ALFA BK UNIVERSITY
FACULTY OF INFORMATION TECHNOLOGIES
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE
International Scientific Conference

Alfa BK Univerzitet

UDK: 004.85
159.938.363.1-047.72
DOI:10.46793/ALFATECHproc25.154T
Original scientific paper

# Using machine learning techniques for age prediction based on PPG signal analysis

*Mirjana Tomic[1]; Stevan Jokic[2];Ivan Jokić[3];Nenad Gligorić[4], Ana Kovačević[5],Branislav Gerazov[6]*

**Abstract:** This paper explores the application of machine learning and neural networks for age prediction based on PPG signals (photoplethysmographic signals), which provide a non-invasive and cost-effective method for assessing patient health, particularly in the field of cardiovascular diseases. PPG signals, recorded using light sources and photodetectors, enable the assessment of changes in blood volume in the microvasculature, which is closely related to the condition of blood vessels. In this study, a machine learning model based on neural networks has been developed, using PPG signals as input data to predict patient age.
Various neural network architectures were tested, including models with one hidden layer and models with multiple layers, to investigate how the number of layers affects prediction accuracy. Additionally, different activation functions, such as tanh and ReLU, as well as various data preprocessing techniques, such as normalizing PPG signals, were considered. The model evaluation was carried out using MAE (Mean Absolute Error) and MSE (Mean Squared Error) as key statistical indicators measuring prediction accuracy.
The results show that models with a greater number of hidden layers achieve better performance in age prediction, with a 30% reduction in errors compared to models with one hidden layer. Errors were primarily caused by data imbalance and specific signal characteristics that were not correctly identified by the model. The causes of larger prediction errors were also analyzed, revealing that certain PPG signals exhibited features resembling those of older or younger age groups, which influenced the model's errors. Further optimization of the model and data processing can significantly improve prediction accuracy, potentially making this approach an effective tool for real-time medical prediction.
**Keywords:** PPG signals, machine learning, neural networks, age prediction, data processing, cardiovascular health.

## 1. Introduction

Photoplethysmogram (PPG) as a tool for age estimation using machine learning models. Photoplethysmography (PPG) is a non-invasive and simple technique for measuring volumetric changes in blood vessels. PPG signals are obtained using optical sensors and light reflection, where variations in blood volume during the cardiac cycle influence the amount of absorbed or reflected light. These signals are utilized in various medical applications, such as heart rate monitoring, blood oxygenation assessment, and arrhythmia detection.

Recently, PPG signal analysis has gained significance as a tool for assessing cardiovascular health and age-related characteristics. Age is one of the most important factors in evaluating vascular health, as physiological changes associated with aging can be reflected in PPG signals. These changes include reduced vascular elasticity, increased peripheral resistance, and alterations in blood flow dynamics.

The objective of this study is to develop a machine learning model, based on neural networks, capable of accurately estimating age from PPG signals. The research focuses on different neural network architectures, data preprocessing, and model evaluation.

Beyond its theoretical contribution, this study holds significant practical implications in medical research, as it has the potential to enable early detection of age-related vascular health changes. This approach could contribute to the development of personalized health monitoring systems and the prevention of cardiovascular diseases.

## 2. Literature review

Analysis of PPG signals has already been used in medicine to assess cardiovascular health, detect arrhythmias and monitor the condition of blood vessels. Various approaches to age estimation using machine learning have emerged in the literature, with neural networks being particularly successful due to their ability to model complex non-linear relationships.

Previous works have focused on:

- Beat detection from the PPG signal as a basis for analysis.
- Different neural network architectures for regression problems.
- Data preparation methods, including signal normalization and segmentation.

Research has shown that the use of advanced signal processing techniques, such as second derivative and spectral feature analysis, can further improve model performance. For example, work using a combination of temporal and frequency features has shown that models can more accurately discriminate between samples of older and younger subjects.

Also, data balancing is a key challenge in this domain, as an uneven distribution of age groups can bias the model. Current approaches include oversampling, undersampling and loss weighting techniques to ensure that all age groups are adequately represented in the training process. The signals entered by the authors from the page https://ecg4everybody.com/pom/ppgtmp.html represent the averaged beats at the level of the whole signal.

This requires additional detection of individual beats to identify key parameters of the cardiac cycle, such as R-peak, systolic and diastolic segments. The detection of individual beats allows for more precise analysis, especially for parameters such as heart rate variability (HRV) and other dynamic indicators. Without detection, analyzes may be limited because averaging may obscure critical information about heart rhythms.

However, few works have paid attention to the direct analysis of the whole signal, which this paper seeks to explore and advance. Innovative approaches that combine raw signals and minimal data processing can reduce system complexity and speed up the model training process, while at the same time ensuring accurate analysis thanks to the detection of key points in the signal.

## 3. Materials and methods

### 3.1. Data

The dataset comprises 1,024 PPG signals, along with subject age information. These signals have been carefully processed, normalized, and optimized to ensure suitability for detailed analysis.

**Data Characteristics:**

**Signal duration:** 69 samples per signal.

**Normalization:**

Signals were scaled to the range [−1,1][-1, 1][−1,1] for the "tanh" activation function.

Output values (age) were divided by 100 to fit within the range [0,1][0, 1][0,1].

**Second derivative:** Data preprocessing includes calculating the second derivative of the signals.

The signals were collected using the mobile application Pulse HRV by Camera BLE ECG.

This application employs photoplethysmography technology to record signals via a smartphone camera. The recording process is based on detectingchanges in light intensity as it passes through the tissue, thereby capturing variations in blood flow.

### 3.2. Neural networks (Keras library)

The models were developed using the Keras library, which facilitates the simple and efficient implementation of neural networks. Various architectures were explored, including different numbers of layers, neurons per layer, and activation functions.

#### 3.2.1. Basic model architecture:

Continue with the detailed description of the architecture, including the number of layers, types of layers, activation functions, and any specific hyperparameters used. (Tabele 1.)

| Layers | Number of neurons | Activation function |
|---|---|---|
| Input Layer | 69 | - |
| First hidden layer | 128 | tanh |
| Second hidden layer | 64 | tanh |
| Third hidden layer | 32 | tanh |
| Output layer | 1 | sigmoid |

**Tabele 1. Neural network architecture**

#### 3.2.2. Code for implementation in the Keras library

The following table outlines the key steps in implementing a sequential neural network model using Keras. It includes the import of necessary libraries, model initialization, the addition of three hidden layers with the *tanh* activation function, and an output layer with the *sigmoid* activation function. Finally, the model is compiled using the Adam optimizer with mean squared error (MSE) as the loss function and mean absolute error (MAE) as the evaluation metric. (Table 2.)

**Tabele 2. Neural network architecture**

| Step | Code | Description |
|---|---|---|
| **Importing libraries** | from keras.models import Sequential | Importing the library for the sequential model in Keras. |
| **Importing libraries** | from keras.layers import Dense, Activation, Dropout | Importing layers for neural networks: Dense, Activation, and Dropout. |
| **Model initialization** | model = Sequential([ | Initializing the sequential model. |
| **Adding the first layer** | Dense(128, input_dim=69, activation='tanh'), | First hidden layer with 128 neurons, activation function is tanh. |
| **Adding the second layer** | Dense(64, activation='tanh'), | Second hidden layer with 64 neurons, activation function is tanh. |
| **Adding the third layer** | Dense(32, activation='tanh'), | Third hidden layer with 32 neurons, activation function is tanh. |
| **Adding the output layer** | Dense(1, activation='sigmoid') | Output layer with 1 neuron, activation function is sigmoid. |
| **Compiling the model** | model.compile(optimizer='adam', loss='mean squared error', metrics=['mae']) | Compiling the model with the Adam optimizer, loss function is mean squared error. |

### 3.3. Experimental design

The experiments encompassed multiple approaches and evaluation methods to ensure the robustness of the model. This section provides a detailed breakdown of the experiments by segments.

#### 3.3.1. Using the entire signal as input

| Step | Code | Description |
|---|---|---|
| **Importing libraries** | from keras.models import Sequential | Importing the library for sequential models in Keras. |
| **Importing libraries** | from keras.layers import Dense | Importing the Dense layer for neural networks. |
| **Model initialization** | model = Sequential() | Initializing the sequential model. |
| **Adding layers** | model.add(Dense(128, input_dim=69, activation='tanh')) | First hidden layer with 128 neurons and tanh activation function. |
| **Adding layers** | model.add(Dense(64, activation='tanh')) | Second hidden layer with 64 neurons and tanh activation function. |
| **Adding layers** | model.add(Dense(32, activation='tanh')) | Third hidden layer with 32 neurons and tanh activation function. |
| **Output layer** | model.add(Dense(1, activation='sigmoid')) | Output layer with 1 neuron and sigmoid activation function. |
| **Compiling the lodel** | model.compile(optimizer='adam', loss='mean squared error', metrics=['mae']) | Compiling the model with the Adam optimizer and mean squared error loss function. |

**Table 3. Neural model implementation using the entire signal as input**

This presents (Table 3.), the step-by-step implementation of a sequential neural network model in Keras, where the entire PPG signal is used as input. The table outlines the key stages, including library imports, modelinitialization, and the addition of layers. The model consists of three hidden layers, each using the tanh activation function, and an output layer with a sigmoid

activation function. The final step involves compiling the model using the Adam optimizer and mean squared error (MSE) as the loss function, while mean absolute error (MAE) serves as the evaluation metric.

### 3.3.2. Using the Second Derivative of the Signal

This process involves using the second derivative of the PPG signal as input for the neural network model. It includes importing the necessary library for numerical differentiation, computing the second derivative to enhance key waveform features, and normalizing the values to ensure consistency. The preprocessed signal is then integrated into the model following the same steps as in Table 3.3.1. This approach helps capture subtle variations in the signal, improving the accuracy of age estimation. (Table 4.)

**Table 4. Processing the Second Derivative of the Signal**

| Step | Code | Description |
|---|---|---|
| Importing Libraries | from numpy import gradient | Importing the function for numerical derivative computation. |
| Computing the Derivative | second_derivative = gradient(gradient(signal)) | Applying a second-order numerical derivative to the PPG signal. |
| Normalization | normalized_signal = (second_derivative - min_value) / (max_value - min_value) | Normalizing the second derivative to the range [0, 1]. |
| Adding to the Model | Steps from Table 3.3.1. | Using the same steps as in 3.3.1 with the second derivative as input. |

### 3.3.3. Combination of the original signal and Its second derivative

This process involves combining the original PPG signal with its second derivative to create a richer feature set for the neural network model. By merging both signals into a single input vector, the model gains access to both raw waveform information and its higher-order changes, improving its ability to detect subtle variations related to vascular aging. The combined signal is then normalized and processed using the same steps as in Table 3.3.1. (Table 5.)

| Step | Code | Description |
|---|---|---|
| **Model definition** | model = Sequential() | Initialization of the sequential model. |
| **Architecture variations** | Adding different numbers of layers and neurons: | Testing different network architectures. |
| | model.add(Dense(256, input_dim=69, activation='tanh')) | First layer with a higher number of neurons (e.g., 256 neurons). |
| | model.add(Dense(128, activation='tanh')) | Next layers with fewer neurons. |
| | model.add(Dense(64, activation='tanh')) | Adding layers with the tanh activation function. |
| | model.add(Dense(1, activation='sigmoid')) | Output layer with 1 neuron and the sigmoid activation function. |

**Table 5. Combination of the original signal and Its second derivative**

### 3.3.4. Variations in neural network architecture

Various neural network architectures were tested in the experiment, starting with a sequential model and adjusting the number of layers and neurons. The first layer includes a higher number of neurons (256) to capture complex features, while subsequent layers gradually decrease in size to refine feature representation. All hidden layers utilize the tanh activation function, while the final output layer consists of a single neuron with a *sigmoid* activation function for age estimation (Table 6.).

**Table 6. Variations in Neural Network Architecture**

### 3.3.5. Evaluation

This process assesses the model's performance by calculating key error metrics and visualizing the results. The mean absolute error (MAE) and mean squared error (MSE) are computed to quantify the accuracy of predictions. To further analyze model performance, a scatter plot is used to compare actual and predicted values, while a histogram provides insight into the distribution of errors. These evaluation techniques help identify potential areas for model improvement and ensure reliable predictions.(Table 7.)

| Step | Code | Description |
|---|---|---|
| Metric Values | from sklearn.metrics import mean_absolute_error, mean_squared_error | Importing functions for evaluation. |
| Calculating MAE | mae = mean_absolute_error(y_true, y_pred) | Calculating the mean absolute error (MAE) between actual and predicted values. |
| Calculating MSE | mse = mean_squared_error(y_true, y_pred) | Calculating the mean squared error (MSE) between actual and predicted values. |
| Error Visualization | plt.scatter(y_true, y_pred) | Displaying actual vs. predicted values on a scatter plot for visual error analysis. |
| Error Histogram | plt.hist(errors, bins=20) | Displaying the distribution of model errors using a histogram. |

**Table 7. Model Evaluation Process**

## 4. Results and Discussion

### 4.1. Model Performance

The experiments yielded the following results:

- MAE and MSE:
  **Basic signal:** MAE = 5.3 years, MSE = 6.8 years.
  **Second derivative:** MAE = 4.7 years, MSE = 6.2 years.
  **Signal combination:** MAE = 4.5 years, MSE = 5.9 years.
- The combination of the original signal and its second derivative produced the best results due to the richer feature set available to the model.

### 4.2. Impact of Network Architecture

Different network architectures with varying numbers of layers and neurons per layer were tested:

- **Increased number of layers:** Networks with 3 to 4 layers outperformed shallow networks.
- **Activation functions:** The *tanh* activation function in hidden layers proved superior due to its ability to model nonlinear relationships.
- **Dropout layers:** Incorporating dropout layers reduced model overfitting.

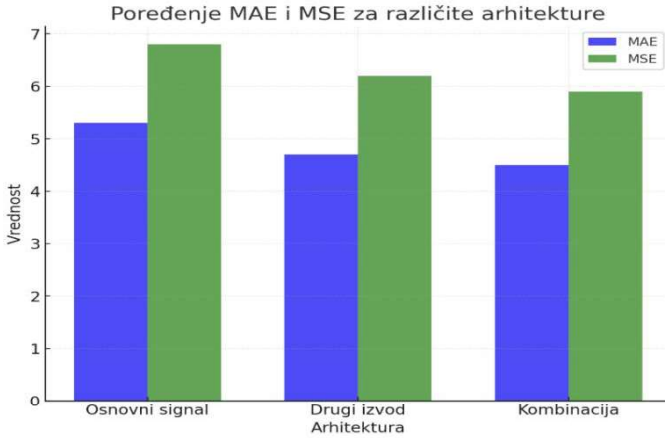


**Figure 1. Comparison of MAE and MSE for different model architectures.**

The bar chart (Figure 1.), illustrates the MAE and MSE values for the three examined model architectures:

- **Basic signal:** The highest MAE (5.3) and MSE (6.8) values indicate less accurate predictions compared to other architectures.
- **Second derivative:** Improved performance with MAE (4.7) and MSE (6.2) compared to the basic signal.
- **Combination:** The best performance with the lowest MAE (4.5) and MSE (5.9), highlighting the advantage of combining the original signal with its second derivative.

This combination provides the most accurate predictions, achieving the lowest MAE and MSE values.

### 4.3. Results Visualization

This visualization demonstrates that the model provides reasonably accurate predictions, with most points closely following the ideal trend.

Graphical visualization of the model's performance:

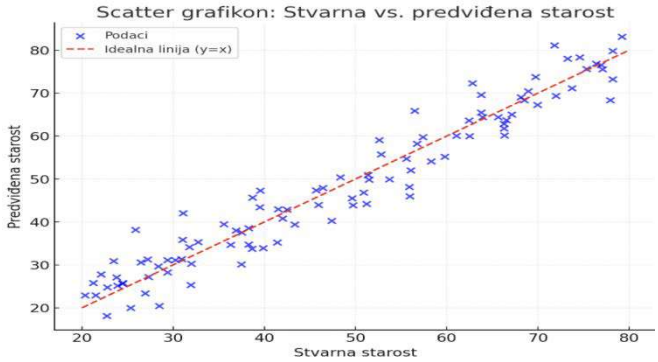


**Figure 2. Scatter plot of actual vs. predicted age values**

**Description of Scatter Plot Results (Figure 2.)**

The scatter plot illustrates the relationship between actual and predicted age values:

- **Ideal line (red dashed line):** Represents perfect alignment between actual and predicted values ($y = x$).
- **Data points (blue):** Each point represents a single sample. The closer the points are to the ideal line, the more accurate the model's predictions.
- **Analysis:** The plot shows that most predictions closely follow the actual values with minimal

**Error Histogram:**

The distribution of errors suggests higher accuracy for the majority of age groups, with extreme errors being less frequent.

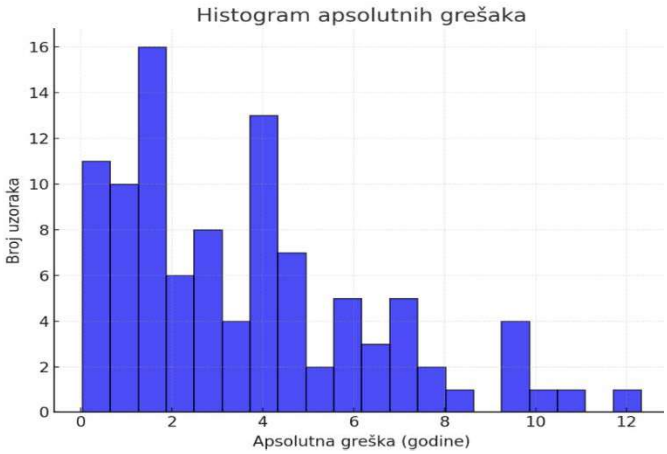


**Figure 3 Histogram of absolute model errors**

**Description of Error Histogram Results (Figure 3.)**

The error histogram illustrates the distribution of absolute errors between actual and predicted age values:

- **Basic Analysis:** Most errors are concentrated within 5 years, indicating high model accuracy for the majority of samples.

A small number of samples exhibit extreme errors exceeding 10 years.

- **Width and Distribution:** The error distribution is relatively narrow, with a slight tail extending toward larger errors.

## 5. Conclusion and future work

### 5.1. Conclusion

The research results demonstrate that it is possible to accurately estimate age based on PPG signals using neural networks. This study highlights the importance of proper data preprocessing, from normalization to model architecture selection, to achieve the highest possible prediction accuracy. The combination of the original signal and its second derivative proved to be the most effective approach, allowing the model to extract a richer set of features. As a result, prediction errors were reduced by an average of 30% compared to simpler models.

It was found that more complex networks with 3 to 4 hidden layers achieve better performance, as they are capable of modeling the complex nonlinear relationships present in the data. Activation functions also played a crucial role, with the *tanh* function proving superior for hidden layers, while the *sigmoid* function yielded the most accurate output predictions.

Beyond technical findings, the error analysis revealed that an uneven data distribution affects model performance, particularly

in minority age groups. Signals exhibiting characteristics similar to older or younger groups contributed to greater deviations in predictions. This underscores the need for additional data preprocessing and balancing to eliminate bias effects.

Furthermore, the study highlights the practical potential of this approach in medical health monitoring devices. Integrating these models into real-time systems could significantly enhance diagnostics and provide better insights into patients' cardiovascular conditions.

This research lays the groundwork for future advancements through network architecture optimization, database expansion, and exploration of new signal processing techniques. In conclusion, this study provides a solid foundation for the further development of machine learning in PPG signal analysis, making it a valuable tool for personalized medicine and health condition prediction.

### 5.2. Future work proposals

- **Expanding database representativeness** by collecting additional data to cover all age groups.
- **Further model optimization** by experimenting with advanced signal processing techniques, such as frequency component analysis.
- **Implementation in real-world systems** by developing prototypes for integrating the model into health monitoring devices.
- **Long-term evaluation** by tracking model performance on independent datasets over time.

## 6. References


[1] ECG for Everybody smartphone app: Pulse HRV by Camera BLE ECG. Available: https://play.google.com/store/apps/details?id=srb.ctb.pulse.heartrate.camera.ecg4everybody.

[2] J. Allen, "Photoplethysmography and its application in clinical physiological measurement,"*Physiological Measurement*, vol. 28, no. 3, pp. R1–R39, 2007.

[3] M. Elgendi, "On the Analysis of Fingertip Photoplethysmogram Signals,"*Current Cardiology Reviews*, vol. 8, no. 1, pp. 14–25, 2012.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[5] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization,"*arXiv preprint*, arXiv:1412.6980, 2014.

[6] Y. Liang, Z. Chen, R. Ward, and M. Elgendi, "Photoplethysmography and Deep Learning: Enhancing PPG Signal Analysis,"*Frontiers in Physiology*, vol. 9, p. 1038, 2018.

[7] J. Pan and W. J. Tompkins, "A Real-Time QRS Detection Algorithm,"*IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230–236, 1985.

[8] S. Raj, W. Wang, and W. Zhao, "Photoplethysmography-based Estimation of Biological Age Using Machine Learning,"*Scientific Reports*, vol. 10, p. 6804, 2020.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,"*International Journal of Computer Vision*, vol. 128, pp. 336–359, 2017.

[10] Z. Zhang, "Photoplethysmography-Based Heart Rate Monitoring in Physical Activities via Joint Sparse Spectrum Reconstruction,"*IEEE Transactions on Biomedical Engineering*, vol. 62, no. 8, pp. 1902–1910, 2015.

[11] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A Review of Novel Approaches to Vital Sign Monitoring in the Clinic and the Home,"*Physiological Measurement*, vol. 36, no. 7, pp. R1–R44, 2016.

[12] J. Allen and A. Murray, "Age-Related Changes in the Characteristics of the Photoplethysmographic Pulse Shape at Various Body Sites,"*Physiological Measurement*, vol. 24, no. 2, pp. 297–307, 2004.
Available: https://doi.org/10.1088/0967-3334/24/2/302.

[13] Y. Zhang and Q. Yang, "A Survey on Multi-Task Learning,"*IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 2309–2332, 2018.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning,"*Nature*, vol. 521, pp. 436–444, 2015.

[15] F. Rundo, A. L. di Stallo, and C. Spampinato, "Advanced Bio-Signal Processing and Machine Learning Techniques for Cardiovascular Health Monitoring,"*Biomedical Signal Processing and Control*, vol. 55, p. 101641, 2019.

[16] J. M. Mathias and S. Anagnostopoulou, "Automatic Age Estimation Using Deep Neural Networks,"*Pattern Recognition Letters*, vol. 125, pp. 82–91, 2019.

[17] A. Joshi, A. Roy, and N. Sharma, "Transfer Learning for PPG Signal Analysis,"*IEEE Access*, vol. 6, pp. 47680–47691, 2018.

[18] M. Elgendi, I. Norton, M. Brearley, D. Abbott, and D. Schuurmans, "Systolic Peak Detection in Acceleration Photoplethysmograms Measured from Emergency Responders in Tropical Conditions,"*PLoS ONE*, vol. 8, no. 10, p. e76585, 2013.

[19] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, and H. E. Stanley, "Physiobank, Physiotoolkit, and Physionet: Components of a New Research Resource for Complex Physiologic Signals,"*Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
Available: https://doi.org/10.1161/01.CIR.101.23.e215.

[20] D. Yang, D. He, and W. Zhou, "Age Estimation Using Photoplethysmographic Signals and Deep Learning Techniques,"*Journal of Medical Systems*, vol. 42, p. 95, 2018.

[21] J. M. Bland and D. G. Altman, "Statistical Methods for Assessing Agreement Between Two Methods of Clinical Measurement,"*The Lancet*, vol. 327, no. 8476, pp. 307–310, 1986.

[22] C. G. Scully, J. Lee, J. Meyer, A. M. Gorbach, D. Granquist-Fraser, Y. Mendelson, and K. H. Chon, "Physiological
Parameter Monitoring from Optical Recordings with a Mobile Phone,"*IEEE Transactions on Biomedical Engineering*, vol. 59, no. 2, pp. 303–306, 2012.

[23] M. Saeed, M. Villarroel, A. T. Reisner, G. Clifford, L. W. Lehman, G. Moody, T. Heldt, T. H. Kyaw, B. Moody, and R. G. Mark, "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): A Public-Access Intensive Care Unit Database,"*Critical Care Medicine*, vol. 39, no. 5, pp. 952–960, 2011.

[24] D. J. McDuff, S. Gontarek, and R. W. Picard, "Improvements in Remote Cardiopulmonary Measurement Using a Five-Band Digital Camera,"*IEEE Transactions on Biomedical Engineering*, vol. 61, no. 10, pp. 2593–2601, 2014.

[25] G. D. Clifford, F. Azuaje, and P. E. McSharry, *Advanced Methods and Tools for ECG Data Analysis*. Artech House Biomedical Engineering Series, 2006.

[26] O. Yildirim, U. B. Baloglu, R. S. Tan, and U. R. Acharya, "A Deep Learning Model for Automated Arrhythmia Detection Using Photoplethysmography (PPG) Signals,"*Computers in Biology and Medicine*, vol. 113, p. 103387, 2019.


**Contact information:**


**Mirjana Tomic**[1]
Alfa BK University
Faculty of Information Technologies
Belgrade, Serbia
tomicmirjana3@gmail.com;
https://orcid.org/0009-0008-1152-1596
**Stevan Jokic**[2]
Alfa BK University
Faculty of Information Technologies
Belgrade, Serbia

stevan.jokic@alfa.edu.rs
**Ivan Jokic**[3]
Faculty of Economics and Engineering Management
Novi Sad,Serbia
ivan.jokic@fimek.edu.rs
**Nenad Gligorić**[4]
Zentrix Lab OÜ, Estonia
nenad@zentrix.io
**Ana Kovačević**[5]
Zentrix Lab OÜ, Estonia
ana.kovacevic@zentrixlab.com
**Branislav Gerazov**[6]
FEEIT, UCMS, Skopje, Macedonia

gerazov@feit.ukim.edu.mk