Cardiac Arrhythmia Classification from 12-lead Electrocardiogram Using a Combination of Deep Learning Approaches

Marko Barišić, Alan Jović
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
marko.barisic22@gmail.com, alan.jovic@fer.hr

***Abstract*—Traditionally, electrocardiogram (ECG) signals are recorded and monitored over a period of time and finally analyzed by an expert. Automatic classification of cardiac arrhythmias has the potential to improve diagnostics. In this work, we explore the use of representation learning from ECG signals for cardiac arrhythmia classification. The dataset consisting of five cardiac rhythm types was created from the CPSC, CPSC-Extra, and The Georgia 12-lead ECG Challenge databases. We use a sophisticated deep learning approach for representation learning and classification, namely a combination of a Convolutional Auto-Encoder (CAE) and a Long Short-Term Memory (LSTM) classifier. CAE was used to compress the input signal that serves as input to the LSTM classifier. We also implemented a CAE-based data augmentation approach to balance the data distribution. The classification results reaching above 90% accuracy show that the use of the complex deep learning approach is suitable for addressing the problem.**

***Keywords*—arrhythmia classification, ECG, deep learning, convolutional autoencoder, LSTM, data augmentation.**

I. Introduction

According to the data published by the World Health Organization, cardiovascular diseases are the leading cause of death worldwide, with an estimated 17.9 million deaths per year, representing 32% of total deaths [1]. Scientific research has played a prominent role in addressing this problem. The basis of scientific research in the field of cardiology is the electrocardiogram, which records the electrical activity of the heart. The data collected, ECG signal recordings, have a wealth of uses in both research and practical applications.

Traditionally, ECG signals are recorded and monitored over a period of time and eventually analyzed by an expert. Nowadays, automatic classification of cardiac arrhythmias has the potential to improve diagnostics and save lives. Such a system can be used in real time in wearable technologies, Holter monitors and medical facilities. The task of arrhythmia classification from ECG signals can generally be divided into three main steps. The first step, similar to other machine learning tasks, is to obtain the data. The second step is feature extraction, which takes place after detecting the fiducial points in the ECG recordings. The third and final step is classification. The traditional rule-based diagnosis paradigm is inefficient when dealing with large amounts of data because it requires significant analysis and expert knowledge [2]. Deep learning, on the other hand, shows outstanding performance on ECG classification studies in recent few years [3]. Its hierarchical architecture and non-linearity enables higher-level features (representations) to be obtained and its strong capabilities in feature extraction contribute greatly to classification tasks [3]. The goal of our work is to use a combination of deep learning approaches in each of the above steps to implement an end-to-end solution for automatic classification of 5 cardiac rhythms (the normal rhythm and 4 common arrhythmias).

This paper is organized as follows. Section II presents some related works. Section III describes the datasets we used for the tasks and the proposed deep learning methods combination. Section IV presents the experimental results obtained with our solution and discusses them. Finally, Section V is the conclusion of this paper.

II. Related Work

We noted that the task of arrhythmia classification can generally be divided into three main steps. Beginning with the second step, the traditional approach is to use deterministic algorithms for feature extraction. The Pan Tompkins QRS detection algorithm [4], dating back to 1985, is the staple algorithm in arrhythmia classification research, as it provided a highly accurate way to detect QRS complexes (R peaks) in ECG. In a more recent work, Elgendi et al. [5], presented an algorithm for accurately detecting other fiducial points in ECG, namely P and T waves. In spite of significant advances in traditional feature extraction, recently, deep learning achieved great results in ECG feature extraction. In their work, Šarlija et al. [6] tackle the problem of QRS complex detection using a 1-D Convolutional Neural Network (CNN) classifier. The classifier consisted of two convolutional layers, 1 max-pooling in between, and two fully connected layers. The CNN served as a decision rule that labeled the QRS complexes with multiple positive outputs. These outputs were then clustered and a central measure was used as the final decision for detecting the QRS complex. The data used in this work were from 44 non-paced records from the MIT-BIH Arrhythmia Database [7]. Their method for detecting the QRS complex achieved a sensitivity of 99.81% and a positive predictive value of 99.93%.

In classification, the third step, the best results are usually obtained with machine learning approaches, especially deep learning. Sannino et al. [8] developed a deep learning model for heartbeat classification and arrhythmia detection with 7 fully connected layers. They obtained the data from the MIT-BIH Arrhythmia Database [7]. The ECG signals were first filtered, denoised, segmented into heartbeats, and finally their features were extracted using deterministic algorithms before being passed to the deep learning model. The input to the deep learning model consisted of 50 heartbeat segments of the original ECG signal and 4 of its temporal features.

Because of the capabilities of deep learning models, particularly CNNs, deep learning was used for both feature extraction and classification. The data does not go through feature extraction algorithms, but is fed directly into the input of the deep learning model, resulting in end-to-end solutions. Ribeiro et al. [9] used four residual blocks placed between a convolutional layer at the beginning and a fully connected layer at the end. They collected a dataset of 2,322,513 ECG recordings from over one and a half million patients from Brazil. This important work classified six types of abnormalities with an F1 score of over 80% and a specificity of over 99%.

Pyakillya et al. [10] presented a deep learning model as an end-to-end solution with seven 1-D convolutional layers for feature extraction followed by three fully connected layers for classification. This work used data from the PhysioNet/Computing in Cardiology Challenge 2017 [11]. The solution classified ECG recordings into four classes; normal sinus rhythm, atrial fibrillation, other rhythm and noisy. They achieved accuracy of 86%. Yildirim et al. [12] used an exciting combination of deep learning approaches that motivated our work. The basis of the experiment was the long short-term memory classifier. A neural network was proposed with an LSTM layer at the beginning, followed by two fully connected layers and a softmax activation function at the end. This classifier was used to automatically classify ECG signals into one of the five available arrhythmia classes. To reduce the computational cost, a nonlinear compression structure was implemented to reduce the input size to the LSTM classifier. For this task, a convolutional autoencoder was implemented to compress (*encode*) and decompress (*decode*) the ECG signal. The compressed ECG signal was used as input to the LSTM classifier, thus cutting down on overall computational resources and training time. This work achieved over 99.0% accuracy and showed that the CAE-LSTM model was about 7 times faster than the LSTM model.

Finally, deep learning can be applied to the first step by augmenting the dataset by generating synthetic data. The term data augmentation refers to any method that artificially inflates the original training set with label-preserving transformations [13]. Shaker et al. [14] addressed the problem of balancing a dataset using generative adversarial networks (GANs). For each arrhythmia class, a specific GAN was trained because the amount of data available for each class was disproportionately large. If a single GAN were trained for the entire dataset, it would focus on generating samples of the dominant class and neglect the minor classes. The dataset used was from the MIT-BIH Arrhythmia Database [7]. This interesting approach and the positive impact on the results also motivated our work. They used a CNN that achieved an overall accuracy of over 98%, a precision of over 90%, a specificity of over 97.4%, and a sensitivity of over 97.7%.

TABLE I: Distribution of the Initial Dataset

Rhythm type	ECG recordings	Heartbeats	Training set heartbeats
NSR	2670	37909	26537
LBBB	467	6872	4811
RBBB	2399	38593	27016
PAC	1852	20610	14427
VPB	387	4068	2848
TOTAL	**7775**	**108052**	**75639**

III. Materials and Methods

In this work, we attempt to combine sophisticated deep learning approaches, the deep CAE-LSTM model for ECG record compression and arrhythmia classification presented in [12], and synthetic ECG record generation based on the work in [14]. However, we did not implement GANs for data augmentation. Instead, we present a different, novel method for data augmentation based on properties of CAE. Additionally, our work and [12] differ in two aspects, the datasets and preprocessing. Subsequently, the works differ in results as well. This is discussed in Section IV.

A. Dataset

We use a different dataset than in [12], this is so that we can experiment how the presented deep learning approaches behave on a different dataset. Namely, ECG signals for our dataset were obtained from a collection of three databases: CPSC Database, CPSC-Extra Database, and The Georgia 12-lead ECG Challenge (G12EC) Database, all available from the PhysioNet/Computing in Cardiology Challenge 2020 at [15]. Each ECG recording obtained was sampled at 500 Hz, and the dataset was created from second-lead ECG signals only (II). The dataset contains recordings of five rhythm types: normal sinus rhythm (NSR), right bundle branch block (RBBB), left bundle branch block (LBBB), premature atrial contraction (PAC), and ventricular premature beat (VPB). The dataset had to be further segmented into heart beats and the ECG signals had to be preprocessed. Afterwards, the dataset was split into a training dataset, containing 70% randomly selected samples, and a test dataset, containing 30% of the samples. The number of recordings for each rhythm are shown in Table I.

B. Preprocessing

Each ECG signal first passes through a bandpass filter (5 – 15 Hz) to remove noise and provide a smoother signal. Each ECG recording is segmented into heartbeats using the Pan-Tompkins algorithm [4] to detect QRS complexes. The segmented heartbeats are 260 samples long, and the detected R peak is located at the 100th sample of the recording. Finally, the ECG signals are normalized between 0 and 1. This method for beat segmentation differs from [12]. In their work, the beat labels were independently annotated by multiple cardiologists, whose consensus resolved differences in the diagnoses [12].

C. Convolutional Auto-encoder (CAE)

An autoencoder is a neural network that is trained to try to copy its input to its output [16]. Internally, it has a hidden layer **h** that describes a code that represents the input [16]. The network consists of two parts: an encoder function **h=f(x)** and a decoder that produces a reconstruction **r=g(h)** [16]. Autoencoders have been successfully used in dimensionality reduction and information retrieval [16]. CAEs benefit from unsupervised learning of convolutional layers [17]. The convolutional part uses a different number of layers with different kernel sizes as filters to create an abstract information for input [18]. Both CNNs and CAEs use convolutional layers, but CAEs are not trained for classification or detection. During the training of CAEs, the goal is to minimize the *reconstruction error*, i.e., the value of the loss function that accounts for the reconstructed and original data.

We have implemented a deep CAE with 16 layers, as shown in Table II. The input to the CAE is a 260-sample ECG signal of a single heartbeat. The CAE model, like any auto-encoder, consists of two parts: encoder and decoder. The encoder has the task of compressing the 260 samples-long signal into 32 samples-long encoded features. Hence, with CAE, we can represent a larger signal with a smaller encoded signal. The decoder part has the task of reconstructing the original signal from the coded signal. The reconstructed signal is compared to the original signal in the loss function during the training phase and used for the reconstruction metrics during the testing and validation phase. We trained our CAE for 50 epochs, with a batch size of 128. The Adam optimizer was used with a learning rate of 0.00036, betas of 0.9 and 0.999, and a weight decay of 1e-05. We used the mean squared error (MSE) as the loss function.

Table I indicates a problem of class imbalance in the dataset. In order to solve this problem, data augmentation had to be implemented. As mentioned earlier, we did not implement GANs. Instead, we turned to our CAE model. We noticed slight imperfections and differences in the reconstructed signals compared to the originals, and this difference was more pronounced for some samples. Fig. 1 shows signal reconstruction examples for each arrhythmia class. We can see, even when there are more pronounced differences in the reconstructed signal, it still retains the general shape of the original ECG signal. These signals are different from the original, but appear unique and natural. As obtaining such signals is the real intent behind data augmentation, we used the obtained reconstructed signals for data augmentation. Table III shows the number of generated synthetic ECG signals and the distribution of the dataset after data augmentation. The generated signals were used to balance the training set only. We can see how the dataset now has a much more balanced distribution of arrhythmia classes.

TABLE II: CAE Architecture

	No	Layer name	Kernel size	Activation function	Trainable parameters	Output size
Input: ECG signal 260 x 1						
E	1	Conv1d	16 x 5	ReLU	96	260 x 16
N	2	MaxPool1d	2	–	0	130 x 16
C	3	Conv1d	64 x 5	ReLU	5,184	130 x 64
O	4	BatchNorm1d	–	–	128	130 x 64
D	5	MaxPool1d	2	–	0	65 x 64
E	6	Conv1d	32 x 3	ReLU	6,176	65 x 32
R	7	Conv1d	1 x 3	ReLU	97	65 x 1
	8	MaxPool1d	2	–	0	32 x 1
Bottleneck: Compressed ECG signal 32 x 1						
D	9	ConvTranspose1d	1x 3	ReLU	4	32 x 1
E	10	ConvTranspose1d	32 x 3	ReLU	128	32 x 32
C	11	Upsample	2	–	0	64 x 32
O	12	ConvTranspose1d	64 x 5	ReLU	10,304	64 x 64
D	13	Upsample	2	–	0	128 x 64
E	14	ConvTranspose1d	16 x 5	ReLU	5,136	128 x 16
R	15	Flatten	–	–	0	2048 x 1
	16	Linear	–	–	532,740	260 x 1
Output: reconstructed ECG signal 260 x 1						

TABLE III: Distribution of the Dataset After Augmentation

Rhythm type	Generated heartbeats	Heartbeats	Training set heartbeats
NSR	0	37909	26537
LBBB	19244	26116	24055
RBBB	0	38593	27016
PAC	14427	35037	28854
VPB	22784	26852	25632
TOTAL	**56455**	**164507**	**132094**

D. LSTM Classifier

LSTM is a part of the recurrent neural network (RNN) family that includes memory blocks, memory cells, and gate units [19]. It has been shown that LSTM networks can learn long-term dependencies more easily than the simple recurrent architectures [16]. In this work, the LSTM network is used for arrhythmia classification from compressed ECG signals. We have implemented a 5-layer deep LSTM classifier. Table IV shows the detailed architecture. The input to the LSTM network is a 32-sample compressed ECG signal, the output of the encoder

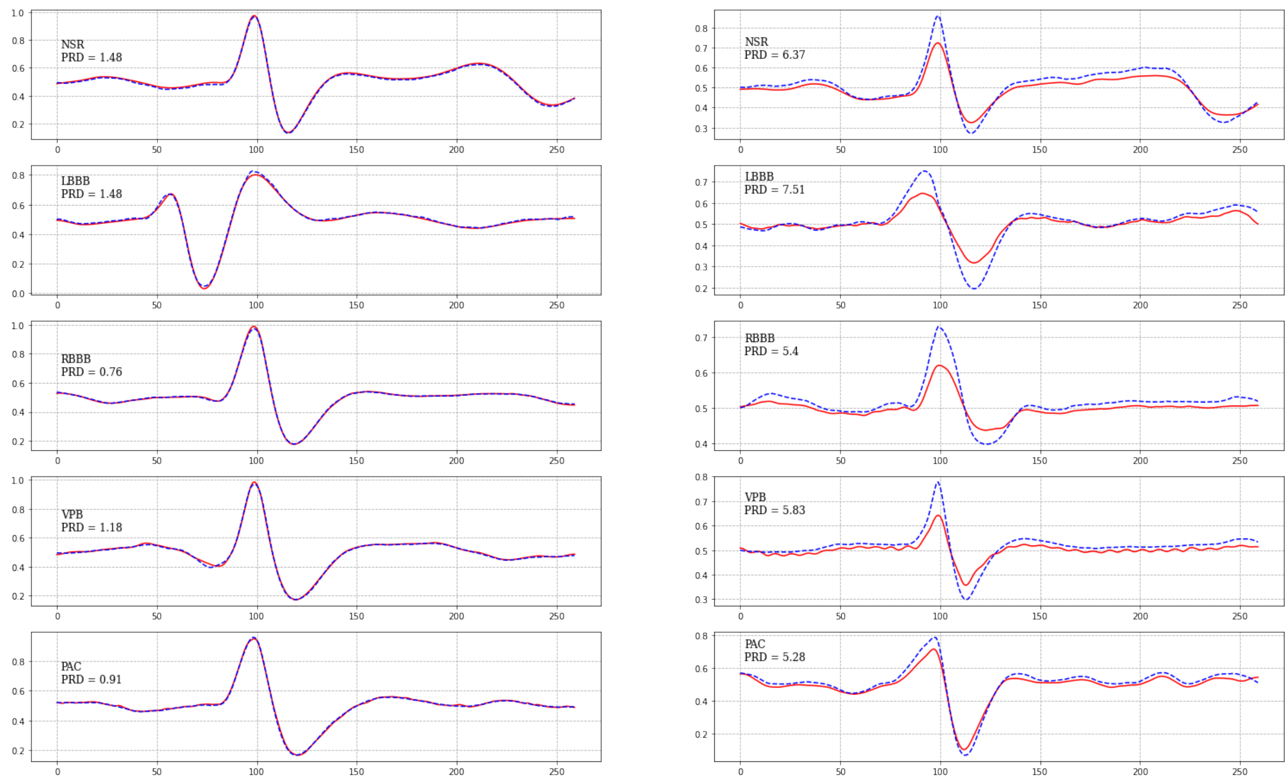

Fig. 1: Examples of the original ECG signals (red line) and the corresponding reconstructed signals (overlapping dashed blue line) for each rhythm. The left column presents examples of very good reconstruction of the original signal, with only a slight deviation. The right column presents examples of reconstruction with pronounced deviations to the original signal, especially along the QRS complex.

TABLE IV: LSTM Classifier Architecture

No	Layer	Unit size	Layer parameter
Input: Compressed ECG signal 32 x 1			
1	LSTM	32	batch first = True
2	Flatten	-	-
3	Linear	128	activation = ReLU
4	Dropout	-	p = 0.1
5	Linear	5	activation = softmax

part of the CAE. At the output of the network is the softmax activation function, which provides a probability prediction for this multiclass classification task (5 rhythm classes). The LSTM classifier was trained for 500 epochs with a batch size of 128. The Adam optimizer was used with a learning rate of 0.0005, betas of 0.9 and 0.999, and a weight decay of 1e-05. We used cross entropy as the loss function. The CAE-LSTM classifier architecture is presented in Fig. 2.

IV. Experimental Results

The experimental results are divided into two categories: compression and classification. The compression results, which come from the CAE evaluation, show the quality of compression and reconstruction of the ECG signals. The arrhythmia classification results were achieved by the LSTM classifier. The experimental setup was as follows. For the experimental implementation, the deep learning tool PyTorch [20] was used along with Python [21]. All experiments were performed on a remote system with 8 CPUs, 30 GB RAM memory and 16 GB NVIDIA Quardo P5000 graphics card.

A. *Compression/Reconstruction results*

The CAE model was used to compress and then reconstruct an ECG signal. The original ECG signal of 260 samples was compressed to 32 samples in the encoder. The decoder reconstructs the compressed ECG signal of 32 samples back to the original 260 samples. During this process, some information of the original signal is lost, as the reconstructed signal is not identical to the original. To evaluate the quality of the reconstructed signal, the following evaluation metrics were used [12]: Root Means Squared (RMS), Percentage RMS Difference (PRD), PRD Normalized (PRDN), Quality Score (QS) and Compression Ratio (CR).

CR is the only criterion that directly relates to compression. It is determined by dividing the size of the original signal with the size of the compressed signal. A higher value of CR means more compression. It is important to choose a good value of CR to get a good reconstruction of the compressed signals. The value of CR was set to 8.125 in this work. PRD is a very important metric for

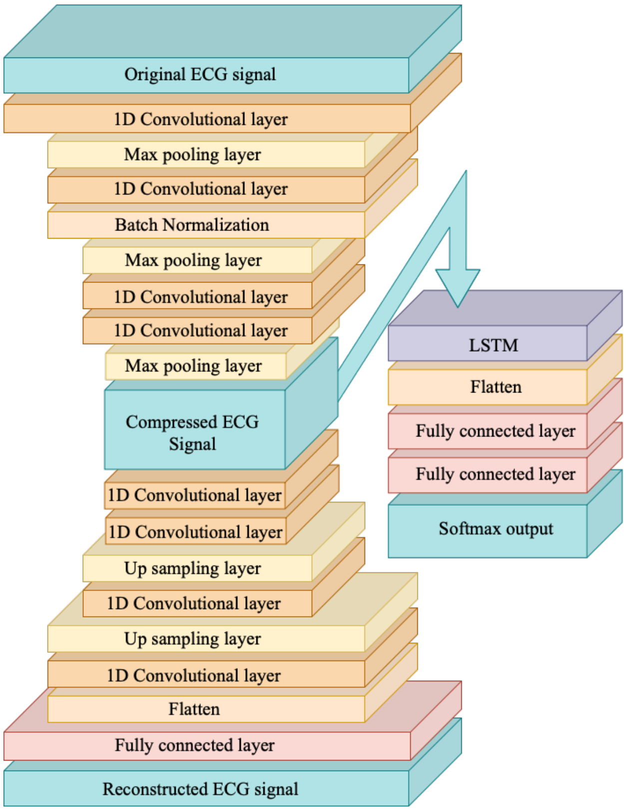

Fig. 2: Block representation of the classifier architecture.

TABLE V: CAE Reconstruction Metrics

Rhythm	PRD	PRDN	RMS	SNR	QS
NSR	**1.31**	**5.72**	0.01	**58.21**	**6.71**
LBBB	1.94	7.61	0.01	53.17	4.69
RBBB	1.76	7.65	0.01	53.73	5.37
PAC	1.43	6.80	0.01	55.23	6.16
VPB	1.66	8.03	0.01	51.98	5.58
Average	1.63	7.16	0.01	54.46	5.7

signal reconstruction, which gives information about the amount of data lost in the reconstructed signal. A lower PRD value means less data loss, thus better reconstruction. The formula for PRD is, as follows:

$$PRD = 100 * \sqrt{\frac{\sum_{i=0}^{N-1}(X_o(i) - X_r(i))^2}{\sum_{i=0}^{N-1}(X_o(i))^2}} \tag{1}$$

The reconstruction results are presented in Table V, where less is better. As we can see from the table, the NSR class has the best overall score for each metric. However, all arrhythmia classes score well on these criteria. The overall PRD score is 1.63 and QS is 5.7. Fig. 1 shows two examples of signal reconstruction for each arrhythmia class, one with a lower PRD and one with a higher PRD.

B. *Classification Results*

The LSTM classifier is tasked with classifying five rhythm classes from the compressed ECG signal. Table VI shows the classification results obtained. These results were obtained using the test dataset presented to the network only during the testing phase. We used the following evaluation metrics: accuracy, positive predictive value (PPV), sensitivity, specificity and negative predictive value (NPV). The formulas for these criteria are, as follows:

TABLE VI: LSTM Classification Results, %

Rhythm	Accuracy	PPV	Sensitivity	Specificity	NPV
NSR	88.4	54.81	81.34	96.84	89.5
LBBB	**96.65**	**94.16**	**88.23**	**97.2**	**98.68**
RBBB	91.58	72.01	84.52	96.61	93.07
VPB	93.91	84.63	84.09	96.15	96.29
PAC	82.94	80.97	57.81	83.48	94.01
Average	90.7	77.32	79.2	94.06	**94.31**

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3}$$

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

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

$$NPV = \frac{TN}{TN + FN} \tag{6}$$

TP denotes the true positives, TN true negatives, FP false positives and FN false negatives. We chose these criteria due to their use in medical diagnostic test evaluation [22]. Sensitivity is the ability to correctly classify the presence of a specific rhythm. Specificity is the ability to correctly classify the absence of a specific rhythm.

The classification results shown in Table VI were achieved after balancing the training dataset. The classification results for the original dataset were much worse. Classification of PAC and VPB was essentially nonexistent with extremely low test dataset results. The results for other rhythms were significantly lower than those obtained after data augmentation. According to Table VI, the detection of the LBBB arrhythmia class showed the best performance with respect to all criteria. Moreover, we see that the specificity and NPV values are better than the sensitivity and PPV values. In practise, this means that our solution is better able to diagnose a patient as *arrhythmia-free* than a patient with arrhythmia. However, according to Table VI, our solution still achieves good results for sensitivity and PPV, especially for LBBB, VPB and even RBBB. As mentioned in Section III under *Preprocessing*, in [12] expert knowledge is used in beat segmentation, and the beat labels were independently annotated by several cardiologists. This method is superior to our method. Due to increased data quality, the results obtained are also better than ours, reaching 99.11% accuracy.

V. Conclusion

In this work, we have implemented a deep CAE-LSTM network model based on coded features, first presented in Yildirim et al. [12]. A deep convolutional autoencoder has been implemented to obtain compressed ECG signals that can be used as input to the LSTM classifier. We also present a novel approach to data augmentation based on CAE. In addition to compression, CAE can also reconstruct the original ECG signal. The reconstructed signal is different from the original and can be considered as a new and unique ECG signal. This was used as a data augmentation approach to balance the initial, significantly unbalanced training dataset. The CAE model compressed and reconstructed the signal with an overall PRD of 1.63 and QS of 5.7. The LSTM classifier used to classify 5 rhythm classes (NSR and 4 arrhythmias) achieved an accuracy of 90.7%, a specificity of 94.06%, a negative predictive value of 94.31%, a sensitivity of 77.32%, and a positive predictive value of 79.2%.

References

[1] *Cardiovascular diseases*, https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1, Accessed: 2022-02-19.

[2] Z. Ebrahimi, M. Loni, M. Daneshtalab, and A. Gharehbaghi, "A review on deep learning methods for ECG arrhythmia classification," *Expert Systems with Applications: X*, vol. 7, p. 100 033, 2020.

[3] X. Liu, H. Wang, Z. Li, and L. Qin, "Deep learning in ECG diagnosis: A review," *Knowledge-Based Systems*, vol. 227, p. 107 187, 2021.

[4] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, no. 3, pp. 230–236, 1985.

[5] M. Elgendi, B. Eskofier, and D. Abbott, "Fast T wave detection calibrated by clinical knowledge with annotation of P and T waves," *Sensors*, vol. 15, no. 7, pp. 17 693–17 714, 2015.

[6] M. Šarlija, F. Jurišić, and S. Popović, "A convolutional neural network based approach to QRS detection," in *Proceedings of the 10th international symposium on image and signal processing and analysis*, IEEE, 2017, pp. 121–125.

[7] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.

[8] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Future Generation Computer Systems*, vol. 86, pp. 446–455, 2018.

[9] A. H. Ribeiro, M. H. Ribeiro, G. M. Paixão, *et al.*, "Automatic diagnosis of the 12-lead ECG using a deep neural network," *Nature Communications*, vol. 11, no. 1, pp. 1–9, 2020.

[10] B. Pyakillya, N. Kazachenko, and N. Mikhailovsky, "Deep learning for ECG classification," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 913, 2017, p. 012 004.

[11] G. D. Clifford, C. Liu, B. Moody, *et al.*, "AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017," in *2017 Computing in Cardiology (CinC)*, IEEE, 2017, pp. 1–4.

[12] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya, "A new approach for arrhythmia classification using deep coded features and LSTM networks," *Computer Methods and Programs in Biomedicine*, vol. 176, pp. 121–133, 2019.

[13] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, 2018, pp. 1542–1547.

[14] A. M. Shaker, M. Tantawi, H. A. Shedeed, and M. F. Tolba, "Generalization of convolutional neural networks for ECG classification using generative adversarial networks," *IEEE Access*, vol. 8, pp. 35 592–35 605, 2020.

[15] *Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020*, https://moody-challenge.physionet.org/2020/, Accessed: 2022-02-19.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[17] V. Turchenko and A. Luczak, "Creation of a Deep Convolutional Auto-Encoder in Caffe," in *2017 9th IEEE Int. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, IEEE, vol. 2, 2017, pp. 651–659.

[18] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, vol. 105, pp. 13–22, 2018.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

[21] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[22] R. Parikh, A. Mathai, S. Parikh, G. C. Sekhar, and R. Thomas, "Understanding and using sensitivity, specificity and predictive values," *Indian Journal of Ophthalmology*, vol. 56, no. 1, p. 45, 2008.