Project 3

Topic: Classification of ECG Arrhythmia using Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN)

Kimia Gholami, 903984681
Ph.D. Student Computer Science

Abstract:

This project aims to use Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) to classify ECG Arrhythmia. Then, the results of different models were used to compare the performance of each model. The popular MIT-BIH Arrhythmia Database is used in this project. The dataset is divided into two sub-data as training and testing datasets, with an 80-20 ratio. 80% for training and 20% for testing the models.

Introduction:

Electrocardiogram (ECG) is a sufficient way to diagnose and detect the abnormal condition of the heart [1]. It is used to diagnose heart and vessel disease [2-3]. The ECG signal (Figure 1) records the heart's bioelectrical activities [4]. Early-stage detection of heart disease improves the quality of living and could be used for appropriate treatment.

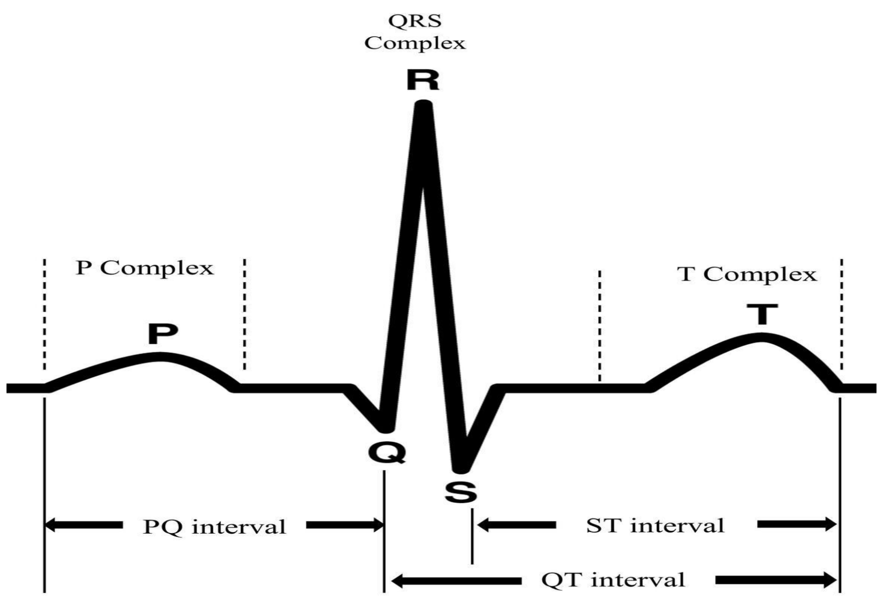


Figure 1. ECG Signal Curve

In this project, two different RNNs, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU), have been used to test the different RNN model's efficiency in classification and prediction. In addition, A CNN model has been used to compare the CNN and RNN results for this dataset. In this project, we have used different CNN and RNN models and modified them to classify the normal and abnormal ECG signals. Also, abnormal ECG signals have been classified into 5 classes representing 5 kinds of ECG signal abnormality to further improve the detection of abnormal heart conditions. Both CNN and RNN models have been tuned with this dataset to represent the best possible results in this project.

Dataset:

The MIT-BIH Arrhythmia database is used in this project for the performance evaluation of the models. This dataset contains normal and abnormal beats in an ECG signal. It is a widely used dataset for testing classification performance. It consists of 48 records, each half an hour in duration. The dataset is classified into 16 types of ECG beats (Figure 2). This dataset shows recording time series of heartbeats in an ECG signal form.

No	Symbol	Annotation Description
1	N	Normal beat
2	L	Left bundle branch block beat
3	R	Right bundle branch block beat
4	V	Premature ventricular contraction
5	/	Paced beat
6	A	Atrial premature contraction
7	f	Fusion of paced and normal beat
8	F	Fusion of ventricular and normal beat
9	!	Ventricular flutter wave
10	j	Nodal (junctional) escape beat
11	x	Non-conducted P-wave
12	a	Aberrated atrial premature beat
13	E	Ventricular escape beat
14	J	Nodal (junctional) premature beat
15	e	Atrial escape beat
16	Q	Unclassifiable beat
		Total 16-class beats

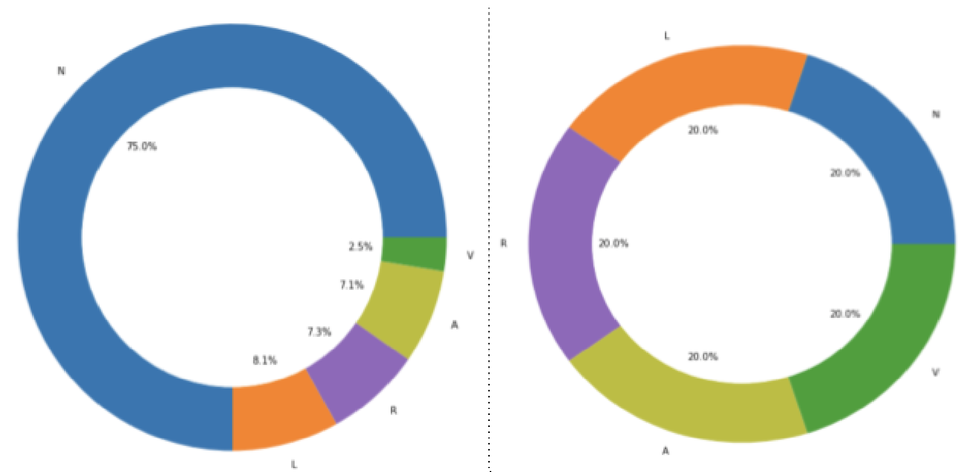
Figure 2. Sixteen classes in MIT-BIH dataset for ECG signals

Of these 16 classes, 5 core diseases have been selected for this project. The target classes are N: Normal beat, L: Left bundle branch block beat, R: Right bundle branch block beat, V: Premature ventricular contraction, and A: Atrial premature contraction.

The distribution of these five classes in the original data is imbalanced. Thus, 30,000 inputs were selected from the original dataset to have a uniform dataset and control the class flow used in this project. Figure 3 shows the class distribution both in the original dataset and the selected inputs. As it is indicated in Figure 3, the selected dataset contains 6,000 data of all classes uniformly. Each class includes 20 % of input data.

The dataset has been split into two sub-datasets, 80% for training and 20% for testing the model's accuracy. Out of 30,000 data, 24,000 have been used for training and 6,000 for evaluation.

The dataset is available online at:
https://www.kaggle.com/datasets/taejoongyoon/mitbit-arrhythmia-database
This dataset has often been used for classification and diagnosis purposes in many papers [5-7].



Class	Distribution	Number of Data
N	75.0%	75,011
L	8.1%	8,071
R	7.3%	7,255
A	7.1%	7,129
V	2.5%	2,546

Class	Distribution	Number of Data
N	20.0%	6,000
L	20%	6,000
R	20%	6,000
A	20%	6,000
V	20%	6,000

Figure 3. Class distribution: (Left) the original dataset. (Right) selected data

The ECG signals might have noises that make the data processing task difficult for the model. So, to further improve the dataset, preprocessing methods have been applied to the signals in the dataset. The z-score normalization and denoising methods have been performed on ECG signals. Then, the normalized and denoised ECG signals with balance class distribution were passed to the models to predict the classes.

Methodology Used:

Recurrent Neural Network (RNN): RNNs are a class of Artificial Neural Networks (ANN) where connections between nodes can create cycles [8-10]. This class of ANN uses the output of some nodes to process the input of the same subsequent. RNNs are known for their ability to analyze sequence data. The process of learning in this class of Neural Networks is shown in Figure 4. They can be used to process, learn and test by a dataset with the order. They can use their internal memory to process a sequence of inputs with time series, and the connections between units form a directed cycle.

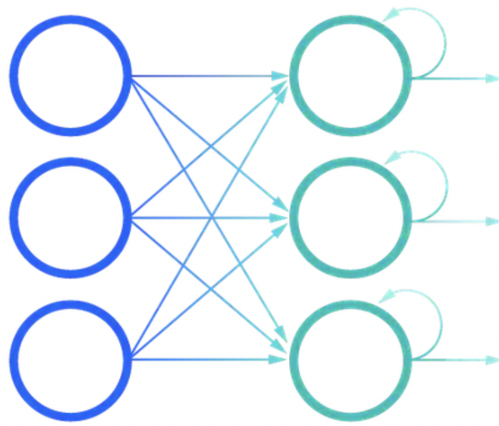


Figure 4. Recurrent Neural Networks

Bidirectional Recurrent Neural Network (BRNN): Bidirectional RNN is an RNN performing in both directions [11]. Figure 5 shows a BRNN with three-time stamps. They will use the RNN architecture in two ways. BRNN connects two hidden layers of opposite directions to the same output. They can improve the model's ability to learn further, especially when input context is needed for prediction. So, the output layer can process the information from the past and future in the data sequence [12].

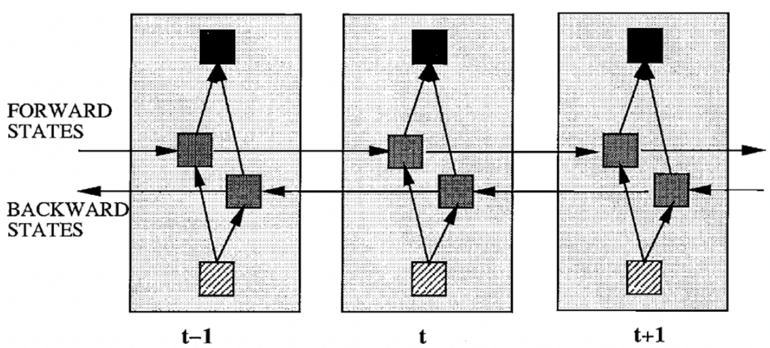


Figure 5. A BRNN with forward and backward RNNs

Long Short-Term Memory (LSTM): LSTM is among the most used RNNs [13]. They were proposed as a solution to the vanishing gradient problem. They are used to address long-term dependencies. They process the information with four units: cell, input gate, output gate, and forget gate. The cell remembers the values, and the three gates control the information flow needed to predict the output. The input and output gates are also known as the learning and use gates. The architecture of LSTM is shown in Figure 6.

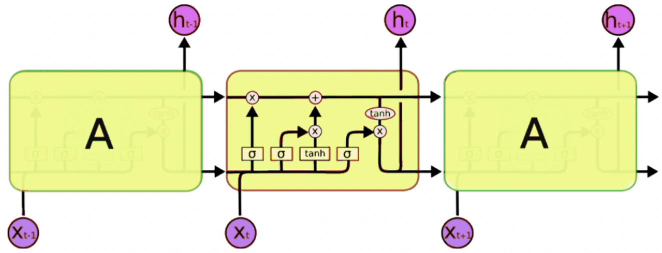


Figure 6. The architecture of LSTM

Model:

In this work, we want to evaluate the result of an RNN model like LSTM which is known for classifying sequential data, and compare its results with a CNN model. A Bidirectional LSTM layer that returns a sequence as output has been used in the RNN model to improve the model's ability to process the information in the sequence of ECG signals. The architecture of the RNN model used in this project is shown in Figure 7. The input has 360 sequences of 1D data for this model.

```
Model: "sequential_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
bidirectional_4 (Bidirection (None, 360, 20)           960
_________________________________________________________________
lstm_20 (LSTM)               (None, 128)               76288
_________________________________________________________________
dense_44 (Dense)             (None, 100)               12900
_________________________________________________________________
dropout_16 (Dropout)         (None, 100)               0
_________________________________________________________________
dense_45 (Dense)             (None, 50)                5050
_________________________________________________________________
dropout_17 (Dropout)         (None, 50)                0
_________________________________________________________________
dense_46 (Dense)             (None, 6)                 306
_________________________________________________________________
dense_47 (Dense)             (None, 5)                 35
=================================================================
Total params: 95,539
Trainable params: 95,539
Non-trainable params: 0
```

Figure 7. The architecture of the RNN model

First, a bidirectional LSTM layer and an LSTM layer were applied to the model. Then, three hidden layers with 100, 50, and 6 neurons were used, respectively. The activation function in the hidden layers is ReLU. Finally, A dense layer with 5 neurons is in the last layer to classify the five normal and abnormal ECG signals. The activation function in the last dense layer is the

Softmax function. As shown in the architecture, two dropouts of rate 0.1 are applied to some layers to prevent the model from overfitting.

Because the data has only one channel, the CNN layers used in the CNN model are convolutional 1D. The architecture of the CNN model is shown in Figure 8. As shown in Figure 8, the number of parameters used in the CNN model is not too large to compare its results with the RNN model.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d (Conv1D)              (None, 360, 16)           160
_________________________________________________________________
average_pooling1d (AveragePo (None, 179, 16)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 179, 24)           4248
_________________________________________________________________
average_pooling1d_1 (Average (None, 89, 24)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 89, 56)            17528
_________________________________________________________________
average_pooling1d_2 (Average (None, 44, 56)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 44, 96)            80736
_________________________________________________________________
average_pooling1d_3 (Average (None, 21, 96)            0
_________________________________________________________________
dropout (Dropout)            (None, 21, 96)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 21, 5)             4325
_________________________________________________________________
global_average_pooling1d (Gl (None, 5)                 0
_________________________________________________________________
softmax (Softmax)            (None, 5)                 0
=================================================================
Total params: 106,997
Trainable params: 106,997
Non-trainable params: 0
```

Figure 8. The architecture of the CNN model

Five 1D convolutional layers have been used in the CNN model with 360, 179, 89, 44, and 21 filters. The activation function in all convolutional layers is ReLU. In addition, an average pooling layer was applied after each convolutional layer. In the last layer, a softmax layer has been added to the model to improve the results. Also, a dropout of rate 0.5 is applied in the model.

The models have been trained and tested with 100 epochs. The loss function used in this work is cross entropy for its ability to calculate the loss in classification. The optimizer used in both models is adam.

Results:

The CNN and RNN models have been trained for 100 continuous epochs. 24,000 data have been used for training and 6,000 data for testing. Figure 9 shows the result and accuracy for the RNN model, and Figure 10 shows the results for the CNN model. The results are also described in the confusion matrix for both models. The accuracy with both the RNN and the

CNN models is 98%. These results on the same validation dataset indicate that the CNN model was as proficient in predicting the classes for ECG signals as the RNN model.

	precision	recall	f1-score	support
N	0.96	0.96	0.96	1179
L	0.99	0.98	0.99	1244
R	0.99	0.99	0.99	1216
A	0.97	0.97	0.97	1171
V	0.98	0.99	0.98	1190
accuracy			0.98	6000
macro avg	0.98	0.98	0.98	6000
weighted avg	0.98	0.98	0.98	6000

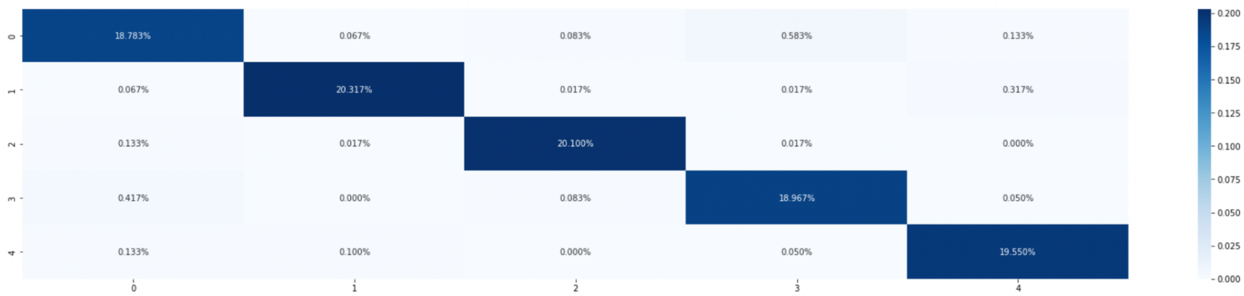


Figure 9. The results and the confusion matrix for the RNN model

As shown in the confusion matrix for both RNN and CNN models, the models successfully predicted the five ECG classes.

	precision	recall	f1-score	support
N	0.98	0.95	0.96	1179
L	1.00	1.00	1.00	1244
R	0.99	1.00	0.99	1216
A	0.95	0.98	0.96	1171
V	1.00	0.98	0.99	1190
accuracy			0.98	6000
macro avg	0.98	0.98	0.98	6000
weighted avg	0.98	0.98	0.98	6000

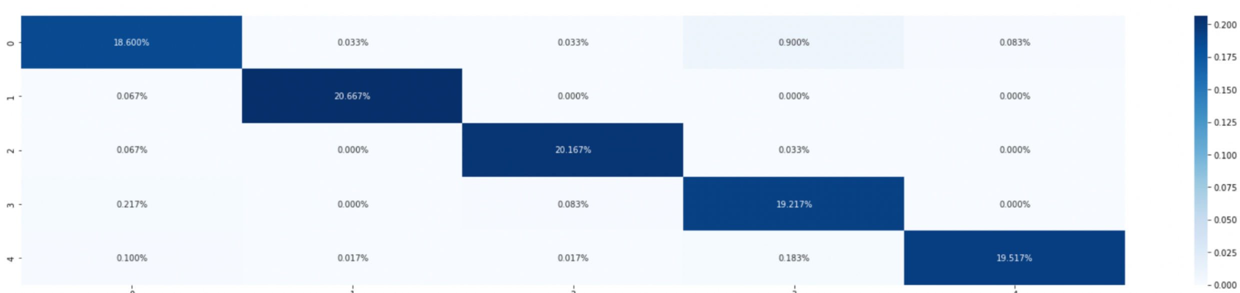


Figure 10. The results and the confusion matrix for the CNN model

Conclusion:

This work applied different models to classify the ECG signals and detect heart disease. Our results show that both the CNN and RNN architecture models had outstanding results for classification. The accuracy calculated for these models does not show a significant difference in the performance of these models. We aim to use Neural Network models in health care and improve the health organizations' ability to detect heart disease at its early stages and improve the quality of human life. However, furthermore, research could be done in the future to either improve the models. A possible path for more research could be to compare different model results to find an even better fit model for heart disease detection purposes. Even testing with different hidden layers and filters could benefit this goal. Another possible path could be applying more complicated layers with more neurons to test if complex models could improve the results for this task.

References:

1. V.M. Uspenskiy. Information function of the heart. a measurement model. MEASUREMENT 2011, 1:383-386, 2011.
2. Wagner Galen and Strauss David. Marriotts Practical Electrocardiography. LWW, 2013.
3. K. Perloff Joseph and J. Marelli Ariane. Clinical Recognition of Congenital Heart Disease. Elsevier, 2012.
4. V.M. Uspenskiy. Diagnostic system based on the information analysis of electrocardiograph. In Embedded Computing (MECO), 2012 Mediterranean Conference on, pages 74-76, June 2012.
5. R. D. Raut and S. V. Dudul, Arrhythmias classification with MLP neural network and statistical analysis, Proc. - 1st Int.
   Conf. Emerg. Trends Eng. Technol. ICETET 2008, pp. 553-558, 2008.
6. B. M. Asl, S. K. Setarehdan, and M. Mohebbi, Support vector machine-based arrhythmia classification using reduced
   features of heart rate variability signal, Artif. Intell. Med., vol. 44, no. 1, pp. 5164, 2008.
7. G. Selvakumar, K. B. Bagan, and B. Chidambararajan, Wavelet Decomposition for Detection and Classification of Critical
   ECG Arrhythmias, pp. 8084, 2007.
8. Dupond, Samuel (2019). "A thorough review on the current advance of neural network structures". Annual Reviews in Control. 14: 200–230.
9. Abiodun, Oludare Isaac; Jantan, Aman; Omolara, Abiodun Esther; Dada, Kemi Victoria; Mohamed, Nachaat Abdelatif; Arshad, Humaira (2018-11-01). "State-of-the-art in artificial neural network applications: A survey". Heliyon. 4 (11): e00938.
10. Tealab, Ahmed (2018-12-01). "Time series forecasting using artificial neural networks methodologies: A systematic review". Future Computing and Informatics Journal. 3 (2): 334–340.
11. Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681.2. Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan
12. Salehinejad, Hojjat; Sankar, Sharan; Barfett, Joseph; Colak, Errol; Valaee, Shahrokh (2017). "Recent Advances in Recurrent Neural Networks". arXiv:1801.01078. Bibcode:2018arXiv180101078S.

13. Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780.
14. P. de Chazal, M. O'Dwyer, and R. B. Reilly, Automatic classification of heartbeats using ECG morphology and heartbeat interval features, IEEE Trans.BiomedEng., 51(7), 2004, 1196-1206.
15. Laura Falaschetti, Michele Alessandrini, Giorgio Biagetti, Paolo Crippa, Claudio Turchetti; "ECG-Based Arrhythmia Classification using Recurrent Neural Networks in Embedded Systems". Procedia Computer Science. Volume 207. 2022. P 3479-3487.