ECG Signal Processing using Virtual Instrument. *International Journal on Recent Trends in Engineering & Technology* 3(2), 2010, 111-114.
- [6] Huang J., Chen B., Yao B., He W. ECG Arrhythmia Classification Using STFT-based Spectrogram and Convolutional Neural Network. *EEE Access* 7, 2019, 92871-92880.
- [7] Kohler B.-U., Hennig C., Orglmeister R.: The principles of software QRS Detection. *IEEE Engineering in Medicine and Biology Magazine* 21(1), 2002, 42-57, [DOI: 10.1109/51.993193].
- [8] Mikhled A., Daqrouq K.: ECG Signal Denoising by Wavelet Transform Thresholding. *American Journal of Applied Sciences* 5(3), 2008. 276-281.
- [9] Park J., Lee S., Jeon M.: Atrial fibrillation detection by heart rate variability in Poincare plot. *Biomed engineering online* 8/38, 2009, 1-12.
- [10] Petrenas A., Marozas V.: Low-complexity detection of atrial fibrillation in continuous long-term monitoring. *Comput in Biology and Medicine* 65, 2015, 184-191.
- [11] Rodenas-Garcia J., Garica M., Alcaraz R., Rieta J.: Wavelet Entropy Automatically Detects Episodes of Atrial Fibrillation from Single-Lead Electrocardiograms. *Entropy* 17, 2015, 6179-6199, [DOI: 10.3390/e17096179].
- [12] Simonyan K., Zisserman A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. *Inter Conf on Learning Representations (ICLR)*, 2015. [arXiv preprint arXiv:1409.1556].
- [13] Tziakouri M., Pitris C., Orphanidou C.: Classification of AF and Other Arrhythmias from a Short Segment of ECG Using Dynamic Time Warping. *Comp in Cardio*, 2017, 1-4.
- [14] Velayudhan A., Peter S.: Noise Analysis and Different Denoising Techniques of ECG Signal – A Survey. *IOSR Journal of Electronics and Communication Engineering*, 2016, 40-44
- [15] Wang Z., Wan F., Wong C.M., Zhang L.: Adaptive Fourier decomposition based ECG denoising. *Computers in Biology and Medicine* 77, 2016, 195–205. <https://keras.io> Keras documentation (available: 19.07.2019).
- [16] <https://numpy.org> NumPy official documentation (available 01.07.2019).
- [17] <https://scikit-learn.org> Scikit-Learn official website (available: 01.07.2019).
- [18] <https://www.python.org> Python programming language (available 30.06.2019).
- [19] <https://www.scipy.org> SciPy official documentation (available 01.07.2019).
- [20] <https://www.tensorflow.org> Tensorflow official page (available: 19.07.2019).
- [21] OMRON HCG801 HearnScan ECG recorder. <https://www.omron-healthcare.com/en/products/electrocardiograph>
- [22] WIWE ECG recording device. <https://shop.mywiwe.com/en/ecg-recording-105>

M.Sc. Viktor Kifer

e-mail: kifervictor@gmail.com

Received a master's degree in Software Engineering from Ternopil Ivan Puluj National Technical University in 2018. Currently, he is a Ph.D. student working on biomedical signal processing.



ORCID ID: 0000-0002-0621-9121

Ph.D. Natalia Zagrodna

e-mail: zagrodna.n@gmail.com

In 2003, graduated in Information Control Systems from Ternopil Ivan Puluj State Technical University. From 2007, PhD in Technical Sciences. The Ph.D. thesis was dealt with mathematic modelling, analysis and prognosis of gas consumption. Current position – Head of Cybersecurity Department.



ORCID ID: 0000-0002-1808-835X

Ph.D. Olena Hevko

e-mail: gevkojenka@gmail.com

In 1995, graduated in medicine from Ivano-Frankivsk State Medical Academy. Obtained a Ph.D. in Internal medicine in 2003. In 2008, received an academic title of assistant professor of the Department of Biotechnical Systems.



ORCID ID: 0000-0003-1427-1699

otrzymano/received: 09.09.2019

przyjęto do druku/accepted: 06.12.2019