IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Electrocardiogram Classification For Cardiac Arrhythmias Using Convolutional Neural Network

Uddhav Bhardwaj

INTRODUCTION

1.1 Objective

We use a deep neural network to classify an electrocardiogram (ECG) because it is one of the most essential biomedical signals. The main objective of this project is to build a model that utilizes a deep neural network (CCN here) to classify the ECG signals by preprocessing the raw signal to identify Cardiac Arrhythmias. We choose the CNN classifier because of its capacity to recognize relevant features automatically without the need for human supervision or feature selection. Because of its basic nature, it could also be easily implemented in a hardware platform. This classifier can automatically generalize similarity-based features and can approximate functions.

1.2 Motivation

One of the significant reasons of death on the planet is accounted to Cardiovascular illnesses (CVDs). Some of the known CVDs are cerebrovascular disease, rheumatic heart disease, coronary artery disease, and various other blood vessel and heart-based illnesses. Arrhythmias are generally innocuous, but some can be harmful or even deadly. If the heartbeat is excessively rapid, too slow, or irregular, the heart may not be able to provide sufficient blood to amount to the body. Some of the popular Arrhythmias are ventricular fibrillation, premature contraction, atrial fibrillation, and tachycardia. Your heart has an irregular, sometimes rapid rhythm when you have atrial fibrillation, a kind of cardiac arrhythmia. A stroke, heart failure, or other heart disorders can all be aggravated by this disease. To manage and prevent CVDs, it is very crucial to monitor the cardiac rhythms regularly. A range of smart devices is now available to help monitor the heart's beats. As a result, in the field of cardiology, the automatic detection of abnormal cardiac rhythms from ECG readings is an important task.

1.3 Background

The ECG signal holds very high significance in the biomedical field. It is used to identify many popular CVDs such as heart attacks, coronary heart diseases, arrhythmias, etc. Cardiac arrhythmia is a condition where the heart could thump exorbitantly sluggish, excessively fast, or sporadic. One of the significant purposes behind death on the planet is accounted to cardiovascular diseases (CVDs). Factually, more than 17.9 million individuals pass on every year because of these sicknesses. The clinical term for a whimsical or erratic heartbeat is Cardiac arrhythmia. At the point when the electrical signs that oversee the heart's beats fall flat, this happens. Arrhythmias are generally innocuous, but some can be harmful or even

deadly. As a result, it is critical to monitor cardiac rhythms regularly to manage and prevent CVDs. The purpose of this study is to classify cardiac arrhythmias using CNNs. This project seeks to use a Convolutional Neural Network to categorize electrocardiograms for Cardiac Arrhythmias to reduce the risk of permanent heart damage. We have used the MIT-BIH database on cardiac arrhythmias as our ECG dataset for this project. This database contains 48 30-minute samples with two ECG leads and 25 samples with unusual but clinically significant arrhythmias. To summarize the steps undertaken in short, the data retrieved from the MIT-BIH database is first preprocessed which is then resampled to remove any bias
introduced by the imbalanced dataset. Now, this dataset is in turn split into two more datasets:
train and test. Then, a CNN model is created utilizing the training dataset. The test dataset is then used to evaluate the model, and thus more conclusions are drawn.

2. PROJECT DESCRIPTION AND GOALS

We started this project with the main aim of categorizing ECGs with the MIT-BIH database
which has data on cardiac arrhythmias. We import the required data from the MIT-BIH
database first and then we preprocess the ECG data to turn the raw data into the required form.
In the dataset, 15 distinct forms of arrhythmia were grouped into five superclasses. I chose six sorts of beats from these five superclasses to be a part of the training and assessment sets, and they are N, L, R, A, V, and/. The data in the dataset is imbalanced, with the number of regular beats significantly outnumbering the number of abnormal beats. The machine learning model will tend towards the majority set in the database if we train imbalanced data. To avoid this, we used resampling, which adds more data points to the dataset, resulting in an equal number of normal and abnormal beats after resampling. After resampling, we separated the
dataset into training and tested for it to be used in the data sets in an 80:20 ratio. Later, we
convert the Non-categorical data to categorical data so that the model can understand it. The TensorFlow and Keras libraries are then used to create a model. To maintain the dataset's flexibility, a complicated convolutional neural network model with more than ten layers and layers such as pooling is developed. Our biggest feature is that each of the five layers of CNN
has its own set of filters or feature detectors. These layers simultaneously have kernel sizes
that depict the length of the 1D convolution window. We have likewise utilized the ELU
(exponential linear unit) activation function for this layer. The dimensions of the training dataset are demonstrated by the input shape. A standardization/normalization layer follows every convolution layer, which normalizes the inputs by keeping the mean output near 0 and the output's standard deviation closer to 1. Three pooling layers are used in the model to downscale the image and extract the most significant information. To enable flexibility, it pulls the most value from the feature map based on filter size and strides. The model also includes a flatten layer that is used to reformat the output by flattening the input tensors into a single dimension. Final layers are dense layers that transmit all of the preceding layer's outputs to all of its neurons, with each neuron sending one response/output to its consecutive layer. We build this model with optimizer adam after creating it because it is the finest adaptive optimizer. The accuracy measure, essentially calculates how many predictions match labels and is then used to calculate accuracy. After this, on the off chance that the epochs show no improvement even after the training, the training will terminate, and a model checkpoint will be added, which also will be the callbacks to save the Keras model or model weights at a certain frequency so that later, the model or weights can indeed be retrieved to resume training from a stored state. After adding callbacks, the model is trained using the fit method which trains the model for a fixed number of epochs. This model is hence evaluated and later confusion matrix is plotted using an evaluate model function.

After incorporating callbacks, the model is trained with the fit method, which runs for a specified number of epochs. Thereafter, we utilize the evaluate-model function to assess the model and then display a confusion matrix. The objective of this project was to create a model that uses a deep neural network (CCN here) to classify the ECG signals, at which we are finally successful.

Serial No.	Superclass Type	Classes included
1	N (Normal)	N L R
2	SVEB (Supraventricular ectopic beat)	A a J S e j
3	VEB (Ventricular ectopic beat)	V E
4	F (Fusion beat)	F
5	Q (Unknown beat)	P / f u

Table1. Superclass and their corresponding classes

Serial No.	Symbol for Beat	Name for Beat
1	N	Normal beat
2	L	Left bundle branch block beat
3	R	Right bundle branch block beat
4	A	Atrial premature beat
5	V	Premature ventricular contraction
6	/	Paced beat

Table2. Symbol of beat and its corresponding names

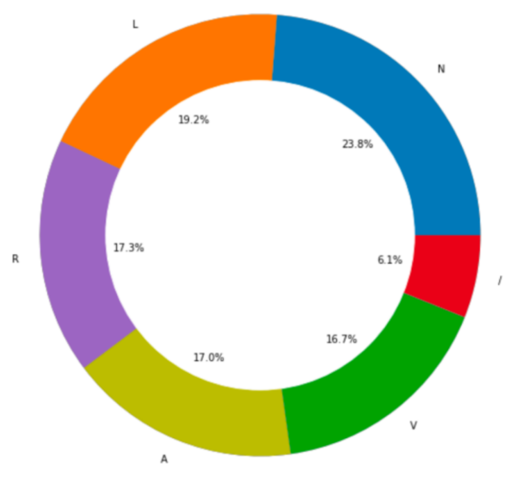

Fig1. Distribution of beats in the dataset

3. TECHNICAL SPECIFICATION

a) H/W Requirements (details about Application-Specific Hardware) -

The whole program is written in python and requires a laptop or a computer to run it. The specs of my laptop, which is used to execute the project, are as follows:

CPU: Intel i7-8750H @ 2.20GHz 2.21 GHz
GPU: Nvidia GeForce RTX 1660 Ti 8 GB
RAM: 8 GB
Storage: 1 TB
OS: Windows 11

b) S/W Requirements (details about Application-Specific Software) -

The program uses Python and its implementation using an anaconda launcher to access the Jupyter notebook to provide a runtime environment for the code. Some of the Libraries used are1) Tensorflow Core - 2.9.1
2) Scikit-Learn - 1.1.1 3) Keras - 2.9.0
4) Numpy - 1.22.4
5) Matplotlib - 3.5.2

4. DESIGN APPROACH AND DETAILS

4.1 Design Approach / Materials & Methods

The most significant requirement was to create the CNN Model in such a way that it produced the highest level of accuracy. We used 11 layers in the model after a lot of trial and error, and it turned out to be the most accurate of all the experiments.

This is the first convolutional layer, and it produces a tensor of outputs by convolving the layer input with the convolution kernel over a single spatial or temporal dimension. This layer has 8 filters or feature detectors, as well as a kernel size of 3 for the 1D convolution window. It also contains the ELU (exponential linear unit) activation function, which itself is employed
since it has -ve values and allows us to reduce mean unit activation to zero, the same as batch
normalization, the only difference is that this has a lesser processing complexity. The dimensions of the training dataset are indicated by the input shape. After the convolutional layer, we've introduced a layer that thus normalizes the inputs. The batch normalization process uses a transformation to keep the mean output around 0 and the output standard deviation near 1. The second layer is max pooling, which is a downscaling method that extracts the most essential characteristics from an image. Then max-pooling is employed to recover the maximum worth out from the feature map in view of the filter size and strides to give more adaptability. The max-pooling window, or pool size, is 3 pixels wide, and the
strides, or how far the pooling window moves for each pooling step, are 2. The padding is the
same, which means that padding is evenly distributed across the width and the height in such a way that the output has similar dimensions as the given input. The third and fourth layers are additionally convolutional, but this time with 64 and 128 filter layers respectively, and a kernel size of 3. The fifth layer, like the second, is a pooling layer with parameters similar to that of the second but here the pool size is two. The sixth and seventh layers are again CNN layers except that these have 256 filter layers. These layers have a kernel size of three. The eighth layer has mostly the same features as the fifth, as it is a pooling layer. The convolution layer and pooling layer are repeated to increase the number of hidden layers in the model and forecast the output as accurately as feasible. The ninth layer is a flattened layer that is used to restructure the output while also flattening the input. It has no effect on the batch size The flatten function reduces multi-dimensional input tensors to a single dimension, allowing us to model our input layer and build our neural network model, then effectively send those inputs

to each and every neuron in the model. The model's tenth layer is dense because it feeds all of the preceding layer's outputs to all of its neurons. A single response/output is sent from each neuron to the consecutive layer. An ELU activation function with a dimensionality of
1024 is then applied to the output space. Because it is the most basic layer of a neural network,
function. We utilize the SoftMax function for scenarios that require multi-class classification with multiple class labels. We build this model with the Adam optimizer after creating it because it is the best adaptive optimizer in usual scenarios and is generally used as a replacement technique for SGD (stochastic gradient descent) in model training. Categorical cross-entropy is picked as the loss function as it is considered valuable for delivering the cross-entropy loss that occurs between various labels and mainly because it is a loss function utilized in multi-class classification scenarios. The accuracy measure is used to calculate how often predictions match labels. Then callbacks are defined as objects that can perform actions at different phases of training and are required to keep the code maintained. When a monitored metric stop improving, an early stopping class is added to end training. The quantity to watch is val loss, which is the cost function's value for our cross-validation data. In this specific scenario, the patience parameter is set to eight, which would be the number of consecutive epochs with no progress beyond which training will now be ended. A model checkpoint is added, which callbacks to store the Keras model or model weights at a predetermined frequency, enabling either model or weights to somehow be loaded later to begin training from a previous condition The path to store the model file is called the file path. The restrictions imposed are saved best only, which always saves whenever the model is regarded as the best and the current best model based on the amount measured will not be overridden. Following the addition of callbacks, the model is prepared(or trained) utilizing the fit technique, which runs for a pre-defined number of epochs. When using evaluate model function, the model is then assessed as well as a confusion matrix is plotted.

Serial No.	Attribute Name	Attribute Explanation
1	MLII	The signal from Lead 2 of the ECG
2	Type	The type of beat
3	Sample #	The sample number of beat

Table3. Name of attributes and their explanations

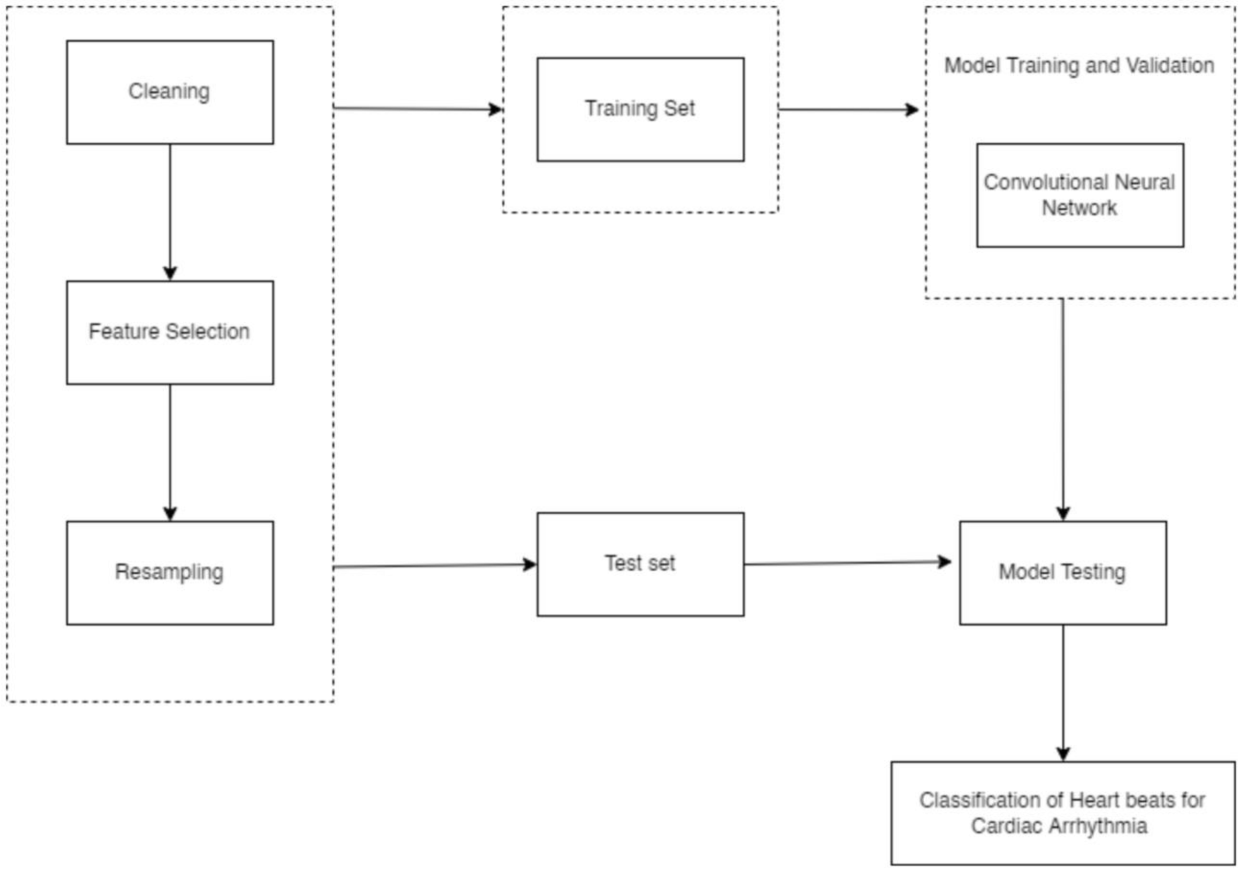

Fig2. Architecture of the project

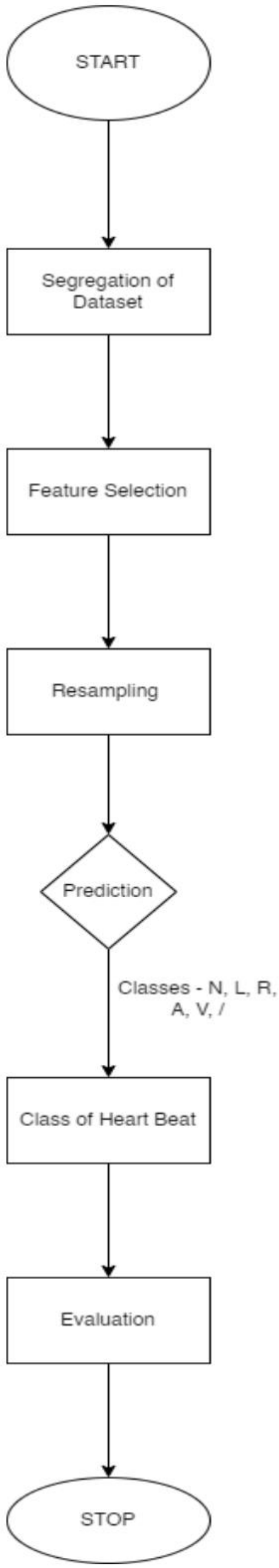

Fig3. Flowchart of the project

4.2 Codes and Standards.

The system must follow certain regulated industrial standards to be able to perform as expected without any glitches. To make sure our system is as bug-proof as possible, we have tried to use the most stable versions of all pip packages such that they function in sync with each other and don’t affect the functioning of one another. We've additionally cross-checked that the code in the .ipynb file is of the highest quality by following the principle of loosely coupling among various cells of the file, such that each cell has inter-dependent code that’s not much affect other cells, as a result, the code has a suitable structure, which improves efficiency and comprehension for the user.

4.3 Constraints, Alternatives, and Tradeoffs.

In this project, there were various restrictions while dealing with an uneven dataset. We have tried to implement resampling to be able to use an imbalanced dataset. It is also important to note here that even resampling has drawbacks, especially when the target class is highly skewed. To begin with, oversampling the minority class may result in overfitting, which means that the model learns patterns that only exist in the oversampled sample. Moreover, under-sampling the majority class might lead to underfitting, which is bad as the main pattern of the data remains undiscovered by the model. We have taken specific efforts in this project to avoid overfitting. When dealing with an uneven dataset, other assessment metrics might be applied. We were able to use the resources we had on hand, so we didn't have to rely on other options or make any compromises.

5. COST ANALYSIS / RESULT & DISCUSSION

The main objective of this project was to classify cardiac arrhythmias using CNNs. Before using a convolutional neural network to develop a model, our technique preprocesses and resamples the dataset. It was shown that the model did attain a 99.4 percent overall accuracy. When evaluated against a single ECG file, we were able to provide details based on the learned model. Eventually, we, at last, made a confusion matrix to cross-verify the representations/ visualizations of the expectations.

```
Accuracy: 99.43%
<keras.callbacks.History object at 0x000002AABF85B610>
```

Fig5. Accuracy

```
Based on the Model trained following are the details for the given ECG
Total number of beat in the given ECG =  2271
Number of Normal beat in the given ECG =  2235
Number of Normal beat in the given ECG =  98.41
Number of Abnormal beat in the given ECG =  36
Number of Abnormal beat in the given ECG =  1.59
```

Fig6. Result of the single ECG record

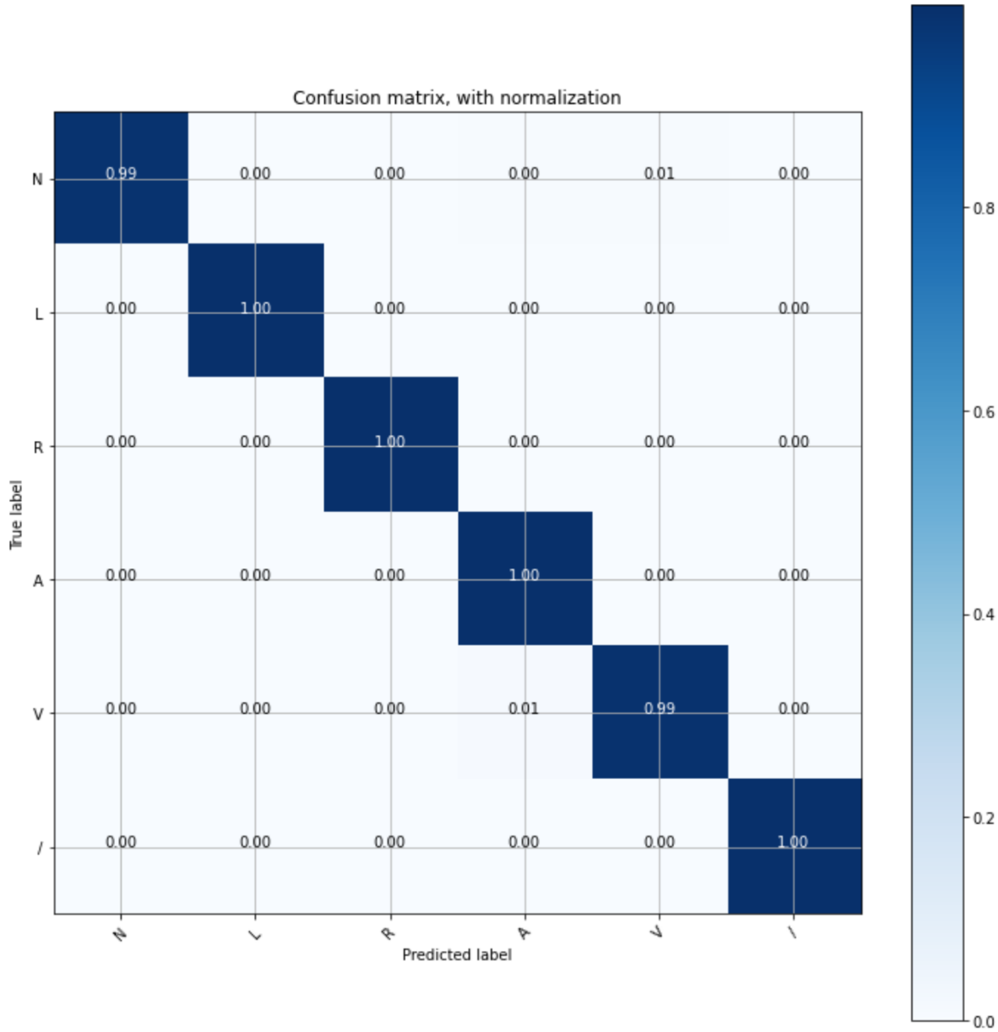

Fig7. Confusion Matrix

6. SUMMARY

The main objective of this project was to classify cardiac arrhythmias using CNNs. Before using a convolutional neural network to develop a model, our technique preprocesses and resamples the dataset. The dataset has been split in an 80:20 ratio so that they can be trained and then tested respectively. A particular ECG file is also evaluated using the learned model. It was shown that the model did attain a 99.4 percent overall accuracy. We believe that our trained CNN model can be used for real-life applications with a slight adaptive adjustment before use. As of now for the algorithm to work we require the user to put the input data in a specific format, namely that consisting of sample ID, MLII, and type in the certain file at a certain row number. This might work for now and provide wonderful results as suggested by the accuracy percentage. In the future, we however can improve by making the input more flexible. By allowing the user to put the data in any format (meaning, we would require the attributes but they should be just present in the table irrespective of the position. This would make the algorithm more efficient in terms of user experience without harming the accuracy.

7. REFERENCES

[1] P. A. P. S. S. R. C. Venkatesan, "ECG signal preprocessing and SVM classifier-based abnormality detection in remote healthcare applications," IEEE Access, vol. DOI 10.1109/ACCESS.2018.2794346, p. 8, 2018.

[2] M. I. O. T. I. M. I. J. H. M. F. I. a. S. T. S. M. I. Carson A. Wick, "Seismocardiography-Based Detection of Cardiac Quiescenc," IEEE Transactions on Biomedical Engineering, vol. DOI 10.1109/TBME.2015.2411155, p. 8, 2015.

[3] S. K. S. R. K. (. M. I. D. M. (. I. CHANDAN NAYAK, "Efficient Design of Zero- Phase Riesz Fractional Order Digital Differentiator Using Manta-Ray Foraging Optimization for Precise Electrocardiogram QRS Detection," IEEE open journal for Circuits and Systems, vol. Digital Object Identifier 10.1109/OJCAS.2020.3035771, p. 13, 10 December 2020.

[4] I. J. H. M. (. I. R. (. I. A. S. T. (. M. I. CARSON A. WICK (Student Member, "A Potential-Based Inverse Spectral Method to Non-Invasively Localize Discordant Distributions of Alternans on the Heart from the ECG," IEEE Transactions on Biomedical Engineering, vol. DOI 10.1109/TBME.2017.2732159, p. 11, 2017.

[5] (. M. I. J. H. M. (. I. R. (. I. A. T. (. M. I. CARSON A. WICK, "Detection of Cardiac Quiescence from B-Mode Echocardiography Using a Correlation-Based Frame-to- Frame Deviation Measure," IEEE journal of Translational Engineering in Health and Medicine, vol. Digital Object Identifier 10.1109/JTEHM.2013.2291555, p. 11, 10 November 2013.

[6] C. L. Y. L. Y. L. J. W. B.-S.L. (. M. I. A. J. L. ZHONGYAO ZHAO, "Noise Rejection for Wearable ECGs Using Modified Frequency Slice Wavelet Transform and Convolutional Neural Networks," IEEE Access, vol. Digital Object Identifier 10.1109/ACCESS.2019.2900719, p. 8, March 21, 2019.

[7] C. W. (. I. Y. Z. A. C. Y. ZHANQUAN SUN, "Multi-Label ECG Signal Classification Based on Ensemble Classifier," IEEE Access, vol. Digital Object Identifier 10.1109/ACCESS.2020.3004908, p. 11, July 7 2020.

[8] I. A. T. G. (. I. A. N. M. K. (. M. I. ZEESHAN AHMAD (Graduate Student Member, "ECG Heartbeat Classification Using Multimodal Fusion," IEEE Access, vol. Digital Object Identifier 10.1109, p. 12, July 22, 2021.

[9] H. L. XUEXIANG XU, "ECG Heartbeat Classification Using Convolutional Neural Networks," IEEE Access, vol. Digital Object Identifier 10.1109/ACCESS.2020.2964749, p. 6, January 15, 2020.

[10] S. M. I. a. S. G. K. S. M. I. Wei Jiang, "Block-Based Neural Networks for Personalized ECG Signal Classification," IEEE TRANSACTIONS ON NEURAL NETWORKS, vol. 18, no. 6, p. 12, NOVEMBER 2007.
24
24/25

[11] M. I. B. a. M. S. M. M. I. Udit Satija, "A New Automated Signal Quality-Aware ECG Beat Classification Method for Unsupervised ECG Diagnosis Environments," IEEE Sensors Journal, vol. DOI 10.1109/JSEN.2018.2877055, p. 9, 2018.

[12] D. I. F. M. I. a. L. K. M. Themis P. Exarchos Costas Papaloukas, "An Association Rule Mining-Based Methodology for Automated Detection of Ischemic ECG Beats," EEE TRANSACTIONS ON BIOMEDICAL ENGINEERING , vol. 53, no. 8, p. 10, AUGUST 2006.

[13] S. M. I. M. M. a. B. B. C. F. I. Sucharita Mitra, "A Rough-Set-Based Inference Engine for ECG Classification," IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT, Vols. DECEMBER , VOL. 55, no. NO. 6, p. 9, 2006.

[14] T. I. a. M. G. F. I. Serkan Kiranyaz, " Real-Time Patient-Specific ECG Classification by 1D Convolutional Neural Networks," IEEE Transactions on Biomedical Engineering, vol. DOI 10.1109/TBME.2015.2468589, p. 8, 2015.

[15] S. M. M.-W. M. S. M. a. C.-C. C. S. M. Sean Shensheng Xu, " Towards End-to-End ECG Classification with Raw Signal Extraction and Deep Neural Networks," JOURNAL OF LATEX CLASS FILES, vol. 14, no. 8, p. 11, AUGUST 2015.

[16] A. S. A. F. S. S. M. I. J. F. a. R. S. S. F. I. Nilanjan Dey, "Developing Residential Wireless Sensor Networks for ECG Healthcare Monitoring," IEEE Transactions on Consumer Electronics, vol. 63, no. 4, p. 8, November 2017.

[17] S. M. I. K. F. T. S. M. I. H. R. C. S. M. I. B. W. K. L. S. M. I. a. C. K. W. Kwok Tai Chui, "An Accurate ECG Based Transportation Safety Drowsiness Detection Scheme," IEEE Transactions on Industrial Informatics, Vols. DOI 10.1109/TII.2016.2573259, , p. 15, 2016.

[18] H. A.-N. Khaled Assaleh, "A Novel Technique for the Extraction of Fetal ECG Using Polynomial Networks," IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, vol. 52, no. 6, p. 5, JUNE 2005.

[19] S. M. I. S. T. S. M. I. W. M. I. a. P. T. B. M. I. J. Yao, "Seismocardiography-Based Cardiac Computed Tomography Gating Using Patient-Specific Template Identification and Detection," IEEE Journal of Translational Engineering in Health and Medicine, Vols. DOI 10.1109/JTEHM.2017.2708100,, p. 14, 2017.

[20] Y. S. D. W. W. Y. A. Y. S. GONG ZHANG, "Automated Detection of Myocardial Infarction Using a Gramian Angular Field and Principal Component Analysis Network," IEEE Acsess, p. 14, December 2019.