A Real-Time Framework for Arrhythmia Classification

W. M. U. P. Jayasinghe

A Real-Time Framework for Arrhythmia Classification

Umesh Jayasinghe
Index : 14000522
Supervisor : Dr. G.D.S.P. Wimalaratne

Submitted in partial fulfillment of the requirements of the
B.Sc. (Hons) in Computer Science Final Year Project (SCS 4124)

December 2018

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Candidate Name: W.M. Umesh Prabushitha Jayasinghe

....
Signature of Candidate
Date:

This is to certify that this dissertation is based on the work of Mr. W.M. Umesh Prabushitha Jayasinghe under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Principle Supervisor's Name: Dr. G.D.S.P. Wimalaratne

....
Signature of Supervisor
Date:

Abstract

Identification of arrhythmia is crucial to recognize cardiovascular diseases early. There has been much research over the past to automate the detection process of arrhythmia. Though most of the recent approaches have given good accuracies, such approaches use high resources which limits the real-time classification in low-end devices. Most of the methods proposed in the literature are not explicitly evaluated to support real-time classification. In this work, we propose a novel hybrid classification framework for real-time arrhythmia classification by using a deep convolutional neural network and dynamic time warping (DTW) distance-based alignment measure. A new data structure based on circular arrays has also been proposed to calculate the alignment scores efficiently. The hybrid classifier is inspired by the similarity of the heart rhythm between adjacent consecutive beats. Incorporating such context of the unknown beat to the classification leads to a significant performance notably in prediction time.

In order to evaluate the complete framework, forty-six ECG sequences in the MIT-BIH database streamed as a continuous time series data. Performance of the framework is tested on accuracy and speed over different hardware configurations. Our classification model achieved state-of-art performance with an average accuracy of over 97.05% for five super-types of arrhythmia defined by AAMI standard. We have compared the performance of our CNN with VGG19 and AlexNet architecture based deep convolutional neural network models. The hybrid classifier prediction time outperformed the prediction time of deep learning methods proposed in literature achieving the same accuracy. As a future work, generalizing the proposed hybrid approach to other domains which uses sequence prediction can be investigated.

Preface

A novel framework for real-time arrhythmia classification by using a deep convolutional neural network and dynamic time warping (DTW) distance based alignment measure is introduced in this dissertation. The concept proposed for the hybrid classifier behind this approach which involved a thorough analysis of the data and is solely my own work. A similar approach has not been proposed in any other study related to the domain of arrhythmia classification. Also, signal to 2D image translation proposed in the beat transformation phase was adapted to this domain from histogram based image processing techniques. A new data structure based on circular arrays has been proposed to calculate the alignment scores efficiently. However convolutional neural networks are recommended in the literature as good performing classifier and used in similar research recently. Preprocessing techniques such as noise reduction and baseline wanderings removal are adapted from the literature. Other research works related to this domain are not specifically evaluated for real-time classification. Therefore, the evaluation method introduced in this dissertation for the classifier performance in terms of speeds for different hardware configurations is a novel evaluation method.

Acknowledgements

I would like to express my sincere gratitude to my research supervisor, Dr. Dr. G.D.S.P. Wimalaratne, senior lecturer of University of Colombo School of Computing and my research co-supervisors from Sri Lanka Heart Association, Dr. Susitha Amarasinghe and Dr. Rohan Gunawardana for providing me continuous guidance and supervision throughout the research.

I would also like to extend my sincere gratitude to Prof N.D. Kodikara, Dr A.R. Weerasinghe and Dr D.A.S. Atukorale for providing me the feedback and showing the best possible paths.

Many thanks to my beloved mother and my dear father for always being my strength, showing me the correct direction, and making me who I am today. This thesis is dedicated to all my family members, to primary and secondary school teachers, to lecturers of University of Colombo School of Computing and to anyone who supported even by words to make my dreams come true.

Table of contents

List of figures

List of tables

Listings

List of acronyms

AAMI Association for the Advancement of Medical Instrumentation.

CNN Convolutional Neural Network.

CPU Central Processing Unit.

DTW Dynamic Time Warping.

ECG Electrocardiogram.

GCP Google Cloud Platform.

GPU Graphics Processing Unit.

LSTM Long Short Term Memory.

MIT BIH Massachusetts Institute of Technology Beth Israel Hospital.

PPV Positive Predictive Value.

ReLu Rectified Linear Unit.

RNN Recurrent Neural Network.

SVM Support Vector Machine.

WFDB Waveform Database.

Chapter 1

Introduction

According to the World Health Organization, 17.7 million people died from cardiovascular diseases in 2015, representing 31% of all global deaths [1]. Cardiovascular diseases are the diseases which involve heart or blood vessels. Heart attack, stroke, heart arrhythmia are common cardiovascular diseases. Over 90% of cardiovascular diseases can be prevented [2]. Most cardiovascular diseases can be prevented by addressing behavioral risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies. As [1] stated, it needs to detect and have medication on cardiovascular diseases early.

Identifying arrhythmia is very important to recognize such cardiovascular diseases early. Many cardiac-related disorders can be found by analyzing the rhythm changes in the heartbeat. Cardiologists find it challenging to make a correct diagnosis for arrhythmia by following conventional techniques like visual analysis as it takes experience and time. Methods proposed in the literature to classify arrhythmia are mostly based on electrocardiographic (ECG) waveform features, but contextual information such as age, gender, medical history, behavioral aspects and continuous rhythm without abnormalities, etc. are also being considered by medical practitioners when classifying arrhythmia.

Patients with heart problems need to undergo a cardiac test at the hospital by using the ECG devices or instruments. Any disorder of heart rate or rhythm, or change in the morphological pattern, is an indication of cardiac arrhythmia, which could be detected by analysis of the recorded ECG waveform. After testing, a cardiologist makes a diagnosis that combines the ECG with the clinical symptoms to take into consideration whether the patient's heart condition has been abnormal. When measuring the ECG patient might not be in an arrhythmia condition which makes it harder for doctors to diagnose. Therefore, evaluation of rhythm disorders usually requires a detailed discussion of symptoms and a physical exam with a health-care professional.

Following the traditional approach, if the rhythm irregularity exists while the ECG is being recorded, it can be identified immediately by consulting with a health-care professional. Otherwise, more specialized testing may be required. A 24-hour (or longer) recording of the heartbeat is often necessary to detect any rhythm problem that occurs daily but not regularly.

Research work which has been reported in the literature on beat detection and classification of ECG signal, described under chapter two. Most of them use either time or frequency domain representation of the ECG waveforms, from which many specific features are defined, allowing the recognition of the beats belonging to different classes. The most challenging problem faced by today's automatic ECG analysis is the considerable variation in the morphologies of ECG waveforms. Moreover, we have to consider the time constraints such as computational time as well [3].

Apart from ECG analysis, recently researchers from Google has developed an algorithm [4] to predict cardiovascular risk factor using images of the retina.

1.1 Domain background

The objective of this work is to provide a computer algorithm which can recognize and classify abnormalities of heartbeats. This section provides a brief introduction to the domain of arrhythmia.

Heartbeat is fabricated by contraction and relaxation of muscles in the heart. Muscle contractions cause an electrical change which is known as depolarization. Such electrical changes can be identified by using electrodes attached to the surface of the body. In an electrical point of view, four chambers of the heart can be observed as two regions where atrium (two upper chambers) contract together and ventricles (two lower chambers) contract together. The electrical activation of the heart normally starts from a special area of the right atrium called the sinoatrial (SA) node shown in figure 1.1. Then takes a small delay to grow into the lower node of the atrium, atrioventricular (AV) node. Thereafter the electrical impulse rapidly propagates down.

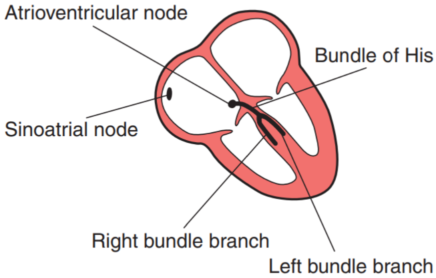

Figure 1.1: Schematic diagram of the heart [5]

The electrical activation may not always start from the SA node. The normal heart rhythm, where the electrical activation begins from the SA node, is called ‘sinus rhythm’. Sinus rhythm is necessary, but not sufficient, for normal electrical activity within the heart [6]. An electrocardiogram (ECG, also known as EKG) is an important tool to identify the origin and the spread of a heartbeat.

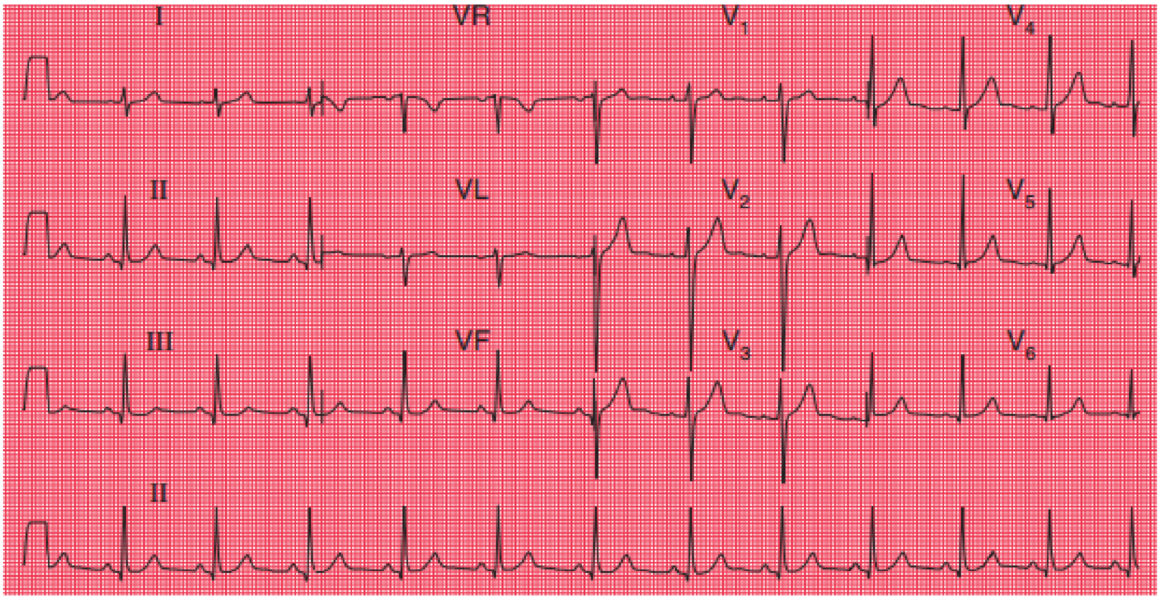

Figure 1.2: A good record of a normal ECG [5]

Electrical activity of the heart can be viewed from ECG (Figure 1.2) which is measured by using leads connected to the surface of different parts of the body. Medical professionals use ECG to diagnose and early recognize cardiovascular diseases and other heart conditions by analyzing the report. i.e. ECG is a matter of pattern recognition. Deflections of the signal are known as waves and arbitrarily chosen set of letters, P, Q, R, S, and T are being used to refer the waves as shown in the figure 1.3.

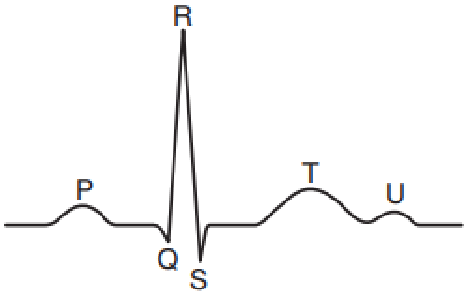

Figure 1.3: Normal ECG, including a U wave [5]

1.2 Research focus and research questions

Real-time analysis has become one of the main component of modern health devices. In modern days, there are different portable devices which have been favored to measure several heart conditions. Heartbeat measuring devices and portable ECG recorders make it easier for an ordinary person to get useful information and get immediate medical consultancy from doctors before following the traditional approach. Existing devices and models take several duration of ECG recorded and then analyze the recording. Indeed there are limitations when processing ECG in such devices. This research would also find the ability to extend the detection model to support streaming recordings with handheld and other devices to fit into a live arrhythmia detection model.

At the time of measuring the ECG, medical professionals may not find exact irregular heartbeats to give a proper diagnosis. However, the ECG signal sequence produces some insight about arrhythmia. Otherwise, ECG has to be measured for a long period of time to get the exact points of irregularities. This work is carried out in order to investigate how to incorporate such inter-related beat information into an automatic arrhythmia classification model.

Medical practitioners use ECG reports and clinical symptoms to analyze whether a patient has arrhythmia (also known as heart rhythm irregularities). Morphological features, interval related features, etc. can be used for automatic analysis heart rhythm [6]. However, hand-crafted features need to identified correctly and most of the times these identified features are not enough to predict all kinds of arrhythmia [7]. Recent works on arrhythmia domain have suggested data-driven approaches provide higher performance than rule-based methods which strongly relies on hand-crafted features. Out of the data-driven methods [8] states that deep learning approaches like convolutional neural networks and recurrent neural networks produce the best results as they are not relying on finding features manually. According to [8] deep-learning methods also removes the burden of preprocessing to a certain extent. But these deep-learning methods are not explicitly evaluated in real-time environments. There are complications in prediction time and space requirements which hinders the usage of deep-learning methods in live systems.

Therefore, this work is mainly focusing on three questions identified from the gaps in the literature.

1. What are the restrictions and obstructions for real-time classification with deep learning classifiers?
2. How to extend arrhythmia classification into a real-time approach by preserving the accuracy?
3. How to identify and incorporate inter-related beats to improve the classifier?

1.3 Aims and objectives

This research aims to support the detection process of heart rhythm irregularities through live monitoring of arrhythmia by incorporating a deep learning classifier and augmenting inter-related features of beats. Objectives of the research can be listed as follows,

- Analyze the drawbacks and limitations of the existing approaches.
- Identify the technical restrictions and obstructions which prevent the development of a real-time ECG arrhythmia detection.
- Incorporate deep-learning classifier for the arrhythmia classification process.
- Evaluate the complexity of deep-learning model for live classification.
- Develop a real-time monitoring and arrhythmia classification framework.
- Evaluate the complete framework on benchmark data.

1.4 Delimitations of scope

This research explores the possibilities to incorporate selected contextual information of inter-related beats along with streaming ECG data into a real-time framework of heart arrhythmia classification. There are different classes of arrhythmia defined in the medical literature and standards like AAMI groups the types of arrhythmia classes into superclasses. The framework will not consider all the types of arrhythmia for the classification. Instead, a selected set of arrhythmia types will be considered. Several additional processing of the ECG signals can be observed when practically testing the framework in a real environment. However, the strength of processing has to be determined with respect to the devices used. In this research, only the core model is concerned without any hardware level implementations.

1.5 Outline of the dissertation

The dissertation is structured as follows. Chapter two explores the existing approaches related to the domain of automatic heart arrhythmia detection. Chapter three describes the proposed research design and methodology. Potential ways of addressing the research problem are justified in this chapter. Chapter four illustrates the implementation details of selected algorithms and data structures in the proposed design. Utilization of software, hardware related configurations and prototype application details are also discussed under chapter four. Chapter five presents the evaluation model and the evaluation results of the proposed approaches. The last chapter, chapter six demonstrates the conclusion of the thesis and outlines the future work.

Chapter 2

Literature review

2.1 Arrhythmia detection

Conventional method to detect and diagnose arrhythmia is by analysing the presence of a particular set of signal features by a healthcare professional. Patients with heart problems need to undergo a cardiac test at the hospital by using the electrocardiographic (ECG) devices or instruments. After testing, a cardiologist makes a diagnosis that combines the ECG with the clinical symptoms to take into consideration whether there has been an abnormality in the patient's heart rhythm. Any disorder of heart rate or rhythm, or change in the morphological pattern, is an indication of cardiac arrhythmia, which could be detected by analysis of the recorded ECG waveform. An ECG signal provides the following information of a human heart which are useful to detect arrhythmia [9], Heart position, relative chamber size, impulse origin and propagation, heart rhythm, conduction disturbances and changes in electrolyte concentrations etc.

RR interval based techniques were used for arrhythmia detection by Donald E.G, Alan S.W, Jyh-Yun W, Malcolm C.L, and John H.T in 1978 [10] where the statistical nature of RR intervals were used for the analysis of abrupt changes. Adaptive recurrent filtering technique has been proposed for arrhythmia detection by Thakor, Nitish V and Y-S. Zhu [11] in 1991 which finds the stability departures in beat to beat morphology through recurrent features of each heartbeat signal complexes and P-QRS-T complex synchronization characteristics. This work is limited to a several set of arrhythmia detection problems, which they mention additional pattern recognition algorithms can be used along with the proposed adaptive recurrent filters to detect vast range of arrhythmia. Due to the complexity of arrhythmia detection which often involving diverse morphologies and rhythms that vary among different subjects as well as over time for the same subject, this work has suggested that data adaptive algorithm are desirable.

PhysioNet/CinC Challenge 2017 (PhysioNet/CinC Challenge is a popular research challenge conducted every year) was based on the theme arrhythmia detection. The challenge encourages the development of algorithms to classify, from a single short ECG lead recording (between 30 s and 60 s in length), whether the recording shows normal sinus rhythm, atrial fibrillation (AF), an alternative rhythm, or is too noisy to be classified. As an entry to the PhysioNet/CinC challenge 2017, a research has been done by Schwab et al [12] to implement a classifier for arrhythmia using recurrent neural networks (RNN). The approach is based on a set of interrelated recurrent neural networks which can jointly identify patterns of ECG segments. This work describes that the significant and most important part is the segmentation of heartbeats in ECG recordings and the identification of independent feature set to train the model. Arrhythmia are classified by segmented heartbeats rather than classifying the raw ECG signals. From the evaluation, class-wise F1 scores of 0.90, 0.79 and 0.68 obtained respectively for normal rhythms, AF and other arrhythmias giving the average score 0.79. For future work they suggests that the contextual information can improve accuracy of the model. Potential contextual information would include additional information about the patient's diagnostic and / or health state, i.e. prior diagnoses, electronic health records, lab results, genomics, etc..

A model which can diagnose irregular heart rhythms from single lead ECG signals using deep convolutional network which can map a sequence of ECG samples to a sequence of arrhythmia annotations has been proposed by Rajpurkar, Pranav, et al. [13]. Network learns to classify and segment 12 arrhythmia types. Neural network is a 34-layer convolutional neural network (CNN) which takes a time-series of arbitrary length raw ECG signal as the input and outputs a sequence of labeled prediction in each second. Each label is one of 14 rhythm classes (12 arrhythmias + sinus + noise). For training and evaluation, this work has used their own dataset having 64,121 ECG records and known to be 500 times larger than the existing datasets like MIT-BIH arrhythmia dataset. These records were captured using single lead continuous monitoring tool called Zio Patch. This model has out performed the average cardiologist score on F1 metrics. This has been done for 12 selected types of arrhythmia and there are some other arrhythmia which were not included in this work. For example this do not detect Ventricular Flutter or Fibrillation. According to the research, extending of the work to the other types of arrhythmia and automatic detection of other forms of heart disease with high-accuracy from single or multiple lead ECG records can be examined further. Similar research has been done to compare CNN and feature-based approaches by Andreotti, Fernando, et al. [14]. A drawback of CNNs is the fact they operate on grid-like structures (e.g. images or fixed segment windows). They have used ResNet with CNN for improved accuracy.

Apart from ECG analysis, recently a research team from google [4] has published an approach based eye's retinal images to find cardiovascular risk factors. Analyzing scans of the back of a patient's eye, the approach is able to accurately deduce data, including an individual's age, blood pressure, and whether or not they smoke. The rear interior wall of the eye (the fundus) is chock-full of blood vessels that reflect the body's overall health. Prediction of arrhythmias are not mentioned in the research but the approach can predict useful contextual information which can be used for arrhythmia detection.

Identification of beats in an ECG signal is a common sub-problem of heart arrhythmia classification. Most visible feature to identify a beat is the QRS complex shown in Figure 3. A real-time algorithm for QRS detection has been proposed by Pan J and Tompkins W.J in 1985 [15] which is based on the digital analysis of slope, amplitude, and width. Algorithm search back for missed beats and periodically adapts each threshold and RR interval limit automatically.

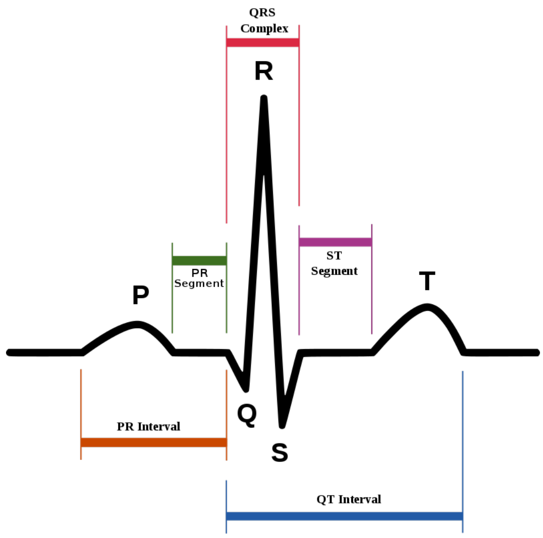

Figure 2.1: Schematic diagram of normal sinus rhythm [16]

This adaptive approach provides for accurate use on ECG signals having many diverse signal characteristics, QRS morphologies, and heart rate changes. Adaptive techniques are advantages because they do not require a prior knowledge of the signal or noise characteristics as do fixed filters. It provide estimated synthesis of desired signal and error feedback to modify the filter parameters. In evaluations using the MIT/BIH arrhythmia database, 99.325% QRS complexes were accurately detected. Algorithm is popular with the name Pan Tompkins algorithm and is widely used in arrhythmia detection and ECG signal processing even for modern research as well. Hidden Markov models with Gaussian observation probability distributions have been applied to the task of beat detection [17]. Artificial neural networks have also been used for the task of beat detection [18]. These models have achieved high-accuracy for some beat types, but they are not yet sufficient for high-accuracy heart arrhythmia classification.

Pan Tompkins algorithm has been used by several works in detecting beats. Patrick S. Hamilton [19] uses Pan Tompkins algorithm and [20] for QRS detection and provides a software tool for ECG analysis. Table 2.1 depicts several algorithmic implementations of beat detection benchmarked on sensitivity and precision against the MIT-BIH arrhythmia database.

Table 2.1: Peak detection algorithms benchmarked on sensitivity and precision

	Hamilton	Christov	Engelse and Zeelenberg
Average Sensitivity	96.2%	92.68%	93.62%
Average Precision	99.79%	99.42%	98.45%
Gross Sensitivity	96.10%	92.23%	93.54%
Gross Precision	99.80%	99.44%	98.56%

Most of the research done in the area have used several datasets. MIT-BIH arrhythmia database [21], [22] is a benchmark dataset with of 15 rhythms available. This database consists of 48 annotated ECG recordings each with 30 minutes long which were recorded using Holter device. There are several derivations of this dataset. MIT-BIH Atrial Fibrillation Database and MIT-BIH Normal Sinus Rhythm Database etc. are also being used by similar work. Apart from the above mentioned work, there has been other related work done in ECG analysis and arrhythmia detection. Table 2.2 compare several data driven approaches carried out for heart rhythm classification.

Table 2.2: Analysis of data driven approaches

Main focus	Study parameters	Description of work	Remarks
Waveform characteristics and phase/rhythm characteristics [23]	1. QRS complex width 2. RR interval 3. Difference in absolute area (ArDiff, using normalized waveform area) 4. Maximal cross-correlation coefficient	1. Decision tree based approach 2. Real-time detection of QRS complexes (using Pan-Tompkins algorithm) 3. Intervention free normal/ abnormal heart beat classification 4. Android-based ECG monitoring application	Used MIT-BIH Arrhythmia and MIT-BIH Supraventricular Arrhythmia databases. Sensitivity for abnormal beat detection was 89.5% with a specificity of 80.6% <u>Limitations</u> Binary classification of beats either Normal or Abnormal
Support vector machines and genetic algorithm [24]	1. Amplitudes of P-peak, Q-valley, R-peak, S-valley and T-peak 2. Positions of P-peak, Q-valley, R-peak, S-valley, T-peak 3. Time ratio between last beat and next beat (RR interval ratio)	Genetic algorithm has been used to obtain optimum SVM parameter values.	Used MIT –BIH arrhythmia database. Overall accuracy of 99.8112% and a good sensitivity of 99.9747% for PVC arrhythmia <u>Limitations</u> No real time analysis. Only for PVC arrhythmia. (Normal, PVC , Other)

Continued on next page...

Table 2.2 – continued from previous page.

Main focus	Study parameters	Description of work	Remarks
Recurrent neural networks with deep long short-term memory (LSTM) [25]	Raw ECG signals	Using LSTM networks, ECG signal can be directly fed into the network without preprocessing as required by other techniques Does not require hand coded features but works directly on raw signals	Used MIT-BIH Arrhythmia Database. 96.45% F-score accuracy for the test set. <u>Limitations</u> No real time analysis, 1 minute recordings of ECG signals. 5 classification classes Normal, PVC, APC, PB , VC
Least square support vector machine with principal component analysis [26]	15 Features extracted from PCA out of 279 attributes in the dataset	Arrhythmia classification using LS-SVM. PCA has been used for feature extraction. Missing values were dealt with probabilistic values considering distribution.	UCI arrhythmia dataset has been used. Obtained accuracies, 96.86% (50-50 train-test) 100% (70-30 train-test) 100% (80-20 train-test) <u>Limitations</u> Dataset is not challenging as all the features are annotated Not considering raw ECG signals

Continued on next page...

Table 2.2 – continued from previous page.

Main focus	Study parameters	Description of work	Remarks
Backpropagation Artificial Neural Network [8]	Initially 4096 features extracted using transferred deep learning. Features were subjected PCA to reduce number of features.	ECG feature extraction using transferred deep learning (AlexNet) 1 min intervals of 3 MLII channel records are used. 100, 118 and 217 (training) 101, 231 and 107 (testing)	Used MIT BIH database. 92% testing accuracy <u>Limitations</u> Image processing techniques reduces the performance Limited number of data has been used from the dataset (6/48 records) Only classifying 2 heart conditions
Optimum-path forest (OPF) classifier [27]	1. DWT 2. RR range 3. Signal energy 4. Morphological	6 different feature extraction mechanisms used. OPF shown to be more efficient than SVM in terms of the computational time for both training and test phases.	Used MIT-BIH arrhythmia database. 90.75% accuracy obtained for 5 classes

Continued on next page...

Table 2.2 – continued from previous page.

Main focus	**Study parameters**	**Description of work**	**Remarks**
Convolutional Neural Networks [13]	Raw ECG signals	Large annotated dataset and a very deep convolutional network has been used.	Introduced a new dataset with 29,163 unique patients and 14 classes. Obtained 80% precision and 78.4% sensitivity. Doesn't detect two main arrhythmias, Ventricular Flutter or Fibrillation.

2.2 Conclusion

There have been many research over the past to automate the detection process of arrhythmia. Almost all the traditional and modern approaches use ECG signals as a key input to determine heart rhythm irregularities and most of the traditional arrhythmia detection approaches were based on the use of R-R interval in ECG waveform, evaluating heart rate variability and determining the presence of morphological characteristics like the absence of P-wave etc. Unlike traditional hand-engineered approaches recently there have been many research on cardiac arrhythmia detection based on artificial neural networks. In the literature, highest accuracies were obtained by using feature annotated datasets. Arrhythmia classification for MIT-BIH arrhythmia dataset still remains challenging as features are not annotated and dataset maps to the real world scenario of data acquisition.

Recurrent neural network with LSTM and Convolutional neural network have been identified from the literature which has given good performance. Real-time application which uses such deep learning techniques on arrhythmia detection are often hindered by high computational complexity and frequent memory accesses.

From above analysis, most of the work only use raw ECG data. Applicability of information such as past medical history, health state, and other contextual information has been poorly considered. Also, past works have not specifically evaluated real-time arrhythmia detection on mobile devices for streaming ECG data. Lack of evaluation of the classifiers, specially deep learning classifiers [28] for the real-timeness along with the accuracy is a major point that has to be addressed.

Chapter 3

Design

3.1 Research methodology

The main aim of this research is to investigate and extend arrhythmia classification into a real-time approach by preserving state-of-art accuracy. To achieve this goal, Design Science [29] with Constructive research approach (a mixed methodology) was carried out which involved in the development of a real-time monitoring and arrhythmia classification proof of concept prototype as well.

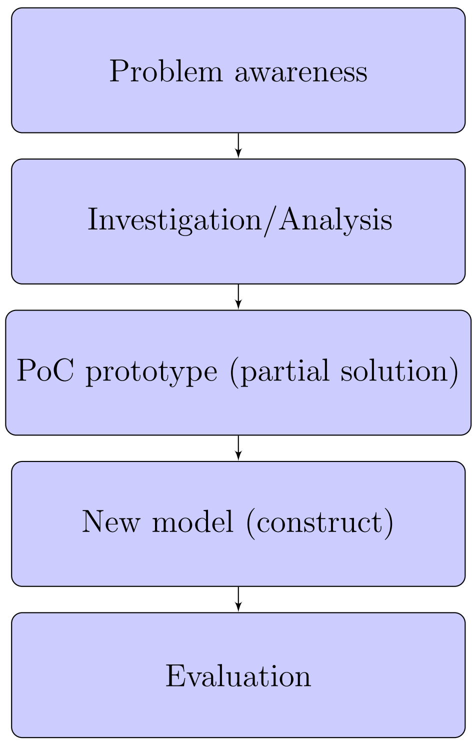

Figure 3.1: Phases aligned with mixed methodology (Design Science and Constructive)

Figure 3.1 shows aligned the phases of Design Science methodology and Constructive methodology according to the purpose of this research. In Design Science research, new scientific knowledge can be generated by means of constructing an artifact, and the core of this approach is a problem-solving process used to develop the artifact. Artifacts can be in the form of constructs, models, methods, instantiations, or better theories and are developed to enable a better understanding of the development, implementation, and use of information systems [30]. The following is a brief overview of the steps in Design Science methodology aligned with the proposed arrhythmia classification framework.

Objective of the solution

The objective is to investigate incorporating electrocardiographic signals with deep learning model in order to classify different arrhythmia types in a real-time approach.

Problem awareness

As described in the 1st chapter, early identification of cardiovascular diseases and risk factors is a problem under investigation. Arrhythmia which is a type of cardiovascular disease also supports in early recognize other cardiovascular diseases. In order to investigate the problem and possible solutions, a comprehensive literature review on the present status of arrhythmia detection and real-time arrhythmia detection frameworks will be performed.

Solution design and implementation

The design is based on a hybrid model combining a deep learning classifier with a rule-based engine to speed up the real-time detection. A proof of concept prototype will be developed as the artifact. Detailed description on design approach is presented in section 3.4.

Demonstration

To demonstrate the feasibility of the proposed work MIT-BIH arrhythmia database will be used. Arrhythmia and cardiology related domain knowledge needs medical professional support. Domain knowledge for the research is facilitated by the Sri Lanka Heart Association (SLHA).

Evaluation

Twofold evaluation will be carried out in order to evaluate the performance of the model and performance of the complete framework. A detailed description of the evaluation plan and evaluation criteria are mentioned in Chapter 5.

3.2 Design considerations

One of the central question in the research design is to identify which machine learning model to be used to develop the classifier. Based on the literature, there are multiple possible models which can be used. Support Vector Machines, Recurrent Neural Networks with Long Short-Term Memory (LSTM) and Convolutional Neural Networks have been identified from the literature which has given an excellent performance (see table 2.2). Though deep learning approaches have given best results, incorporating a deep learning model to a real-time classifier is a challenging task.

SVMs are inherently for two-class classification, where the best accuracies have been given for a maximum of 3 classes. The issue with SVM is that it is hard to engineer kernel when the number of classes is high. Problem with Recurrent Neural Network with LSTM is that the network builds up memory. Therefore, a high memory requirement exists. When compared with CNN, RNN with LSTM also perform more computations.

Another major concern is the incorporation contextual information into the classifier model. Though there are enough heartbeats to train a classifier (MIT-BIH dataset contains over 105, 000 heartbeats), there are only a limited number of subjects (MIT-BIH dataset contains 48 subjects) which would be not enough to incorporate contextual information with a machine learning classifier. Therefore, the classifier would use a rule-based approach and a machine learning approach which combines to form a hybrid model.

3.3 ECG data preprocessing

MIT-BIH Arrhythmia Database contains 48 two-channel recordings that were obtained by the Beth Israel Hospital Arrhythmia Laboratory between 1975 and 1979. Each recording is 30 minutes (or slightly higher), recorded with a Holter monitor device. The subjects were 25 men aged 32 to 89 years, and 22 women aged 23 to 89 years where 60% out of all the subjects were inpatients.

Out of the 48 recordings in the MIT-BIH dataset, we have used 46 records. Reason for the elimination of two records is that this work only uses a single channel (single lead). Except for two recordings (record 102 and record 104) all the other 46 recordings has MLII channel. Record 102, 104, 107 and 114 contains a high amount of unknown beats. Therefore, for the final result, the two eliminated records have less significance.

3.3.1 Noise removal

Preprocessing of the ECG beats needs to be performed to address the most common issues with the ECG signals. Prominent issues identified from the literature are noise and baseline wanderings. According to the literature most of the noise is located outside of the interval of 1.5 Hz to 50 Hz [31]. A low pass filter with a cut off frequency of 50 Hz and a high pass filter with a cut off frequency of 1.5 Hz has been used to remove the noise artifacts.

Some records in the dataset are unstable and contain power line interference, baseline wanderings which caused by muscular movements [32] etc. as shown in the figure 3.2.

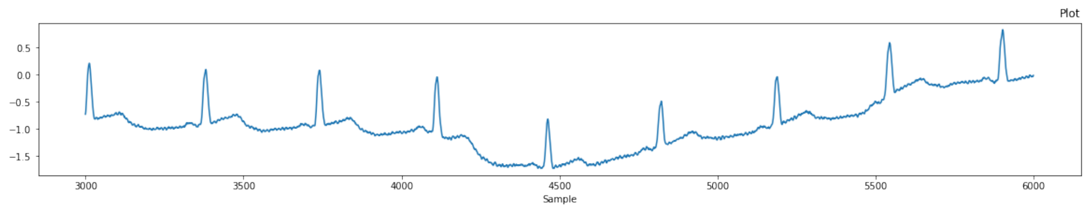

Figure 3.2: MIT-BIH record 121 containing baseline wanderings

The baseline can be identified by applying a median filter over the signal. By subtracting the signal amplitude values from the baseline gives a baseline wandering free signal output (equation 3.1).

$$Signal_{new} = Signal_{old} - Baseline \qquad \text{(equation 3.1)}$$

3.3.2 R-peak correction

Apart from noise, baseline wanderings and amplitude inconsistencies of ECG signals, beat annotations are also not aligned with the R-peaks as shown in figure 3.3. Simple linear search around ten samples of the annotated beat can be used to correct the R-peak location. Therefore before the segmentation of the beats, R-peaks has to be adjusted accordingly.

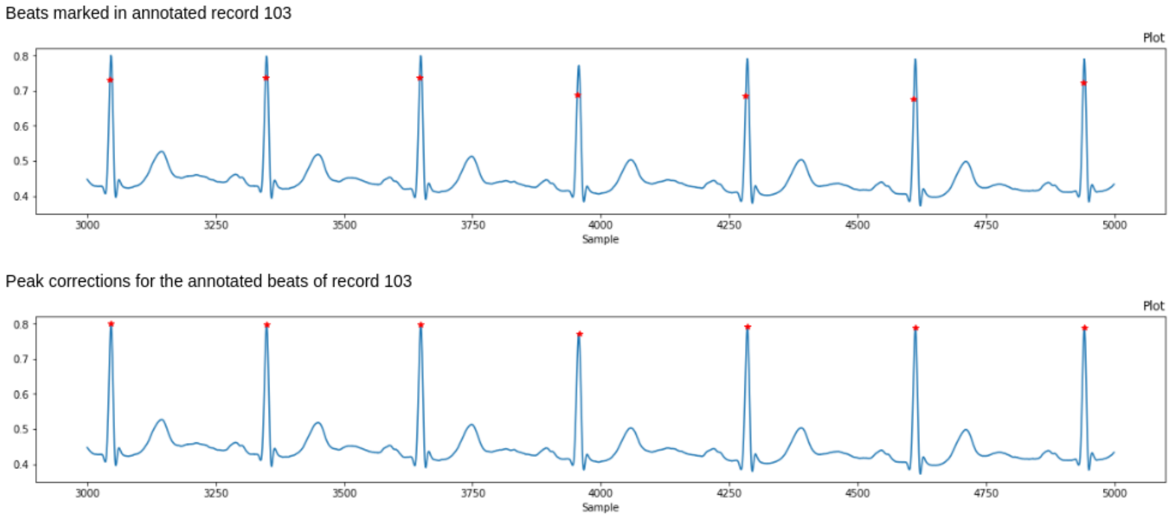

Figure 3.3: On top, MIT-BIH record 103 with annotated beat positions. On bottom, corrected R-Peak positions

Approach for the adjustment of R-peak positions of MIT-BIH dataset is follows.

Let Xi be annotated beat position and Yi be value at that position

1. Take the interval $[Xi - Thresh, Xi + Thresh]$ (Thresh is set to 10)
2. Find the maximum value Ymax and Minimum value Ymin in that interval (where Xmax and Xmin are corresponding sample positions).
3. Xmin is the beat position (Xbeat) if Ymin is existing near Yi and values increasing after Ymin. Otherwise Xmax is the beat position
   i.e.
   if $|Ymax - Yi| < |Ymin - Yi|$ and $gradien(Xmin + 1) > 0$;
   then Xbeat = Xin ;
   else Xbeat = Xmax ;

3.3.3 Beat selection

There are 17 beat types defined in the entire MIT-BIH database. Table 3.1 shows the list of all beat types and number of beats found for each beat type per record. Below table depicts the total number of beats per each type found in the entire database. (Refer Appendix A : table A.1 for the meaning of each shortcode)

Table 3.1: Beat types with respective beat count

Beat type (short)	Number of beats
N	74722
V	7122
A	2544
a	150
F	802
R	7231
S	2
j	226
J	83
E	105
L	8067
e	16
/	3616
f	260

Above table clearly shows the imbalanced nature of beat types in the dataset. Further, the class imbalance has a significant effect on heartbeat classification because some beat classes do not have a significant amount of beats to facilitate the classification process. Clear solution would be to group beat classes. Association for the Advancement of Medical Instrumentation (AAMI) has introduced five heartbeat super classes [33] which group 15 existing classes of 5 datasets including MIT-BIH arrhythmia database. Below shows the AAMI standard for heartbeat classification.

N = Normal Beat; (N L R)
SVE = Supraventricular ectopic beats; (A a J S e j)
PVC = Premature ventricular contraction beats; (V E)
FV = Fusion of ventricular; (F)
Q = Unknown beats; (P / f u)

When considering the AAMI classification, dataset still has 90,020 Normal-type beats while other four types have less than 10,000 beats. Therefore, for the training phase of the model, we have randomly selected 27,804 beats which cover all five types of beats.

3.4 Approach

The core of the arrhythmia detection classifier is based on Convolutional Neural Network, which often used as a deep learning network. In order to achieve real-time classification, We propose a novel method which wraps a Convolution Neural Network with a sequence alignment engine.

Before the classification of ECG signals, beat segmentation and image creation has to be performed. Figure 3.4 shows the high-level flow of the proposed approach.

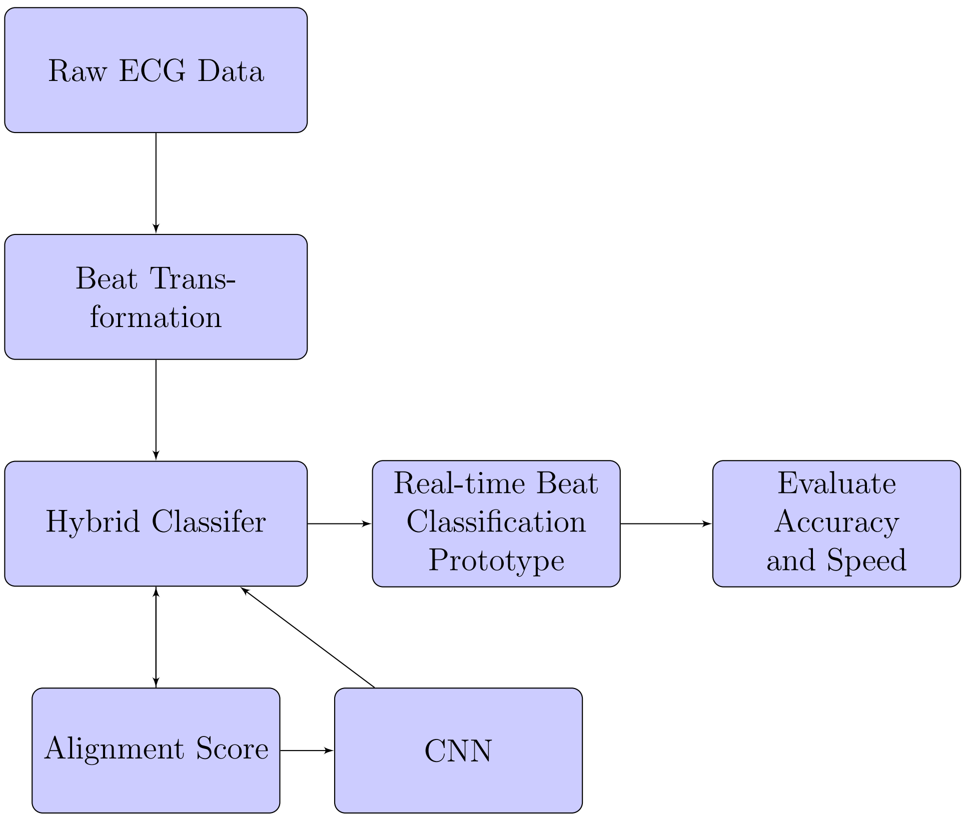

Figure 3.4: High level work flow diagram

3.4.1 Data acquisition

The approach does not depend on any device or sensor. Therefore ECG data can be acquired from any standard ECG measuring instrument. Raw data input methods are mainly considered in two different ways: Streaming ECG data and static ECG data. Streaming data are only considered for the real-time classification model.

CNN classifier is trained over a set of static ECG data. In the proposed approach, it considers only a single channel ECG stream.

3.4.2 Beat transformation

This work does not consider R-peak detection since there is highly accurate algorithms literature. Instead annotated R-peaks from the dataset are being considered. Here, in the proposed approach a beat is defined as 180 sample width window centered to R-peak. See figure 3.5. Therefore, the R-peak gets centered at 90th sample. Reason for selecting such a fixed window is to achieve time normalization by removing the effect of the beat miss alignment.

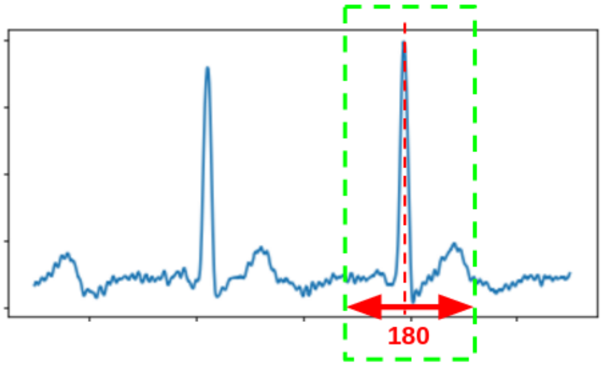

Figure 3.5: Illustration of beat definition

ECG data are inherently 1-dimensional (1D). Signal to 2-dimensional (2D) representation is necessary to classify the beats using a 2D CNN. Therefore, it needs to grip two representations of a beat. 1D representation passes through the sequence alignment engine whereas the 2D translation used to derive the actual classification.

In order to translate a beat signal into a 2D representation, this approach proposes a direct sample matching through a range stretching technique. Range stretching is a widely used method in the image processing field where it has been successfully used to enhance distorted images. Contrast stretching is one such application based on range stretching technique. To translate amplitude to 2D representation, initially, it considers a matrix with the dimensions 190x100. The width of 190 represents the 180 samples and 5 sample padding from both the ends. Amplitude values are range stretched from 3 to 93., i.e. height component (100 pixels) has a padding of 3 pixels from the bottom and 7 pixels from the top. In another aspect, the above approach also leads to amplitude normalization. For each 180 samples (as Y_{in}), calculation of the pixel position (Y_{out}) is done using the equation 3.2. $CONST_{MIN}$, $CONST_{MAX}$ are referred to the minimum and the maximum amplitude values in a beat.

$$Y_{out} = 90 - \frac{Y_{in} - CONST_{MIN}}{CONST_{MAX} - CONST_{MIN}} * 90 + 3 \qquad \text{(equation 3.2)}$$

All the Y_{out} pixels are colored black and eroded with a filter of size 3x3. Figure 3.6 shows the resulting translation.

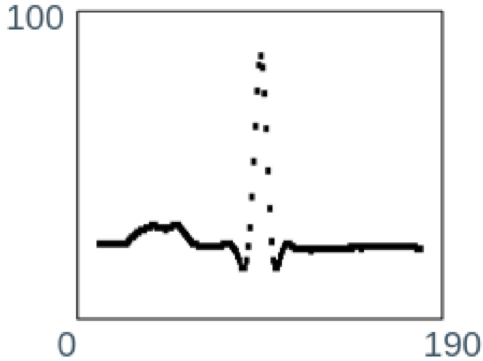

Figure 3.6: 2D translated beat

3.4.3 Hybrid classifier model

In this work, we propose a novel hybrid classifier model to classify arrhythmia. The model contains three components namely, sequence alignment engine, decision layer and convolutional neural network (CNN) based deep learning classifier. The goal of the proposed hybrid model is to gain a good prediction time without a significant loss in the accuracy of the deep learning classifier. The sequence alignment engine and the decision layer is used in order to minimize the number of CNN classifications and make use of the previously classified beats to predict new beats.

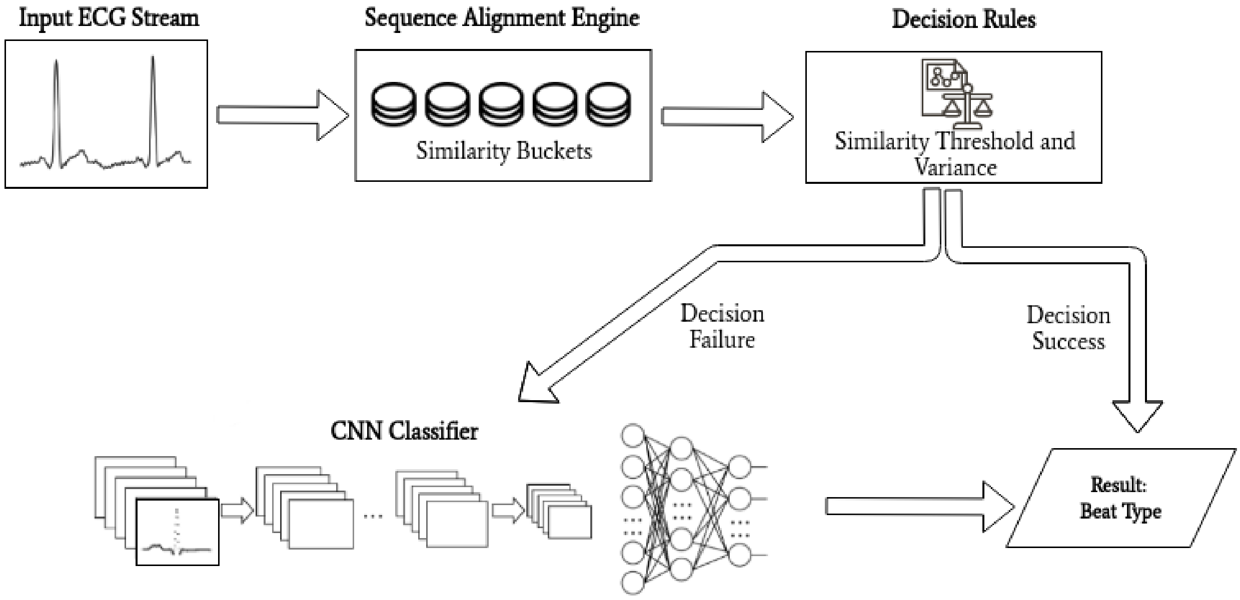

Figure 3.7: Highlevel design of the proposed hybrid classifier

The similarity between adjacent consecutive beats has inspired the proposed architecture. By analyzing the heart rhythm of a person, it is evident that there is a similarity between adjacent beats. Usually, normal heartbeats of a person are likely to have a similar morphology. Therefore, there is no point of classifying every beat in a sequence of streaming beats. Instead, the possibility of classifying a single beat and measuring the similarity of that beat with the next beat can be considered. Such a concept is useful for a real-time classification system as we need to consider both accuracy and the speed of classification.

After the beat transformation stage, two representations of a beat are stored. i.e. 1-dimensional array and 2-dimensional array (matrix). When a beat is given to classifying, if there are no prior CNN classifications for previously encountered set of beats then that beat is classified using CNN. When there's a CNN classified beat within the last set of classifications, we measure the similarity between the CNN classified beats and the beat to be classified. If the similarity value is above a threshold value, CNN based classification is performed on that beat.

Dynamic time warping (DTW) algorithm is used to wrap the decision layer with the sequence alignment engine. DTW algorithm measures the similarity between two beats using a distance measure called DTW distance. Complete architecture of the classifier model is shown in the figure 3.7.

3.4.3.1 Sequence alignment engine

In the literature there have been several works carried out entirely using sequence alignment and similarity measures [34, 35, 36] to classify different arrhythmia. However, the performance is not up to the level of the performance recorded by the deep learning classifiers. In our work arrhythmia type is decided by the CNN deep learning classifier and the similarity is determined with previous CNN classified beats. Adjacent heartbeats have a higher tendency to be aligned than other beats. Since the focus of this work is to classify real-time arrhythmia on streaming ECG data, such adjacent beats can be incorporated for the classification.

In the sequence alignment engine, there is a set of fixed-size storage buckets, which we name as similarity buckets. Each of these similarity buckets is assigned to only one beat type. i.e., a single similarity bucket holds only one type of beats. Therefore, for the classification of 5 beat types, there should be five similarity buckets. When a beat is classified through the CNN classifier, then the classified beat is added to the corresponding similarity bucket. The task of a similarity bucket is to store the most recent beats of a particular beat type. Each of these similarity buckets is a first-in-first-out (FIFO) circular queue. Therefore, the least recent beat is replaced by the new beat. Though the bucket has a set of beats, the output of a single bucket is the median beat.

	col1	col2	col3	...	coln
beat1	a11	a12	a13	...	a1n
beat2	a21	a22	a23	...	a2n
beat3	a31	a32	a33	...	a3n
beat4	a41	a42	a43	...	a4n
beat5	a51	a52	a53	...	a5n

Figure 3.8: A similarity bucket

Here the beat i is a record in the bucket which defined as $ai1, ai2, .., ain$

The median beat of a bucket is determined by the set of medians for each column as expressed in the equation 3.3. n refers to the number of samples (columns) of a beat, and *median* function calculates the median value of a set.

$$Beat_{median} = \bigcup_{j=1}^{n} median(\{a1j, a2j, a3j, a4j, a5j\}) \qquad \text{(equation 3.3)}$$

Once the median beat of a bucket is calculated, a similarity value is calculated by using an alignment measure.

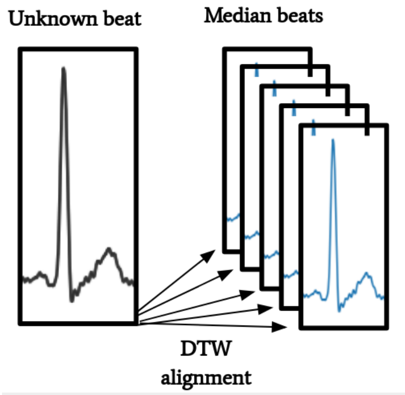

Figure 3.9: Unknown beat with median beats

As figure 3.9 depicts, an unknown beat is subjected to alignment with each median beat obtained from the similarity buckets. Fast dynamic time warping algorithm (DTW) is used to calculate the similarity. Dynamic time warping algorithm tries to align two sequences using dynamic programming. Figure 3.10 shows how two sequences (beat samples), X and Y are aligned using DTW method. The similarity value is calculated by using a matrix, and the resulting similarity is given by the alignment value of the last sample of X with the last sample of Y.

Y \ X	0	1	2	3	4	5	6
6	∞	26	23	21	20	18	20
5	∞	18	19	16	14	19	22
4	∞	16	14	14	13	15	19
3	∞	12	10	12	11	15	17
2	∞	9	7	8	13	14	18
1	∞	3	8	9	11	15	19
0	0	∞	∞	∞	∞	∞	∞

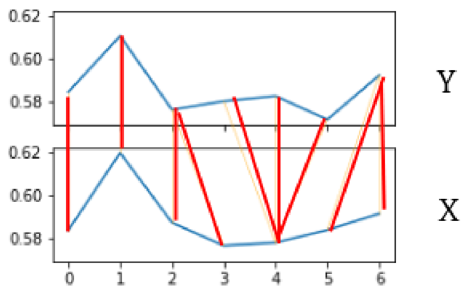

Figure 3.10: Visual representation of DTW algorithm

Alignment between any given sample of the sequence X and any given sample of the sequence Y is given by equation 3.4. Samples of X are represented by i and samples of Y are represented by j.

$$f_{i,j} = \|x_i - y_j\| + minimum\{f_{i,j-1}, f_{i-1,j-1}, f_{i-1,j}\} \qquad \text{(equation 3.4)}$$

$$base \ \ values$$

$$f_{0,0} = 0$$

$$f_{0,j} = f_{i,0} = \infty \quad \forall i, j$$

As an example, for the two sequences given in the figure 3.10, alignment for 3^{rd} sample position of X with 2^{nd} sample position of Y is, $f_{3,2} = 8$. Similarity value for the two sequences is $f_{6,6} = 20$. DTW algorithm has a quadratic time and space complexity of $O(n^2)$. In this work, we have used FastDTW [37] algorithm which is an approximation of DTW algorithm. The FastDTW algorithm provides a linear time and space complexity through a multilevel approach. Another approach which we have used to reduce the time and space is subsampling. In work [35], a prominent finding is that sub-sampling has improved the accuracy of the model. Our proposed sequence alignment engine also uses a similar mechanism of subsampling.

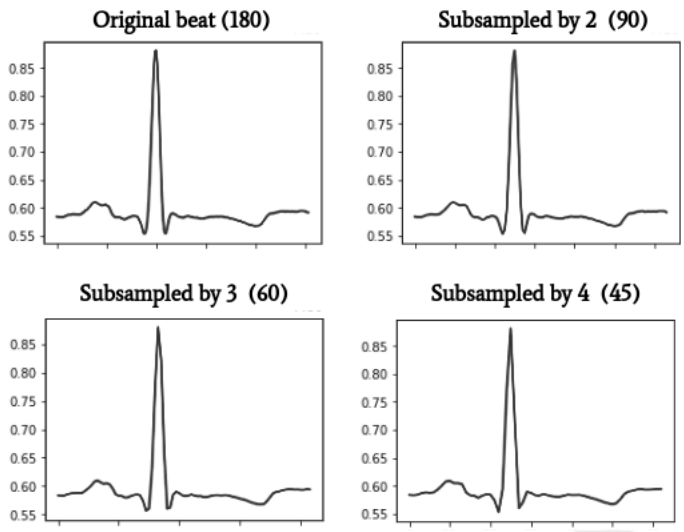

Figure 3.11: Sub-sampled outputs of a beat

Subsampling reduces the number of samples in a sequence by skipping samples. When a beat which has 180 number of samples is subsampled by 2, a single sample is skipped every time resulting in a sequence with only 90 samples. By following a trial and error basis for 2, 3, 4 and 5 skipping, the optimal number is chosen to be four. After four skippings accuracy tends to decrease. [35] also states that the optimal number of skippings is four. Although the number of samples reduced by subsampling is significant, there's a less visual difference in the plots as seen in the figure 3.11. Morphology and most of the features of a beat are preserved without losing information.

Subsampled beats are stored in similarity buckets, and unknown beat which is to be classified is also subjected to subsampling before measuring the similarity.

3.4.3.2 Decision layer

Decision layer decides whether a beat should be sent to the CNN classifier or not. In a straightforward approach, it is possible to label the unknown beat to the beat type of the similarity bucket which gives the maximum similarity value. However, there can be complications regarding performance as it omits different parameters like variance between the similarity values. One preferred approach would be to identify 'normal' and 'other' beats (which may or may not be abnormal) and then if the beat is labeled to be the type 'other', then we can classify that beat using a CNN classifier. Therefore, for the decision layer, it is possible to plug-in and plug-out different algorithms.

In this work, a rule-based algorithm and a support vector machine (SVM) based classification have been tested on taking decisions. An SVM model has been trained for binary classification using the same dataset for normal and abnormal beats. Beat samples are subjected to discrete Daubechies wavelet decomposition and the resulting approximated wavelet has been used as the input feature set for SVM.

Apart from the SVM classifier, a rule-based model has been also tested as the core decision-making algorithm. Sequence alignment engine provides a set of alignment scores for each beat type for the unknown beat. Decision rules are set according to several threshold values incurred upon the similarities. Figure 3.12 shows the flow of identifying whether the beat is a normal beat or not using the alignment scores. Parameters have been chosen by trial and error basis. Although in this work we focus mainly on finding normal beats in the decision layer, it possible to extend the decision layer to check for other beat types as well.

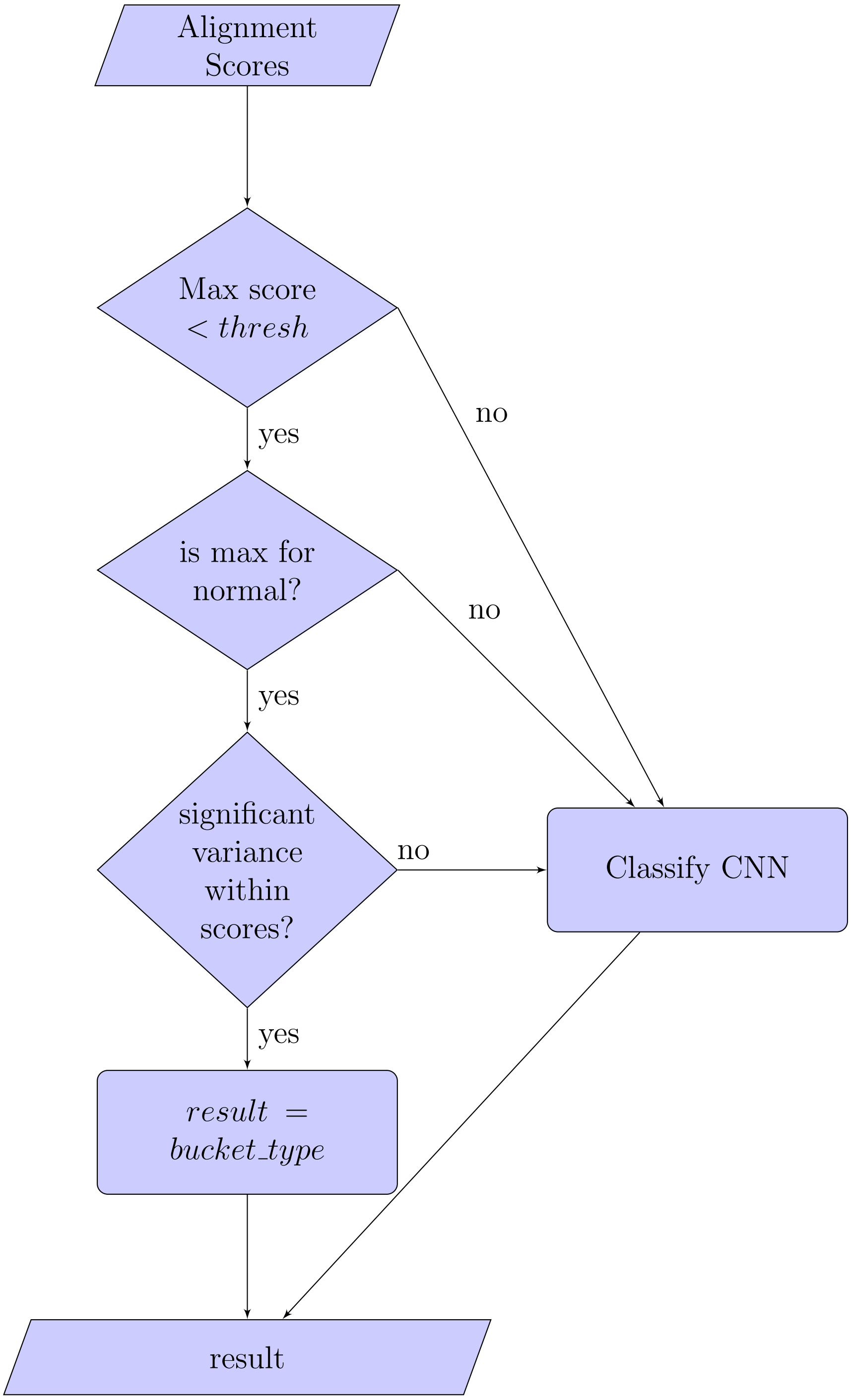

Figure 3.12: Decision rules for the rule based model

3.4.3.3 Convolutional neural network classifer

Deep learning classifier which is used in this work is a convolutional neural network. Usually, a deep learning neural network refers to an artificial neural network with multiple layers between the input and output layer. Convolutional neural networks can be taken inherently as a deep learning network as it contains at least one convolution layer between the input and output layer. A convolution layer is similar to an ordinary layer which is made up of neurons. What changes is that the CNNs make the assumption that the inputs are images which allows embedding several image processing operations on each neuron.

As described in chapter two, there have been several research recently on applying deep learning models to arrhythmia classification. A significant performance has been achieved by these methods when compared to traditional methods. However, high resource consumption and prediction time is a major downside which obstructs the use of such networks in real-time frameworks. As shown in the figure 3.13, computational cost of standard convolution is $HWNMK^2$ (refer the figure for notations). Apart from fine-tuning CNNs for high performance, we use a specific feature of the problem to gain good prediction time performance. i.e., the similarity between adjacent heartbeats.

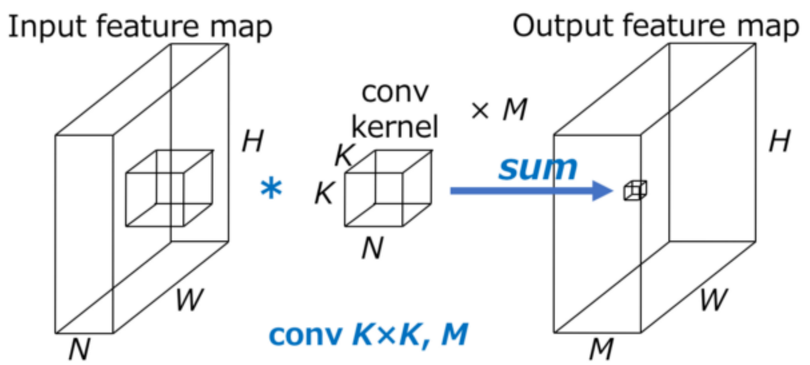

Figure 3.13: Visualization of the complexity of convolution according to [38]

Reason to select a convolutional neural network is that out of most deep learning methods CNNs has given good accuracy as well as it does not build up memory compared to Recurrent LSTM networks.

Network Configurations
64x64 input image
Conv2D (3x3, ReLu)
MaxPooling
Conv2D (3x3, ReLu)
Conv2D (2x2, ReLu)
MaxPooling
Dropout
Flatten
Dense (ReLu)
Dropout
Dense (ReLu)
Dropout
Dense (Softmax, Output layer)

Figure 3.14: Modified LeNet model

Convolution Layer - A weight layer in which the weights are adjusted by convoluting with a filter (kernel).

Dense Layer - A fully connected weight layer in which every node in the layer is connected to every node in the preceding layer.

Pooling - A technique which reduces the number of parameters by reducing the spatial size of the image representation.

Dropout - A technique which defines ignore neurons in a neural network by adding a penalty to the loss function.

As the core classifier, a modified version of LeNet architecture is used. LeNet is importantly a lightweight architecture having a lesser number of layers. Therefore, the calculations are done with fewer computations. Figure 3.14 shows the structuring of layers in the architecture. Chapter 4 further describes the convolutional neural network and parameters in detail.

Chapter 4

Implementation

This chapter elaborates the implementation details of the proposed solution. In the below sections, code segments of non-generic data structures proposed by this work, critical algorithms, and parameters used in the model are explained along with the details about the prototype, software tools, and services used.

4.1 Software tools

The proposed solution was implemented using python version 3.6. WFDB Software Package is used for dataset related operations, signal processing, automated analysis, annotation, and interactive analysis of waveform data (LightWave). MIT-BIH dataset parsing has been done using the WFDB package. Google Colab, which runs entirely in the cloud facilitated GPU computations for the Convolutional Neural Network. Colaboratory tool has been used to train the neural network model in less time. Hardware configurations are simulated by using Google Cloud Platform (GCP) virtual machines.

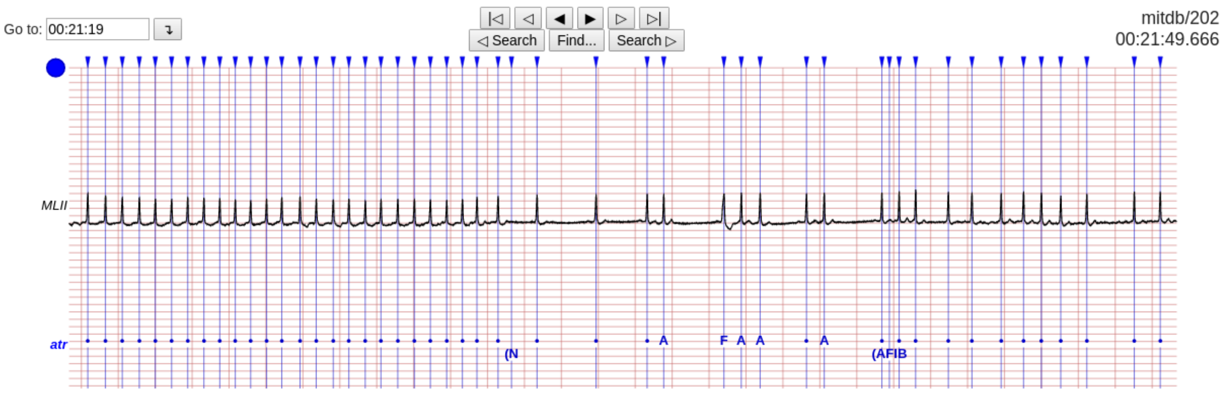

Figure 4.1: Interface of LightWave application

4.2 Dataset improvements

The MIT-BIH dataset has 48 subject records each recorded from Holter monitor. Some records in the dataset are unstable and contain power line interference, baseline wanderings, and noise. In order to overcome such issues, a median filter is used to calculate the drift in the baseline and subtracted from the original signal to eliminate baseline wanderings as implemented in the below function.

Listing 4.1: Baseline alignment

```
def remove_baseline_wandering(channel):
    res = []
    # median_filter1D
    baseline = medfilt(channel, 71)
    baseline = medfilt(baseline, 215)
    # Remove Baseline
    for i in range(0, len(channel)):
        res.append(channel[i] - baseline[i])
    return res
```

As described in section 3.3.2, R-peaks are not accurately annotated in the dataset. Searching for a maximum or minimum turning has been done for a specific threshold range around the annotated R-peak as shown in the code segment below. For a given sample sequence and annotated beat position, below function outputs the corrected R-peak position

Listing 4.2: R-peak correction

```
def rpeaks_annotation_correction(beat, cpos):
    max_pos = cpos
    min_pos = cpos

    for j in range(cpos-10, cpos+10):
         # Threshold from peak is taken as 10
        if beat[j] > beat[max_pos]:
            max_pos = j

        if beat[j] < beat[min_pos]:
            min_pos = j

    if beat[max_pos]-beat[cpos] > beat[cpos]-beat[min_pos] \
      and beat[min_pos]-beat[min_pos+1] < 0 :
        return min_pos
    else:
        return max_pos
```

4.3 Input translation

One of the innovative approaches in this work is input translation. In order to translate a beat signal into a 2D representation, the proposed direct sample matching through range stretching technique has been implemented as below. For a given 1-D array, this function will find an appropriate 2-D representation where a 3x3 erosion filter follows each mapped coordinate.

Listing 4.3: Input translation with erosion

```
def image_translation(samples):
    max_val = max(samples)
    min_val = min(samples)

    img = Image.new('1', (190, 100))
    pixels = img.load()
    for i in range(img.size[0]):
        for j in range(img.size[1]):
            pixels[i, j] = 255

    y_positions = [map_y_pixel(x, min_val, max_val, 90) \
                   for x in samples]
    for i in range(180):
        xpos = i+5
        ypos = (90-y_positions[i])+3

        # erosion filter
        pixels[xpos-1, ypos-1] = 0
        pixels[xpos-1, ypos] = 0
        pixels[xpos-1, ypos+1] = 0

        pixels[xpos, ypos-1] = 0
        pixels[xpos, ypos] = 0
        pixels[xpos, ypos+1] = 0

        pixels[xpos+1, ypos-1] = 0
        pixels[xpos+1, ypos] = 0
        pixels[xpos+1, ypos+1] = 0

    return img

def map_y_pixel(val, min_val, max_val, height):
    return int((val-min_val)*height/(max_val-min_val))
```

4.4 Alignment scoring

A primary component of the design proposed by this work is sequence alignment engine. A new data structure based on circular arrays has been implemented as shown in listing 4.4. A bucket structure is a fixed size matrix which holds a set of sequences. The least recent sequence in the bucket structure gets invalidated and substituted by the new sequence which is pushing into the bucket. The function 'getMedian' outputs a sequence with the median value of each column in the bucket matrix.

Listing 4.4: Bucket data structure

```
class AlignmentBucket:
    ... # more code here
    def push(self, data):
        self.data[self.nextpos] = data
        self.nextpos = (self.nextpos+1)%self.max_size
        self.median_arr = []

    def getMedian(self):
        if len(self.median_arr) > 0:
            return self.median_arr

        result = []
        bucket_matrix =  np.array(self.data[0:self.length()])
        if len(bucket_matrix) > 0:
            for i in range(len(bucket_matrix[0])):
                result.append(median(bucket_matrix[:,i]))
            self.median_arr = result
        return result
```

Listing 4.4 shows how bucket structure is used to find the alignment scores with unknown sequences with FastDTW algorithm.

Listing 4.5: Measurng alignment scores

```
similarities = []
for bucket in self.buckets:
    if bucket != None:
        bucket_beat = bucket.getMedian()
        distance = dtw.distance_fast(bucket_beat, input_beat)
        similarities.append(distance)
    else:
        similarities.append(math.inf)
```

4.5 CNN model

A model similar to LeNet Convolutional neural network architecture has been trained. The model consists of three convolution layers and three fully connected dense layers. Convolution layers are followed by a max pooling layer which reduces the number of dimensions. Initially, we start with 64x64 beat image. First convolution layer convolves the image with a 3x3 filter. For convolutional layers, we use a ReLu activation function. Convolution is followed by a pooling operation with a 2x2 filter. Pooling operation with 2x2 filter reduces the number of dimensions by half. After two convolutions and two max pooling operations, a fully connected hidden layer has been added with ReLu activation. Finally, a fully connected output layer with 5 output neurons (number of beat rhythm types) has been added. In order to get probabilities for each output neurons (between 0 and 1), a softmax activation function is used in the output layer. Implementation of the proposed CNN model used Keras machine learning libraries and GPU-based Tensorflow backend as shown in the listing 4.5

Listing 4.6: Convolutional neural network structure

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(Conv2D(128, (2, 2), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.1))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(units=100, activation='relu'  ))
model.add(Dropout(0.3))
model.add(Dense(num_classes, activation='softmax'))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])
```

4.6 Prototype application

After the evaluation, the proposed hybrid model is tested on a real-time ECG streaming prototype application. This application is also implemented using python 3.6. The prototype is a three threaded application to perform the tasks of ECG reading, classification, and visualization.

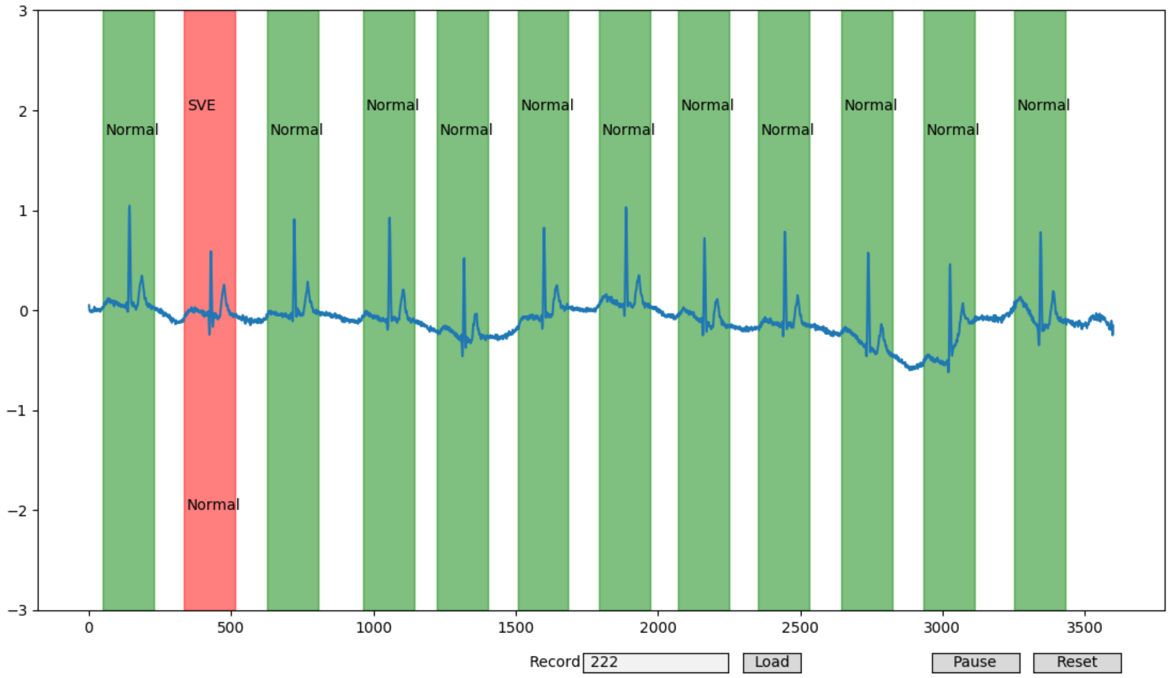

Figure 4.2: Graphical-user interface of the prototype application

A user can input any ECG recording, and the recording will be passed to the system as a stream. If there is an annotated value, both prediction and annotation are shown.

4.7 Virtual machine configurations

Google Cloud Platform offers (GCP) virtual machines that can customize the hardware as required. Ubuntu 16.04.5 LTS (GNU/Linux 4.15.0-1026-gcp x86_64) with 10 GB Standard persistent disk virtual machine has been used with different configurations of hardware. Virtual machines were configured to use 10 GB, 2 GB, 1 GB and 1 GB main memory with 4 vCPUs, 2 vCPUs, 1 vCPU, and 1 vCPU respectively.

Chapter 5

Results and evaluation

This chapter presents how the results evaluate the levels of the proposed solution. Evaluation is based on Massachusetts Institute of Technology Beth Israel Hospital (MIT-BIH) arrhythmia database [21], a benchmark dataset for arrhythmia. Forty-six recordings of the MIT-BIH arrhythmia database are considered for the classification of five heartbeat types as per recommendation by the ANSI/AAMI standards [33]. Although AAMI standard groups the sixteen beat types to five major groups, it is harder to generalize the features for a dominant group. Classification model which consider sixteen classes instead of the five super-classes may increase the accuracy. However, due to the lack of an adequate number of beats for several classes, it is not valid to use sixteen classes with a deep learning classifier.

5.1 Dataset

The subjects were 25 men aged 32 to 89 years, and 22 women aged 22 to 89 years. About 60% of the records were obtained from inpatients.

Two possible methods of splitting the dataset are as follows.

1. Split by record (subject)
   Example: Out of 48 records 34 records for training and 14 records for testing

2. Split by interval
   Example: Extract different time intervals from each record and split into the training and testing sets.

Split by record method tend to make the classifier more personalized. However, such an approach needs more subjects. Since the dataset contains only 46 subjects, training and evaluating based on subjects is not a good choice. Therefore, the splitting of the dataset by considering the time interval is the most appropriate option. From each 30 minute recording in MIT-BIH dataset, a set of intervals has been selected for the evaluation. Selected intervals are categorized based on the beat rhythms consist of each interval. From each category, random intervals have been selected to train the classification model and to test the classification model.

As the deep learning arrhythmia classification model can be independently fused into any other frameworks, streaming of data is not considered when evaluating the performance of the model. Figure 5.1 depicts the flow of evaluating the classification model.

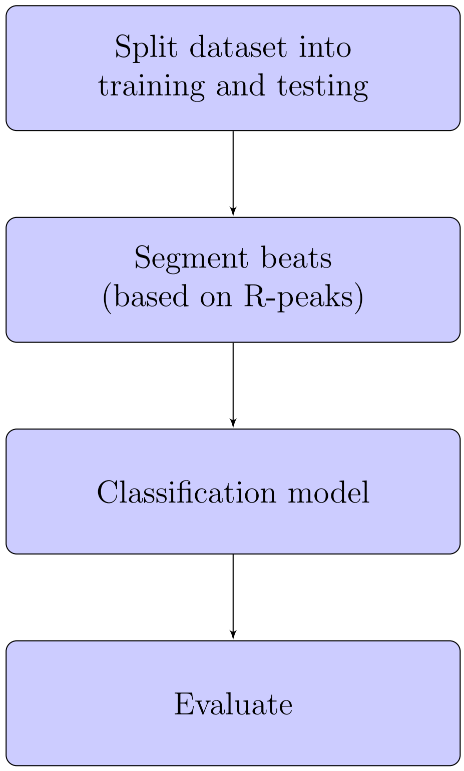

Figure 5.1: Classifier highlevel flow

5.2 Evaluation protocol

Two-fold evaluation where the performance of the arrhythmia classifier and performance of the real-time arrhythmia detection framework with respect to time is evaluated separately.

5.2.1 Performance of the classifier according to AAMI standard

According to the measures recommended by AAMI [33], Accuracy, Sensitivity and Positive Predictive Value (PPV) are measured in order to access the performance of the classifier. True positive (TP), true negative (TN), false positive (FP) and false negative (FN) values are used to calculate the three performance measures as mentioned in equation 5.1, equation 5.2 and equation 5.3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{(equation 5.1)}$$

$$Sensitivity = \frac{TP}{TP + FN} \quad \text{(equation 5.2)}$$

$$PPV = \frac{TP}{TP + FP} \quad \text{(equation 5.3)}$$

As the dataset is an imbalance, the accuracy of separate classes is evaluated in terms of balanced accuracy. Balanced accuracy can be denoted as in the equation 5.4.

$$Accuracy_{balanced} = \frac{Sensitivity + Specificity}{2} \quad \text{(equation 5.4)}$$

where specificity is defined as,

$$Specificity = \frac{TN}{FP + TN} \quad \text{(equation 5.5)}$$

For the CNN model, training and testing datasets are created in the proportion of 7:3. i.e., 70% of the data for training and rest 30% for testing. Performance is compared with standard CNN models.

The hybrid classifier is then evaluated with compared to the proposed CNN classifier. As the hybrid classifier closely related with adjacent beats, the comparison is made by passing the complete dataset as a sequence to both of the classifiers. The primary objective of this comparison is to check whether if there is a performance increase in the hybrid classifier than using only CNN classifier.

5.2.2 Prediction time of the classifier

In order to evaluate the complete framework, ECG data in the MIT-BIH dataset or similar is streamed as continuous time series data. Performance profiling is a crucial standard in evaluating prediction time, CPU consumption and memory consumption of the proposed framework. The significance of the framework is based on the accuracy of the framework with respect to the prediction time.

5.3 Results

In this section, results are expressed independently for the convolutional neural network classifier and the hybrid framework. However, the CNN model is used to compare the speed of the hybrid classifier.

5.3.1 Cross-validation and hyper-parameters

Before the evaluation of the hybrid classifier, a set of parameters needed to be tuned. These parameters are called hyper-parameters. It includes filter sizes for convolution, filter sizes for pooling, number of neurons in a layer, batch size, number of epochs in CNN classifier and threshold values for the rule-based decision module. Tuning hyper-parameters is an iterative process that generally takes a significant amount of time to obtain a top performance. Both automated and manual approaches were followed in order to set the values for hyper-parameters.

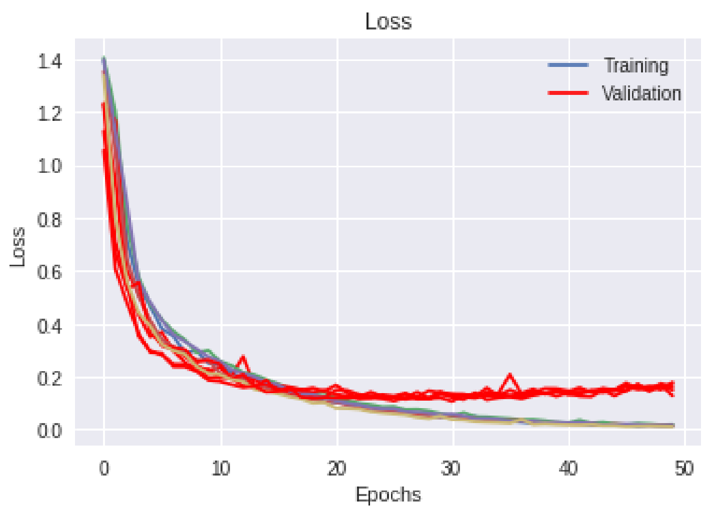

Figure 5.2: CNN loss for 5-fold cross-validation

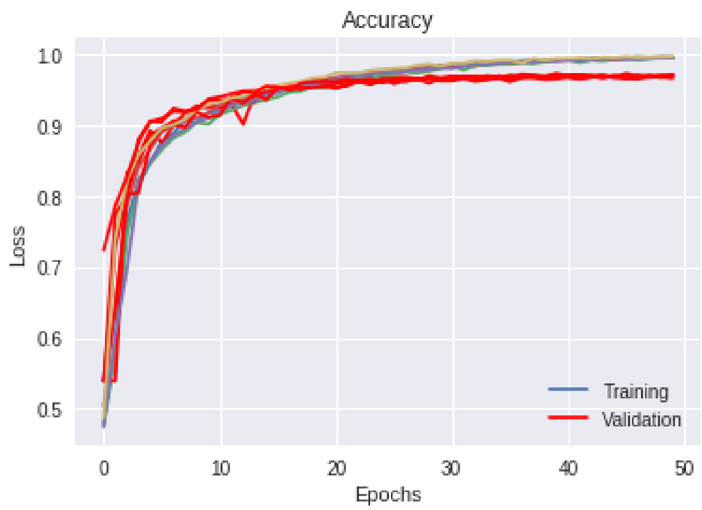

Figure 5.3: CNN accuracy for 5-fold cross-validation

Convolutional neural network model is subjected to cross-validation in order to assess how the results of analysis generalize to an independent dataset. Cross-validation results are also used to set different parameters of CNN model. 70% of the beats were taken for the training and the rest 30% as testing dataset. 70% of the training data were again split into 5 chunks and a single chunk is chosen to be the validation dataset at each iteration. 5 iterations were done and results were plotted into two graphs, loss and accuracy.

Loss graph for the modified LeNet CNN model is shown in the figure 5.2. Here the graph is drawn for training loss and validation loss (in red). Although the training loss decreases with the number of epochs, validation loss is being constant and tends to increase slightly after 20 epochs. Towards 40 to 50 epochs, there is around 0.2 loss gap and between training and validation which indicates a possible overfitting. Therefore, 22 number of epochs would be a good valid estimate for this model.
Figure 5.2 and 5.3 shows that the loss and accuracy always follow a similar path without major outliers. Therefore the cross-validation results suggest that the proposed model gives consistent results. Using manual trial and error methods batch size is chosen to be 1000.

5.3.2 Convolutional neural network performance

As described in the section 5.2, CNN model is based on AAMI/ANSI standards. Figure 5.5 and figure 5.4 shows loss and accuracy respectively for the instance of the CNN classifier which is using in the hybrid classifier. Model accuracy and loss are similar in the curve as the cross-validation results.

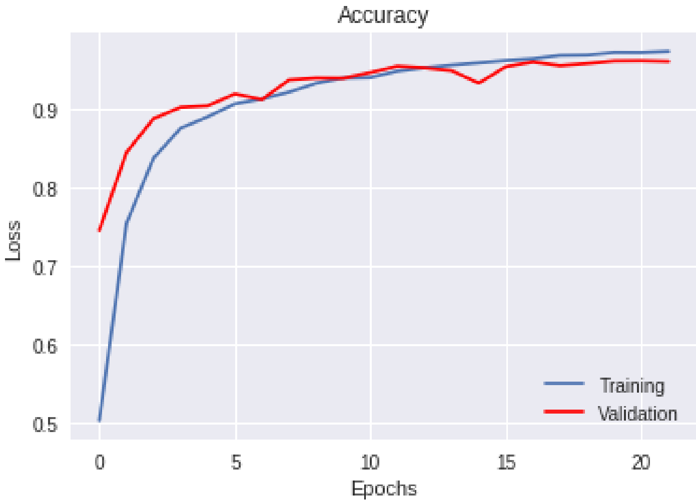

Figure 5.4: CNN model accuracy against number of epochs

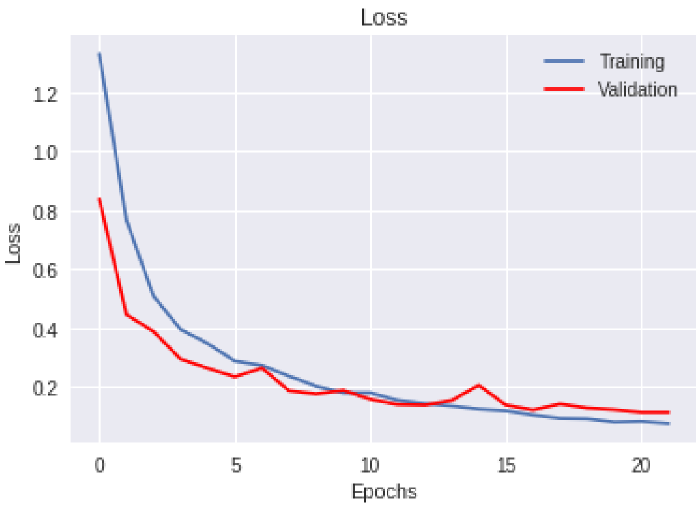

Figure 5.5: CNN model loss against number of epochs

Training of the model was carried out for 22 epochs and results of training and validation has a very slight loss difference of 0.04. Therefore, the predictions from the model can be taken as well generalized.

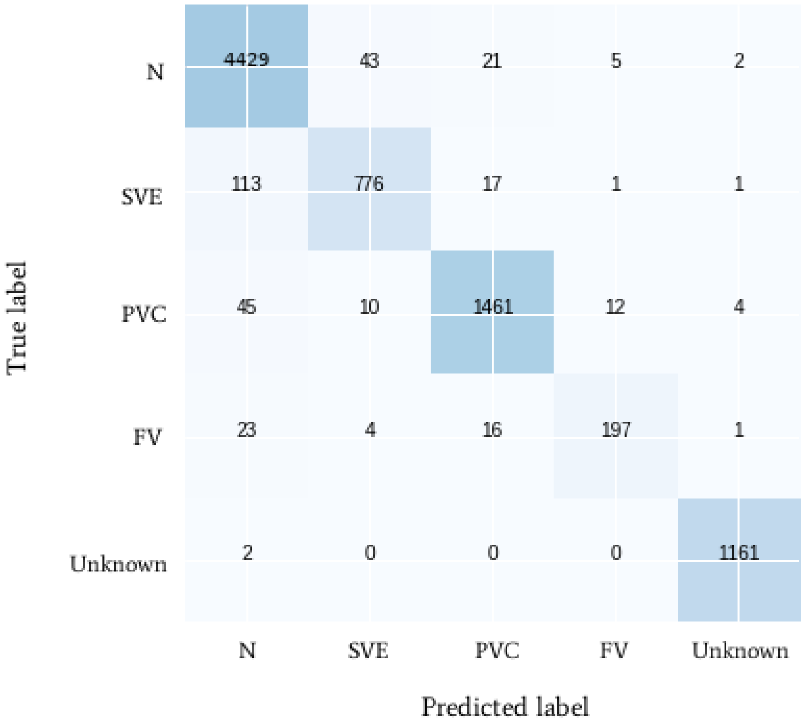

Figure 5.6: Confusion matrix for the testing dataset

The confusion matrix for the test data is shown in the figure 5.6. 19,460 samples were trained and tested on 8,344 samples. Performance metrics are listed in the table 5.1.

Table 5.1: Performance of the CNN model for the evaluation metrics

	Accuracy (%)	Sensitivity (%)	Precision (%)	Specificity (%)
N	96.8%	98.4%	96%	95.2%
SVE	92.4%	85.5%	93.2%	99.2%
PVC	97.3%	95.4%	96.4%	99.2%
FV	90.7%	81.7%	91.6%	99.8%
Unknown	99.8%	99.8%	99.3%	99.9%

In the table 5.1, accuracy is denoted in terms of balanced accuracy. Sensitivity for Supraventricular Ectopic (SVE) beats and Fusion of Ventricular (FV) beats have the lowest scores. All the ECG records in MIT-BIH dataset were individually evaluated, and almost all the records hit an accuracy above 95% except the record 222. Record 222 gained an accuracy of 74% which is highly deviated from the performance of other records. A large number of SVE beats are present in the record 222 (420 SVE beats). When considering the noise, record 222 is the noisiest record out of the 46 subjects. Filtering noise approach through a band-pass filter which is used in this work has reduced general noise in most of the beats, but since the record 222 contains abnormal noise condition, it has failed to achieve an approving result for the record 222. Fusion of Ventricular beats are less in number. There are only 802 beats in the whole dataset, which is the minimum count for a beat type. Not having adequate number of data for FV could have been a reason for such a low performance.

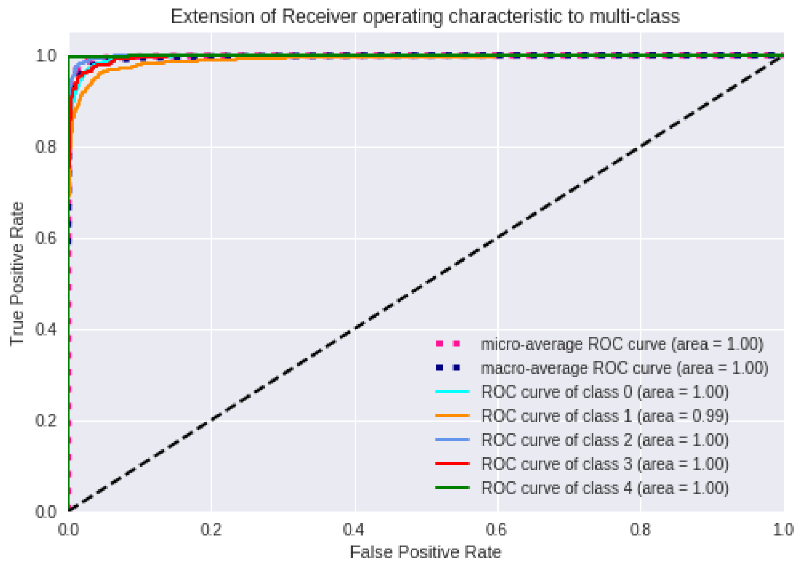

Figure 5.7: ROC curves for the five classes
Classes 0, 1, 2, 3 and 4 represents N, SVE, PVC, FV and Unknown respectively.

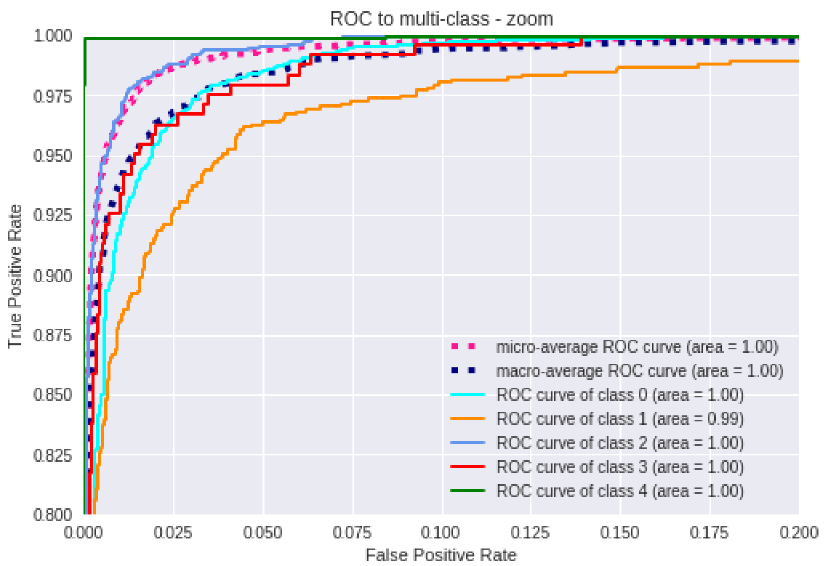

Figure 5.8: ROC curves for the five classes - zoomed to the top left

An overall measure of test performance can be visualized using Receiver Operating Characteristic (ROC) curves. According to the ROC curves which are shown in the figure 5.7, area under the curve (AUC) is almost 1 for all the classes. (Top left corner focused ROC curve on figure 5.8) Standards of AUC defines that area of 1 represents a perfect test while the area of 0.5 and below represents a worthless test/test by chance. Here the proposed classifier has obtained an excellent test performance.

With the proposed CNN classifier model, following two CNN models were chosen to evaluate. VGG19 and AlexNet. One of the focus of this work is to reduce the number of computations to speed-up the prediction time. Proposed CNN model only uses only three convolutional layers but still have able to achieve the state-of-art performance.

Table 5.2: CNN model accuracy comparison
Prediction times are evaluated on a virtual machine with 4 vCPUs, 10 GB memory.

Model	Convolutional layers	Accuracy	Avg. Prediction time (1000 beats)
VGG19	15	**97.59**%	40.15 sec.
AlexNet	5	97.04%	14.13 sec.
Proposed CNN model	3	96.45%	**8.12 sec.**
Proposed CNN model (without erosion)	3	94.9%	**8.12 sec.**

When comparing erosion of sample inputs proposed in this work with non-eroded inputs, the proposed erosion method has given a prominent accuracy gain. Notably, the prediction time remains the same in both instances. It concludes that the erosion has enhanced the features for convolution. Prediction time of the CNN classifier model is not fine-tuned and deeply evaluated as this research introduces a hybrid classifier which is an envelope on the CNN classifier.

5.3.3 Hybrid classifier performance

In order to evaluate the hybrid classifier, each record is streamed to the classifier as a sequence. Therefore, while making the predictions classifier consider the context of recent beats. For the comparison, all 105,063 beats from 46 ECG recordings are used. Table 5.3 depicts the sensitivity and specificity scores for the two classifiers. Note that, 105,063 beats were considered in evaluating the metrics (complete dataset).

Table 5.3: Sensitivity and specificity comparison for CNN and hybrid classifiers

	Sensitivity (%)		Specificity (%)	
	CNN	Hybrid	CNN	Hybrid
N	96.2%	**97.0%**	**97.1%**	95.9%
SVE	**89.0%**	87.4%	98.2%	**98.5%**
PVC	**98.2%**	97.8%	98.9%	**99.1%**
FV	**83.9%**	80.6%	99.3%	**99.6%**
Unknown	**99.8%**	99.2%	99.9%	**99.9%**

The hybrid classifier correctly classified 101,964 out of 105,063 beats while CNN classifier predicted 101,079 beats accurately. i.e., **97.05%** accuracy for the hybrid classifier and 96.20% accuracy for the CNN classifier. Around 1% gain in accuracy is observed here. Although the specificity of the hybrid classifier is higher than the specificity of the CNN classifier (except for normal beats), sensitivity loss is present for SVE, PVC, FV and Unknown beats. 0.8% sensitivity gain is present for normal beats. The main reason for the gain is that in the hybrid classifier, what it considers is finding the normal beats. From the alignment scores, decision rules of the classifier are all based on the score for the normal beats. If the decision layer considers finding all the beat types (using the alignment bucket scores), sensitivity could be further improved. However, the hybrid classifier is introduced mainly to improve the prediction time. Therefore, such alternative approaches are not evaluated in this work.

Table 5.4: Prediction time comparison for CNN and Hybrid classifiers

Hardware configuration	CNN	Hybrid
4 vCPUs, 10 GB memory	854.27 sec.	**414.01 sec.**
2 vCPUs, 2 GB memory	876.95 sec.	**457.25 sec.**
1 vCPUs, 2 GB memory	886.95 sec.	**464.87 sec.**

It can be observed in table 5.4 that prediction time of the proposed hybrid classifier is approximately two times faster than a classification with CNN only classification. CNN classifier takes around 14-15 minutes for the classification while Hybrid classifier classifies within 6-7 minutes.

Significance in the prediction time is due to the consideration of the similarity and the alignment score of adjacent beats. It is proven when evaluating each record against the hybrid classifier. Table 5.5 shows how the similarity and alignment scores affected the prediction time. From the table, records having a higher number of beat classifications by similarity has less prediction time. Records 208, 222, 228, 217, 232, 200 and 107 are having a higher number of beats classified with CNN than the alignment scores thus considerable time is taken.

Table 5.5: Number of beats classified by similarity for each record

Record (time)	Total beats	Classifications by similarity	Record (time)	Total beats	Classifications by similarity
202 (15 s)	2136	797	116 (10 s)	2412	1557
215 (9 s)	3363	2831	113 (4 s)	1795	1543
230 (8 s)	2256	1629	217 (16 s)	2208	159
231 (3 s)	1571	1382	112 (8 s)	2539	1956
119 (11 s)	1987	855	111 (9 s)	2124	1359
208 (24 s)	2953	466	100 (8 s)	2273	1449
108 (10 s)	1763	820	201 (15 s)	1963	579
115 (4 s)	1953	1729	209 (19 s)	3005	1019
117 (6 s)	1535	991	109 (7 s)	2532	2227
105 (12 s)	2567	1637	221 (17 s)	2427	793
124 (9 s)	1619	921	101 (16 s)	1863	253
210 (14 s)	2650	1566	233 (19 s)	3079	1320
121 (12 s)	1863	836	114 (8 s)	1879	1246
222 (19 s)	2483	492	118 (10 s)	2278	1470
207 (10 s)	1860	937	212 (6 s)	2748	2362
213 (23 s)	3251	1176	232 (14 s)	1780	94
220 (5 s)	2048	1682	214 (8 s)	2260	1683
223 (21 s)	2605	600	123 (9 s)	1518	699
205 (7 s)	2656	2285	103 (3 s)	2084	1862
228 (16 s)	2053	418	219 (14 s)	2154	1019
122 (4 s)	2476	2228	106 (12 s)	2027	806
234 (8 s)	2753	2214	200 (22 s)	2601	229
203 (27 s)	2976	534	107 (17 s)	2137	0

Above results are based on the hybrid classifier with CNN and rule-based model in the decision layer. When replacing the rule-based model with SVM classifier to identify normal beats in the decision layer, accuracy tends to decrease. Achieved accuracy by using the SVM model is 95.76%. Employing many hand-engineered features in the decision layer is a cost because the decision layer is a wrapper around the CNN model. Therefore, only two feature extraction methods were employed for the SVM model which are namely decomposed wavelets and re-sampling. Trained linear SVM model to identify 'normal' and 'other' beats have given an accuracy of 84.4% in identifying the two classes. Incorporating more features would have increased the accuracy, but higher cost is undoubtedly a limitation. To achieve 95.76% accuracy hybrid classifier with SVM model took 702 seconds. Therefore, the most suitable method for the decision layer from the two approaches is the rule-based classifier.

5.4 Summary

This chapter presented the proposed evaluation model for the CNN classifier and the Hybrid classifier. CNN classifier performance was slightly lesser with a gap of 1.14% with the best performing model, but five times computationally efficient due to the lesser convolution layers. The proposed hybrid classifier which is a wrapper on the CNN model has decreased the computational time significantly and slightly increased the classification accuracy as well. Decision layer of the hybrid classifier is the crucial intersection which balances the performance and prediction time. For the decision layer, a rule-based method and SVM method has been tested and evaluated. The rule-based method gave the top performance with lesser cost than the SVM model.

Chapter 6

Conclusions and future work

This chapter includes a review of the research aims and objectives, research problem, limitations of the current work and implications for further research.

6.1 Conclusion

In this research, we have analyzed the applicability of deep convolutional neural network based classifiers for the real-time classification of arrhythmia. When the number of convolutions increases, the prediction time has much effect due to the increase in parameters. However, there is a significant improvement in accuracy. Classifiers with less number of convolutions have given a good prediction time but with a slight loss in accuracy.

The proposed solution employs a hybrid classifier for the question of extending arrhythmia classification into a real-time approach by preserving accuracy. In practice, without a proper understanding of the problem, it is difficult to propose a solution. With domain advisory from medical professionals and in-depth analysis of the patterns, we proposed a solution to real-time arrhythmia classification framework using the similarity between adjacent beats. This kind of classifiers not only achieve the state-of-the-art in accuracy but also outperforms the state-of-art performance concerning time. 96.45% accuracy for the proposed CNN model and 97.05% overall accuracy for the hybrid classifier was achieved. The accuracy obtained surpasses most of the machine learning and rule-based techniques in the literature (refer the table 2.1) for five class classification. The hybrid classifier was able to speed up the prediction time by twice in times than using only a CNN classifier. Thus, it can be concluded that the proposed approach can be a viable solution in incorporating deep learning classifiers for real-time classification in this domain.

Detection of arrhythmia from streaming electrocardiographic signals can be efficiently refined by considering the nature of adjacent beats and calculating the alignment scores for each rhythm class. However, the accuracy of the base classifier model should be highly accurate. The hybrid classifier uses a convolutional neural network as the base classifier and focuses on determining the similarity through approximating a beat for each rhythm class. This method of approximating a beat using the median values of most fresh beats limit propagating CNN erroneous classifications. Therefore, we have introduced a concept of alignment buckets to measure the alignment scores of ECG beats using the DTW algorithm. This approach is a significant contribution to real-time sequence classification as it does not compare with each sequence, but approximate for a median sequence in a FIFO method.

This study also contributed to the domain of arrhythmia analysis by introducing two non-mutually exclusive approaches, a CNN and an alignment scoring wrapper on CNN. The two proposed approaches are capable of yielding predictions in lesser time while preserving the accuracy. We have also shown that the erosion of input can improve the accuracy of the prediction. Also, comparison with the existing hybrid approaches, the combined approach do not require hand-crafted feature extraction. Proposed classifier is extensible and can be further extended to improve the results by adjusting the defined parameters according to the requirement of the classification. The primary requirement of this work is to balance the performance with prediction time, and the proposed approach has given high-grade results.

6.2 Limitations

The evaluation results showcased that there are certain special scenarios where noise affected a strong accuracy degrade. Since this work does not consider any specific hardware devices to capture ECG recordings, the level of required noise filtering cannot be determined. Although a band-pass filtering method is employed to reduce the most common noise, a sound noise reduction needs to be incorporated. Proposed hybrid classifier model does not provide a mechanism to detect R-peaks from a sequence of ECG data. Model is relying on the annotated peak positions.

6.3 Future work

Although accuracy, specificity and prediction time of the proposed hybrid classifier has achieved the state-of-art, sensitivity has slightly degraded when compared to the proposed CNN classifier. Sensitivity would have further increased if additional rules appended to the decision layer. Current decision rules focus mainly on identifying normal beats. Therefore, the sensitivity for the normal beats is higher. As future works, decision rules can consider incorporating other beat types as well using the alignment scores. Apart from a rule-based model, data-driven approaches can be considered for the decision layer if the requirement is to improve the accuracy further, instead of the prediction time.

This method is not applicable to all the other domains which use convolutional neural networks with real-time needs. Our approach is based on a specific problem, i.e., identifying arrhythmia in which we have used a concept of similarity. However, in-depth analysis of the problem domain may derive a mapping to the design of the proposed hybrid classifier.

In this work, the MIT-BIH benchmark dataset has been used to evaluate the model. Evaluation model can also be extended to evaluate for the generalizability of the proposed approach by incorporating the classifier with a live system.

References

[1] "Cardiovascular diseases (cvds)." [Online]. Available: http://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[2] H. C. McGill Jr, C. A. McMahan, and S. S. Gidding, "Preventing heart disease in the 21st century: implications of the pathobiological determinants of atherosclerosis in youth (pday) study," *Circulation*, vol. 117, no. 9, pp. 1216–1227, 2008.

[3] C.-C. Chiu, T.-H. Lin, and B.-Y. Liau, "Using correlation coefficient in ecg waveform for arrhythmia detection," *Biomedical Engineering: Applications, Basis and Communications*, vol. 17, no. 03, pp. 147–152, 2005.

[4] R. Poplin, A. V. Varadarajan, K. Blumer, Y. Liu, M. V. McConnell, G. S. Corrado, L. Peng, and D. R. Webster, "Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning," *Nature Biomedical Engineering*, vol. 2, no. 3, p. 158, 2018.

[5] J. R. Hampton, *ECG MADE EASY*. Elsevier, 1997.

[6] M. Gertsch, *The ECG*. Springer, 2004.

[7] T. Ince, M. Zabihi, S. Kiranyaz, and M. Gabbouj, "Learned vs. hand-designed features for ecg beat classification: A comprehensive study," in *EMBEC & NBC 2017*. Springer, 2017, pp. 551–554.

[8] A. Isin and S. Ozdalili, "Cardiac arrhythmia detection using deep learning," *Procedia Computer Science*, vol. 120, pp. 268–275, 2017.

[9] A. J. Moss and S. Stern, *Noninvasive electrocardiology*. W.B. Saunders, 1996.

[10] D. E. Gustafson, A. S. Willsky, J.-Y. Wang, M. C. Lancaster, and J. H. Triebwasser, "Ecg/vcg rhythm diagnosis using statistical signal analysis-i. identification of persistent rhythms," *IEEE Transactions on Biomedical Engineering*, no. 4, pp. 344–353, 1978.

[11] N. V. Thakor and Y.-S. Zhu, "Applications of adaptive filtering to ecg analysis: noise cancellation and arrhythmia detection," *IEEE transactions on biomedical engineering*, vol. 38, no. 8, pp. 785–794, 1991.

[12] P. Schwab, G. Scebba, J. Zhang, M. Delai, and W. Karlen, "Beat by beat: Classifying cardiac arrhythmias with recurrent neural networks," *arXiv preprint arXiv:1710.06319*, 2017.

[13] P. Rajpurkar, A. Y. Hannun, M. Haghpanahi, C. Bourn, and A. Y. Ng, "Cardiologist-level arrhythmia detection with convolutional neural networks," *arXiv preprint arXiv:1707.01836*, 2017.

[14] F. Andreotti, O. Carr, M. A. Pimentel, A. Mahdi, and M. De Vos, "Comparing feature-based classifiers and convolutional neural networks to detect arrhythmia from short segments of ecg," *Computing*, vol. 44, p. 1, 2017.

[15] J. Pan and W. J. Tompkins, "A real-time qrs detection algorithm," *IEEE Trans. Biomed. Eng*, vol. 32, no. 3, pp. 230–236, 1985.

[16] "Qrs complex," Jan 2007. [Online]. Available: https://goo.gl/vxQMGG

[17] D. A. Coast, R. M. Stern, G. G. Cano, and S. A. Briller, "An approach to cardiac arrhythmia analysis using hidden markov models," *IEEE Transactions on biomedical Engineering*, vol. 37, no. 9, pp. 826–836, 1990.

[18] S. Melo, L. Caloba, and J. Nadal, "Arrhythmia analysis using artificial neural network and decimated electrocardiographic data," in *Computers in Cardiology 2000*. IEEE, 2000, pp. 73–76.

[19] P. Hamilton, "Open source ecg analysis software documentation, ep limited, 2002."

[20] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of qrs detection rules using the mit/bih arrhythmia database," *IEEE transactions on biomedical engineering*, no. 12, pp. 1157–1165, 1986.

[21] 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.

[22] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[23] S. Gradl, P. Kugler, C. Lohmüller, and B. Eskofier, "Real-time ecg monitoring and arrhythmia detection using android-based mobile devices," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE.* IEEE, 2012, pp. 2452–2455.

[24] A. Khazaee and A. Ebrahimzadeh, "Heart arrhythmia detection using support vector machines," *Intelligent Automation & Soft Computing*, vol. 19, no. 1, pp. 1–9, 2013.

[25] S. Chauhan and L. Vig, "Anomaly detection in ecg time signals via deep long short-term memory networks," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on.* IEEE, 2015, pp. 1–7.

[26] K. Polat and S. Güneş, "Detection of ecg arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine," *Applied Mathematics and Computation*, vol. 186, no. 1, pp. 898–906, 2007.

[27] V. H. C. de Albuquerque, T. M. Nunes, D. R. Pereira, E. J. d. S. Luz, D. Menotti, J. P. Papa, and J. M. R. Tavares, "Robust automated cardiac arrhythmia detection in ecg beat signals," *Neural Computing and Applications*, vol. 29, no. 3, pp. 679–693, 2018.

[28] S. Anwar, K. Hwang, and W. Sung, "Structured pruning of deep convolutional neural networks," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 13, no. 3, p. 32, 2017.

[29] R. J. Wieringa, *Design science methodology for information systems and software engineering.* Springer, 2014.

[30] A. Hevner and S. Chatterjee, "Design science research in information systems," in *Design research in information systems.* Springer, 2010, pp. 9–22.

[31] P. S. Gokhale, "Ecg signal de-noising using discrete wavelet transform for removal of 50hz pli noise," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, no. 5, pp. 81–85, 2012.

[32] E. Butkeviciute, L. Bikulciene, and K. Poderiene, "Training system methodology using ecg signal," in *Precision Medicine Powered by pHealth and Connected Health*. Springer, 2018, pp. 261–265.

[33] "Testing and reporting performance results of cardiac rhythm and ST segment measurement algorithms," 2012.

[34] J.-H. Bae, S.-J. Lim, J.-J. Kim, S.-D. Park, and J.-D. Kim, "The classification of arrhythmia using similarity analysis between unit patterns at ecg signal," *The KIPS Transactions: PartD*, vol. 19, no. 1, pp. 105–112, 2012.

[35] B. Raghavendra, D. Bera, A. S. Bopardikar, and R. Narayanan, "Cardiac arrhythmia detection using dynamic time warping of ecg beats in e-healthcare systems," 2011.

[36] J. Park and K. Kang, "Heartsearcher: finds patients with similar arrhythmias based on heartbeat classification," *IET systems biology*, vol. 9, no. 6, pp. 303–308, 2015.

[37] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis*, vol. 11, no. 5, pp. 561–580, 2007.

[38] M. Lin, Q. Chen, and S. Yan, "Network in network," *arXiv preprint arXiv:1312.4400*, 2013.

Appendix A

MIT-BIH database annotations

Table A.1: Beat annotations

Symbol	Meaning
N or .	Normal beat
L	Left bundle branch block beat
R	Right bundle branch block beat
B	Bundle branch block beat (unspecified)
A	Atrial premature beat
a	Aberrated atrial premature beat
J	Nodal (junctional) premature beat
S	Supraventricular premature or ectopic beat (atrial or nodal)
V	Premature ventricular contraction
r	R-on-T premature ventricular contraction
F	Fusion of ventricular and normal beat
e	Atrial escape beat
j	Nodal (junctional) escape beat
n	Supraventricular escape beat (atrial or nodal)
E	Ventricular escape beat
/	Paced beat
f	Fusion of paced and normal beat
Q	Unclassifiable beat
?	Beat not classified during learning

Table A.2: Non-beat annotations

Code	Description
[	Start of ventricular flutter/fibrillation
!	Ventricular flutter wave
]	End of ventricular flutter/fibrillation
x	Non-conducted P-wave (blocked APC)
(	Waveform onset
)	Waveform end
p	Peak of P-wave
t	Peak of T-wave
u	Peak of U-wave
‘	PQ junction
’	J-point
∧	(Non-captured) pacemaker artifact
—	Isolated QRS-like artifact
	Change in signal quality
+	Rhythm change
s	ST segment change
T	T-wave change
*	Systole
D	Diastole
=	Measurement annotation
”	Comment annotation
@	Link to external data