XLVI CILAMCE
VITÓRIA - ES - BRAZIL - 2025
XLVI IBERO-LATIN AMERICAN CONGRESS ON COMPUTATIONAL METHODS IN ENGINEERING

ECG Heartbeat Classification using Artificial Neural Networks: a comparative study of PyTorch and TensorFlow

Italo F. Di Paolo[1], Hian M. S. C. Barroso[2], Marta de O. Barreiros[1], Adriana R. G. Castro[3]

[1]*Dept. of Computer Systems and Infrastructure, Pará State University*
Tv. Dr. Enéas Pinheiro, 2626, Marco, 66095-015, Belém/PA, Brazil
itflexa@uepa.br, marta.do.barreiros@uepa.br
[2]*Software Engineering Course, Pará State University*
R. Pedro Porpino da Silva, 1181, São José, 68745-000, Castanhal/PA, Brazil
hian.correa@gmail.com
[3]*Postgraduate Program in Electrical Engineering, Federal University of Pará,*
R. Augusto Corrêa, Campus Universitário do Guamá, Instituto de Tecnologia, 66075-110, Belém/PA, Brazil
adcastro@ufpa.br

Abstract. The automatic classification of arrhythmias from ECG signals transformed into images represents a major advance in computer vision for cardiology, enabling faster and more accurate diagnoses. This study presents a comparative analysis of the deep learning frameworks PyTorch and TensorFlow for arrhythmia classification, using the publicly available PTB Diagnostic ECG Database. ECG signals were segmented and labeled into two categories (Abnormal and Normal), generating 14,552 vectors with 187 positions each at 125 Hz, divided into training and testing sets (80/20). Different signal representations were evaluated, including one-dimensional (1D) and two-dimensional (2D) formats, with Recurrence Plot used for image transformation. Implemented models included multilayer Perceptron's and convolutional neural networks (ResNet-18 and AlexNet). Performance was evaluated using various metrics, while computational efficiency was measured by training time and hardware utilization. The results showed that AlexNet in PyTorch achieved the best performance (99.55% accuracy, 99.49% precision, 99.38% recall, 99.44% F1 score, 0.9889 MCC), outperforming TensorFlow (99.17%, 98.97%, 98.97%, and 98.97%, 0.9784). The comparative analysis showed that PyTorch proved more efficient and accurate under identical conditions.

Keywords: Arrhythmia classification; Electrocardiogram; Neural networks; Computational performance.

1 Introduction

The electrocardiogram (ECG) records the electrical activity of the heart, providing relevant information regarding its dynamics and rhythm. These recordings enable specialists to identify a wide range of cardiac abnormalities. However, long-duration examinations, such as 24-hour ECGs—which may contain more than 100,000 heartbeats—pose significant challenges to manual analysis, rendering the process time-consuming, repetitive, and prone to errors [1]. In this context, automated systems emerge as promising tools to support clinical diagnosis, allowing for more accurate and efficient analyses of the information collected by monitoring devices.

The earliest approaches to automatic ECG classification were based on supervised learning, in which studies employed techniques such as linear discriminant analysis, decision trees, and rough sets to extract morphological features from the signals and subsequently apply Artificial Neural Networks (ANNs) for classification [1]. With the advancement of Deep Neural Networks (DNNs), the application of architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has become increasingly prevalent in ECG analysis. CNNs have demonstrated effectiveness in both one-dimensional (1D) signals and two-dimensional (2D) representations, the latter obtained through transformations that convert temporal signals into matrices or images. Deep learning models operate in an end-to-end manner, automatically extracting hierarchical features from raw data. Studies suggest that 2D representations may yield higher accuracy in heartbeat classification compared to

1D representations [2]. Nevertheless, further investigation is required into deep learning techniques applied to the automatic classification of cardiac arrhythmias, particularly by exploring different representations of ECG signals. Moreover, it is essential to establish a comparative analysis of computational tools implemented in the Python environment.

Therefore, the objective of this study was to identify, between PyTorch and TensorFlow, the framework with superior performance and computational efficiency in the automatic classification of cardiac arrhythmias. To this end, the public PTB Diagnostic ECG Database (PTB) was employed, applying three distinct neural network architectures: a Multilayer Perceptron (MLP) and two Convolutional Neural Networks (CNNs), namely AlexNet and ResNet-18. Performance was evaluated using metrics such as accuracy, precision, recall, F1-score, and the Matthews Correlation Coefficient, with comparisons to the state of the art, in addition to an analysis of computational resource usage under different framework configurations.

This article is organized into four sections: Section 1 presents this Introduction; Section 2 details the Methodology employed, including the database, tools, and techniques; Section 3 discusses the experimental Results and their implications; and Section 4 provides the Conclusions of the study.

2 Methodology

This section describes the procedures adopted in the development of the present study, from data selection and preparation to model configuration and evaluation. The methodology is structured into five main components: (i) description of the database employed; (ii) the artificial neural network models implemented; (iii) the data preprocessing strategy; (iv) the training and evaluation process of the models; and (v) the description of the computational environment used for the experiments.

2.1 Database

The Association for the Advancement of Medical Instrumentation (AAMI), founded in 1967 and composed of more than 10,000 professionals, is a reference institution in the definition of technical standards, such as ANSI/AAMI EC57 [3], which is dedicated to reproducible testing for ECG analysis [4]. In this study, the PTB Diagnostic ECG Database (PTB), recommended by AAMI for scientific applications, was employed. This database, made available by Bousseljot et al. [5] and Goldberger et al. [6], comprises 549 records from 290 patients aged between 17 and 87 years (mean age of 57.2 years), including 209 males (55.5 ± 12.4 years) and 81 females (61.6 ± 15.5 years).

For this work, the subset of the PTB database pre-processed by Kashuee, Fazeli, and Sarrafzadeh [7] was used, which includes only heartbeat records labeled with myocardial infarction (MI) diagnosis and healthy controls. This subset contains 14,552 heartbeats extracted from 290 patients, of which 148 belong to the MI class and 52 to the healthy class. Each record consists of 188 samples of a single heartbeat, with a sampling frequency of 125 Hz. The study exclusively employed lead II signals, as this lead is among the most widely used in both clinical and experimental ECG analysis. The division of data into training and testing sets, as well as the binary categorization of beats (Abnormal ‘A’ and Normal ‘N’), will be detailed in Section 2.4

2.2 Artificial Neural Network Models

Artificial Neural Networks (ANNs) represent a computational approach widely applied to the modeling of complex patterns, particularly due to their ability to learn from examples. In this study, three distinct architectures were employed with the aim of comparing their performance in heartbeat classification: a Multilayer Perceptron (MLP) with 100 neurons in the hidden layer, and two Convolutional Neural Networks (CNNs), namely AlexNet [8] and ResNet-18 [9].

2.3 Preprocessing Strategy

The preprocessing stage of the electrocardiogram signals was essential to adapt the data to the different neural network models employed. As described in Section 2.2, one-dimensional models, such as the MLP, and two-

dimensional convolutional models, such as AlexNet and ResNet-18, were utilized, requiring specific data representation strategies. Heartbeats were initially preserved in their 1D form to feed the MLP, which is suitable for capturing sequential patterns in temporal signals. In parallel, to exploit the spatial feature extraction capabilities of CNNs, the same heartbeats were converted into 2D representations through visual transformation techniques, enabling the learning of more complex structural patterns.

Section 2.3.1 describes the construction of these representations, including the use of recurrence plots for image generation, while Section 2.4 addresses supervised training, the cost function, and the evaluation metrics.

2.3.1 1D and 2D Transformation

A Recurrence Plot (RP) is a technique that represents the recurrence of patterns in a temporal signal, highlighting relationships between points in the series and revealing changes in the signal's dynamics. The RP is represented as an image (matrix) derived from the time series, where each element encodes the pairwise distances between points in the series [10][11].

Considering a temporal signal represented by N points $x = [x_1\ x_2\ x_3\ ...\ x_N]$, the RP is mathematically defined as a square matrix of size $N \times N$, where each element of the matrix is given by:

$$\mathrm{R_{ij}}(\varepsilon) = \Theta(\varepsilon - \|\vec{\mathrm{x_i}} - \vec{\mathrm{x_j}}\|) \tag{1}$$

for $i,j = 1, \ldots, N$, where ε represents the threshold or tolerance limit within which two points of the series are considered recurrent, and Θ denotes the Heaviside step function, which returns 1 (recurrent points) if $\varepsilon - \|x_i - x_j\| > 0$ and 0 if $\varepsilon - \|x_i - x_j\| < 0$. Here $\|.\|$ denotes the norm, with the most used being the minimum, maximum, and Euclidean norms. In this case, the RP generates a black-and-white image.

In this study, a modified version of the RP [12] was adopted, which produces a color image, where each element of the matrix is obtained as follows:

$$\mathrm{R_{ij}} = \|\mathrm{x_i} - \mathrm{x_j}\| \tag{2}$$

for $i,j = 1, \ldots, N$, where $\|.\|$ corresponds to the Euclidean norm.

Figure 1 illustrates the recurrence plots of one record from each class, with $\varepsilon = 0$. and a rainbow colormap, as employed by Di Paolo and Castro [13], with each record converted into an image for input into the CNN.

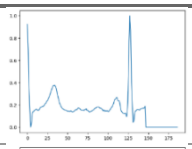
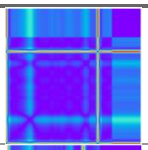
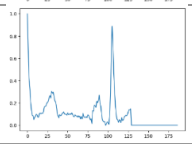
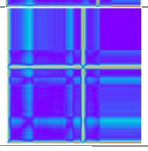
	Time serie	**Recurrence Plot**
Class "N" (Normal)		
Class "A" (Abnormal)		

Figure 1. Recurrence plots for one record from each class.

It is observed that the standard version of the Recurrence Plot employs the Heaviside step function, which generates a binary matrix in which each element indicates whether the distance between two points of the signal is below a given threshold. In contrast, the adapted version used in this study applies a continuous rescaling of distances, producing a matrix with gradual values and thereby mapping color images with a richer visual pattern. This approach preserves more information from the signal, beyond binary recurrence, which is particularly useful for CNNs, as it highlights subtle differences between classes. The drawback, however, lies in the need for tuning additional hyperparameters, requiring further optimization experiments.

2.4 Training and Evaluation Process

After preparing the data as images, as described in Section 2.3 and organized according to Table 1 in an 80/20 ratio, the models were subjected to a supervised training process aimed at learning patterns distinguishing normal from abnormal heartbeats. Each neural network architecture was trained independently, using the same datasets

and validation criteria, in order to ensure a fair comparison among models. Training employed the Binary Cross-Entropy (BCE) loss function to measure the discrepancy between predictions and true labels, iteratively adjusting the network parameters across epochs.

Table 1. Quantitative distribution of the study dataset.

	A	N	Total
Train	8,404	3,237	11,641
Test	2,102	809	2,911
Total	10,506	4,046	14,552

Source: Data extracted from [14].

Section 2.4.1 presents the metrics employed in the evaluation of classifications, namely accuracy, precision, recall, F1-score, and the Matthews Correlation Coefficient (MCC), which are essential for comparative analysis, particularly in cases of class imbalance.

2.4.1 Performance Indicators

The ANSI/AAMI EC57:2012 R2020 [3] standard specifies the indicators that must be used for the performance evaluation of arrhythmia classifiers. The selected indicators include accuracy, precision (positive predictive value), and sensitivity (or recall), which can be calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$Specificity = \frac{TN}{TN + FP} \tag{6}$$

where TP denotes True Positives, TN denotes True Negatives, FP denotes False Positives, and FN denotes False Negatives. Thus, as observed in the literature and recommended for evaluation on highly imbalanced datasets, the harmonic mean between precision and sensitivity, known as the F1-Score, and the Matthews Correlation Coefficient (MCC) can be calculated as follows:

$$F1_Score = 2.\frac{Precision * Recall}{Precision + Recall} \tag{7}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TN + FP) \times (TP + FN) \times (TN + FN)}} \tag{8}$$

3 Results and Discussion

This section presents and analyzes the results obtained from the application of proposed neural architecture to the PTB Diagnostic ECG Database, as previously described. The experiments were conducted with the dataset randomly divided into training (80%) and testing (20%) sets. Three models were employed: a Multilayer Perceptron (MLP) for one-dimensional (1D) signal classification, and two Convolutional Neural Networks (CNNs), AlexNet and ResNet-18, for classification based on two-dimensional (2D) representations generated by recurrence plots. The models were implemented in Python 3.10.9, using specialized libraries such as Pandas, NumPy, TensorFlow 2.16.2, Keras, PyTorch 2.3.1, and Scikit-learn. The experiments were conducted in a local environment equipped with an AMD Ryzen 5 5600G processor, integrated Radeon Graphics, 16 GB of RAM, and a 128 GB SSD. The following subsections detail the performance indicators evaluated (Section 3.1) and a comparison with related literature (Section 3.2), based on the classification models implemented (Section 2.2).

3.1 Performance Indicators

The models were trained from randomly initialized weights, without the use of pre-training. For each architecture and framework, 30 runs were performed, retaining the results with the highest F1-Score, and the mean of these values is reported here. Performance was assessed based on classification metrics (accuracy, precision, recall, F1-score, and MCC) and computational efficiency indicators (average time per epoch).

Table 2 presents the average times per epoch for the three models tested. The MLP implemented in TensorFlow was the most efficient (0.09 s), whereas the same architecture in PyTorch required significantly more time (3.40 min). For the convolutional networks, TensorFlow also exhibited lower average times per epoch, particularly for AlexNet (7.23 min) and ResNet-18 (20.26 min), compared to PyTorch (15.19 min and 24.32 min, respectively). These differences reflect the internal characteristics of each framework and parameter configurations, including the Adam optimizer, learning rates for each model ($1e^{-3}$ for MLP, $1e^{-4}$ for CNNs), 100 epochs for CNNs and 300 for MLP, and variations in batch size—where only a value of 128 could be used for CNNs in PyTorch, while TensorFlow required reduction to 32 due to high memory consumption..

Table 2. Average Time per Epoch.

Model	Average time per epoch (minutes)
MLP Tensorflow	0.09
MLP Pytorch	3.40
ResNet-18 Tensorflow	20.26
ResNet-18 Pytorch	24.32
AlexNet Tensorflow	7.23
AlexNet Pytorch	15.19

Table 3 summarizes the overall classification results. The AlexNet architecture implemented in PyTorch achieved the best overall performance, with an accuracy of 99.55% and an F1-score of 99.44%. ResNet-18 in PyTorch also achieved high performance (accuracy of 99.31%, F1-score of 99.14%). The MLP model, although efficient, yielded lower performance, with accuracy values of 97.18% (TensorFlow) and 97.56% (PyTorch).

Table 3. Classification performance indicators for each model and framework.

Modelos	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	MCC
MLP Tensorflow	97.18	96.30	96.72	96.51	0.9302
MLP Pytorch	97.56	96.88	97.06	96.97	0.9394
ResNet-18 Tensorflow	99.17	98.97	98.97	98.97	0.9795
ResNet-18 Pytorch	99.31	99.29	99.00	99.14	0.9829
AlexNet Tensorflow	99.11	99.19	98.58	98.88	0.9770
AlexNet Pytorch	99.55	99.49	99.38	99.44	0.9889

Framework performance differed: TensorFlow tended to exhibit higher memory consumption, requiring parameter adjustments to avoid failures. PyTorch demonstrated greater flexibility in configuration, as well as competitive training times, particularly with larger batch sizes, which could not be handled by TensorFlow.

3.2 Comparison with the Literature

Table 4 compares the results of this study with previous works using the PTB database, also employing an 80/20 train-test split, thereby illustrating the evolution of models over time. For instance, Diker et al. [15] reported an accuracy of 83.82%, whereas Ahmad et al. [2] achieved 99.20%, reflecting advances in deeper architectures, optimized preprocessing, and increased computational capacity.

Table 4. Comparison of heartbeat classification results on the PTB database with previous studies.

Previous Results	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Sharma, Tripathy, Dandapatl (2015) [16]	96.00	-	93.00	-
Acharya et al. (2017b) [17]	95.22	-	95.49	-
Chen et al. (2018) [18]	96.18	-	93.67	-
Kachuee, Fazeli, Sarrafzadeh (2018) [7]	95.90	95.20	95.10	-
Liu et al. (2018) [19]	96.00	-	95.40	-
Diker et al. (2019) [15]	83.82	82.35	95.45	88.42
Cao et al. (2020) [20]	96.65	-	-	-
Ahamed et al. (2020) [21]	97.66	96.90	97.06	-
Ahmad et al. (2021) [2]	99.20	98.00	98.00	-
Lakhdari, Saeed (2022) [22]	99.00	99.00	98.00	99.00
Kim et al. (2023) [23]	96.53	95.54	95.85	95.69
Victor et al. (2023) [24]	98.45	98.45	98.45	98.45
El-Ghaish, Eldele (2024) [25]	-	99.35	99.46	99.41
Yousuf et al. (2024) [26]	99.84	-	99.70	-
This work (AlexNet Pytorch)	99.55	99.49	99.38	99.44

The model proposed in this study (AlexNet with PyTorch) achieved a precision of 99.55%, matching or surpassing other works in the literature on heartbeat classification using this database. The performance indicators consistently outperformed those of the compared studies, with particular emphasis on precision (99.49%), F1-Score (99.44), and MCC (0.9889), indicating a low rate of false negatives.

The study sought to maximize the performance of each architecture within its environment, showing that TensorFlow required batch size adjustments due to memory limitations, whereas PyTorch allowed greater flexibility and the use of larger batches, resulting in shorter iteration times and superior performance. Therefore, the results are consistent with metrics reported in previous works and reinforce the feasibility of the proposed model in real-world contexts, providing methodological insights, particularly for the PyTorch environment.

4 Conclusions

This study analyzed tools and methods based on artificial neural networks for the automatic classification of cardiac arrhythmias. Three objectives were defined and achieved: selecting a reliable database, implementing different neural network models in distinct frameworks, and comparing their performance. Accordingly, the public PTB Diagnostic ECG Database provided a representative variety of cardiac signals, serving as a solid foundation for model training and validation.

Classifiers were developed using MLP, AlexNet, and ResNet-18, implemented in TensorFlow and PyTorch. The results demonstrated superior performance with PyTorch, particularly in CNNs, with higher accuracy, precision, recall, F1-score, and MCC, as well as greater efficiency in computational resource usage, with lower memory consumption and greater flexibility in hyperparameter tuning. TensorFlow required parameter adjustments, such as batch size reduction, due to hardware limitations.

The strengths of this study include the use of a public database and the comparison between two deep learning frameworks, with results surpassing those of previous works. The main limitation was the absence of a dedicated GPU, which restricted the number of epochs and prevented testing with deeper networks or larger batch sizes.

As future work, it is proposed to expand the evaluation of models to other databases, such as MIT-BIH or PhysioNet Challenge, as well as to test more advanced architectures, including RNNs or hybrids combining CNNs with SVMs/KNNs, in addition to experiments with dedicated GPU usage. It is also suggested to investigate more robust preprocessing techniques, feature engineering approaches, ensemble learning, and mechanisms for continuous model adaptation, aiming to enhance robustness in clinical applications.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property and authorship of the authors.

References

[1] SUN, Li et al. ECG Analysis Using Multiple Instance Learning for Myocardial Infarction Detection. IEEE Transactions on Biomedical Engineering, v. 59, n. 12, p. 3348-3356, dec. 2012. https://www.doi.org/10.1109/TBME.2012.2213597

[2] AHMAD, Zeeshan et al. ECG Heartbeat Classification Using Multimodal Fusion. IEEE Access, Vol. 9, 2021. https://doi.org/10.1109/ACCESS.2021.3097614

[3] ANSI/AAMI EC57:2012 (R2020). Testing And Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. AAMI, 2020. https://webstore.ansi.org/Standards/AAMI/ANSIAAMIEC572012R2020
[4] LUZ, E. J. S. Classificação automática de arritmias: um novo método usando classificação hierárquica. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Ouro Preto, Ouro Preto, 2012. https://www.repositorio.ufop.br/items/4ed58a08-ec05-44a1-af95-49e43b0f7505
[5] BOUSSELJOT, R., KREISELER, D.; SCHNABEL, A. Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. Biomedizinische Technik (Biomedical Engineering), Vol. 40, n. s1, 1995. https://doi.org/10.1515/bmte.1995.40.s1.317
[6] GOLDBERGER, A. L. et al. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. Circulation, v. 101, n. 23, p. e215-e220, 2000. https://physionet.org/content/ptbdb/1.0.0
[7] KACHUEE, Mohammad; FAZELI, Shayan; SARRAFZADEH, Majid. ECG Heartbeat Classification: a deep transferable representation. In: Proceedings of the IEEE International Conference on Healthcare Informatics (ICHI), New York, NY, USA, 4–7 Jun. 2018. https://doi.org/10.1109/ICHI.2018.00092
[8] KRIZHEVSKY, Alex. One weird trick for parallelizing convolutional neural networks. Neural and Evolutionary Computing, 2014. https://doi.org/10.48550/arXiv.1404.5997
[9] HE, Kaiming et al. Deep Residual Learning for Image Recognition. 2015. https://doi.org/10.48550/arXiv.1512.03385
[10] ECKMANN, J.-P.; KAMPHORST, S. Oliffson; RUELLE, D. Recurrence Plots of Dynamical Systems. Europhysics Letters, Vol. 4, n. 9, 1987. https://doi.org/10.1209/0295-5075/4/9/004
[11] CASDAGLI, M. C. Recurrence plots revisited. Physica D: Nonlinear Phenomena, Vol. 108, n. 1–2, 1997. https://doi-org.ez3.periodicos.capes.gov.br/10.1016/S0167-2789(97)82003-9
[12] FARIA, Fabio A. et al. Fusion of time series representations for plant recognition in phenology studies. Pattern Recognition Letters, Vol. 83, Parte 2, 2016. https://doi.org/10.1016/j.patrec.2016.03.005
[13] DI PAOLO, Ítalo Flexa; CASTRO, Adriana Rosa Garcez. Intra- and Interpatient ECG Heartbeat Classification Based on Multimodal Convolutional Neural Networks with an Adaptive Attention Mechanism. Applied Sciences, Vol. 14, n. 20, oct. 2024. https://doi.org/10.3390/app14209307
[14] ECG Heartbeat Categorization Dataset. 2018. [Online]. https://www.kaggle.com/shayanfazeli/heartbeat
[15] DIKER, Aykut et al. A novel application based on spectrogram and convolutional neural network for ECG classification. In: Proceedings of the 1st International Informatics and Software Engineering Conference (UBMYK), Ankara, Turkey, nov. 2019. https://doi.org/10.1109/UBMYK48245.2019.8965506
[16] SHARMA, L. N.; TRIPATHY, R. K.; DANDAPAT, S. Multiscale energy and eigenspace approach to detection and localization of myocardial infarction. IEEE Transactions on Biomedical Engineering, Vol. 62, n. 7, 2015. https://doi.org/10.1109/TBME.2015.2405134
[17] ACHARYA, U. Rajendra et al. Application of deep convolutional neural network for automated detection of myocardial infarction using ECG signals. Information Sciences, Vol. 415–416, nov. 2017b. https://doi.org/10.1016/j.ins.2017.06.027
[18] CHEN, Yufei et al. Multi-channel lightweight convolution neural network for anterior myocardial infarction detection. In: Proceedings of the IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), Guangzhou, China, oct. 2018. https://doi.org/10.1109/SmartWorld.2018.00119
[19] LIU, Wenhan et al. Real-time multilead convolutional neural network for myocardial infarction detection. IEEE Journal of Biomedical and Health Informatics, Vol. 22, n. 5, 2018. https://doi.org/10.1109/JBHI.2017.2771768
[20] CAO, Yangjie et al. Multi-channel lightweight convolutional neural network for remote myocardial infarction monitoring. In: IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Seoul, Korea (South), apr. 2020. https://doi.org/10.1109/WCNCW48565.2020.9124860
[21] AHAMED, Atik et al. ECG heartbeat classification using ensemble of efficient machine learning approaches on imbalanced datasets. In: Proceedings of the 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh, nov. 2020. https://doi.org/10.1109/ICAICT51780.2020.9333534
[22] LAKHDARI, Kheira; SAEED, Nagham. A new vision of a simple 1D Convolutional Neural Networks (1D-CNN) with Leaky-ReLU function for ECG abnormalities classification. Intelligence-Based Medicine, Vol. 6, 2022. https://doi.org/10.1016/j.ibmed.2022.100080
[23] KIM, Eunchan et al. TinyML-Based Classification in an ECG Monitoring Embedded System. Computers, Materials & Continua, Vol. 75, n. 1, 2023. https://doi.org/10.32604/cmc.2023.031663
[24] VICTOR, Johnson Olanrewaju et al. A Cost-Based Dual ConvNet-Attention Transfer Learning Model for ECG Heartbeat Classification. Journal of Informatics and Web Engineering, Vol. 2, n. 2, 2023. https://doi.org/10.33093/jiwe.2023.2.2.7
[25] EL-GHAISH, Hany; ELDELE, Emadeldeen. ECGTransForm: Empowering adaptive ECG arrhythmia classification framework with bidirectional transformer. Biomedical Signal Processing and Control, Vol. 89, 2024. https://doi.org/10.1016/j.bspc.2023.105714
[26] YOUSUF, Asim et al. Inferior Myocardial Infarction Detection from Lead II of ECG: A Gramian Angular Field-Based 2D-CNN Approach Sensor applications, Vol. 8, n. 10, 2024. https://doi.org/10.1109/LSENS.2024.3450176