IJCRT.ORG ISSN : 2320-2882

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

AN ARRHYTHMIA CLASSIFICATION METHOD BASED ON CONVOLUTIONAL NEURAL NETWORKS INTERPRETATON OF ELECTROCARDIOGRAM IMAGES

Teancy Jennifer R
Student, 8th sem, CSE,
CITech, k.r puram,
Bangalore, Karnataka.

Sehba Seher
Student, 8th sem, CSE,
CITech, k.r puram,
Bangalore, Karnataka.

Khushal Vaishnav
Student, 8th sem, CSE,
CITech, k.r puram,
Bangalore, Karnataka.

Manish. S
Student, 8th sem, CSE,
CITech, k.r puram,
Bangalore, Karnataka.

Prof. Shilpa S. B
Assistant Professor,
Dept. of CSE., CITech,
Bangalore, Karnataka.

Abstract: A new method for classifying cardiac abnormalities is here proposed based on the electrocardiogram (ECG). The ECG may manifest abnormal heart patterns, which are generally known as arrhythmias. MIT-BIH arrhythmia database and AAMI standards are used for machine learning purposes considering the patient-oriented scheme. Heartbeat time intervals and morphological features processed by a 2-D time-frequency wavelet transform of ECG signals are combined into an image, which carries relevant information from each heartbeat. These dataset images are used as input to train and evaluate the classifier, which is essentially a 6 layers convolutional neural network(CNN),resulting in powerful artifact discrimination. The training set is artificially augmented to reduce the imbalance of the five heartbeat classes, achieving better results. A significant achieved overall accuracy of 95.3% of the proposed method, compared to some of the most relevant published methods, permits to expect effective results when applied to real patients.

Index Terms **- ECG classification, feature extraction, wavelet transform, convolutional neural network**

I. INTRODUCTION

Heart bioelectricity study has always been of keen importance and dates back to the 19th century as a way of understanding the heart physiological performance and function. Electrocardiogram is the recording of the heart bioelectrical activity, which has evolved over the years into a powerful tool for assessing heart pathologies due to its easily recognized characteristic signature. Cardiac abnormalities reveal themselves as slight ECG visual artifacts, which can conduct to automatic heart diagnosis through signal processing methods. Considering that heart dysfunctions may imply life-threatening possibilities, careful procedures for automatic diagnosis methods constitute nowadays an important research area. ECG artifact classification may conduct to heart diagnosis through the application of machine learning methods. Classifiers need to be trained in order to learn how to recognize its classes, and only after arriving at an acceptable performance level it may then be applied to interpret real ECG signals. A report from the Association for the Advancement of Medical Instrumentation (AAMI) suggests using some specific internet available ECG signals databases with classification based on five cardiac arrhythmias: normal beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V),fusion beats (F), and unclassified beats (Q). This standardization proposal has become widely adopted in recent studies due to propitiate a good comparison among methodologies and results. Two approaches have been used for grouping the training labeled sample: beat-oriented or patient-oriented. Beat-oriented is a well-known approach where the dataset division is taken considering the heartbeat classes. Patient oriented approach deals with heartbeats from different subjects for training and testing datasets.

This method was proposed and presents the advantage of faithfully portraying the scenario if compared to the previous approach. Automatic heartbeat clustering is usually achieved through a sequence of feature extraction, classification, and evaluation. There have been used several feature extraction techniques, but the main one among them is the heartbeat intervals to discriminate heartbeat types. Wavelet transforms (WT) are used as a feature extraction method conducting the exposing of information simultaneously from both frequency and time domains. Different classification methods have been used, such as support vector machines (SVM), artificial neural networks (ANN), linear discriminant (LD), decision trees, convolutional neural networks (CNN), among others.

The AAMI recommended databases are often used for evaluating automatic arrhythmia classification methods, such as AHA DB (The American Heart Association Database for Evaluation of Ventricular Arrhythmia Detectors) and MIT-BIH Arrhythmia Database, which is the most used one in the literature. A heartbeat classification new approach is here presented. The patient-oriented division is adopted as the method for training the classifier. A new method of classifying images is here investigated through a compact CNN. Time features are extracted from the heartbeat intervals while morphological features are represented by WT scales. These features are combined into images to be clustered in the classification stage. Wavelet transforms generate a 2-D image of heartbeats that concentrate both time and frequency information about the patient ECG. This processed image is submitted to the trained CNN scrutiny that outputs the respective classification of each heartbeat. The MIT-BIH Arrhythmia Database is used to evaluate the performance of the proposed system considering the AAMI recommended classes. Up to the knowledge of the authors, this is the first study that implements a heartbeat classification using wavelet analysis to transform 1-D heartbeat signals in 2-D images as input to a CNN architecture.

II. PROTOTYPE

Several methods have been proposed for arrhythmia classification. Khorrami et.al. [12] proposed the use of the continuous wavelet transform in ECG arrhythmia classification. The two classifiers used were MLP that was trained using backpropagation (BP) and SVM . Along with the normal ECG beat, four different arrhythmia types were taken from the MIT arrhythmia database. Ten types of ECG arrhythmia obtained from MIT database were used.

According to the results, the proposed approach achieved an average accuracy of 99.33%. Arif et.al [15] presented a fuzzy K-nearest neighbor approach for arrhythmia beats classification. They achieved a classification accuracy of 97%. Elly Matul et. al. The classification was performed using 12 types of arrhythmia that also included the normal beat using the MIT arrhythmia database. The Accuracy obtained in this study was 92%. ECG signals with five heartbeat classes obtained from the MIT database were used. The overall accuracy that was obtained from the proposed system was 99.8%. In [19] a genetic-SVM approach for cardiac arrhythmia classification is described. Four types of arrhythmia obtained from the MIT database are classified with a success rate of 95%. Finally, in [20], cardiac arrhythmia classification is achieved by a generalized feed-forward neural network classifier.

The classification accuracy obtained in this paper is 82.35%. To the best of our knowledge, the only publication that utilized a hierarchical classification model for arrythmia classification is [21]. The authors used several evolutionary algorithms to classify six types of arrhythmia that they obtained from the UCI repository [22]. This database had 452 instances (i.e. heart beats). They reported overall accuracy rates in the range of 98%. However, given the fact that the classes were skewed, in the sense that the normal class had 245 beats and the abnormal classes had a number of beats ranging from 50 to 15, it is difficult to analyze the reported results because using sensitivity and specificity is necessary to measure the performance in terms of identifying positive and negative labels. We have designed and implemented a comprehensive study that uses the MIT database with 11 classes and features extracted from 108,232 heart beats. Moreover, this paper reports and compares 4 machine learning techniques via sensitivity, specificity, and accuracy.

III. PROPOSED SYSTEM

Problem Statement

Predict if a heartbeat from a ECG signal has an arrhythmia for each 6 second window centered on the peak of the heartbeat. To simplify the problem, we will assume that a QRS detector is capable of automatically identifying the peak of each heartbeat. We will ignore any non-beat annotations and any heart beats in the first or last 3 seconds of the recording due to reduced data. We will use a window of 6 seconds so we can compare the current beat to beats just before and after. This decision was based after talking to a physician who said it is easier to identify if you have something to compare it to.

Proposed Work

By analyzing the electrical signal of each heartbeat, i.e., the combination of action impulse waveforms produced by different specialized cardiac tissues found in the heart, it is possible to detect some of its abnormalities. In the last decades, several works were developed to produce automatic ECG-based heartbeat classification methods. In this work, we survey the current state-of-the-art methods of ECG-based automated abnormalities heartbeat classification by presenting the ECG signal preprocessing, the heartbeat segmentation techniques, the feature description methods and the learning algorithms used.

IV. HARDWARE AND SOFTWARE SPECIFICATIONS

a. *Hardware specifications:*

CPU : Intel 2.1 GHZ
Memory : 4GB
Disk : 100GB
Display : 15 inch color

b. *Software specifications:*

Coding : Python
Platform : python 3.7
Tool : Spyder
OS : Windows 7

c. *Resource requirements:*

SPYDER

Spyder, the Scientific Python Development Environment, is a free integrated development environment (IDE) that is included with Anaconda. It includes editing, interactive testing, debugging, and introspection features. After you have installed Anaconda, start Spyder on Windows, macOS, or Linux by running the command spyder. Spyder is also pre-installed in Anaconda Navigator, which is included in Anaconda. On the Navigator Home tab, click the Spyder icon. For more information about Spyder, see the Spyder web page or the Spyder documentation. Anaconda command prompt is just like command prompt, but it makes sure that you are able to use anaconda and conda commands from the prompt, without having to change directories or your path. These locations contain commands and scripts that you can run.

MATLAB

MATLAB ® combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook

Professionally Built

MATLAB toolboxes are professionally developed, rigorously tested, and fully documented. With Interactive Apps. MATLAB apps let you see how different algorithms work with your data. Iterate until you've got the results you want, then automatically generate a MATLAB program to reproduce or automate your work. And the Ability to Scale Scale your analyses to run on clusters, GPUs, and clouds with only minor code changes. There's no need to rewrite your code or learn big data programming and out-of-memory techniques.

Python Programming

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

4.1 Data and Sources of Data

For this study MIT-BIH arrhythmia database and AAMI standards are used for machine learning purposes considering the patient-oriented scheme. A report from the Association for the Advancement of Medical Instrumentation (AAMI) [2] suggests using some specific internet available ECG signals databases with classification based on five cardiac arrhythmias: normal beats (N), supraventricular ectopic beats (S), ventricular ectopic beats (V), fusion beats (F), and unclassified beats (Q). This standardization proposal has become widely adopted in recent studies due to propitiate a good comparison among methodologies and results [3]–[5].

4.2 Theoretical framework

The study used pre-specified method for the selection of variables. For the proper working of the proposed method, the pre-specified dataset called the MITBIH under the AAMI standards have been taken and considered. However, the accuracy is more important as it determines the worth of the proposed system. The accuracy is directly proportional to the number of epoch. Finally, a model that has to be built in order to be trained on the dataset that has been considered which should also be able to classify the arrhythmia, we are ready with the proposed system.

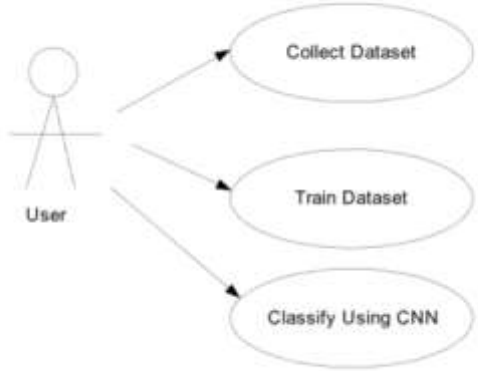

Fig 4.2.1 Theoretical Framework

4.3 Methodologies

4.3.1 Data Preparation

- Let's get started by making a list of all the patients in the data_path.
- Load dataset path and make list of patients
- Use package wfdb for loading the ecg and annotations.
- Let's load all the annotations and see the distribution of heart beat types across all files.
- We can make a list of the non-beat and abnormal beats now:
- We can group by category and see the distribution in this dataset

A function to load a single patient's signals and annotations need to be written. The annotation values are the indices of the signal array.

4.3.2 Make Dataset

A dataset that is centered on beats with +- 3 seconds before and after is made and split on patients not on samples. All the patients need to be processed. Once the processing of all the patients is done, we are ready to build our first dense NN. The software used to build the NN would be Keras for simplicity.

4.3.3 Building the model

Keras is a simple-to-use but powerful deep learning library for Python. In this post, we'll see how easy it is to build a feedforward neural network and train it to solve a real problem with Keras. Every Keras model is either built using the Sequential class, which represents a linear stack of layers, or the functional Model class, which is more customizable. We'll be using the simpler Sequential model, since our network is indeed a linear stack of layers. The Sequential constructor takes an array of Keras Layers. Since we're just building a standard feedforward network, we only need the Dense layer, which is your regular fully-connected (dense) network layer. A three layered NN can be built by the following:

```
model = sequential([
        Dense(64, activation='relu', input_shape=(784, )),
        Dense(64, activation ='relu'),
        Dense(10, activation='softmax')
        ])
```

The first two layers have 64 nodes each and use the ReLU activation function. The last layer is a Softmax output layer with 10 nodes, one for each class. The details on the input to keras needs to be specified. This can be done by specifying an input_shape to the first layer in the Sequential model. Once the input shape is specified, Keras will automatically infer the shapes of inputs for later layers.

4.3.4 Compilation steps

- The **optimizer**. We'll stick with a pretty good default: the Adam gradient-based optimizer.
- The **loss function**. Since we're using a Softmax output layer, we'll use the Cross-Entropy loss.
- A list of **metrics**. Since this is a classification problem, we'll just have Keras report on the accuracy metric.

```
model.compile(
        Optimizer: 'adam',
        Loss: 'catergorical_crossentropy',
        Metrics: ['accuracy'],
      )
```

4.3.5 Training the model

Training a model in Keras consists only of calling fit() and specifying some parameters. The input to train the model would be:

- The **training data** (images and labels), commonly known as X and Y, respectively.
- The **number of epochs** (iterations over the entire dataset) to train for.
- The **batch size** (number of samples per gradient update) to use when training.

```
model.fit(
        train_images, # training data
        train_labels, #training targets
        epochs=5,
        batch_size=25
   )
```

Keras expects the training targets to be 10-dimensional vectors, since there are 10 nodes in our Softmax output layer, but we're instead supplying a single integer representing the class for each image. Conveniently, Keras has a utility method that fixes this exact issue: to_categorical . It turns our array of class integers into an array of one-hot vectors instead. For example, 2 would become [0, 0, 1, 0, 0, 0, 0, 0, 0, 0] (it's zero-indexed).

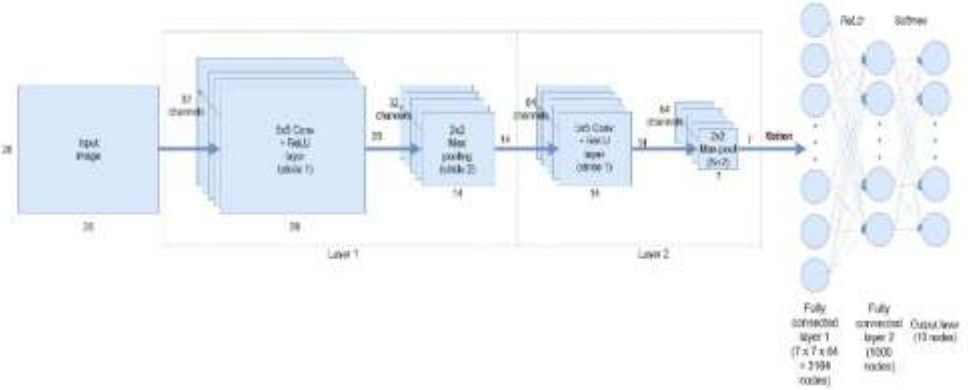

Fig 4.3.5.1 Classification model

4.3.6 Performance

We can build some functions for metrics reporting. Metrics report has accuracy, precision, recall parameters. We can build some functions for metrics reporting. Finally, a threshold has to be set at the prevalence of the abnormal beats from which the report can be calculated.

V. RESULTS AND DISCUSSION

5.1 Results of Study Variables

Table 5.1.1: Descriptive Statics

RESULTS ACCORDING TO THE AAMI STANDARD CONSIDERING THE DS2 DATASET

Class	Se(%)	+P(%)	FPR(%)
N	97.1	97.8	17.4
S	76.1	56.6	2.2
V	93	95	0.3
F	0.5	1.3	0.3
Q	0.0	-	0.0
Overall accuracy = 95.3%			

Table 4.1 presents the results obtained here, according to AAMI standard and the DS2 dataset. The absolute numbers of the five classes are illustrated in a confusion matrix shown in Table IV. Some heartbeats which do not have previous or posterior R-R intervals are rejected in DS2, respectively 91, 3, 6, 0, and 1 samples for classes N, S, V, F, and Q. All the performance is analyzed considering the parameterization explained in Section III. The training time for each epoch is around 15s, considering a machine with an Intel® Core™ i7-7500U CPU@2.70GHz, 16GB of memory and a graphics card GeForce 940MX/PCIe/SSE2.

Table 5.1.2: Comparative analysis

COMPARATIVE RESULTS

Method	N (%)		S (%)		V (%)	
	Se	+P	Se	+P	Se	+P
Chazal et al. [3]	86.9	99.2	75.9	38.5	77.7	81.9
Llamedo et al. [4]	95	98	77	39	81	87
Ye et al. [7]	88.5	97.5	60.8	52.3	81.5	63.1
Afkhami et al. [5]	97.4	98.4	86.5	90.9	96.0	77.6
Proposed	97.1	97.8	76.1	56.6	93	95

Table 4.2 presents the comparative results of this study with four published algorithms, considering the indicators of sensitivity and positive predictivity of classes N, S, and V. The proposed algorithm obtained a better result for positive predictivity in ventricular class, with 95% followed by Llamedo et al. [4] with 87%. According to positive predictivity of class N, all algorithms obtained close results, with emphasis to Chazal et al. [3] algorithm. It should be noted the high sensibility values of classes N and V for the algorithm proposed by Afkhami et al. [5], closely followed by the proposed algorithm. It was highlighted the Afkhami et al. [5] algorithm for the positive predictivity of class S, with the expressive value of 90.9%, if compared to the second and third values (56.6% and 52.3%) reached by the proposed and Ye et al. [7] algorithms, respectively. Comparing the overall results of S and V classes obtained by the Afkhami et al. [5] and the proposed algorithms, it is evident the false positives in class V of Afkhami et al. [5] algorithm, while in the proposed algorithm the evidence of the false positives occurs in class S.

Other hyperparameters were considered in order to tune the CNN, as different learning rates, number of epochs, batch size, number of hidden layers, number of neurons and filters in each layer, weight initialization, and dropout for regularization. It was observed bias results in some configurations, likely due to the imbalance of the heartbeats in different classes. Efforts were made in order to reduce false positives of class S through the following changes:

- Using different mother-wavelets, such as the first derivative of Gaussian and Morlet functions;
- Considering different number of heartbeats to provide the R-R mean calculations;
- Changing the formula of θ through the use of linear functions;
- Through the other possibilities of architecture, parameterization, weight initialization, and the filter kernel size of the classifier.

These efforts provided positive effects in reducing the false positives of the class S but with deteriorate results from the other classes. CNN allows data augmentation, which had a considerable impact on performance in this work. This strategy is important when dealing with imbalance classes in a dataset, as happens in MIT-BIH Arrhythmia Database.

VI. Conclusion and future work

The present study proposes a CNN-based 2-D classifier which performs ECG classification based on the patient oriented scheme and AAMI guidelines, validated over the MIT-BIH Arrhythmia Database. The overall system is composed of three stages. First, a signal preprocessing is used in order to remove the DC offset, followed by feature extraction through the application of the wavelet transform in the segmented heartbeats and the heartbeat intervals used to rotate the images generated by the WT. Finally, a classifier is implemented, which is composed of two convolutional layers, two max-pooling layers, one fully-connected layer and the output layer. This classifier receives as input the images from the previous stage and discriminates each heartbeat input data onto five classes, according to the aforementioned guidelines. The training dataset is used to model the classifier and the testing dataset used independently to assess the performance of the optimal model. As future work objective, extracted features on both leads will be used. For this purpose, the classifier should be modified in order to receive both images, combining these features for a final classification. Additionally, an effort will be exerted to improve the results of the positive predictivity in the supraventricular class. Furthermore, studies addressing different feature extraction methods and other classifiers architectures will attempt to improve the present results.

VII. ACKNOWLEDGMENT

We would like to place on record our deep sense of gratitude to Shri. D. K. Mohan, Chairman, Cambridge Group of Institution, Bengaluru, India for providing excellent Infrastructure and Academic Environment at Cambridge Institute of Technology without which this work would not have been possible. We are extremely thankful to Dr. Suresh L., Principal, CITech., Bengaluru, for providing us the academic ambience and everlasting motivation to carry out this work and shaping our careers. we express our sincere gratitude to Dr. Shashikumar D. R., HOD, Department of Computer Science and Engineering, CITech., Bengaluru, for his stimulating guidance, continuous encouragement and motivation throughout the course of present work. We also wish to extend our thanks to Mrs. Shilpa S. B., Assistant Professor, Department of Computer Science and Engineering, CITech., Bengaluru, for her guidance and impressive technical suggestions complete our paper publication.

REFERENCES

[1] F. Mahananto, T. Igasaki, and N. Murayama, "Cardiac arrhythmia detection using combination of heart rate variability analyses and puck analysis," in Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE, pp. 1696–1699, July 2013.

[2] B. Vuksanovic and M. Alhamdi, "Ecg based system for arrhythmia detection and patient identification," in Information Technology Interfaces (ITI), Proceedings of the ITI 2013 35th International Conference on, pp. 315–320, June 2013.

[3] K. Grauer and R. Curry, "Clinical electrocardiography: A primary care approach," Blackwell Scientific Publication, Boston, USA, 1987.

[4] S. Jadhav, S. Nalbalwar, and A. Ghatol, "Arrhythmia disease classification using artificial neural network model," in IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2010, pp. 1 –4, dec. 2010.

[5] G. B. Ajalmar Neto, Ricardo Sousa and J. S. Cardoso, "Diagnostic of pathology on the vertebral column with embedded reject option," in Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA), pp. 588–595, 2011.

[6] N. Thakor and Y.-S. Zhu, "Applications of adaptive filtering to ecg analysis: noise cancellation and arrhythmia detection," IEEE Transactions on Biomedical Engineering, vol. 38, pp. 785 –794, aug. 1991.

[7] D. Coast, R. Stern, G. Cano, and S. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," IEEE Transactions on Biomedical Engineering, vol. 37, pp. 826 –836, sept. 1990.

[8] C. Lima and M. Cardoso, "Cardiac arrhythmia detection by parameters sharing and mmie training of hidden markov models," in 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2007. EMBS 2007., pp. 3836 –3839, aug. 2007.

[9] M. Mitra and R. Samanta, "Cardiac arrhythmia classification using neural networks with selected features," Procedia Technology, vol. 10, no. 0, pp. 76 – 84, 2013. First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013.

[10] S. Yang and H. Shen, "Heartbeat classification using discrete wavelet transform and kernel principal component analysis," in TENCON Spring Conference, 2013 IEEE, pp. 34–38, April 2013.

[11] H. M. Masih Tavassolia, Mohammad Mehdi Ebadzadeha, "Classification of cardiac arrhythmia with respect to ecg and hrv signal by genetic programming," Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition, vol. 3, January 2012.

[12] H. Khorrami and M. Moavenian, "A comparative study of dwt, cwt and dct transformations in ecg arrhythmias classification," Expert System Application, vol. 37, pp. 5751–5757, Aug. 2010.

[13] S. Jadhav, S. Nalbalwar, and A. Ghatol, "Ecg arrhythmia classification using modular neural network model," in IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES), pp. 62 –66, December 2010.

[14] R. Ceylan, Y. Ozbay, and B. Karlik, "A novel approach for classification ¨ of ecg arrhythmias: Type-2 fuzzy clustering neural network," Expert System Application, vol. 36, pp. 6721–6726, Apr. 2009.

[15] M. Arif, M. Akram, and F. Afsar, "Arrhythmia beat classification using pruned fuzzy k-nearest neighbor classifier," in International Conference of Soft Computing and Pattern Recognition, 2009, pp. 37 –42, dec. 2009.

[16] I. Matul, I. Setiawan, A. Febrian, and W. Jatmiko, "Arrhythmia heart beats classification using mahalanobis generalized learning vector quantization (mahalanobis glvq)," in 2011 International Symposium on Micro-NanoMechatronics and Human Science (MHS), pp. 355 –360, nov. 2011.

[17] M. Setiawan, E. Imah, and W. Jatmiko, "Arrhytmia classification using fuzzy-neuro generalized learning vector quantization," in 2011 International Conference on Advanced Computer Science and Information System (ICACSIS), pp. 385 –390, dec. 2011.

[18] N. Emanet, “Ecg beat classification by using discrete wavelet transform and random forest algorithm,” in Fifth International Conference on Soft Computing, Computing with Words and Perceptions in System Analysis, Decision and Control, 2009. ICSCCW 2009., pp. 1 –4, sept. 2009.

[19] J. Nasiri, M. Sabzekar, H. Yazdi, M. Naghibzadeh, and B. Naghibzadeh, “Intelligent arrhythmia detection using genetic algorithm and emphatic svm (esvm),” in Third UKSim European Symposium on Computer Modeling and Simulation, 2009, pp. 112 –117, nov. 2009.

[20] S. Jadhav, S. Nalbalwar, and A. Ghatol, “Generalized feedforward neural network based cardiac arrhythmia classification from ecg signal data,” in 2010 6th International Conference on Advanced Information Management and Service (IMS), pp. 351 –356, 30 2010-dec. 2 2010.

[21] K. Waseem, A. Javed, R. Ramzan, and M. Farooq, “Using evolutionary algorithms for ecg arrhythmia detection and classification,” in Natural Computation (ICNC), 2011 Seventh International Conference on, vol. 4, pp. 2386–2390, July 2011.

[22] K. Bache and M. Lichman, “UCI machine learning repository,” 2013.