DEPARTMENT OF COMPUTER SCIENCE ENGINEERING,
SCHOOL OF ENGINEERING AND TECHNOLOGY,
SHARDA UNIVERSITY, GREATER NOIDA

DETECTION AND CLASSIFICATION OF CARDIAC ARRHYTHMIAS USING DEEP LEARNING

A project submitted
In partial fulfillment of the requirements for the degree of
Bachelor of Technology in Computer Science and Engineering

By
ZEESHAN AHMAD (2018009161)
ADITYA RAJ (2018007210)
SHIVANK GOEL (2018006884)
SARIK ANWAR (2018015190)

Supervised by:
Mr. Laxman Singh, Assoc. Prof. (CSE)

May, 2022

DECLARATION

I hereby declare that the Final Year Project task named "**DETECTION AND CLASSIFICATION OF CARDIAC ARRHYTHMIAS USING DEEP LEARNING**" is my original work. I further certify that I have followed all academic honesty and integrity norms and have not misrepresented, fabricated, or manipulated any idea, data, fact, or deliver in my work. I understand that any breach of the foregoing may result in disciplinary action by the Institute, as well as penal action from the assets that haven't been properly referred to or from whom proper authorization hasn't been obtained when required.

Name of Students
Zeeshan Ahmad (2018009161)
Shivank Goel (2018006884)
Aditya Raj (2018007210)
Sarik Anwar (2018015190)

CERTIFICATE

This is to certify that the project work entitled **"DETECTION AND CLASSIFICATION OF CARDIAC ARRHYTHMIAS USING DEEP LEARNING"** is completed by “**ZEESHAN AHMAD (2018009161), SHIVANK GOEL (2018006884), ADITYA RAJ (2018007210), SARIK ANWAR (2018015190)** ” to Sharda University, towards the fulfillment of requirements of the degree of “**Bachelor of Technology**” is record of bonafide final year Project work carried out by him in the “Department of Computer Science and Engineering, School of Engineering and Technology, Sharda University” in the year 2021-22
The results/findings contained in this Project have not been submitted in part or full to any other University/Institute for award of any other Degree/Diploma.

Signature of Supervisor

Name: Mr. Laxman Singh
Designation: Asst. Prof. (CSE)

Signature of Head of Department

Name: Prof. (Dr.) Nitin Rakesh
Place:
Date:

Signature of External Examiner

Date:

ACKNOWLEDGEMENT

A major project is a golden opportunity for learning and self-development. We consider our self very lucky and honored to have so many wonderful people lead us through in completion of this project. First and foremost we would like to thank **Dr. Nitin Rakesh, HOD (CSE)** who gave us an opportunity to undertake this project. My grateful thanks to **Mr. Laxman Singh** for their guidance in our project work. **Mr. Laxman Singh**, who in spite of being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where we would have been without her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

Name and signature of Students

ZEESHAN AHMAD (2018009161)
ADITYA RAJ (2018007210)
SHIVANK GOEL (2018006884)
SARIK ANWAR (2018015190)

ABSTRACT

Arrhythmia is a coronary heart rhythm circumstance that reasons abnormal heartbeats. The electrocardiographic (ECG) sign can screen abnormalities with inside the conduction machine. An electrocardiogram (ECG) is an vital diagnostic device for detecting coronary heart arrhythmias in medical practise. Due to the very low amplitudes, visually assessing the ECG alerts may be hard and time-consuming. Implementing an automatic method with inside the medical context ought to probably accelerate and decorate the accuracy of arrhythmia diagnosis. In this paper, we advocate an automatic machine for detecting ordinary sinus rhythm, R-on-T Premature Ventricular Contraction (R-on-T PVC), Supra-ventricular Premature or Ectopic Beat (SP or EB), Unclassified Beat (UB) and untimely ventricular contraction (PVC) on ECG alerts the usage of a protracted short-time period memory (LSTM). The primary cause of this look at is to create a deep mastering method for categorizing one-of-a-kind forms of arrhythmia this is simple, dependable, and clean to use. In order to categorize ordinary and pathological beats in an ECG, recurrent neural networks (RNN) have been used. The major intention of this studies changed into to make it feasible to routinely distinguish among everyday and abnormal beats. The beat type overall performance is assessed the usage of the MIT-BIH Arrhythmia database. As inputs to the Long Short Term Memory Network, a big quantity of popular data, consisting of ECG time-collection data, is used. The dataset changed into separated into education and trying out sub-data. The proposed technique done nicely in phrases of type, with a ninety seven percentage accuracy rate. Our proposed method can help clinicians in as it should be detecting not unusualplace arrhythmias.

Keywords: Deep Learning, ECG Detection and Classification, Recurrent Neural Networks, Long Short Term Memory

CONTENTS

Title Page……………………….………………………………………………….i
DECLARATION……………………………………………………………...…ii
CERTIFICATE…………………………………………………………...…..iii
ACKNOWLEDGEMENT……………………………………………...…….iv
ABSTRACT…………………………………………………………………...v
Chapter 1: INTRODUCTION……………………………………………...……7
1.1 Motivation………...…………………………………………………………9
1.2 Problem Statement……………………………………………………………..9
1.3 Problem Formulation/Objectives……………………………………………..10
1.4 Project Overview/Requirement Specifications………………………………10
1.5 Hardware Requirements………………………………………………………12
1.6 Software Specifications……………………………………………………….13
Chapter 2: LITERATURE REVIEW…………………………………………...16
2.1 Existing Systems…………………………………………………………...16
2.2 Proposed Model……………………………………………………………20
2.3 Feasibility Analysis………………………………………………………...23
Chapter 3: SYSTEM DESIGN AND ANALYSIS...26
3.1 System Design...26
3.2 Classification of Cardiac Arrhythmia..26
3.3 Working of Algorithm..27
Chapter 4: IMPLEMENTION……………..........……………………………………32
Chapter 5: RESULT………………………………………………………..............47
Chapter 6: CONCLUSION……………………………………………………..52
Chapter 7: REFERENCES……………………………………………………..53
Chapter 8: APPENDIX………………………………………………….........57

Chapter 1: INTRODUCTION

Electrocardiography (ECG) is an important and effective diagnostic tool for diagnosing heart disorders. The electrocardiogram (ECG) signal is an example of the bioelectrical functions of the heart. An electrocardiogram is a useful tool for diagnosing a person's condition. It provides complete facts about the processes of the body within the human body and therefore can be considered as a tool for the ability to assess health. Early diagnosis of coronary heart disease (abnormal) at an early age can help grow and improve unique life. Changes or confusion within the ECG signal seen by the observer was a common way to detect heart attacks. As a result, it is important for miles to improve the accuracy and performance of the signal and bit rate.

The default class of cardiac arrhythmia will provide objective diagnostic results and save time for cardiologists. These benefits have stimulated the agency's interest within the phases and analysis of ECG facts using tangible power.

The purpose of this study was to use the RNN Short-Term Memory Community to access arrhythmia from ECG alerts successfully. The ECG signal is performed from 5 single heartbeat styles, which can be divided into groups: daily and arrhythmia. Standard (N), R-on-T Premature Ventricular Contraction (R-on-T PVC), Premature Ventricular Contraction (PVC), Supra-ventricular Premature or Ectopic Beat (SP or EB), and Unclassified Beat (UB) many styles of arrhythmia heartbeats.

Cardiologists, who have spent years studying bias and arrhythmic beats, have repeatedly failed because of human nature, considering further research and development in the field of biotechnology. To diagnose arrhythmia, a wide range of learning and study methods have been used, and many of them have surpassed heart doctors. Now we will find various fashion learning systems to diagnose arrhythmia with the aim of gaining a higher knowledge of fashion and gaining details of what should and should not be suitable for this course.

To eliminate the need for human detection of arrhythmic rhythms within the ECG, we used a system learning gadget to automatically beat abnormal beats, which could also be further referred to a cardiologist for confirmation and further research. The accuracy of the type may be seen to some degree, and this paper can help the

documents simplify their drawings and be considered in the development and development of the conclusion.
An arrhythmia is a problem with the rate or rhythm of your heartbeat. It means your heart is beating very fast, very slowly, or with an unusual pattern. When the heart beats faster than normal, it is called tachycardia. When the heart is beating too slowly, it is called bradycardia.
An arrhythmia is a heart attack that affects the charge or rhythm where the heartbeat is. Electrical impulses can also appear to be very fast, very slow, or inconsistent - causing the coronary heart to break down too fast, too slowly, or even unevenly. When the heart is not beating properly, it is unable to pump blood properly. The arrhythmias may be pure or precarious. An electrocardiogram, also called an ECG, is often used to diagnose arrhythmia. The surgeon attaches electrodes to the chest, fingers, or legs to that degree and draws the electrical activity of the heart. In this exercise, we can use the ECG (continuous electrical signal of the coronary heart) and teach neural networks to anticipate coronary heart arrhythmias. We will find and differentiate Cardiac Arrhythmias from the use of Deep Learning.The heart is a pump made up of muscle tissue. Like all other tissues, the heart needs energy and oxygen to function. The heart pumping system is controlled by a circulatory system that controls access to various chambers of the heart.
Arrhythmias, also known as cardiac arrhythmias, cardiac arrhythmias, or dysrhythmias, are abnormalities in heartbeat, including when they are very fast or very slow. The fastest heartbeat - more than 100 beats per minute in adults - is called tachycardia, and the slowest heartbeat - less than 60 beats per minute - is called Some types of bradycardia arrhythmias have no symptoms. Symptoms, if any, may include palpitations or hallucinations during a heartbeat. In severe cases, there may be a mild headache, dizziness, shortness of breath or chest pains Although many forms of arrhythmia are not so serious, some cause complications such as stroke or heart failure. Some may cause sudden death.
The arrhythmias are usually divided into four groups: hypertension, supraventricular tachycardia, ventricular arrhythmias and bradyarrhythmias. Additional beats include premature atrial fibrillation, premature ventricular contraction and premature junctional congestion, Supraventricular tachycardias include atrial fibrillation, atrial fibrillation. Ventricular arrhythmias include ventricular filbrillation and ventricular tachycardia. Bradyarrhythmias are caused by sinus node dysfunction or

atrioventricular conduction disorders. Arrhythmias are caused by problems with the cardiovascular system. Many tests can be helpful in diagnosing, including electrocardiogram (ECG).

Many arrhythmias can be treated successfully. Treatment may include medication, medical procedures such as pacemaker implants, and surgery. Rapid heart rate medications may include beta blockers or antiarrhythmic agents such as procainamide, which attempt to restore normal heart rhythm. The latter group may have very important side effects, especially if taken for a long time. Pacemakers are often used for slow heartbeats.

1.1 Motivation

Arrhythmia is a term used to describe a set of disorders characterized by abnormal heartbeat that is very rapid or very slow. Also called cardiac arrhythmia or cardiac arrhythmia. Tachycardia is defined as more than 100 beats per minute in adults.

An arrhythmia may not be noticeable in some cases. In severe cases, mild headaches, fainting, shortness of breath or chest pain may occur. It can lead to stroke or heart failure in the worst case scenario. It is best to detect unusual heartbeats from an early age in these situations so that they can be dealt with before they become serious.

Arrhythmia is a common condition that affects millions of people worldwide. Arrhythmia is the cause of sudden cardiac death in 80 to 90 percent of cases. It could happen at any time.

Eliminate the need for a person to detect arrhythmic strokes on the ECG and help physicians make their job easier and consider future development and improvement.

Exploring various machine learning models for arrhythmia to get a better understanding of the models and get details of what is appropriate and appropriate for this field of study.

1.2 Problem Statement

An arrhythmia is a condition of the heart that causes the heart rate or rhythm to change. Arrhythmia is defined as any deviation from the normal sequence of electrical impulses. The lungs, brain, and other organs would not function properly when the heart was unable to pump blood properly, and it may shut down or damage it.

1.3 Problem Formulation/Objectives

The goal is to find the best state of the art accuracy class measure as possible, which should be better than the recently reported results in classifying similar types of arrhythmias. Other indicators, such as sensitivity and clarity, should also be high, suggesting that the proposed method is effective in predicting the type of cardiac arrhythmia that causes patients pain and discomfort in a timely manner, allowing them to seek treatment for the disease.

1.4 Project Overview/Requirement Specifications

1.4.1 Functional Requirements

1.4.1.1 Input

ECG Reading data

ECG studies can reveal heart rhythm disorders, sometimes known as arrhythmias. A thorough evaluation of ECG symptoms is essential for the accurate diagnosis of heart problems in critically ill and chronic patients. We have developed a two-dimensional (2-D) (CNN) model for dividing ECG signals into categories: normal, premature ventricular contraction, mobility, and more.

1.4.1.2 Processing

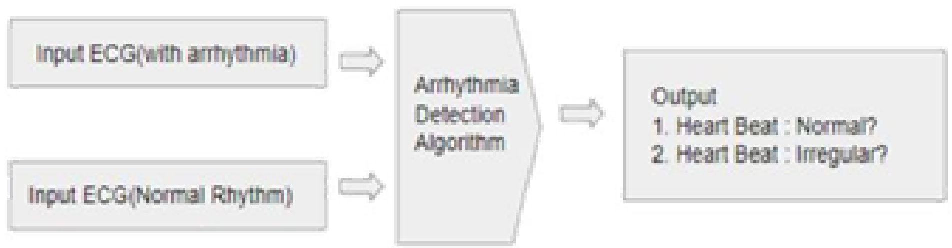


Fig. 1: Processing

1.4.2 Normal Requirement

These are customer-specific requirements, so customer satisfaction requirements must be met.

N1: The system should provide a user interface.

N2: The system should provide the most accurate results.

N3: The system must be functional and efficient.

1.4.3 Non Functional Requirements

These are specifications, as the name implies, that can be directly related to the specific functions offered by the device.

1.4.3.1 Performance Requirements

Relationship-based killings, high-end PC killings may include one or more concomitant: immediate response time for a small task. Compared to the time and resources used, the performance is defined by the amount of useful work performed by the PC framework or PC system.

1.4.3.2 Reliability

Fixed quality is the property of any PC-related component (for example, editing, or hardware, or a system) that operates consistently as its determination shows. For a time, it was seen as one of the three related factors to consider when making, purchasing, or using an object or component of a PC.

1.4.3.3 Availability

Accessibility is a common concept used in PC systems and system management to define the average time within a period of one year when frame assets are acquired after a partial system failure. A building with all its ever-present structures seems to bear fruit.

1.4.3.4 Security

Security (or PC security) on registration is a way to ensure that information stored on a PC cannot be accessed or negotiated without the consent of anyone. Data encryption and passwords are many PC attempts to improve security. Encryption is the definition of data into an invisible structure without a separation method. A watchword is a mysterious word or phrase that gives a client access to a specific project or structure.

1.4.3.5 Maintainability

It is defined as an opportunity within the given time to perform a successful repair work. Thus, usability checks precision and speed at which, after a disappointment, the system can be restored to working order. Convenience is a trademark of the PC application in the case where it can be used instead of the one in which it is built as part of applications without the need for major reconstruction. Porting is the function of performing any function required to keep a PC system running in a new location.

1.4.3.6 Ability of Learning

It is easy to work with and reduces learning activity

1.5 Hardware Requirements

Minimum Requirements	Windows
Operating System	Windows 7
Processor	Dual core, Intel i3
RAM	2 GB RAM
DISK Space	The amount of disc space available depends on the partition size and whether or not online help files are allowed. The Math Works installer would tell you how much disc volume your partition needs.
Graphics Adapter	8-bit graphics adapter and display (for 256 simultaneous colors)
CD ROM drive	For installation from CD

Recommended Requirements		Windows		
	Processor	RAM	DISK Space	Graphics Adapter
PYCHARM-Python IDE	Intel i3	2 GB	1 GB for Pycharm only. 5 GB for a typical installation	A 32-bit or 64-bit OpenGL capable graphics adapter is strongly recommended

1.6 Software Specifications

- **Tensorflow-** : TensorFlow is an open source planning data flow plan for all areas of operations. A number-related library is used for AI applications such as neural networks.

- **Google Collaboratory-** Google Colaboratory (open-supply Jupyter Notebook interface with excessive GPU space) - Google Colab / Colaboratory is an unencrypted Jupyter package that does not need to be set up and works perfectly in the cloud. [5] With Colab, you can really write and use code, save and analyze percentages, Access functional computing software, all at no cost in the browser. Jupyter Notebook is an effective way to duplicate and write in your Python code to analyze facts. Instead of rewriting and rewriting the entire code, you can actually write the code types and apply them at a time.

- **Seaborn-** Seaborn is a Python library for comprehension by viewing matplotlib. It provides an undeniable level of interaction in drawing attractive and useful measurable drawings. For a brief introduction to the ideas behind the library, you can look up the first notes or paper.

- **OS**: The OS module in Python enables enabling and completing enrollment (envelope), importing, modifying and separating the current catalog, and more.
- **Matplotlib:** Matplotlib is a Python programming language library and extension of NumPy. Provides article-based API for installing episodes in applications using the most useful GUI toolboxes such as Tkinter, wxPython, Qt, or GTK.
- **Pandas**: Pandas explain () is used to view certain basic mathematical details such as percentage, total, std etc. of a data framework or a series of numerical values. When this method is used in a series of character units, it returns the different output that is shown in the examples below. Return type: Summary of data frame statistics.
- **NumPy:** NumPy is a Python library used for running with collections. It additionally has the cappotential to paintings withinside the area of direct

polynomial calculations, 4 variables, and frames. NumPy become evolved in 2005 with the aid of using Travis Oliphant. It is an open supply enterprise and you could use it sparingly. NumPy stands for Numerical Python.

- **Kaggle** was used to find an online database that we used as a mit-bih database.
- **GitHub** and **Stackoverflow** were used for reference in case of syntax editing errors.

Language

- **Python 3** - We have used Python which is R-mathematical programming language instead of MATLAB for the following reasons:

1. Python code is much more understandable than MATLAB
2. Python record format improved in MATLAB
3. Keras (with backend of TensorFlow 2.3.zero version) - Keras is a neural community API that includes TensorFlow, CNTk, Theano etc. Python programs such as Numpy, Matplotlib, Pandas math and editing graphs,

Chapter 2: LITERATURE SURVEY

2.1 Existing Systems:

Paper 1: Automatic discovery of cardiac arrhythmia the usage of In- Depth Literacy Strategies.

It is a condition in which the blinkling is not usual. The cause of this paper is to use in- intensity analyzing techniques for the opinion of cardiac arrhythmia the usage of ECG alerts with as littlepre-processing information as possible. We use convolutional neural community (CNN), repetitious systems comparable as nonstop neural community (RNN), short- time period memory (LSTM) and reopened intermittent unit (GRU) and CNN mongrel and repetitious systems to descry abnormalities itself. Unlike traditional evaluation patterns, in- intensity literacy algorithms don`t have affair- grounded evaluation patterns. The accurate parameters for in- intensity literacy patterns are named via way of means of appearing a number of evaluation patterns. All check tracks are operated via way of means of one thousand a while in line with grade analyzing position (0.01-0.5). We locate afive-foldcross-sectional delicacy of0.834 in ordinary and unusual separation (cardiac arrhythmia) ECG and CNN-LSTM. In addition, the delicacy attained via way of means of different intertwined systems of in- intensity literacy algorithms in comparison to CNN-LSTM. Cardiac arrhythmia is mainly a abnormal twinkle. Some types of cardiac arrhythmia can cause headaches comparable as stroke, coronary heart assault and can certainly cause unexpected cardiac death. Thus, well timed discovery and opinion of arrhythmia are usually important. Once the arrhythmia has been detected, the approaching level of identity of the arrhythmia level may be performed. We created anon-invasive device grounded on deep literacy networks to carry out introductory bracket of ECG information surpassed as ordinary or unusual ECG (arrhythmia) ECG the usage of the in detail to be had MIT-BIH arrhythmia point. We in comparison overall performance the usage of a number of in- intensity analyzing codecs for CNN, CNN-RNN, CNN-LSTM and CNN-GRU and attained0.834 delicacy.

Plagiarized Concerned with the price of computations, we`re unfit to educate complex systems. Reported troubles may be constantly bettered via a complicated in- intensity literacy armature. Complex community systems may be educated the usage of superior address and following a considerably dispensed schooling device undeserving to try.

We've bandied a part of in- intensity literacy techniques comparable as CNN and repetitious systems in arrhythmia bracket paintings. The spotlight of the proposed device is that it does not endure noise filtering and engineering features. The consequences attained verify that the effectiveness of our device is higher than different posted consequences in efficiently classifying ECG as ordinary. Although deep literacy networks produce notable consequences, corruption remains now no longer sufficient to manual the complicated inner approaches of deep literacy system. This may be triumph over via way of means of reconsidering deep round networks into direct shape via way of means of gathering motorized eigenvalues and eigenvectors at exclusive time intervals (20). Unborn paintings may be the gathering of actual- global information from hospitals with cardiac care devices and using analogous patterns in actual information sets.

Paper 2: In-depth Reading with 2-D ECG Spectral Image Representation for Cardiac Arrhythmia Classification

The electrocardiogram (ECG) is one of the most widely used diagnostic tools for diagnosing and predicting cardiovascular disease (CVDs). ECG symptoms can trigger heart rate abnormalities, often called arrhythmias. Careful screening of ECG symptoms is essential for accurate diagnosis of heart conditions in acute and chronic patients. In this study, we propose a convolutional neural network (CNN) model with a two-dimensional (2-D) division of ECG signals into eight classes; that is, the normal bit, the bit of ventricular contraction, the movement of the rhythm, the bit of the right lobe block, the bit of the left lobe block, the rhythm of the atrial contraction, the bit of the ventricular flutter wave, and the bit of ventricular escape. Signals of the one-sided ECG timeline are converted to 2-D spectrograms by the Fourier transition time. The 2-D CNN model that combines four convolutional layers and four layers of compound design is designed to extract solid elements from input spectrograms. Our proposed method is tested on a publicly available MIT-BIH arrhythmia. We found a high degree of accuracy of 99.11%, which is better than the results recently reported in classifying arrhythmias. Performance is also important in other indicators, including sensitivity and clarity, which indicate the success of the proposed approach. In this study, we proposed a 2-D CNN differential model for the automatic detection of cardiac arrhythmias using ECG signals. An accurate taxonomy of ECG symptoms is very helpful in the prevention and diagnosis of CVDs. Deep CNN proved to be useful in improving the accuracy of diagnostic algorithms in the integration of medicines and modern machine learning technologies. The proposed CNN-based separation algorithm, using 2-D images, can distinguish eight types of arrhythmia, namely, NOR, VFW, PVC, VEB, RBB, LBB, PAB, and APC, and obtain sensitivity average 97.91%, 99.61% specificity, average accuracy 99.11%, and positive prediction value of 98.59% (accuracy).

These results suggest that predicting and classifying arrhythmia with 2-D ECG represented as spectrograms and the CNN model is a reliable method for diagnosing CVD. The proposed system can help professionals evaluate CVDs by referring to the automatic separation of ECG signals. The current study uses only one ECG signal. The impact of more ECG lead data on improving test conditions will be studied in future work.

Paper 3: Discovering Arrhythmia Through In-Depth Reading and More Representation

An electrocardiogram (ECG) is a sign of a time collection represented through one-sided information (1- D). The illustration of the most length includes extra data reachable to the function release. Hidden variables inclusive of fre- quency relation and phase morphology aren't without delay reachable on the time zone. In this paper, the information of the 1-D time collection is transformed to polygonal illustration withinside the shape of 2-D photographs of a couple of channels. Following that, in-intensity take a look at changed into used to teach the neural community primarily based totally classifier to hit upon arrhythmias. The simulation effects at the check web website online show the effectiveness of the proposed approach through demonstrating the extremely good effectiveness of the type in comparison to different to be had strategies and annotations made through licensed cardiologists.

We have educated an in-intensity gaining knowledge of version wherein its effectiveness is simplest in detecting some arrhythmias in ECG records. The primary concept that affects the overall performance of multilateral illustration and the in-intensity gaining knowledge of version of a couple of layers. It can retrieve 1-D information shape and match the version primarily based totally on education information. On the alternative hand, the invention of extra not unusualplace arrhythmias stays a prime challenge. For example, Torsades de pointes have a completely unique shape and a function which could effortlessly be studied in-intensity reading, however the confined quantity of information to be had makes it hard to teach. Additionally, Torsades have a morphology much like sound shape. It is viable that with out ok education information, in-intensity gaining knowledge of will omit out on seeing Torsades as sound and vice versa.

A spontaneous analysis can assist each medical doctor and heart specialist lessen the quantity of time spent on ECG evaluation. In addition, in an technology whilst touch and put on are affordable, it opens up an entire new international in which far off evaluation may be carried out in an area in which the heart specialist is inaccessible.

Paper 4: The opinion of Cardiac arrhythmia using an in- depth study system

An electrocardiogram (ECG) is an important individual tool for diagnosing cardiac arrhythmias in a clinical setting. In this study, an in- depth study frame that was preliminarily trained in the standard image data set was transmitted for the automatic opinion of ECG arrhythmia by dividing the case`s ECGs into coherent cardiac conditions. The deep convolutional neural transmission network (i.e. AlexNet) is used as a point affair and the uprooted features are incorporated into the neural network of simple neural propagation to make the final separation. Three different ECG waveform scripts are named from the MIT-BIH arrhythmia database to estimate the proposed frame. The main focus of this study is to use a simple, dependable and easy-to- read literacy system to distinguish three different heart conditions. The results attained showed that the in- depth literacy point uprooted by the normal neural propagation network was suitable to achieve veritably high situations of performance. The loftiest role of popularity attained is98.fifty one while testing delicacy is roughly 92. Grounded on these findings, in- depth refreshed literacy has proven to be an effective way to diagnose cardiac arrhythmia while removing the burden of training the deep convolutional neural network from the morning furnishing a more effective approach.

ECG data attained from the MIT-BIH database ispre-analyzed, QRS complexes are attained and features during R-T are uprooted using three different styles. After all these way, three different networks that use these different features as inputs are designed, trained, tested and estimated for pattern recognition and ECG signaling by three different cardiac conditions. Table 2 summarizes the test results attained by the three different networks.

When all tested networks were plant to be networks grounded on the in- depth literacy point (N-Fc6 and N-Fc7) they achieved roughly 100 situations of recognition and delicacy over 96 in the training phase. For this reason it isn`t plant necessary to change any parameters similar as peak times, error limits or learning situations. In comparison, thenon-core literacy network (N-tst) achieved a position of recognition of 92 and an delicacy of about 90 in the training phase.

At the testing stage, N-Fc7 attain a test discovery rate of98.51 and an average delicacy of about92.4 in its excellent performance. N-Fc6 figures were97.53 recognition and91.2 delicacy. N-tst achieved a recognition rate of91.58 but 85 delicacy.

As the results attained easily show, ECG arrhythmia discovery styles grounded on the release of in- depth reading point perform better than in- depth study with a positive endpoint. Also, the results attained by networks using features deduced from AlexNet`s full deep convolutional (6th and 7th) layers have proven that deep layers of the convolutional neural network trained in a large data set defined can be general enough to transmit and initiate ECG separation function.

2.2 Proposed Model

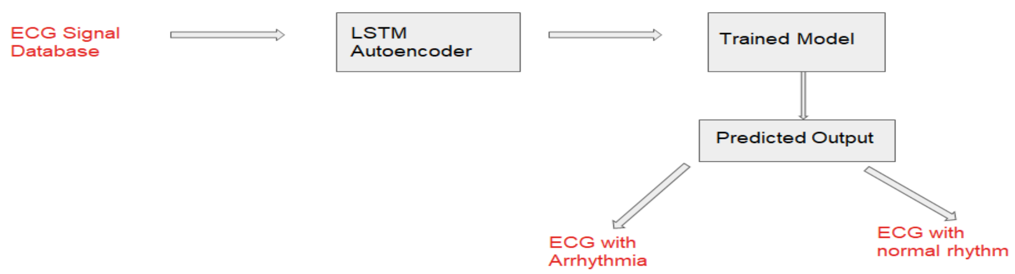


Fig. 2: Proposed Model

2.2.1 Technical Flow

- Preparing a Time Series Data Detection Anomaly data set
- Using PyTorch to create LSTM Autoencoder
- Teaching and modeling
- Choosing a confusing discovery limit

- Unseen examples should be described as normal or abnormal.

2.2.2 Dataset

This dataset carries a sequence of CSV documents. Each of those CSV documents carries a matrix, every row representing an instance in that part of the database. The remaining part of every line refers back to the paragraph that the instance is a part of.

It then converts all WFDB files to CSV files with the same name (example from 100.dat, 100.hea, 100.atr 100.csv file is generated).

Each row in a CSV file is accompanied by a single heartbeat and is as follows:

Divide the MIT-BIH record into R-tops into single heartbeat records.

Each heartbeat record is linked to the first 40 readings of the next heartbeat record to include the complete QRS Complex.

Repeat for each heartbeat record from 360Hz to 125Hz.

Make mV readings normal at 0-1 range.

Records of more than 187 heartbeats are discarded.

The record for heartbeat is zero and eventually includes exactly 187 values.

The heart rate from the annotations is reduced to Normal and Unusual and added at the end of each heartbeat record (0 is normal, 1 is uncommon). Each line then contains exactly 188 values.

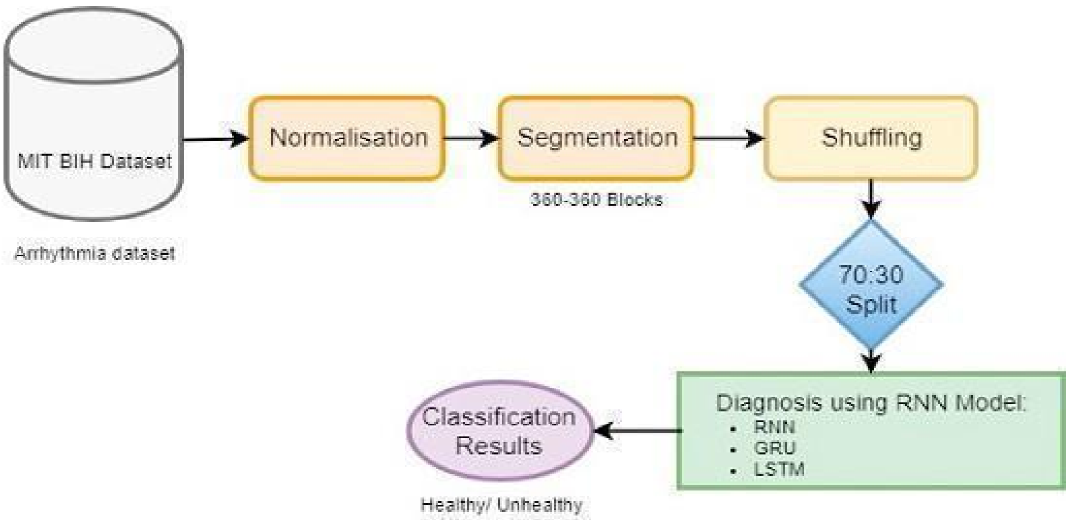


Fig. 3: Flowchart

2.2.3 Types of heartbeats

- Typical (N)
- Premature Ventricular Contraction (R-on-T) (R-on-T PVC)

- Ventricular Premature Contraction (PVC)
- Ectopic or Supra-ventricular Premature Beat (EB or SB)
- Unclassified Heart Beats

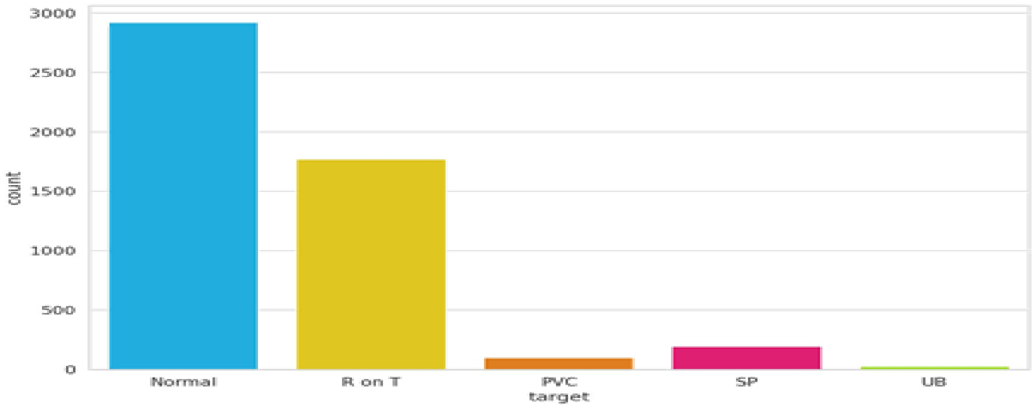


Fig. 4: Dataset Composition

2.2.4 Assumption

Each heartbeat, or heart rate, takes about 0.8 seconds to complete a healthy heart rate and an average of 70 to 75 beats per minute. 60-100 times per minute frequency (People) 0.6-1 second in length (People).

Records of heartbeat outside the category are discarded.

The purpose of these CSV files will be used to train your ECG model to classify the heartbeat as normal or abnormal

2.2.5 LSTM ENCODER

- The Autoencoder function is to input input data, run it by model, and re-input the input. It is important to use a modest number of parameters for the model to learn input input.
- Generally, the Autoencoder format is divided into two categories. The encoder presses the data input, and the recorder records the output data.
- To compress Time Series data input, Encoder uses two LSTM layers.
- Two layers of LSTM and one layer of extraction in the output generate the final reconstruction.
- Autoencoder transfers input to encoder and Decoder.

2.3 FEASIBILITY ANALYSIS

Feasibility study is the first survey of users' knowledge that may estimate service needs, costs, benefits, and feasibility. The feasibility study examines the implementation and operation of the system due to a number of issues. Resources needed for use, such as computer equipment, personnel, and costs, are estimated in this section. Estimated costs are compared to available resources, and system cost analysis is performed. The main objective of a feasibility study is to determine the feasibility of the project, the feasibility of the project, the feasibility of the project, and the feasibility of the planning. Its purpose is to ensure that all project input data is available. As a result, we checked

- Technology
- Finance
- Service and time

2.3.1 Technical Feasibilities:

Diagnosis and Classification of Heart Disease using Deep Learning is a complete project based on In-Depth Reading with minimal Machine Learning.

Technologies and associated tools:

CNN

Tensorflow

Google Collaboratory

Seaborn

2.3.2 Resource Feasibility:

- Programming device (Laptop)
- Programming tools (freely available)
- Tensorflow
- Google Collaboratory
- Dataset
- Programming individuals

2.3.3 Financial Feasibility:

It will cost a total of 0 rupees for a project based on Deep Reading (simultaneous).
Bug fixes and repair works will have lower costs associated with them.
Customers will reap many benefits in addition to the costs associated with it

2.3.4 Time Feasibility:

According to the Gantt chart plan, the project timeline is appropriate.
Because this is a ten-month plan, there will be many deadlines and deliveries that will be organized accordingly.
As a result, all project progress is well planned, no delays are anticipated, leading to rapid completion.
Diagnosis and Classification of Cardiac Arrhythmia Through In-Depth Study has a required period of time.

2.3.5 Risk Feasibility:

- Risks related to size
- Risk-related risks to the Business
- Customer related risks
- Risks related to the process
- Risks associated with technical problem
- Technological Risks

2.3.5.1 Risk associated with the sizes:

The average product size is between 10 and 100MB.
The average product size requires at least 1GB of RAM.
Size increases with the number of users.

2.3.5.2 Risk Feasibility with impact on business:

Sensibility of delivery deadlines:
Delivery deadlines are well thought out and logical and taken after considering all the possible factors involved.
Strengthening end users:
Low complexity, high usability
Real-time monitoring and forecasting

The value and quality of product documents to be created and delivered to my customer is as follows:

User instructions

Delivery costs:

Product costs

Cost of hosting

Maintenance costs

2.3.5.3 Risks associated with Customers:

Insufficient service provider

Insufficient privacy and protection of customer information.

2.3.5.4 Risks associated with Process

Technical errors or security risks

Process Quality

2.3.5.5 Risks associated with Technical issue

Are there any coded conventions that are being advertised and followed?

Code documents will be provided. The rules of coding are followed.

During the software development process?

2.3.5.6 Risks associated with Technology

Will the technology / tools be used fresh?

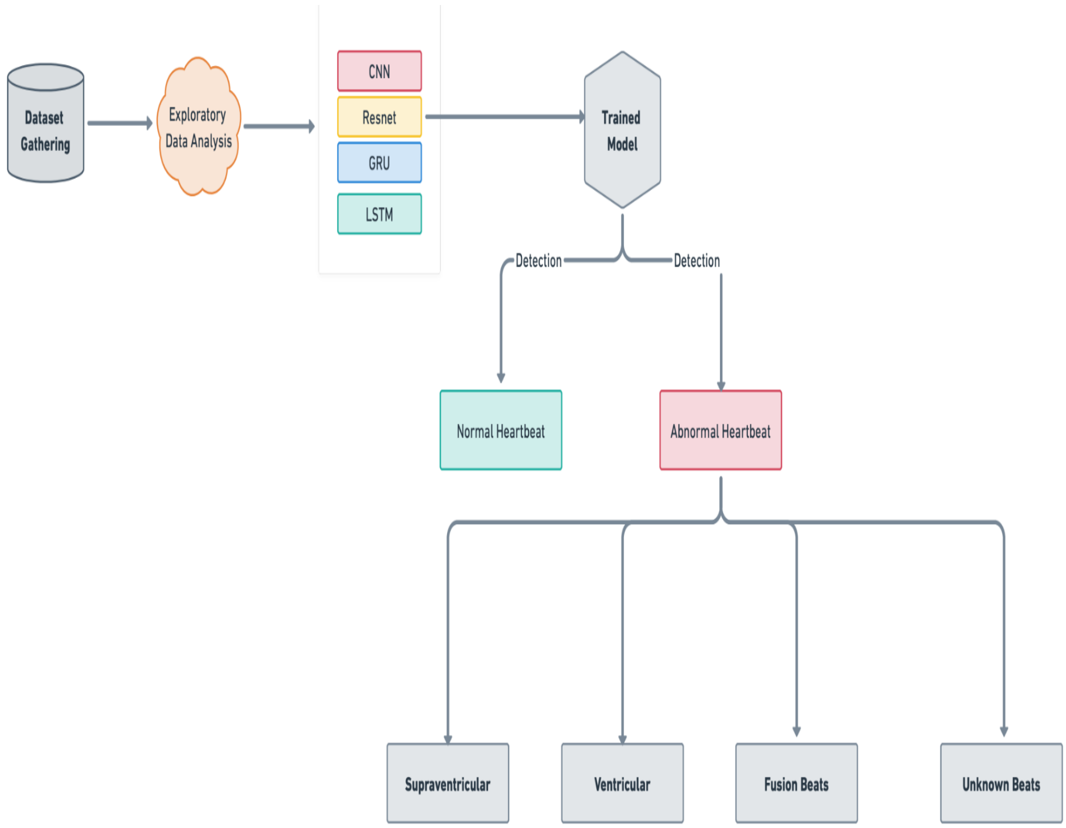
All the tools are well known and the best in their class.

Is it necessary to develop new algorithms, input or output technologies?

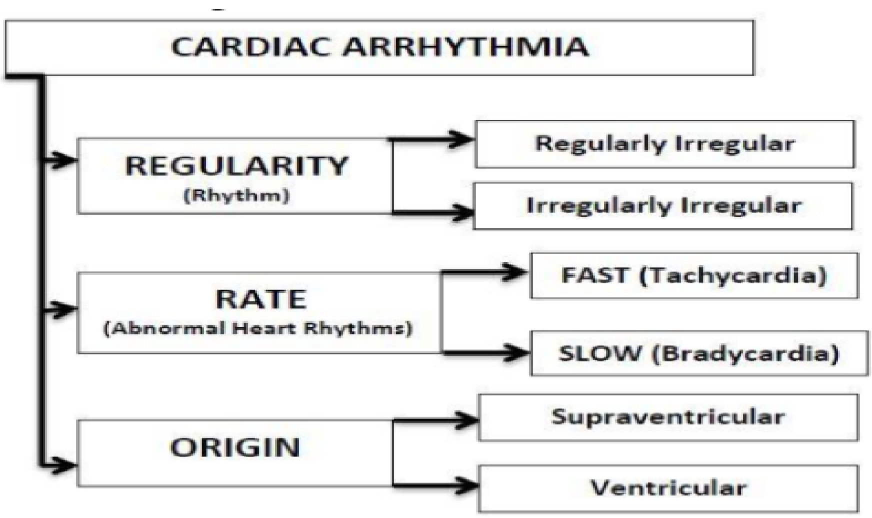
It has a few algorithms for running applications in microprocessor, capturing data, providing real-time results and diagnosing diseases

Chapter 3: SYSTEM DESIGN AND ANALYSIS

3.1 System Design



3.2 Classification of Cardiac Arrhythmia



3.3 Working of the algorithms

1) CNN Model-In deep learning, the convolutional neural network (CNN / ConvNet) is a class of deep-seated networks, widely used to analyze visual images.

Each neuron receives multiple inputs and takes a weighted load over you, which transmits it to the opening function and response output. CNNs are primarily used to classify images, group them together, and then create object recognition.

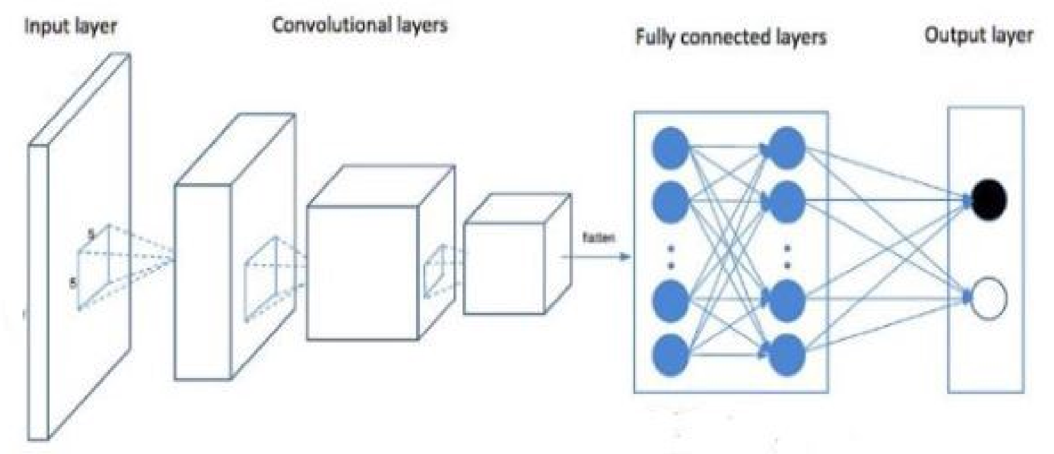


Fig. 5: Basic CNN architecture

Layer of CNN model:

1. Convolution 2D
2. MAX Poolig2D
3. Dropout
4. Flatten
5. Dense
6. Activation

A. **Convolution 2D:** In the Convolution 2D extract the featured from enter image. It given the output in matrix form.

B. **MAX Pooling2D**: In MAX Pooling 2D it captures very large details on a modified performance map.

C. **Cessation**: Cessation is randomly determined by neurons that are skipped during training.

D. **Flatten**: Complete the feed output into a fully integrated layer. Provides statistics in the listing form.

E. **Density**: A Linear function where each installation is attached to each exit in a weighted manner. It is compatible with an indirect activation feature.

F. **Getting Started:** Use the Sigmoid feature and expect a zero chance and 1. In the merger version we used binary entropy because we have zeros and layers 1. We used the Adam optimizer for compilation version. [17] Adam: -The second variable variable. It is used for non-convex abrasion as a clear cut for use.

2) GRU (Gradient Recurrent unit)- At every timestamp t, it takes an enter Xt and the hidden nation Ht-1 from the preceding timestamp t-1. Later it outputs a brand new hidden nation Ht which once more passes to the subsequent timestamp.

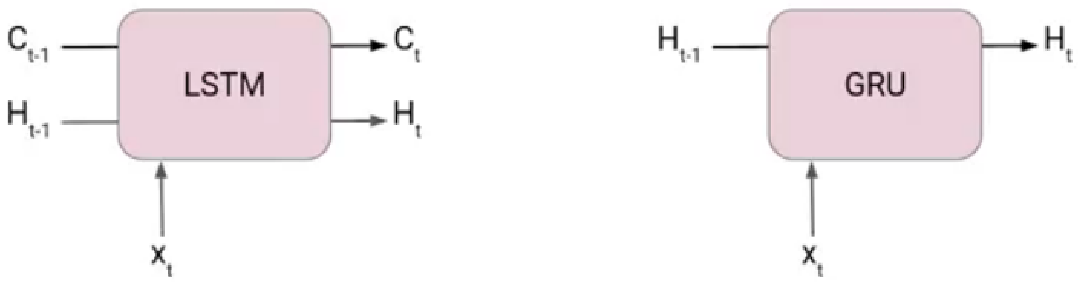


Fig. 6: Gated recurrent unit (GRU)

Unlike LSTM, it has only three gates and does not keep Internal Cell State. Information stored in the Internal Cell State in the continuous LSTM unit is encrypted in the encrypted Gated Recurrent Unit. This collected information is transferred to the next Gated Recurrent Unit. The various GRU gates are described below: -

Refresh Gate (z): Determines how much of the past information needs to be passed on in the future. It is similar to the Output gate in a continuous LSTM unit.

Reset Gate (r): Decides how much previous information you should forget. It is similar to the combination of Input Gate and Forget Door in a continuous LSTM unit.

Current Memory Gate (): Often overlooked during normal conversation with the Gated Recurrent Unit Network. Installed in the Reset Gate as just the Input Exchange Gate is an integral part of the Input Gate and is used to introduce something that is not linear input and input means Zero. Another reason to make it a small part of the Reset gateway is to reduce the impact that previous information has on current information that is passed on in the future.

The basic functionality of the Gated Recurrent Unit Network is similar to that of the Basic Recurrent Neural Network when shown, the main difference between the two is in the internal functioning within each normal unit as the Gated Recurrent Unit networks include model gates. current input and previous hidden status.

3) Normal Neural Network

We have used Recurrent Neural Networks to further distinguish and analyze arrhythmic bits in this paper. Percentage accuracy is used to measure RNN performance — primarily based entirely on the heart rate. Careful assessment of the drawings completed within the challenge is also carried out, taking into account the key factors. When the need arose for a series of drawings, which included handwriting and speech recognition, a sensitive type of emotional networks emerged. Recurrent Neural Networks is a type of synthetic neural community that can create and differentiate random sequences of inputs using their internal memory, and communication between gadgets shapes the targeted cycle.

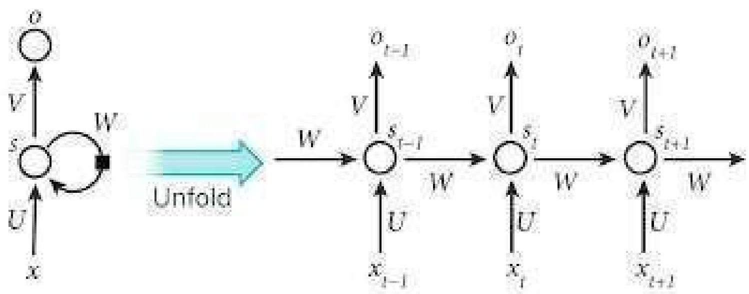


Figure 7: Recurrent Neural Network (Source: ICCIDS)

Gaining in-depth knowledge is a completely interesting topic, and is employed in the ways of other educators to improve and improve performance and overall accuracy standards. RNN and CNN are the top attractive Deep Learning domain names, and each is used to differentiate ECG arrhythmias. However, when CNN is used to split the ECG, it breaks the bits into parts of a fixed period, which reduces the performance of the phase as a whole. The overall functionality of the RNN may be enhanced by providing custom divider functions, making it superior in a number of ways. We use RNN to analyze key and positive robotic residues in a way that feeds modern rhythm and final rhythm, i.e.T.

Because of their unusually flexible function, repetitive sensory networks emerge, while multi-layer feed feeds have a vertical map. RNNs were employed in a number of fields and included packages for integrated memory, optimization, and general performance. RNNs are the first measure to separate time collection statistics because current day comments and costs are re-entered into the network, and the output includes the types of values stored within the memory, which enhances the performance of the whole class and gives superior results. there are standard feed forwarding networks.

4) Long Term Memory Network

The Long-Term Memory (LSTM) shape is a shape of neural recurrent network (RNN).LSTMs were created to produce a sequential version of the sequence, and the dependence of the longer RNN versions and a copy of the memory played an important role, making them more efficient and powerful than conventional RNNs. The method is used after the calculations have been processed in advance to remove any unwanted, missing, or invalid values.

Three layers of RNN – LSTM were used in this study, with 64, 256, and 100 neurons throughout the layer, respectively, with 5 repetitions. After each layer, 0.2

Withdrawal of funds. The loss factor became the MSE, at the same time as the implementation became sigmoid.

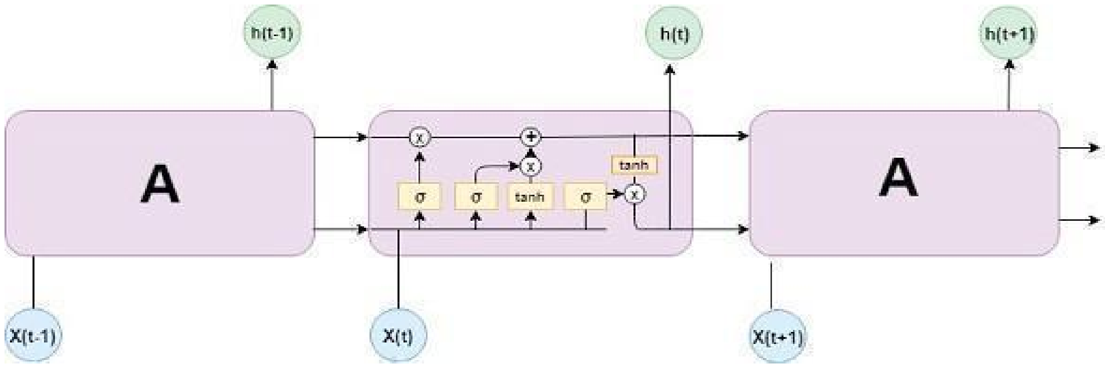


Figure 8: Long Short Term Memory (Source: ICCIDS)

5) Autoencoder

The Autoencoder`s assignment is to obtain a few enter facts, run it via the version, and recreate the enter. The reconstruction ought to be as near the authentic as possible. The key's to restrict the quantity of parameters to your version in order that it is able to examine a compressed illustration of the facts.

Autoencoders, in a sense, attempt to examine most effective the maximum tremendous factors of the facts (compressed version). We`ll study the way to feed Time Series facts to an Autoencoder on this section. To seize the temporal dependencies of the facts, we're going to utilise more than one LSTM layers (hence the LSTM Autoencoder). We have selected a threshold in which a heartbeat is appeared strange to discover a series as regular or strange. The intention of Autoencoder schooling is to reconstruct the enter as as it should be as feasible. This is carried out via the usage of a loss characteristic this is minimised (much like in supervised learning). Reconstruction loss is the call given to this characteristic. Examples encompass cross-entropy loss and suggest squared error.

There are components to the Autoencoder structure in general. The enter is compressed through an encoder, and the output is decoded through a decoder. To compress the Time Series facts enter, the Encoder employs LSTM layers.

Two LSTM layers and an output layer offer the very last reconstruction in our Decoder.

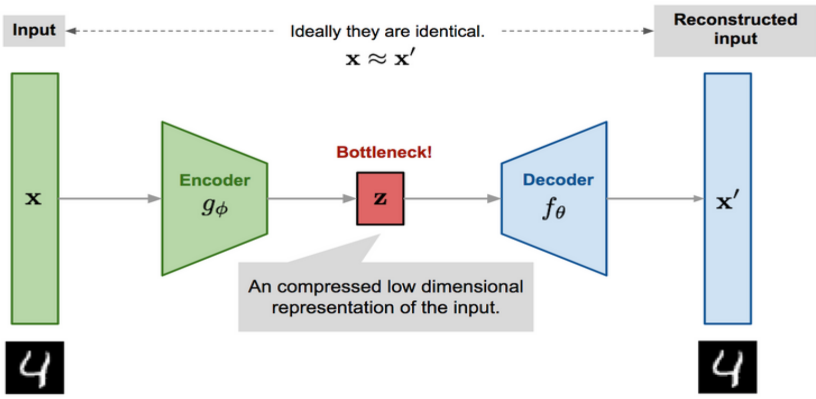


Figure 9: Autoencoder (Source: Curiosity.com)

Chapter 4: IMPLEMENTATION

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.metrics import f1_score
from sklearn.metrics import confusion_matrix
from keras.utils.np_utils import to_categorical
from sklearn.utils import class_weight
import warnings
warnings.filterwarnings('ignore')
```

```python
import tensorflow as tf
tf.test.gpu_device_name()
```

```
'/device:GPU:0'
```

```python
train_df=pd.read_csv('/content/drive/MyDrive/data/mitbih_train.csv',header=None)
test_df=pd.read_csv('/content/drive/MyDrive/data/mitbih_test.csv',header=None)
```

```python
train_df[187]=train_df[187].astype(int)
equilibre=train_df[187].value_counts()
```

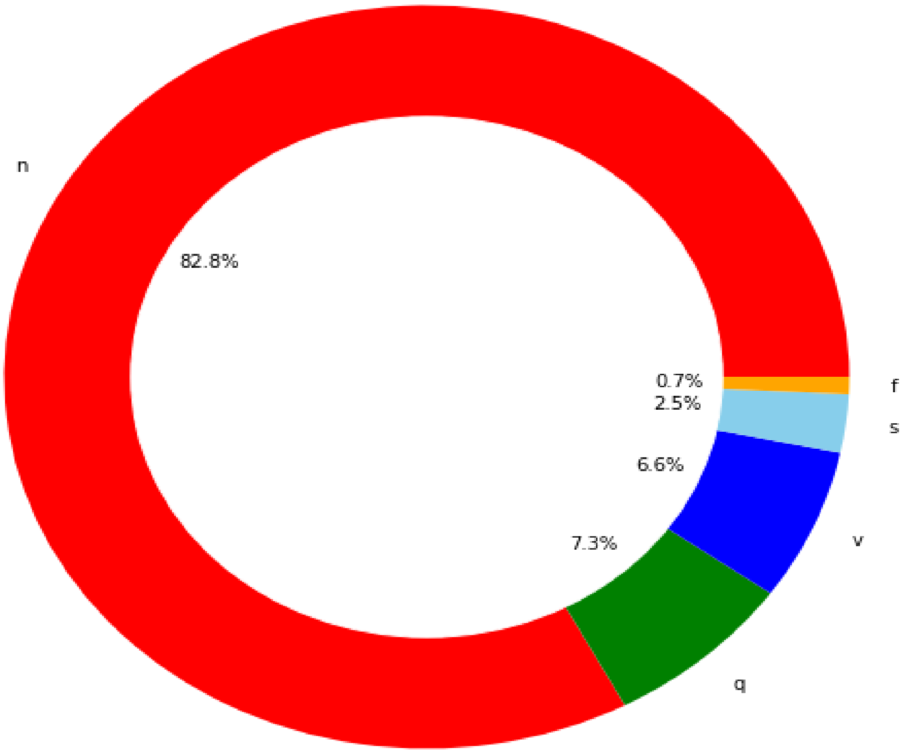

```
'/device:GPU:0'
```

```python
train_df=pd.read_csv('/content/drive/MyDrive/data/mitbih_train.csv',header=None)
test_df=pd.read_csv('/content/drive/MyDrive/data/mitbih_test.csv',header=None)
```


```python
train_df[187]=train_df[187].astype(int)
equilibre=train_df[187].value_counts()
print(equilibre)
```

```
0    72471
4     6431
2     5788
1     2223
3      641
Name: 187, dtype: int64
```

```python
plt.figure(figsize=(20,10))
my_circle=plt.Circle( (0,0), 0.7, color='white')
plt.pie(equilibre, labels=['n','q','v','s','f'], colors=['red','green','blue','skyblue','orange'],autopct='%1.1f%%')
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



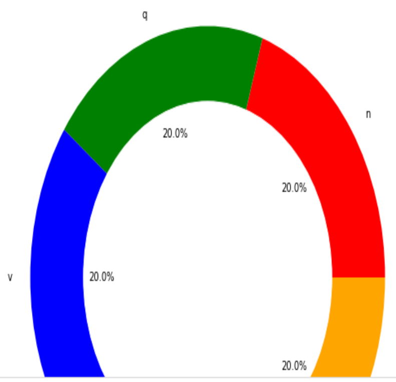

```
from sklearn.utils import resample
df_1=train_df[train_df[187]==1]
df_2=train_df[train_df[187]==2]
df_3=train_df[train_df[187]==3]
df_4=train_df[train_df[187]==4]
df_0=(train_df[train_df[187]==0]).sample(n=20000,random_state=42)

df_1_upsample=resample(df_1,replace=True,n_samples=20000,random_state=123)
df_2_upsample=resample(df_2,replace=True,n_samples=20000,random_state=124)
df_3_upsample=resample(df_3,replace=True,n_samples=20000,random_state=125)
df_4_upsample=resample(df_4,replace=True,n_samples=20000,random_state=126)

train_df=pd.concat([df_0,df_1_upsample,df_2_upsample,df_3_upsample,df_4_upsample])
```

```
equilibre=train_df[187].value_counts()
print(equilibre)
```

```
0    20000
1    20000
2    20000
3    20000
4    20000
Name: 187, dtype: int64
```

+ Code + Text Copy to Drive Connect Editing

```
plt.figure(figsize=(20,10))
my_circle=plt.Circle( (0,0), 0.7, color='white')
plt.pie(equilibre, labels=['n','q','v','s','f'], colors=['red','green','blue','skyblue','orange'],autopct='%1.1f%%')
p=plt.gcf()
p.gca().add_artist(my_circle)
plt.show()
```



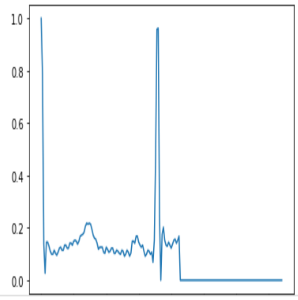
+ Code + Text Copy to Drive Connect Editing

45055	1.000000	0.786260	0.137405	0.026718	0.145038	0.145038	0.129771	0.110687	0.099237	0.099237	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
73341	0.921147	0.724014	0.207885	0.064516	0.093190	0.164875	0.150538	0.168459	0.236559	0.222222	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
78106	0.000000	0.072115	0.171474	0.283654	0.371795	0.426282	0.479167	0.536859	0.588141	0.661859	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
80731	1.000000	0.840502	0.277778	0.091398	0.060932	0.032258	0.016129	0.000000	0.008961	0.003584	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3
85323	0.586103	0.492447	0.453172	0.432024	0.432024	0.404834	0.386707	0.323263	0.283988	0.202417	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4

5 rows × 188 columns

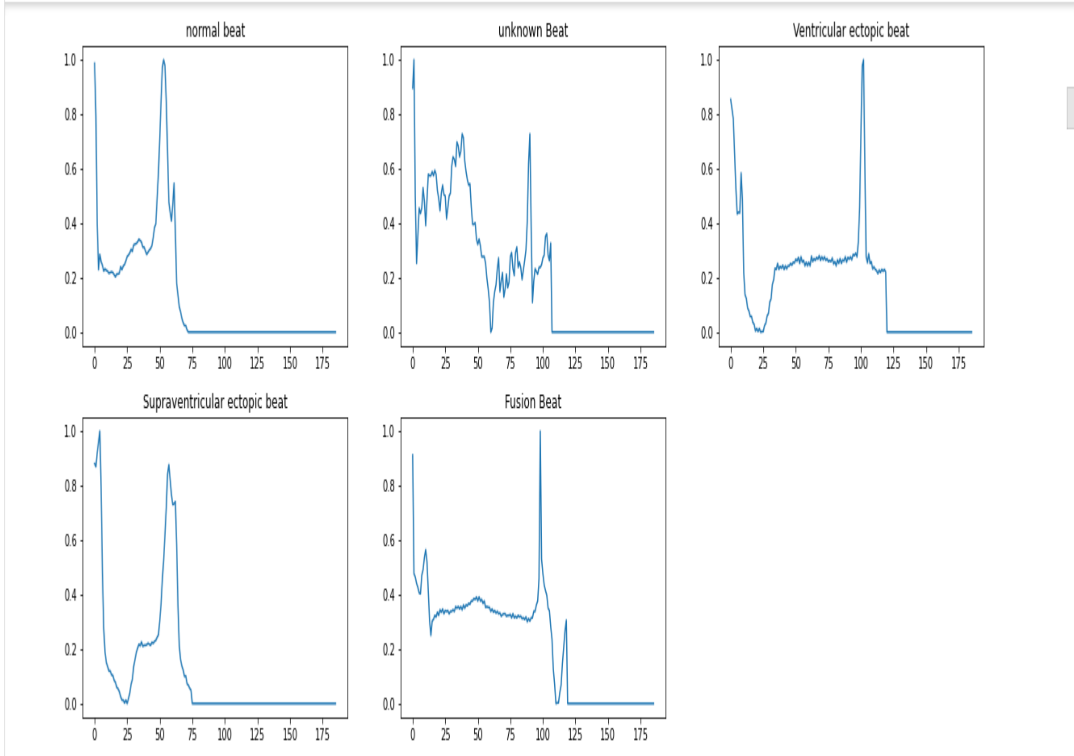
```
plt.plot(c.iloc[0,:186])
```

[<matplotlib.lines.Line2D at 0x7fdad63d1910>]



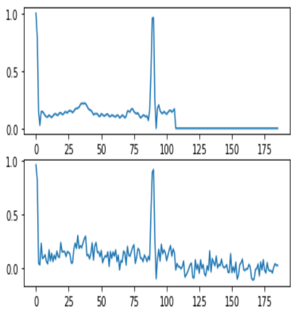
```
# randomly sampling from each class
classes=train_df.groupby(187,group_keys=False).apply(lambda train_df : train_df.sample(1))
```

```
# plotting classes ECG
plt.figure(figsize=(20,8))
# normal
plt.subplot(2, 3, 1)
plt.plot(classes.iloc[0,:186])
plt.title('normal beat')
# unknown
plt.subplot(2, 3, 2)
plt.plot(classes.iloc[1,:186])
plt.title('unknown Beat')
# veb
plt.subplot(2, 3, 3)
plt.plot(classes.iloc[2,:186])
plt.title('Ventricular ectopic beat')
plt.show()
plt.figure(figsize=(20,8))
plt.subplot(2, 3, 4)
plt.plot(classes.iloc[3,:186])
plt.title('Supraventricular ectopic beat')
# fusion
plt.subplot(2, 3, 5)
plt.plot(classes.iloc[4,:186])
```

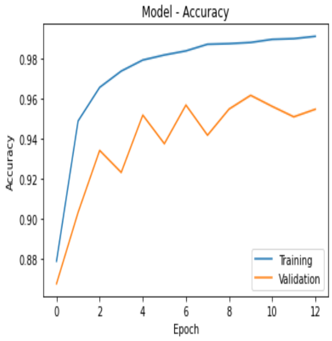


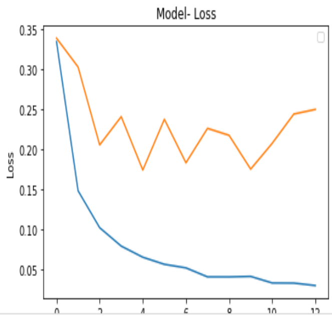
```
def add_gaussian_noise(signal):
    noise=np.random.normal(0,0.05,186)
    return (signal+noise)
```

```
tempo=c.iloc[0,:186]
bruiter=add_gaussian_noise(tempo)

plt.subplot(2,1,1)
plt.plot(c.iloc[0,:186])

plt.subplot(2,1,2)
plt.plot(bruiter)

plt.show()
```

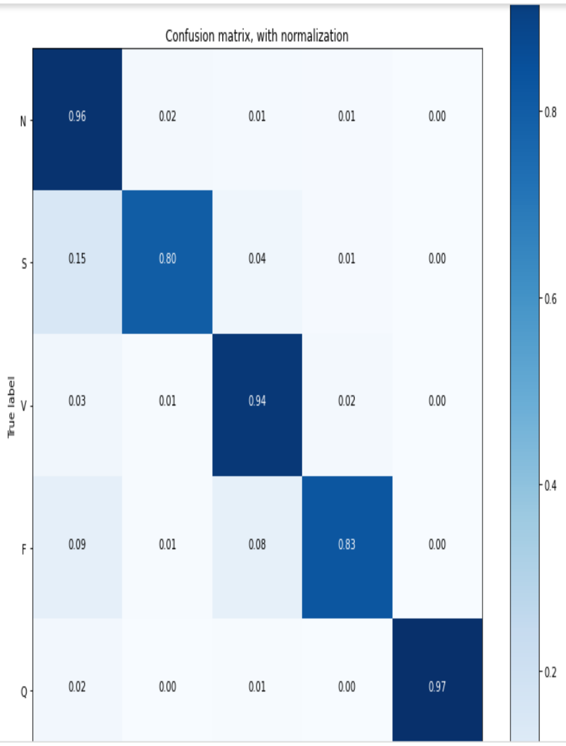


```
Accuracy: 95.18%
<keras.callbacks.History object at 0x7fdad3484e50>
```





```python
import itertools
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
```



bold text RESNET MODEL

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('whitegrid')
plt.style.use('seaborn-deep')
plt.style.use('fivethirtyeight')
plt.rcParams['font.size'] = 14
plt.rcParams['axes.labelsize'] = 12
plt.rcParams['axes.titlesize'] = 12
plt.rcParams['xtick.labelsize'] = 12
plt.rcParams['ytick.labelsize'] = 12
plt.rcParams['legend.fontsize'] = 12
plt.rcParams['figure.titlesize'] = 14
plt.rcParams['figure.figsize'] = (20,8)
plt.rcParams["ps.useafm"] = True
import random
```

\+ Code + Text Changes will not be saved Connect

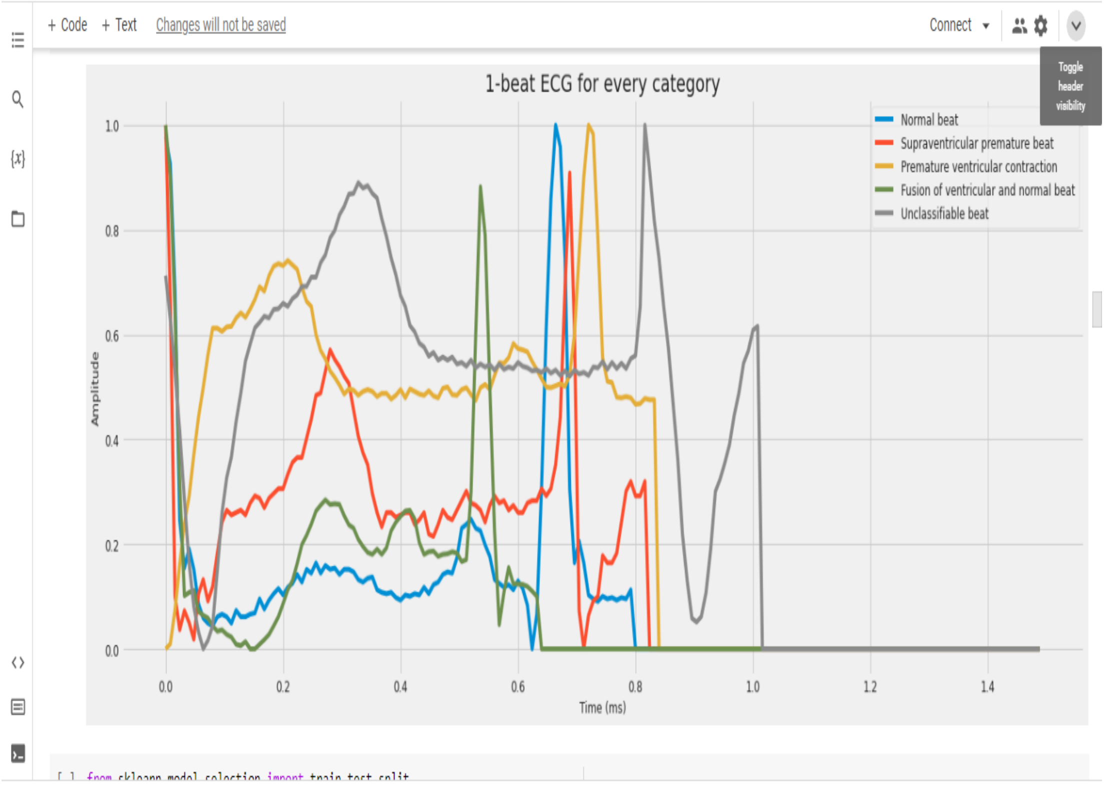
```
df[187] = df[187].astype('int64')
target_col = df[187]
labels = ['Normal beat','Supraventricular premature beat','Premature ventricular contraction','Fusion of ventricular and normal beat','Unclassifiable beat
```

```
df[187] = df[187].astype('int64')
target_col = df[187]
labels = ['Normal beat','Supraventricular premature beat','Premature ventricular contraction','Fusion of ventricular and normal beat','Unclassifiable beat']
```

```
X = df.drop(187,axis=1)
y = target_col
print(X.shape)
print(y.shape)

x_test = df_test.drop(187,axis=1)
y_test = df_test[187].astype('int64')
print(x_test.shape)
print(y_test.shape)
```

```
(87554, 187)
(87554,)
(21892, 187)
(21892,)
```

```
C0 = (target_col == 0)
C1 = (target_col == 1)
C2 = (target_col == 2)
C3 = (target_col == 3)
C4 = (target_col == 4)
```



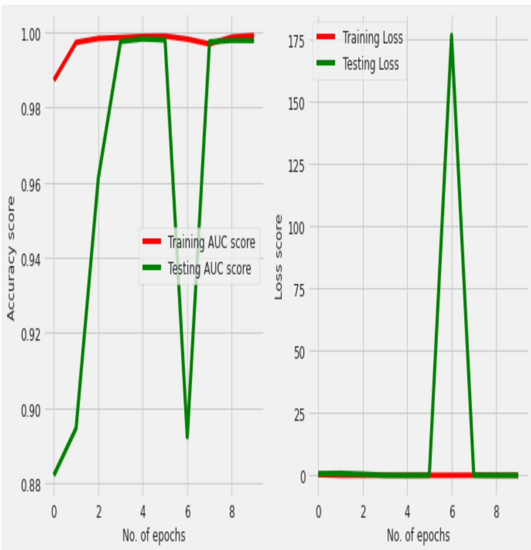

```
def ResNet(input_shape=(feature,depth)):

    X_input = Input(input_shape)

    X = ZeroPadding1D(3)(X_input)

    X = Conv1D(64, 7, strides=2, name='conv1', kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=2, name='bn_conv1')(X)
    X = Activation('relu')(X)
    X = MaxPooling1D(3, strides= 2)(X)

    X = convolutional_block(X, f=3, filters=[128, 128, 256], stage=2, block='a', s=1)
    X = identity_block(X, 3, [128, 128, 256], stage=2, block='b')
    X = identity_block(X, 3, [128, 128, 256], stage=2, block='c')


    X = convolutional_block(X, f=3, filters=[128, 128, 512], stage=3, block='a', s=2)
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='c')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='d')

    X = convolutional_block(X, f=3, filters=[256, 256, 1024], stage=4, block='a', s=2)
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='b')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='c')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='d')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='e')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='f')

    X = X = convolutional_block(X, f=3, filters=[512, 512, 2048], stage=5, block='a', s=2)
```

+ Code + Text Changes will not be saved Connect

```
[ ] model.save("./model.h5")
```

```
## Model auc score and loss visualization
auc = h.history['auc']
val_auc=h.history['val_auc']
loss=h.history['loss']
val_loss=h.history['val_loss']

epochs=range(len(auc)) #No. of epochs

plt.figure(figsize=(10,6))
ax0 = plt.subplot(1,2,1)
ax1 = plt.subplot(1,2,2)
ax0.plot(epochs,auc,'r',label='Training AUC score')
ax0.plot(epochs,val_auc,'g',label='Testing AUC score')
ax0.legend()
ax0.set_xlabel('No. of epochs')
ax0.set_ylabel('Accuracy score')


ax1.plot(epochs,loss,'r',label='Training Loss')
ax1.plot(epochs,val_loss,'g',label='Testing Loss')
ax1.set_xlabel('No. of epochs')
ax1.set_ylabel('Loss score')
ax1.legend()
plt.show()
```

+ Code + Text Changes will not be saved Connect

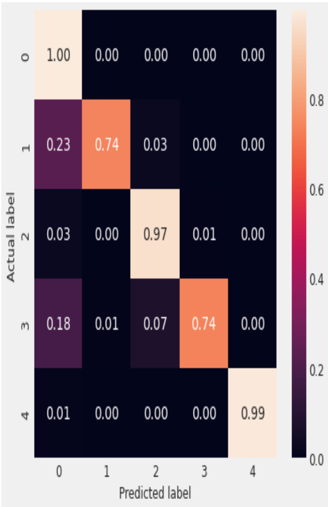


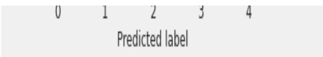

```
[ ] model.evaluate(x_val,y_val)

685/685 [==============================] - 23s 33ms/step - loss: 0.0609 - auc: 0.9979
[0.06094751134514809, 0.9978828430175781]
```

```
[ ] model.evaluate(x_test,y_test)
```

+ Code + Text Changes will not be saved Connect

```python
fig = plt.figure(figsize=(6, 6))
ax= plt.subplot()
cm = confusion_matrix(y_true=np.argmax(y_val,axis=1),y_pred=np.argmax(y_pred_val,axis=1),normalize="true")
sns.heatmap(cm, annot=True, ax = ax, fmt = '.2f',)
ax.set_xlabel('Predicted label')
ax.set_ylabel('Actual label')
plt.show()
```






```python
from sklearn.metrics import precision_score
print("The precision score : ",precision_score(y_true=y_val,y_pred=y_pred_val,average='weighted'))
```

```
The precision score :  0.9842206659506656
```

```python
for i in range(len(labels)):
    print("The f1-score for class "+str(i)+" ("+str(labels[i])+") : ",f1_score_per_class_validation[i])
```

```
The f1-score for class 0 (Normal beat) :  0.9918305597579425
The f1-score for class 1 (Supraventricular premature beat) :  0.8336713995943206
The f1-score for class 2 (Premature ventricular contraction) :  0.9555707450444292
The f1-score for class 3 (Fusion of ventricular and normal beat) :  0.8082191780821918
The f1-score for class 4 (Unclassifiable beat) :  0.9897483690587139
```

```python
for i in range(len(labels)):
    print("The f1-score for class "+str(i)+" ("+str(labels[i])+") : ",f1_score_per_class_test[i])
```

```
The f1-score for class 0 (Normal beat) :  0.9912613355317395
The f1-score for class 1 (Supraventricular premature beat) :  0.8052738336713996
The f1-score for class 2 (Premature ventricular contraction) :  0.9493055555555556
The f1-score for class 3 (Fusion of ventricular and normal beat) :  0.7845659163987139
The f1-score for class 4 (Unclassifiable beat) :  0.9872552067143301
```

LSTM

```
[ ] # importing libraries
    import numpy as np # linear algebra
    import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
    import os, tqdm, re, time, itertools, sys
    import seaborn as sns
    from sklearn.model_selection import train_test_split
    from sklearn.svm import SVC
    from sklearn.ensemble import RandomForestClassifier, ExtraTreesClassifier
    from sklearn.metrics import accuracy_score, classification_report, confusion_matrix, plot_confusion_matrix
    from keras.layers import Conv2D, Conv1D, MaxPooling2D, MaxPooling1D, Flatten, BatchNormalization, Dense
    from keras.layers import LSTM, GRU, Dropout
    from keras.utils.np_utils import to_categorical
    from tensorflow.keras.utils import plot_model
    from keras.models import Sequential
    from keras.callbacks import CSVLogger, ModelCheckpoint
    import matplotlib.pyplot as plt
```

```
[ ] import warnings
    warnings.filterwarnings('ignore')
```

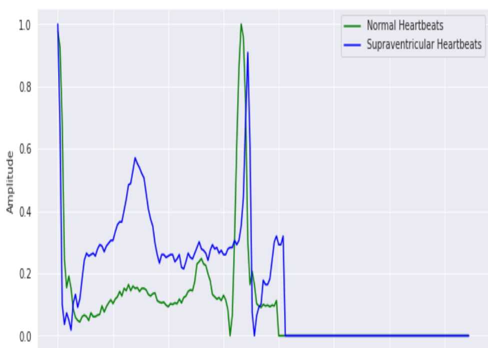
```
[ ] data_train=pd.read_csv('/content/drive/MyDrive/data/mitbih_train.csv',header=None)
    data_test=pd.read_csv('/content/drive/MyDrive/data/mitbih_test.csv',header=None)
```

```
[ ] # making the class labels for our dataset
```

+ Code + Text Cannot save changes Connect

```
[ ] # making the class labels for our dataset
    data_1 = data_train[data_train[187] == 1]
    data_2 = data_train[data_train[187] == 2]
    data_3 = data_train[data_train[187] == 3]
    data_4 = data_train[data_train[187] == 4]

    sns.set_style('darkgrid')
    plt.figure(figsize=(10, 5))
    plt.plot(data_train.iloc[0, 0:187], color='green', label='Normal Heartbeats')
    plt.plot(data_1.iloc[0, 0:187], color='blue', label='Supraventricular Heartbeats')
    plt.xlabel('Time')
    plt.ylabel('Amplitude')
    plt.legend()
```

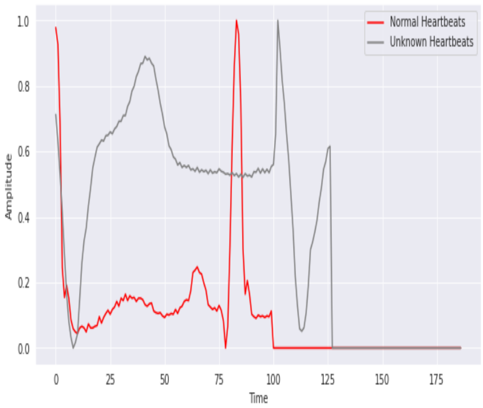
<matplotlib.legend.Legend at 0x7f10b6bd72d0>



+ Code + Text Cannot save changes Connect

```
[ ] sns.set_style('darkgrid')
    plt.figure(figsize=(10, 5))
    plt.plot(data_train.iloc[0, 0:187], color='red', label='Normal Heartbeats')
    plt.plot(data_4.iloc[0, 0:187], color='grey', label='Unknown Heartbeats')
    plt.xlabel('Time')
    plt.ylabel('Amplitude')
    plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f10b66de2d0>
```

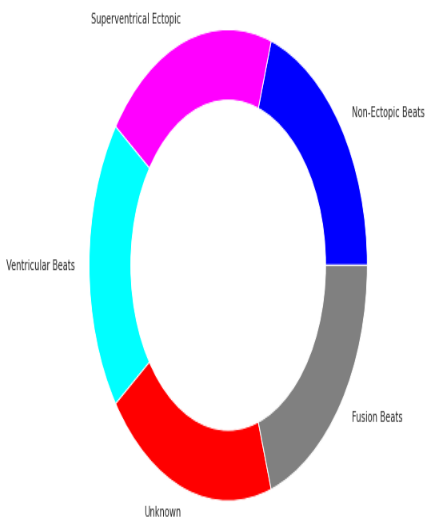



```
[ ] from sklearn.utils import resample
    data_1_resample = resample(data_1, n_samples=20000
```

+ Code + Text Cannot save changes Connect

```
    'Unknown', 'Fusion Beats'], colors=['blue', 'magenta', 'cyan',
                                        'red', 'grey'])
    p = plt.gcf()
    p.gca().add_artist(circle)
    # viewing the distribution of beats in our dataset
```

```
<matplotlib.patches.Circle at 0x7f10b6676b90>
```



+ Code + Text Cannot save changes Connect

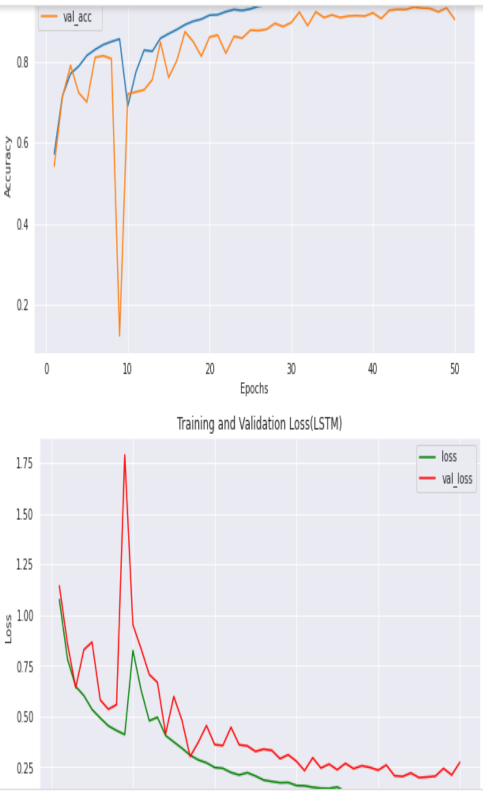
```python
def lstm_model():
    model = Sequential()
    model.add(LSTM(64))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(5, activation='softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])
    return model
```

```python
# apply function
model = lstm_model()
```

```python
# setup logger object
logger = CSVLogger('logs.csv', append=True)

# fit lstm on training data and validate on validation data
history_lstm = model.fit(X_train, y_train, epochs=50, batch_size=32,
          validation_data=(X_test, y_test), callbacks=[logger])
```

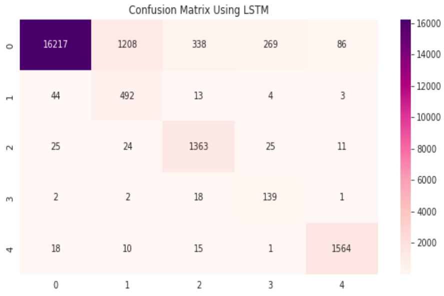
```
Epoch 1/50
3125/3125 [==============================] - 169s 51ms/step - loss: 1.0768 - accuracy: 0.5724 - val_loss: 1.1441 - val_accuracy: 0.5421
Epoch 2/50
3125/3125 [==============================] - 154s 49ms/step - loss: 0.7825 - accuracy: 0.7143 - val_loss: 0.8576 - val_accuracy: 0.7119
Epoch 3/50
3125/3125 [==============================] - 154s 49ms/step - loss: 0.6480 - accuracy: 0.7697 - val_loss: 0.6418 - val_accuracy: 0.7901
Epoch 4/50
```

+ Code + Text Cannot save changes Connect



```python
# plot confusion matrix
plt.figure(figsize=(10, 5))
plt.title('Confusion Matrix Using LSTM')
sns.heatmap(confusion_matrix(np.argmax(y_test, axis = 1), y_hat_lstm),
            annot=True,
            fmt='0.0f',
            cmap='RdPu')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f11a66becd0>
```



GRU

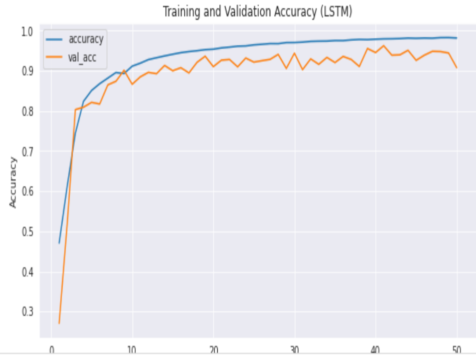
```python
def gru_model():
    model = Sequential()
    model.add(GRU(64))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.3))
    model.add(Dense(5, activation='softmax'))

    model.compile(loss='categorical_crossentropy',
                  optimizer='adam',
                  metrics=['accuracy'])
    return model
```

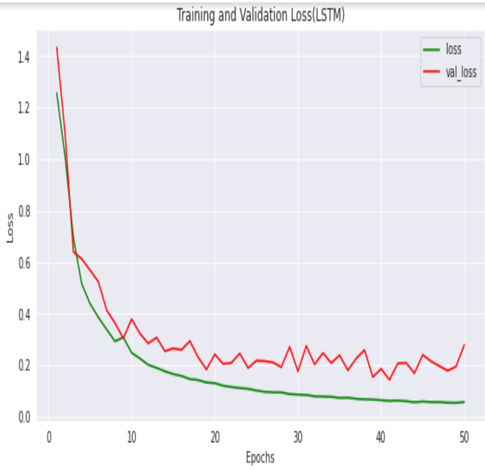
```python
# apply function
model = gru_model()
```

```python
# setup logger object
logger = CSVLogger('logs.csv', append=True)

# fit lstm on training data and validate on validation data
```

```
# loss/epoch plot
plt.figure(figsize=(10, 5))
plt.title('Training and Validation Loss(LSTM)')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.plot(epochs, loss, label='loss', color='g')
plt.plot(epochs, val_loss, label='val_loss', color='r')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7f103dbb7c50>
```






```
[ ] # make predictions and confusion matrix
    y_pred_gru = model.predict(X_test)
    y_hat_gru = np.argmax(y_pred_gru, axis = 1)
    confusion_matrix(np.argmax(y_test, axis = 1), y_hat_gru)
```

```
array([[16318,   908,   307,   443,   142],
       [   52,   474,    18,     6,     6],
```

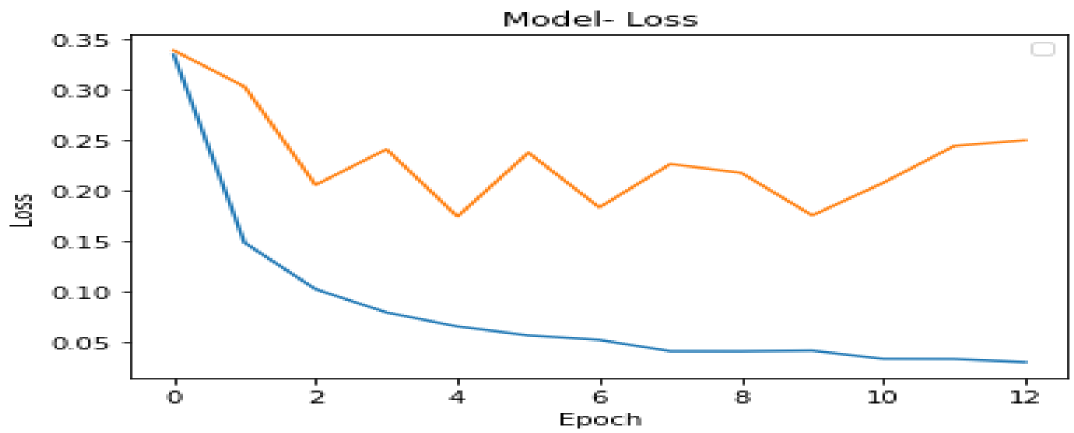
Chapter 5: RESULT

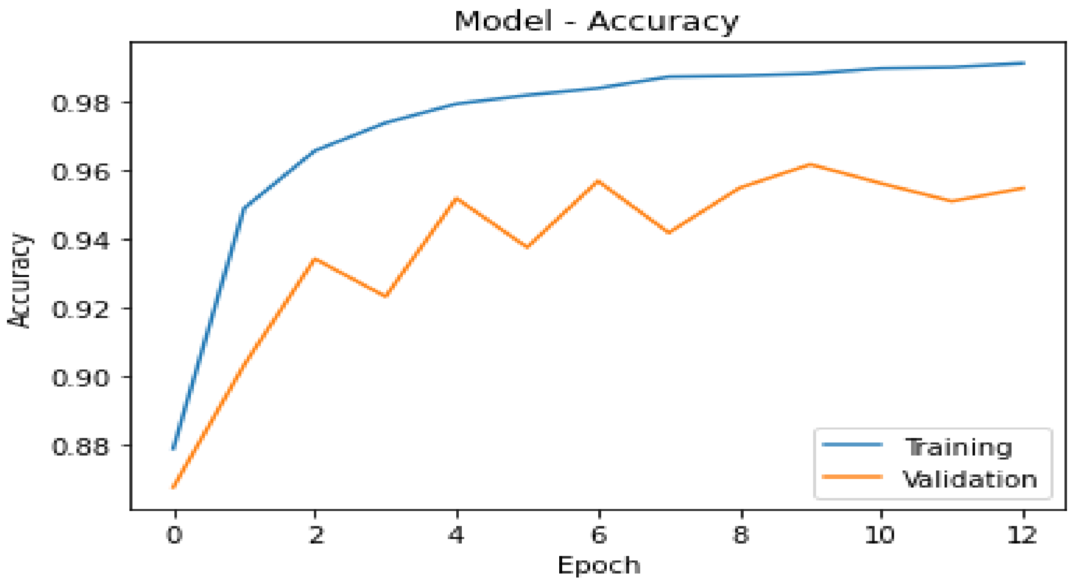
The classification methods employed, the quality of the ECG signal, the derived features used to define the beat, and the test data used in classification algorithm training all have an impact on performance of heartbeat classification method. The MIT-BIH arrhythmia database was leveraged to get the ECG signals used in the testing. To validate the proposed ECG beat classification system's classification performance, 109446 samples of the five most prevalent ECG beat types labeled in the dataset were used. These datasets' training and testing sections have previously been denoised, and the training and testing portions are in the form of conventional ECG heartbeats.

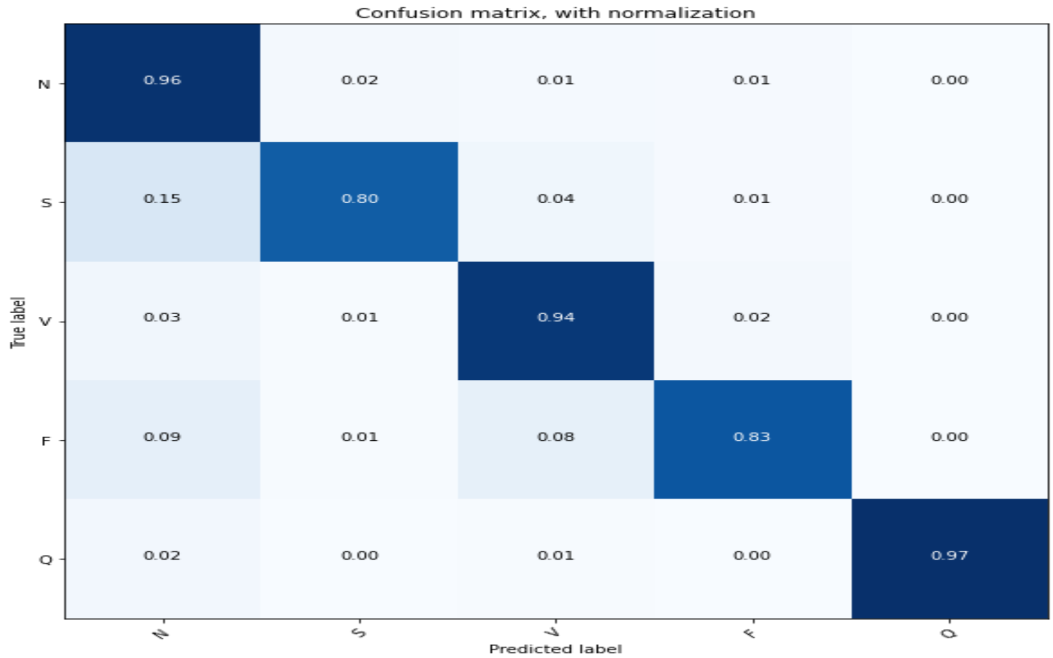
Model	**Accuracy**
Neural Network	95.18
Resnet Model	89.42
GRU	94.77
LSTM	90

Table 1: Accuracy

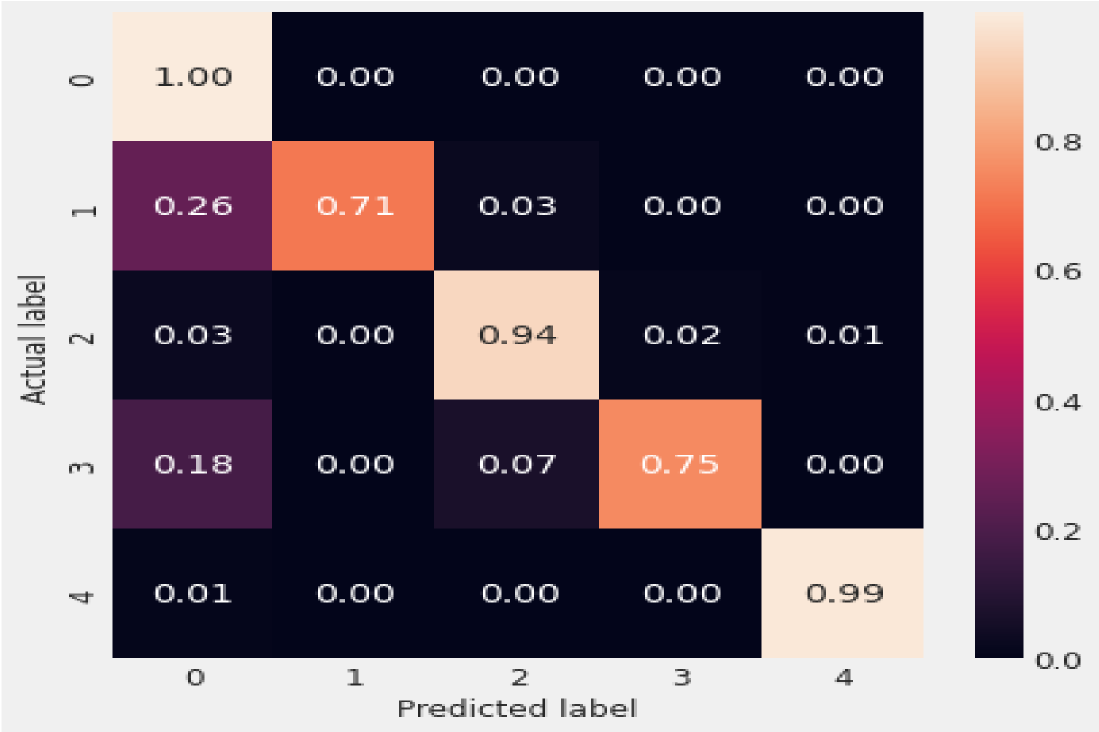
Neural Network

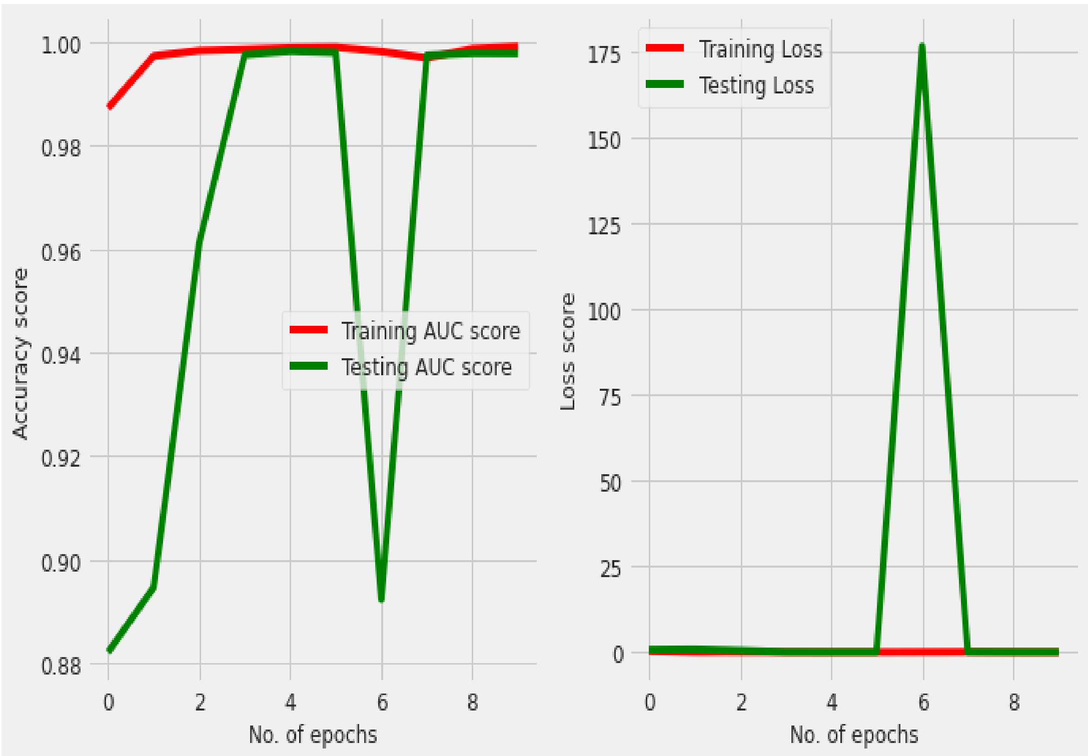



Model - Accuracy
Accuracy
0.98
0.96
0.94
0.92
0.90
0.88
0
2
4
6
8
10
12
Epoch
Training
Validation

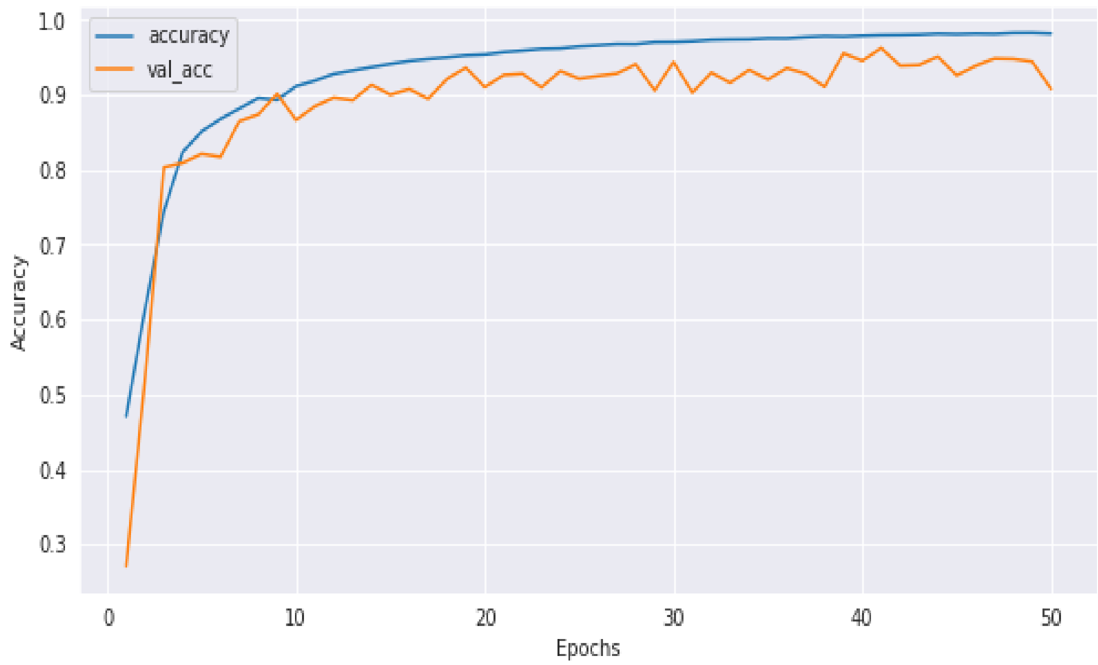

Confusion matrix, with normalization
True label
Predicted label
N
S
V
F
Q
0.96 0.02 0.01 0.01 0.00
0.15 0.80 0.04 0.01 0.00
0.03 0.01 0.94 0.02 0.00
0.09 0.01 0.08 0.83 0.00
0.02 0.00 0.01 0.00 0.97

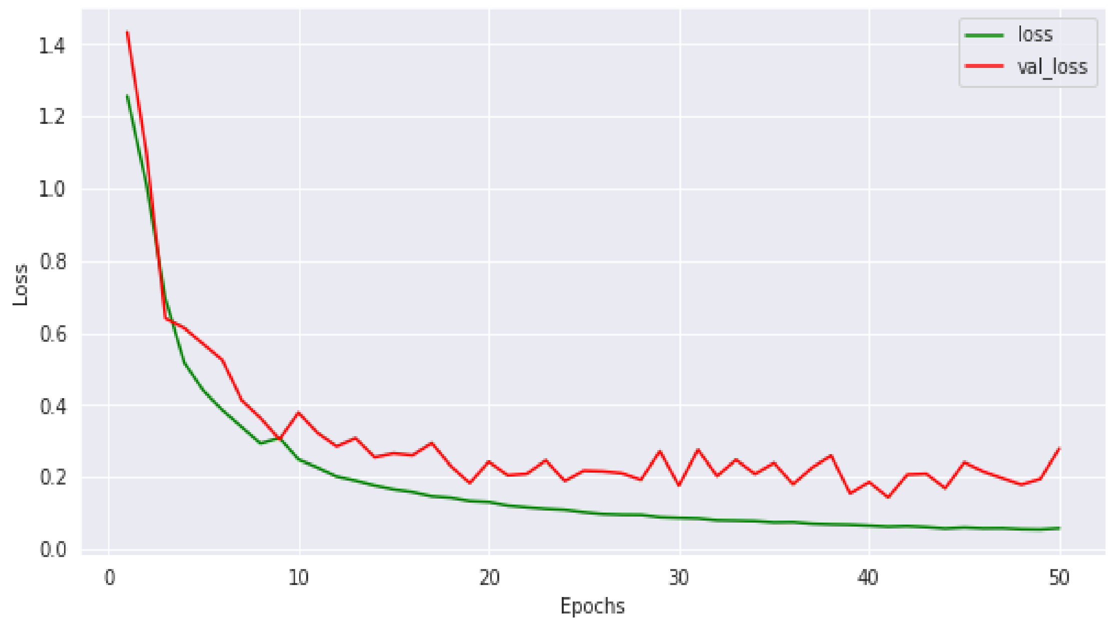
Resnet

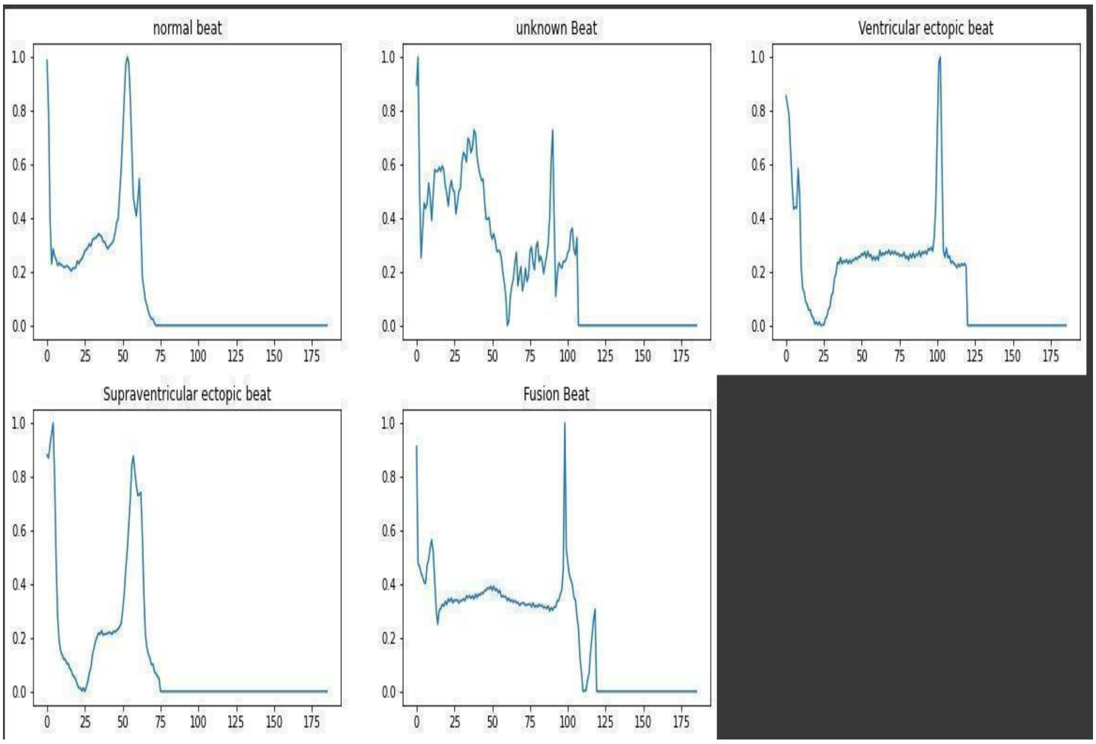

Actual label
Predicted label
0
1
2
3
4
1.00 0.00 0.00 0.00 0.00
0.26 0.71 0.03 0.00 0.00
0.03 0.00 0.94 0.02 0.01
0.18 0.00 0.07 0.75 0.00
0.01 0.00 0.00 0.00 0.99
0.0
0.2
0.4
0.6
0.8

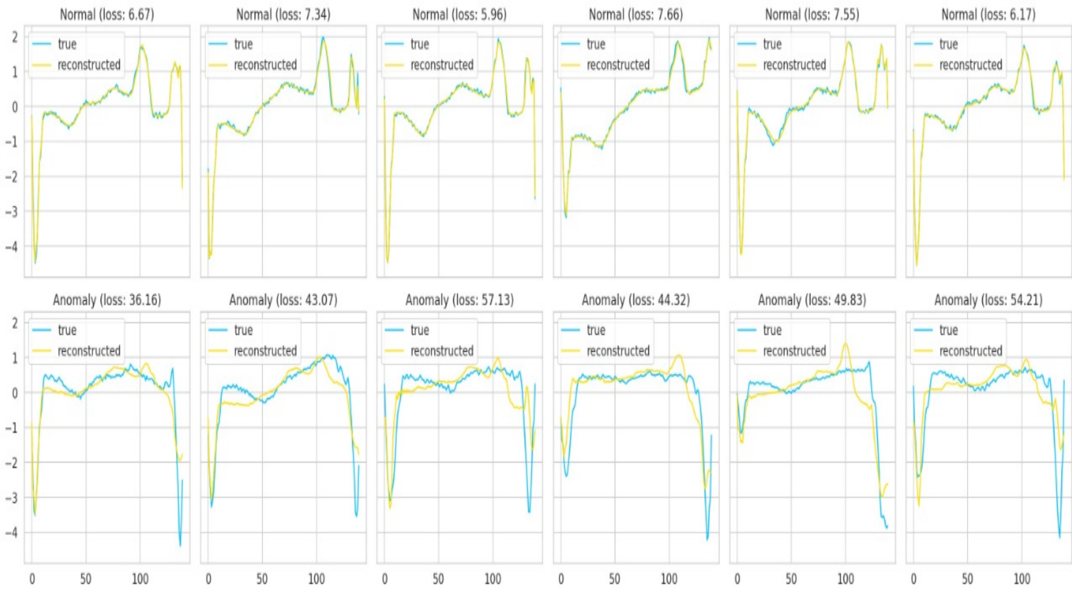

Accuracy score
No. of epochs
Training AUC score
Testing AUC score
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0
2
4
6
8
Loss score
No. of epochs
Training Loss
Testing Loss
175
150
125
100
75
50
25
0

GRU


accuracy
val_acc
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
Accuracy
0
10
20
30
40
50
Epochs


loss
val_loss
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Loss
0
10
20
30
40
50
Epochs


normal beat
unknown Beat
Ventricular ectopic beat
Supraventricular ectopic beat
Fusion Beat


Normal (loss: 6.67)
Normal (loss: 7.34)
Normal (loss: 5.96)
Normal (loss: 7.66)
Normal (loss: 7.55)
Normal (loss: 6.17)
Anomaly (loss: 36.16)
Anomaly (loss: 43.07)
Anomaly (loss: 57.13)
Anomaly (loss: 44.32)
Anomaly (loss: 49.83)
Anomaly (loss: 54.21)
true
reconstructed

Chapter 6: CONCLUSION

We have educated a deep mastering version wherein the overall performance of it's far advanced for detecting numerous arrhythmias from ECG records. The foremost concept that contributes to the overall performance is the multi-dimensional illustration and multilayer deep mastering version. It can get better the shape of 1-D records and match the version primarily based totally at the education records. On every other aspect, the detection of very uncommon prevalence arrhythmias stays a totally difficult problem. For example, the small wide variety of to be had records makes it tough to train. Additionally, It is probably that with out sufficient education records, the deep mastering will leave out discover arrhythmias as noise and vice versa. Automated prognosis can assist clinician and heart specialist to lessen the quantity of time spent on studying ECG. Furthermore, withinside the generation in which connectivity and wearable is affordable, it opens a brand new opportunity in which far flung evaluation may be achieved in an area in which heart specialist isn't always accessible.

Chapter 7: REFERENCES

[1] V.M. Uspenskiy. Information function of the heart. a measurement model. MEASUREMENT 2011, 1:383-386, 2011.

[2] V.M. Uspenskiy. Diagnostic system based on the information analysis of electrocardiograph. In Embedded Computing (MECO), 2012 Mediterranean Conference on, pages 74-76, June 2012.

[3] Wagner Galen and Strauss David. Marriotts Practical Electrocardiography. LWW, 2013.

[4] K. Perloff Joseph and J. Marelli Ariane. Clinical Recognition of Congenital Heart Disease. Elsevier, 2012.

[5] Babak Mohammadzadeh Asl, Seyed Kamaledin Setarehdan, and Maryam Mohebbi. Support vector machine-based arrhyth- mia classification using reduced features of heart rate variability signal. Artificial Intelligence in Medicine, 44(1):51- 64, 2008.

[6] Mi Hye Song, Jeon Lee, Sung Pil Cho, Kyoung Joung Lee, and Sun Kook Yoo.Support vector machine based arrhythmia classification using reduced features. International Journal of Control, Automation, and Systems, 3:571-579, 2005.

[7] A.J. Joshi, S. Chandran, V.K. Jayaraman, and B.D. Kulkarni. Hybrid svm for multiclass arrhythmia classification. In Bioinformatics and Biomedicine, 2009. BIBM 09. IEEE International Conference on, pages 287-290, Nov 2009.

[8] V.Ahanathapillai and J.J. Soraghan. Myocardial ischemia detection algorithm(mida): Automated echocardiography se- quence analysis for diagnosis of heart muscle damage. In Computing in Cardiology, 2010, pages 405-408, Sept 2010.

[9] T. Konttila, M. Stenroos, H. Vaananen, H. Hanninen, M. Lindholm, I. Tierala, and T. Katila. Support vector classification of acute myocardial ischemia with reduced sets of body surface potential map electrodes. In Computers in Cardiology, 2005,pages 869-872, Sept 2005.

[10] P Ranjith, P.C Baby, and P Joseph. ECG analysis using wavelet transform: application to myocardial ischemia detection. ITBM-RBM, 24(1):44 - 47, 2003.

[11] Ismail Babaoglu, Mustafa Servet Kiran, Erkan Ulker, and Mesut Gunduz.

Diagnosis of coronary artery disease using artificial bee colony and k-nearest neighbor algorithms. International Journal of Computer and Communication Engineering, 2:56-59, 2013.

[12]R. Alizadehsani, M.J. Hosseini, Z.A. Sani, A. Ghandeharioun, and R. Boghrati. Diagnosis of coronary artery disease using cost-sensitive algorithms. In Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on, pages 9-16, Dec 2012.

[13]A. Kampouraki, G. Manis, and C. Nikou. Heartbeat time series classification with support vector machines. Information Technology in Biomedicine, IEEE Transactions on, 13(4):512-518, July 2009.

[14]Adam Gacek and Witold Pedrycz. ECG Signal Processing, Classification and Interpretation. Springer-Verlag London, 2012.

[15]L. Yash, L. Pallavi, and A. Sameer. Ecg in non cardiac disorders.

[16]Mieghem Carlos, Sabbe Marc, and Knockaert Daniel. The clinical value of the ecg in noncardiac conditions. Chest, 125:1561- 1576, 2004.

[17]John S. Child, Joseph K. Perloff, Philip M. Bach, Allan D. Wolfe, Susan Perlman, and R.A. Pieter Kark. Cardiac involvement in friedreichâĂŹs ataxia: A clinical study of 75 patients. Journal of the American College of Cardiology, 7(6):1370-1378, 1986.

[18]Mikael Henaff, Kevin Jarrett, Koray Kavukcuoglu, and Yann Lecun. Unsupervised learning of sparse features for scalable audio classification.

[19]Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. Convolutional deep belief networks for scalable unsu- pervised learning of hierarchical representations.

[20] Hoo-Chang Shin, M.R. Orton, D.J. Collins, S.J. Doran, and M.O. Leach. Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1930-1943, Aug 2013.

[21]S. Osowski, L. T. Hoai, and T. Markiewicz, Support vector machine-based expert system for reliable heartbeat recognition, IEEE Trans.BiomedEng., 51(4), 2004, 582-589.

[22] P. de Chazal, M. O'Dwyer, and R. B. Reilly, Automatic classification of heartbeats using ECG morphology and heartbeat interval features, IEEE Trans.BiomedEng., 51(7), 2004, 1196-1206.

[23] E. J. da S.Luz, T. M. Nunes, V. H. C. de Albuquerque, J. P. Papa, and D. Menotti, ECG arrhythmia classification based on optimum-path forest, Expert Systems with Applications, 40(9),2013,3561-3573.

[24] O. T. Inan, L. Giovangrandi, and G. T. A. Kovacs, Robust neural-network-based classification of premature ventricular contractions using wavelet transform and timing interval features, IEEE Trans.BiomedEng., 53(12), 2006, 2507-2515.

[25] i. GUIer and E. D. Obeyli, ECG beat classifier designed by combined neural network model, Pattern Recognition, 38(2),2005, 199-208.

[26] Predicting the severity of Breast Cancer masses with ensemble of Bayesian Classifiers Journal of Computer Science 2010.

[27] R. D. Raut and S. V. Dudul, Arrhythmias classification with MLP neural network and statistical analysis, Proc. - 1st Int. Conf. Emerg. Trends Eng. Technol. ICETET 2008, pp. 553-558, 2008.

[28] B. M. Asl, S. K. Setarehdan, and M. Mohebbi, Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal, Artif. Intell. Med., vol. 44, no. 1, pp. 5164, 2008.

[29] G. Selvakumar, K. B. Bagan, and B. Chidambararajan, Wavelet Decomposition for Detection and Classification of Critical ECG Arrhythmias, pp. 8084, 2007.

[30] L. Khadra, A. Fraiwan, and W. Shahab, Neural-wavelet analysis of cardiac arrhythmias.

[31]P. Melin, J. Amezcua, F. Valdez, and O. Castillo, A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias, Inf. Sci. (Ny)., vol. 279, pp. 483-497, 2014.

[32] S. Yu and Y. Chen, Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network, vol. 28, pp. 1142-1150, 2007.

[33] M. Engin, ECG beat classification using neuro-fuzzy network, vol. 25, pp. 1715-1722, 2004.

[34] R. Silipo, G. Bortolant, and C. Marchesi, Supervised and unsupervised learning for ciiagnostic E, Evaluation, pp. 931-934, 1996.

[35] T. K. Toshifumi Sugiural, Hisashi Hiratal, Yukio Harada2, AUTOMATIC DISCRIMINATION OF ARRHYTHMIA, vol. 20, no. I, pp. 108-111, 1998.

[36] N. Kohli, N. K. Verma, and A. Roy, SVM based Methods for Arrhythmia

Classification in ECG, pp. 486-490, 2010.

[37] S. M. Jadhav, S. L. Nalbalwar, and A. A. Ghatol, Artificial Neural Network Based Cardiac Arrhythmia Disease Diagnosis, 2011 Int. Conf. Process Autom. Control Comput., pp. 16, 2011.

[38] Shraddha Singh, Saroj Kumar Pandey, Urja Pawar, Rekh Ram Janghel. "Classification of ECG Arrhythmia using Recurrent Neural Networks" , Procedia Computer Science, 2018

[39] Harmandeep Singh Gill, Osamah Ibrahim Khalaf, Youseef Alotaibi, Saleh Alghamdi, Fawaz Alassery. "Multi-Model CNN-RNN-LSTM Based Fruit Recognition and Classification" , Intelligent Automation & Soft Computing, 2022

[40] Joao Pereira, Margarida Silveira. "Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection" , 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), 2019

Chapter 8: APPENDICES

S. No.	Abbreviation	Meaning
1	CNN	Convolutional Neural Network
2	RNN	Recurrent neural network
3	CRNN	Convolutional Recurrent Neural Network
4	GRU	Gated Recurrent Unit
5	AE	Auto Encoder
6	LSTM	Long-short-term Memory

Table 2: Abbreviation