DESIGN AND DEVELOPMENT OF AN INTELLIGENT SYSTEM FOR MEDICAL DIAGNOSIS BASED ON MULTI-DIMENSIONAL ANALYSIS

Thesis
Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
by
SHRIVATHSA T V
(Registration No: 177084ME018)

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA
SURATHKAL, MANGALORE-575 025
OCTOBER 2023

DECLARATION

By the Ph.D. Research Scholar

I hereby declare that the Research Thesis entitled **"DESIGN AND DEVELOPMENT OF AN INTELLIGENT SYSTEM FOR MEDICAL DIAGNOSIS BASED ON MULTI-DIMENSIONAL ANALYSIS"** which is being submitted to the **National Institute of Technology Karnataka, Surathkal** in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** in **Mechanical Engineering** *is a bonafide report of the research work carried out by me.* The material contained in this Research Thesis has not been submitted to any other University or Institute for the award of any degree.

Register Number : **177084ME018**

Name of the Research Scholar : **Shrivathsa TV**

Signature of the Research Scholar:

Department of Mechanical Engineering

Place: NITK-Surathkal
Date: 04 October 2023

CERTIFICATE

This is to certify that the Research Thesis entitled **"DESIGN AND DEVELOPMENT OF AN INTELLIGENT SYSTEM FOR MEDICAL DIAGNOSIS BASED ON MULTI-DIMENSIONAL ANALYSIS"** submitted by **Mr. SHRIVATHSA TV (Reg. No. 177084ME018)** as the record of the research work carried out by him, is accepted as the Research Thesis submission in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy**.

Prof. Shrikantha S. Rao
Research Guide
Date: 04/10/2023

Dr. Navin Karanth P
Research Guide
Date: 04/10/2023

Chairman-DRPC
Date: 10·10·2023

राष्ट्रीय प्रौद्योगिकी संस्थान, कर्नाटक
यांत्रिकी अभियांत्रिकी विभाग
Dept. of Mechanical Engineering
National Institute of Technology Karnataka

ACKNOWLEDGMENT

Undertaking this PhD has been a truly life-changing experience for me and it would not have been possible to do without the support and guidance that I received from many people.

The author, **Mr. Shrivathsa TV** wishes to express his deep sense of gratitude to his guide and supervisor, **Prof. Shrikantha S Rao**, Professor, and **Dr. Navin Karanth P**, Associate Professor, Department of Mechanical Engineering, National Institute of Technology Karnataka (N.I.T.K), Surathkal for their invaluable guidance, motivation, untiring efforts and meticulous attention at all stages of this research work. Their constant encouragement, help, and review of the entire work during the course of the investigation are invaluable. Without their guidance and constant feedback this PhD would not have been achievable.

I am immensely grateful to **Prof. Ravikiran Kadolli**, Professor and Head, Department of Mechanical Engineering for extending the Departmental facilities, which ensured the satisfactory progress of my research work.

I would like to thank my co-researchers and friends for their kind help, encouragement for successful completion of this research work.

For this dissertation, I would like to thank my RPAC committee members: **Prof. Ramesh Kini M and Prof. S M Murigendrappa** for their time, interest, and helpful comments. I feel infinitely blessed for all the opportunities and good moments that I have had during these years of my Ph.D.

I would also like to thank the specialist and faculty members of KMC Mangaluru, **Dr. Padmanabha Kamath and Dr. Chakrapani M** for their time and guidance. I thankful to KMC Mangaluru, for providing wonderful opportunity to work with live data and ethical clearance for the project.

I would like to thank all the Teaching and non-Teaching staff members of the Department of Mechanical Engineering, NITK and KMC Mangaluru for their continuous help and support throughout the research work.

My sincere thanks go to all my family members, for their unwavering support, encouragement, and love. Without their support, this accomplishment would not have been possible.

I am also grateful to my friends for their support, understanding and cheering me during the stressful times of my research.

The list goes on and there are many others I should mention. There are people who have helped me all the way and provided me support when I didn't even realize I needed it, or needed it now, or needed it constantly. Listing all of them would fill a book itself, so I merely will have to limit myself to a few words: I THANK YOU ALL……!

(Shrivathsa TV)

ABSTRACT

The advancement of healthcare prediction systems has revolutionized the medical field, enabling to predict and prevent diseases severity, improve patient care, and enhance healthcare efficiency. This requires proper study of historic data in the related field and thorough analysis. Greater emphasis is laid on relevance of live data rather than repository data available in scholarly database. Again, the causes of a disease may vary geographically due to distinct living conditions or environmental conditions. At the same time, the ability of a medical practitioner to decipher information out of diagnosis procedure followed will be limited by his expert knowledge or experience. It is in such situations that a reliable accurate prediction system based on Artificial Intelligence (AI) comes as an assisting tool to the medical fraternity in conflict resolution. An AI-based diagnostic system will definitely help the medical expert in arriving at remedial solution, since knowledge base contained in it is based on sound design. The prediction system attempted in the present work consists of two stages. In the first stage, prediction system was developed for classification of undifferentiated fever symptomatic disease. The motivation of good results at this stage led to the development of full-fledged end-to-end predictive system for identification and classification of coronary artery disease (CAD), with consideration of electrocardiogram (ECG) and treadmill test electrocardiogram (TMT-ECG, stressed ECG) signals. It is then validated with angiography results.

Accurate diagnosis of undifferentiated fever symptomatic disease at the earliest is a challenging task necessitating extensive diagnostic tests. The aim of the present study was to apply Artificial Intelligence (AI) algorithm using temperature information for the prediction of major categories of diseases among undifferentiated fever symptomatic disease cases. Illnesses like tuberculosis, non-tubercular bacterial infection, dengue fever, and non-infectious diseases have regular manifestations of fever symptoms. The present work uses only temperature data of the patient being referred in predicting the nature of fever symptomatic disease, with the highest degree of accuracy, instead of several self-defined parameters over an interval of time. This was an observational study carried out in tertiary care hospital and validated with the

help of experienced physicians. Back-propagation algorithm was used to train the network. A good relation was found between the target data set and output data set, purely based on the observed 24 hrs. continuous tympanic temperature of the patients. An accuracy of 99% was achieved from the Artificial Neural Network (ANN) prediction model. Prediction model with different classifiers (logistic regression, decision tree classifier, k-nearest neighbor's classifier, linear decrement analysis, Gaussian Naive Bayes classifier, and Support Vector Machine) were experimented for optimization. The optimized prediction model deals with lesser time intervals and shows good performance of results when it is combined with additional medical parameters which may be considered during medical testing. A result of predictive system defines with a good classifier adaptation will show a strong performance in identification of fever-symptomatic diseases. Accuracy score and other salient parameters describe the complete picture of the system. No other investigation has ever been carried out so far taking temperature as the only parameter in classification of diseases achieving an accuracy of as high as 99.9%.

Based on the success attained here, a more complicated problem is taken up for investigation related to coronary artery disease. Coronary artery disease (CAD) is one of the major cardiovascular diseases and is a cardiac condition where plaque formed in arteries leads to death worldwide. The identification of CAD in the traditional approach needs a report of ECG, TMT ECG, Pharmacological test, and echocardiogram. The confirmation of CAD leads to the next stage of cardiac catheterization. An accurate prediction system that can detect the existence of CAD with an initial test like an ECG or TMT ECG report can assist doctors during periodic health monitoring of patients. It may be challenging and time-consuming to visually assess the ECG signals. Identification of abnormal ECG morphology, especially in low amplitude curves may be prone to error. Initially, an image processing method has been developed and implemented for the extraction of data from ECG and TMT-ECG reports. The 12 lead TMT-ECG report provides cardiac information of abnormality under medication. This information plays a vital role in automated cardiac analysis. Any small discontinuity in the ECG/TMT-ECG images will be patched up by the developed method. The data extraction method involves scanning of ECG and TMT-ECG images, masking, binarization, and morphological operation, etc. These extracted data are compared with the available output of commercial software (IM2GRAPH) In addition to data extraction, a part of the algorithm based on hybrid method is used to identify and classify important major features namely P, Q, R, S, T, PQ segment, QRS complex, QT segment, and ST segment. A convolutional neural network model is developed which works on the data extracted from ECG signals (one-dimensional data). The developed Convolutional Neural Network (CNN) architecture deals with single-lead and multi-lead (12 Lead) ECG and TMT-ECG data effectively. The highlight of the CNN system developed is that entire data is collected from the clinical lab of a renowned neighboring hospital. The automated computer-assisted system helps in the detection of CAD with an accuracy of 99%.

The study also focused on developing a prediction system for CAD disease based on raw and filtered, single-lead and twelve-lead ECG signal images (two-dimensional), by passing data extraction. The algorithm results are compared with transfer learning algorithms. The novelty of the work is highlighted by the fact that the prediction accuracy of the developed algorithm, with a single lead and twelve lead ECG or TMT ECG signals (accuracy of approximately 93.5% for single lead and 94% for twelve lead) is much higher compared to transfer learned algorithms. The developed model exhibited better accuracy with lesser number of layers compared to deeper pre-trained algorithms. Further improvement is achieved by developing a novel multi-headed model which deals with both one-dimensional data and two-dimensional data simultaneously. This hybrid deep multi-headed model is built with a combination of two prediction models which work parallelly. The outcomes of these models are concatenated at the end part of model before flowing to the output layer. This process helps to extract and collect more featuristic information related to disease with all possibilities during prediction. To generalize this methodology, it is further tested over a repository dataset and has shown good performance and acceptable results. For good accessibility, a user-friendly Graphical User Interface (GUI) is developed based on proposed algorithm to support healthcare experts in classifying CAD ECG signals without much effort. The prototype model which is developed can be tested with a still larger dataset before implementation for clinical usage.

Keywords: Diseases, Prediction, Fever, Artificial Intelligence, Neural network, classifiers, Coronary Artery Disease (CAD), ECG, TMT-ECG, Digitization, Convolutional Neural Network (CNN), Prediction algorithm Transfer learning, Pre-processing, GUI, etc.

TABLE OF CONTENTS

Sl. No.	Contents	Page No.
	Declaration	
	Certificate	
	Acknowledgments	
	Abstract	
	Table of Contents	i-vi
	List of Figures	vii-xiv
	List of Tables	xv-xviii
1	Introduction	1-14
	1.1 Introduction of Artificial Intelligence	1
	1.2 Classifiers	1
	1.2.1 Logistic Regression (LR):	3
	1.2.2 K-Nearest Neighbor Classifier (KNN)	3
	1.2.3 Linear Discriminant Analysis (LDA)	4
	1.2.4 Gaussian Naive Bayes (Gaussian NB)	4
	1.2.5 Support Vector Machine (SVM)	5
	1.2.6 Decision Tree Classifier (CART)	6
	1.3 Deep Learning Model	6
	1.4 Convolutional Neural Networks	7
	1.5 Transfer Learning Approach of Pre-Trained Models	7
	1.6 Fever Symptomatic Diseases	8
	1.6.1 Tuberculosis (TB):	8
	1.6.2 Nontuberculous Mycobacteria (NTM)	9
	1.6.3 Dengue	9
	1.6.4 Non-Infectious Disease	10
	1.7 Heart Diseases	11
	1.7.1 Arrhythmia	11

1.7.2 Coronary Artery Disease (CAD) 11
1.7.3 Arterial Fibrillation 12
1.7.4 Myocardial Infarction 12
1.8 Electrocardiogram (ECG or EKG) 12
1.9 Tread Mill Test - Electrocardiogram (TMT-ECG) 14

2 Literature Review 15-47
2.1 Undifferentiated Fever Symptomatic Disease 15
2.2 Application of Classifiers in Disease Prediction 17
2.3 Literature Study Related to Heart Diseases 20
2.4 ECG Digitization 27
2.5 Application Neural Network on Heart Disease 37
2.6 Transfer Learning Approach 41
2.7 Multi-Headed Model Approach 41
2.8 Summary of The Literature 42
2.8.1 Summary of Undifferentiated Fever Diseases Study 42
2.8.2 Summary of Heart Diseases Study 42
2.8.2.1 Sources of Data 42
2.8.2.2 A Number of Leads 43
2.8.2.3 Duration 43
2.8.2.4 Annotations 43
2.8.2.5 Pre-Processing 44
2.8.2.6 Prediction Algorithms 44
2.9 Research Gap 45
Research Objectives 47

3 Research Methodology 49-68
3.1 Developed ECG System- An Experimental Setup Of ECG 49
3.1.1 Arduino Uno Microcontroller 49
3.1.2 AD8232 Sensor (3 Lead ECG Sensor) 50
3.1.3 Plot Obtained Using Single AD8232 Sensor 51

- 3.2 Agatsa (Sanket Life) ECG Sensor 52
- 3.3 GE MAC 5500HD Machine 53
- 3.4 Clinical Dataset 54
  - 3.4.1 Temperature Data 54
    - 3.4.1.1 Tuberculosis Diseases 55
    - 3.4.1.2 Non-Tubercular Bacterial Infection 56
    - 3.4.1.3 Dengue Fever Case 57
    - 3.4.1.4 Non-Infectious Diseases 58
  - 3.4.2 ECG and TMT-ECG Data 59
  - 3.4.3 Repository Data 60
- 3.5 Undifferentiated Fever Analysis 61
  - 3.5.1 24-Hour Temperature Data Pattern Analysis 61
  - 3.5.2 Refinement of Undifferentiated Fever Analysis 64
  - 3.5.3 Prediction System Development Based on Machine Learning Technique 66

4 Digital Image Processing 69-88
- 4.1 ECG Image Pre-processing and Digitization 70
  - 4.1.1 Image Scanning 73
  - 4.1.2 Image Masking 74
- 4.2 Threshold Setting Or Color Image Segmentation 75
- 4.3 Binarization 77
- 4.4 Open Area Deletion 78
- 4.5 Morphological Operation 79
- 4.6 Filtering 80
- 4.7 Text Detection and Removal 81
- 4.8 Dilation and Multi-Cropping. 82
- 4.9 Signal Data Storage Using Pixel Indexing: 83
- 4.10 ECG Feature Parameter Extraction 84
  - 4.10.1 Low Pass and High Pass Filters 85
  - 4.10.2 Squaring 86

4.10.3 Moving Window Integration 86
4.10.4 Adaptive Thresholding 86
4.10.5 Feature Extraction 87

5 Prediction Model for Coronary Artery Disease 89-120
5.1 One-Dimensional ECG Prediction Analysis 89
5.1.1 Single-Lead ECG/TMT-ECG-Based CNN Architecture 90
5.1.2 Multi (12)-Lead ECG/TMT-ECG Based CNN Architecture 97
5.2 Two-Dimensional CAD ECG Image Prediction Analysis 104
5.2.1 Two-Dimensional ECG Prediction Model 105
5.2.2 Transfer Learning Approach 107
5.2.2.1 VGG16 Transfer Learning 107
5.2.2.2 MobileNetV2 Transfer Learning 108
5.2.2.3 Inception Transfer Learning 109
5.2.2.4 ResNet Transfer Learning 109
5.2.2.5 EfficientNet Transfer Learning 110
5.3 Multi-Headed Prediction Model 110

6 Results and Discussions 121-182
6.1 Fever Symptomatic Disease Prediction System 121
6.2 Refinement of Fever Prediction System 127
6.2.1 Diagnostic prediction system with temperature data alone 128
6.2.2 Diagnostic Prediction System with Temperature Data and Additional Features 130
6.3 One Dimensional Data Based AI System for Large Data Handling 133
6.3.1 Correlation study on ECG signals 133
6.3.2 Performance metrics evaluation of one-dimensional data analysis 142
6.3.3 Identification of the Optimal Model for one-dimensional ECG data 147

    - 6.3.4 k-Fold Cross Validation Technique 150
- 6.4 Analysis of the Two-Dimensional Prediction Model 151
    - 6.4.1 Analysis of Single Lead 152
    - 6.4.2 Analysis of Twelve Lead 158
- 6.5 Multi-Headed Model 164
- 6.6 Graphical User Interface (GUI) 178

Research Summary 183-185

Conclusions 187

Scope for Future Work 189

Reference 191-213

List of Publications 215

Appendices

- Appendix -I Algorithmic Code for Multi-dimensional Analysis 217-250
- Appendix -II Working Principle of Deep Learning Models 251-265
- Appendix -III Ethical Certificates 267-268

Bio-data 269

LIST OF FIGURES

Figure No.	Figure Description	Page No.
1.1	Data classification in kNN classifier	3
1.2	Data classification in SVM classifier	5
1.3	Flow chart of Decision tree classifier (Mahamdi et al. 2022)	6
1.4	Arrhythmia heart condition	11
1.5	Plaque formation under CAD	11
1.6	Structure of the heart and ECG (Al-ani 2014)	13
2.1	Literature study layout related focused area of research	16
2.2	Overall literature study layout focused on heart related issues	20
2.3	Flow chart of literature study focused on digitisation and prediction models	27
3.1	The three sensors connected in parallel	50
3.2	Circuit design for AD8232's in parallel using Arduino and breadboard	51
3.3	AD8232 single senor output	52
3.4	Output from three AD8232 sensors in parallel	52
3.5	Agatsa ECG sensor report	53
3.6	sample datasets temperature distribution pattern for Tuberculosis disease	55
3.7	sample datasets temperature distribution pattern for Non-tubercular bacterial infections diseases	56
3.8	sample datasets temperature distribution pattern for dengue fever diseases.	57

3.9 sample datasets temperature distribution pattern for non-infectious diseases 58
3.10 The scanned ECG signal of 12 lead signals 59
3.11 The scanned paper ECG of 12 lead signals for normal (Left) and confirmed CAD (Right) which have resting and treadmill test ECG signal images 60
3.12 ECG distribution plot over a time of period for the repository dataset 61
3.13 Flow chart of undifferentiated fever prediction analysis 63
3.14 Simple neural network 63
3.15 Generalized methodology of fever symptomatic disease analysis 64
3.16 Temperature variation versus time for ten random samples of (a).Tuberculosis, (b).Non-tubercular bacterial infection, (c).Dengue fever and (d).Non-infectious diseases 65
3.17 Prediction model methodology 67

4.1 RGB color distribution in M*N*3 array 70
4.2 Overview of the block diagram related to pre-processing 72
4.3 Single-lead TMT-ECG digitization 72
4.4 Twelve-lead TMT-ECG ECG report 73
4.5 (a) Skewed Image of ECG (b) De-skewed image of ECG 74
4.6 Masking image 75
4.7 ECG graph after adding the mask 75
4.8 Histogram of RED color of ECG 76
4.9 Binary ECG image with noise 77
4.10 Binary image of ECG 78
4.11 Clean binary image of ECG 79

4.12 After a morphological operation 80
4.13 Filtered ECG image 80
4.14 Text detected image 81
4.15 Text erased ECG image 82
4.16 Cropped single lead images 82
4.17 Flow chart for pixel indexing 83
4.18 12 Lead ECG data comparison between original data and extracted data 84
4.19 (a) Before application of low pass filter, (b) After application of low pass filter, and (c) After application of high pass filter 85
4.20 Application Moving Window Integration 86
4.21 QRS Feature parameters 87
4.22 Extracted feature data of the ECG 88

5.1 Generalized three-layer convolutional neural network architecture 90
5.2 (a) Single CNN layer, (b) double CNN layer, and (c) triple CNN layer for the single-lead ECG/TMT-ECG 93-95
5.3 The proposed CNN architecture for single-lead and multi-lead (twelve lead) ECG/TMT-ECG respectively 96
5.4 (a) Single CNN layer, (b) double CNN layer, and (c) triple CNN layer for the single-lead ECG/TMT-ECG 98-100
5.5 Generalized architecture of developed deep neural network for single lead and twelve lead ECG/TMT-ECG signal images 104
5.6 Layer details of deep convolutional neural network 106
5.7 Developed deep convolutional neural network layer wise input and output parameter details 106
5.8 Generalized architecture of pre-trained neural network adopted for transfer learning approach 107

5.9 Pre-processed noise-free ECG/ TMT-ECG signal image extraction for multi-headed hybrid model 111
5.10 Generalized architecture of multi-headed hybrid ECG/ TMT-ECG prediction model 112
5.11 Multi headed deep neural network model architecture for (a) ECG signal image (2D data) +Extracted ECG time series Data (1D data) (b) ECG signal image (2D data) + ECG flattened image (1D data) 116
5.12 Multi headed deep neural network model architecture for ECG signal image (2D data) + ECG flattened image (1D data) 117
5.13 Multi headed deep neural network model architecture for ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data 118

6.1 ROC plot of 24-hour temperature prediction model during overall condition 123
6.2 Performance plot of 24-hour temperature prediction model 123
6.3 Error histogram plot of 24-hour temperature prediction model 124
6.4 (a) Precision score and (b) Recall ability of diseases with different classifier conditions. 126
6.5 Evaluation of F1 scores for Dengue fever, Non-infectious diseases, Non-tubercular bacterial infection, and tuberculosis disease under the different prediction systems. 130
6.6 Comparison ECG plot obtained different methods for all limb leads (I, II, III), augmented leads (aVR, aVL, aVF) and Precordial leads (V1, V2, V3, V4, V5, V6) 134

6.7	Comparison Stressed ECG (TMT-ECG) plot obtained different methods for all limb leads (I, II, III), augmented leads (aVR, aVL, aVF) and Precordial leads (V1, V2, V3, V4, V5, V6)	135
6.8	Comparison (a) data flow and (b) data distribution for all the three types of data extraction method	136
6.9	(a) Plot of Area of interest selected from Physio-net data set, (b) Comparison plot of extracted result and original data related to area of interest, (c) Comparison plot of extracted result and original data for large data	138
6.10	Comparison of features of the ECG sample and digitized ECG for the sample data	139
6.11	Extracted feature data of TMT-ECG	140
6.12	The best CNN model for the single-lead ECG and the multi (12)-lead ECG	148
6.13	(a) Accuracy and (b) validation loss of developed model for single-lead ECG and TMT-ECG	149
6.14	(a) Accuracy and (b) validation loss of developed model for twelve-lead ECG and TMT-ECG	149
6.15	Five-fold cross-validation technique adopted for single lead ECG and TMT-ECG	150
6.16	Accuracy plot and Validation loss of developed network for filtered single lead two-dimensional image data	153
6.17	(a-f): Receiver Operating Characteristics (ROC) plot of transferred learning model and developed model for single lead filtered two-dimensional data	154-155

6.18 (a) Accuracy plot and (b) Validation loss of developed network for twelve lead two dimensional image data 160

6.19 (a-f): Receiver Operating Characteristics (ROC) plot of transferred learning model and developed model for twelve lead filtered two-dimensional signal images 161

6.20 Clinical data multi headed model performance based on single lead with consideration of ECG and TMT signal image (2D) + Extracted ECG and TMT time series data (1D) +ECG flattened image (1D) 167

6.21 Repository data multi headed model performance based on single lead with consideration of ECG signal image (2D) + Extracted ECG time series data (1D) + ECG flattened image (1D) 167

6.22 ROC plot for ECG signal image (2D data) +Extracted ECG time series Data (1D data) of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model 169

6.23 ROC plot for ECG signal image (2D data) + ECG flattened image (1D data) of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model 170

6.24 ROC plot for ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model 171

6.25 ROC of Repository data for single lead hybrid multi headed model 172

6.26 GUI home page of prediction model 179

6.27 Single lead ECG image selection page 180

6.28 Twelve lead ECG image selection page 181

6.29 Indication model results in terms of "Predicted Positive" for Single lead and multi lead ECG/ TMT-ECG selected image 182
6.30 Indication model results in terms of "Predicted Negative" for Single lead and multi lead ECG/ TMT-ECG selected image 182

LIST OF TABLES

Table No.	Table Description	Page No.
2.1	Application of Machine learning and AI approach for classifications of Undifferentiated fever	19
2.2	Prediction of heart disease based on different feature parameters and smart approaches	22
2.3	Prediction of heart disease based on different parameters and traditional approaches	23
2.4	Prediction of heart disease based on different machine learning and AI approaches	24
2.5	Parameters and techniques adopted for digitisation ECG	29
2.6	Prediction of arrythmia disease based on different techniques and different parameters	32
2.7	Prediction of coronary artery disease (CAD) based on different techniques and different parameters	33-34
2.8	Prediction of myocardial infarction (MI) disease based on different techniques and different parameters	35
2.9	Prediction of atrial fibrillation (AF) disease based on different techniques and different parameters	36
2.10	ANN and DNN Techniques adopted to develop the prediction model for CAD	40
5.1	Detailed CNN architecture for a single-lead ECG with one-convolutional, two-convolutional, and three-convolutional layers respectively	91
5.2	Detailed CNN architecture for multi (12)-lead ECG with one-convolutional, two-convolutional, and three-convolutional layers, respectively	103

6.1 Confusion matrix of 24-hour temperature prediction model during (a) training, (b) validation, (c) testing, and (d) overall, respectively 122
6.2 Comparison of results between discriminative type classifier and generative type classifier-based model 125
6.3 Confusion matrix results for best classifier algorithm 127
6.4 Accuracy Score obtained for all the classifiers under all four prediction system categories. 129
6.5 Confusion Matrix of Decision Tree (CART), kNN and Support vector machine for Dengue fever, Non-infectious diseases, Non-tubercular bacterial infection, and tuberculosis disease under thirty minutes with additional features 131
6.6 Behavior of kNN and CART-based prediction system over unseen data with consideration of additional features. 132
6.7 Correlational results of the developed method and IM2GRAPH method with respect to original data 137
6.8 Feature parameter of Extracted data and Manual data for single lead TMT-ECG. 139
6.9 Correlation of feature parameter values of hybrid method 140
6.10 Correlation coefficient values of developed method for physio-net data and results developed by Baydoun. 141
6.11 Results of the single-lead ECG for the single-, two-, and three-convolutional layer networks 143
6.12 Results of the multi (12)-lead ECG for the single-, two-, and three-convolutional layer networks 144
6.13 Confusion matrix of developed model for the single-lead ECG. 146

6.14 Confusion matrix of developed model for the 12-lead ECG 146

6.15 Accuracies of the k-fold cross validation over single lead and twelve lead ECG and TMT-ECG data 151

6.16 Comparison of transfer learned and developed models' accuracies of raw data for single lead signal images 152

6.17 Comparison of Transfer learned and developed models' accuracies of filtered data for single lead signal images 153

6.18 Confusion matrix plot of transferred learning model and developed model for single lead filtered two-dimensional data 155-156

6.19 Precision, Recall and F1 Score of transfer learned and developed network for single lead signal image data 157

6.20 Comparison of Transfer learned and developed models' accuracies of raw data for twelve lead signal images 159

6.21 Comparison of Transfer learned and developed models' accuracies of filtered data for twelve lead signal images 159

6.22 Confusion matrix plot of transferred learning model and developed model for twelve lead filtered two-dimensional signal images 162-163

6.23 Precision, Recall and F1 Score of transfer learned and developed network for 12 lead signal image data 163

6.24 Performance accuracy of multi headed model with all the data types during different combinational approach 166

6.25 Accuracies of the k-fold cross validation over multi headed model 168

6.26 Precision, Recall and F1 Score of multi headed model with all the data types during different combinational approach 173

6.27 Confusion matrix of ECG signal image (2D data) +Extracted ECG time series Data (1D data) for (a) single 175 lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

6.28 Confusion matrix of ECG signal image (2D data) + ECG flattened image (1D data) for (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model 176

6.29 Confusion matrix of ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data for (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model 177

6.30 Confusion matrix of repository data for single lead hybrid multi headed model 178

CHAPTER 1

INTRODUCTION

Artificial Intelligence (AI) based prediction systems are gaining prominence in each and every field, assisting the domain experts in evaluation and inferencing. These systems are backed by huge amount of context sensitive data and decision making capabilities, along with domain knowledge of the concerned experts. Nowadays, these techniques are popular in the healthcare domain to identify and classify diseases. Based on this concept, the present research work is carried out in two stages. At first, an AI system for classifying different types of fever symptomatic disease is developed to explore the relevance of AI in medical diagnosis. In the next stage, a full-fledged prediction system for cardiovascular disease detection has been developed and validated, with live data from a reputed hospital. As a result, an attempt is made to investigate the relevance of AI in medical diagnosis, both in terms of one-dimensional and multidimensional data analysis.

1.1 INTRODUCTION OF ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) is an ability of system or machine to mimic or improve human performance, such as reasoning and experience-based learning. Machine learning (ML) is defined as development of systems that are able to learn and adopt without following explicit programming. These ML models are classified into four categories i.e. supervised learning, unsupervised learning, reinforcement learning and artificial neural networks (ANN) to handle both structured data (data classification, segmentation, and anomaly detection, etc.) and unstructured data (text analysis, speech recognition and image classification, etc.).

1.2 CLASSIFIERS

According to the supervised learning technique, each and every set of patterns is predefined with a known class. These class labels will represent the output decision.

(Boden 2014) states that input data should match with given labelled data to obtain good accuracy so that newer observations will effectively be classified based on past learnt experience.

A classifier based on the machine learning algorithm is a supervised learning technique that automatically arranges or categorizes data into one or more of a set of "classes". (Hartmann et al. 2019) mentioned that comparative results of classifiers are to be studied for understanding the behavior of every classifier. There are many statistical and machine learning approaches hidden in the principle of these classifiers. All the major classifiers are considered in implementation to classify fever symptomatic disease.

(Fujino et al. 2007) segregated classifiers into two bases: discriminative model classifier and generative classifier. The term discriminative model describes a group of statistical classification models that primarily use supervised machine learning techniques. As a result of their ability to identify the borders between classes or labels in a dataset, these models are often referred to as conditional models. In discriminatory modelling $P(\mathrm{y}|\mathrm{x})$, matching the specified unobserved variable (target) x into a class label y based on the observed variables is represented as equation 1.1,

$$P(\mathrm{y}|x) = \mathrm{x} \tag{1.1}$$

A Generative Model learns from the joint probability distribution p(x,y) which helps to distribute each class. This model predicts conditional probability using Bayes Theorem (Stylianides and Kontou 2020). For a variable X (features) and target variable Y (labels), a generative model is a statistical model of the joint probability distribution on X × Y and is represented as,

$$P(X, Y) \tag{1.2}$$

The major classifiers which are leading in both the categories are as discussed below:

1.2.1 Logistic Regression (LR):

In LR method, the model is used to rectify the data, which belong to the same and other categories. The probabilistic score which varies from 0 to 1 is assigned based on the type of the data. (Marker et al. 2019) explained classifications are made depending on the probabilistic score. The binary logistic regression model (equation 1.3) works on the decision of main content (whether pass or fail). Logistic regression for multi-label categorization is done in two ways: either by Multinomial or by Ordinal Logistic Regression. The general logistic function $\sigma: R \rightarrow (0,1)$ is defined as,

$$\sigma(t) = \frac{e^t}{e^t+1} = \frac{1}{1+e^{-t}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots++\beta_n X_n)}} \qquad (1.3)$$

Where, β_i are parameters of the model and $X_i$are the observation of the logistic model.

1.2.2 K-Nearest Neighbor Classifier (KNN):

The k-nearest neighbor algorithm, also referred to as KNN or k-NN, is a supervised learning classifier that uses similarity to make classifications or predictions about the grouping of every data point. k-nearest neighbor is known as one of the non-parametric methods of classifier. (Adeniyi et al. 2016; Hmeidi et al. 2008) assigns the available data to one particular class, on the basis of the majority of votes of its neighbors (Figure 1.1). It is commonly employed as a classification algorithm because it relies on the assumption that comparable points can be discovered close to one another.

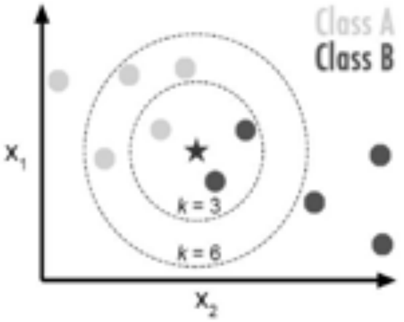


Figure 1.1 Data classification in kNN classifier

1.2.3 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) or Discriminant function analysis is a generalization of Fisher's linear discriminant (Rani et al. 2013), a method used in statistics, pattern recognition, and machine learning to find the linear combination of features that characterize the object. It is preferable during is the presence of more than two classes of target values.

When there are more than two classes, the analysis used in the derivation of the Fisher discriminant (equation 1.4) can be extended to find a subspace which appears to contain all of the class (C) variability. The scatter between class variability may be defined by the sample covariance of the class means μ in a direction ψ,

$$J(\psi) = \frac{\psi^T S_B \psi}{\psi^T S_W \psi} \tag{1.4}$$

Where,

$$S_B = \sum_{i=1}^{C} n_i(\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^{C} S_i \sum_{X \in D_i} n_i(x - \mu_i)(x - \mu_i)^T$$

Class scatters matrix S_B and S_W defined with the function of μ are the k-dimensional samples. Mean for the whole set, μ_i is the sample mean and S_i is the scatter matrix for i^{th} class. D_i and n_i are discriminant class and number of samples in class I respectively.

1.2.4. Gaussian Naive Bayes (Gaussian NB):

In the Gaussian Naive Bayes algorithm, (Haq et al. 2018) explains probabilities of each attribute which belong to each class being considered for a prediction. The assumption of the algorithm is, all the other attributes are not depending on that the probability of a given class value attribute (equation 1.5). In case, the value of the attribute is known, then the probability of a class value is defined as the conditional probability. Prediction is based on the calculation of each class instance probability and selection of the highest probability class value.

$$P(M/N) = P(N/M) * P(M)/P(N) \tag{1.5}$$

Where, $P(M/N)$ is the probability of attribute M given the data (values) N. This is called the posterior probability. $P(N/M)$ is the probability of data N given that the attribute M was true. $P(M)$ is the probability of attribute M being true (regardless of the data). This is called the prior probability of M. $P(N)$ is the probability of the data or values (regardless of the attribute).

1.2.5. Support Vector Machine (SVM)

Support vector machine algorithm or SVM is one of the most well-known supervised machine learning techniques. It is helpful in both classification and regression-type problems. The objective of SVM is identification of a hyperplane in N-dimensional space to distinctly classify the data (Figure 1.2). (Saleh 2011) considered N-dimensional feature space to plot each data value as a point, with the value of each feature as a particular coordinate. Hyper-plane is used to get better output of classification. In the case of SVM training, samples are classified into different subsets as support vectors. The decision function is specified by these support vectors.

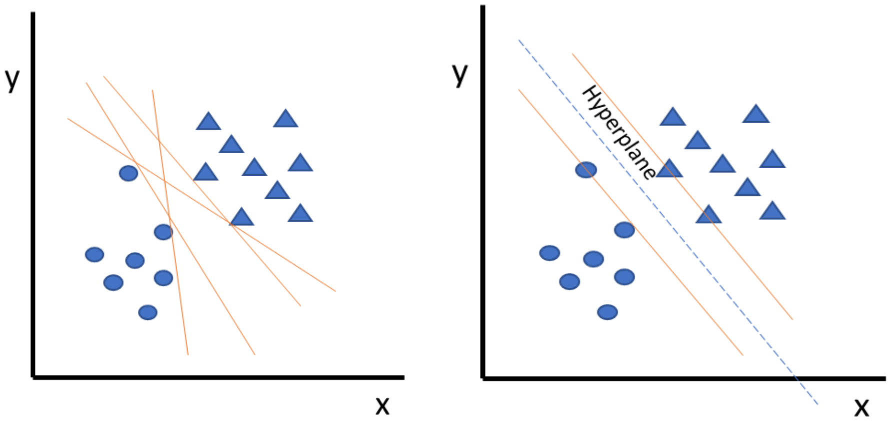


Figure 1.2 Data classification in SVM classifier

1.2.6. Decision Tree Classifier (CART)

Another well-known supervised learning technique is called as decision tree algorithm which suits both classification and Regression problems and is most preferable for classification problems. (Lee et al. 2017) explains about this prediction modelling approach which flows from source observation to the target conclusion. (Seera and Lim 2014) Figure out in the classification tree (Figure 1.3), leaves will represent class labels, and branches will be conjunctions of the features.

Among well-known classifiers such as logistic regression (Belavagi and Muniyal 2016), decision tree classifier, k-neighbors classifier (Trstenjak et al. 2014), linear decrement analysis (Rani et al. 2013), Gaussian Naive Bayes (Haq et al. 2018), SVM (Saleh 2011), etc, the SVM and decision tree classifiers (Classification and regression tree (CART) algorithm) are considered as a major classifier (Lee et al. 2017).

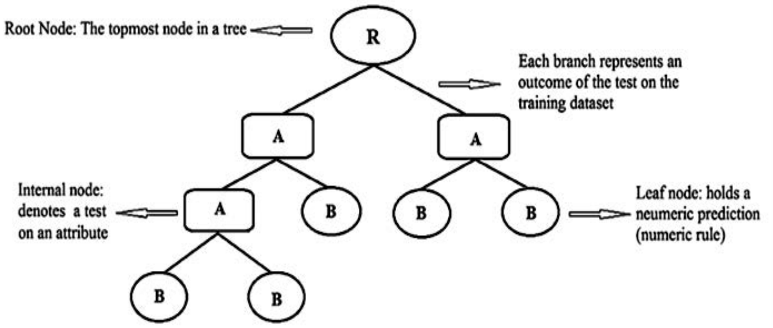

Root Node: The topmost node in a tree
R
Each branch represents an outcome of the test on the training dataset
A
A
Internal node: denotes a test on an attribute
A
B
B
B
Leaf node: holds a neumeric prediction (numeric rule)
B
B

Figure 1.3 Flow chart of Decision tree classifier (Mahamdi et al. 2022)

1.3 DEEP LEARNING MODEL

Deep learning is a machine learning system that triggers the computer or system to learn from experience and interpret in terms of a hierarchy of concepts. These models are cable to extract the low- and high-level features automatically which are required for categorization. Deep learning is made up of a number of layers of neurons, to learn a structured representation of big data, layer by layer. Generally, for analyzing structured data types, either dense deep neural network or the one-dimensional deep convolutional neural network is used. But during analysis of unstructured data types, (like images, speech recognition, etc.) two-dimensional convolutional networks are used popularly to automatically detect the features and classify the categories.

1.4 CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) are feed-forward Artificial Neural Networks (ANNs) which consist of alternative convolutional and subsampling layers. which are used for deep learning algorithms and for computer vision techniques over the past ten years. A two-dimensional deep convolutional network (Conv2D) with multiple hidden layers and more feature parameters has the ability to learn complex objects, images, and patterns with the availability of large data. The trained system can play a significant role in many technical applications for 2D signals, such as photos. However, these systems are not a feasible alternative for one-dimensional signals or data, particularly if the training data is limited. 1D CNNs have recently been proposed as a solution to the above problem and have already attained state-of-the-art performance levels in a number of applications, including the classification and early diagnosis of personalized biomedical data, health monitoring, and anomaly detection. Another significant benefit of Conv1D is it can easily be incorporated into real-time analysis because Conv1D is configured only with 1D convolutions. Hence it is simple to monitor various hyperparameters and is widely used for data analysis. The various hyper parameters generally used in deep learning algorithms are discussed in appendix.

1.5 TRANSFER LEARNING APPROACH OF PRE-TRAINED MODELS

Transfer learning approach is similar to person sharing the knowledge of known things to unknown one. Similarly here, if the model is trained over large dataset, the gained knowledge is compiled as weights of the model. These weights can be extracted and transferred to any other network model known as transfer learned feature network. This will help to avoid training it from scratch and obtain better results. The best known pre-trained models are as discussed in appendix.

1.6 FEVER SYMPTOMATIC DISEASES

A fever is an indication of body temperature higher than normal and even known as high temperature, hyperthermia, or pyrexia. Normal body temperature differs from person to person and lies within the range of 97 to 99 ^{o}F. If the temperature is 100.4 ^{o}f or higher, it is considered as fever. A body temperature is regulated by the hypothalamus, a region of the brain. The hypothalamus can reset the body temperature and the failure of this will result in an increase in temperature.

Nowadays, fever diseases are common and universal. Symptoms (Frieden et al. 2014) of many fever diseases (majorly Tuberculosis, Non-tubercular bacterial infections, Dengue fever, Non-infectious diseases, etc.) are similar and the prediction of the exact type of fever will be based on several types of tests. This procedure is time-consuming and cost-intensive. Even inaccurate identification may lead to some other type of problem. To bring down such uncertainties, many researchers have worked on the accurate prediction of diseases using different parameters. In this study, only temperature data as a parameter is considered, for the prediction of Tuberculosis, Non-tubercular bacterial infections, Dengue fever, and Non-infectious diseases, which is very unique development.

1.6.1 Tuberculosis (TB):

From the World Health Organization (WHO) report, it is observed that Tuberculosis (TB) is one of the top ten diseases causing death worldwide. From a survey in 2017, it is found that approximately 10.5 million people get affected by Tuberculosis, and nearly two million people die due to this. Death counts are much more in low and middle-income countries (nearly 95%). (Doshi et al. 2017) Out of this, India with 64% of the total count leads the world, followed by China, Indonesia, Pakistan, the Philippines, Nigeria, and South Africa.

Artificial Intelligence (AI) and machine learning play an important role in identifying different types of TB like latent TB and active TB (Luger 2005; Samuel 1969) (Machuca et al. 2018). In recent days, diagnosis of tuberculosis is effectively done through Chest X-rays, especially in the case of pulmonary tuberculosis, which is one of the most common forms of tuberculosis. (Lakhani and Sundaram 2017) Although chest X-ray does not provide the root cause for confirming TB, they still offer a high sensitivity method for detecting tuberculosis-related abnormalities in the lungs.

1.6.2 Nontuberculous mycobacteria (NTM)

Nontuberculous mycobacteria (NTM), which is ubiquitous and affects both airways and lung tissue will cause inflammation of the respiratory system. NTM lung infection occurs when a person inhales the organism from the surrounding environment. In the case of a few susceptible individuals, the slowly progressive and destructive disease can occur (Falkinham 2003; Mirsaeidi et al. 2014).

Additionally, (Marras et al. 2007; Mirsaeidi et al. 2014) studies show that NTM-related lung infections, particularly in persons over 50 years of age, are on the rise in North America. Some patients, however, do not require treatment in case of less severe infections (Lyman et al. 2017).

1.6.3 Dengue

Compared to the above two types of diseases, dengue fever stands second on the list of dangerous diseases. Dengue fever is a mosquito-borne tropical disease caused due to the dengue virus (Diagnosis 2009; Scott 2009). Symptoms are headache, high fever, vomiting, muscle & joint pain, and characteristic rashes on the skin (Dr.N.Kannathasan 2018). WHO stated that dengue fever widely spreads all over tropical areas, with local risk variations affected by rainfall, temperature, and unwell-planned urbanization. Mosquitoes are known as one of the causes of human suffering than any other organism. A large number of people die from mosquito-borne diseases worldwide each year (Scott 2009). Increasing environmental temperature due to global warming could cause the growth rates of larval mosquitoes (Bayoh and Lindsay 2003, 2004), resulting in a large number of adult mosquitoes leading to mosquito-borne diseases (Nazareth et al. 2016).

In a few cases (Selvaretnam et al. 2016; Singla et al. 2015), the life-threatening dengue hemorrhagic fever is developed because of this, resulting in continuous bleeding, blood platelet levels becoming low, and leakage of blood plasma. Also, in some situations, blood pressure drops to a dangerously low level, resulting in dengue shock syndrome (Lee et al. 2017), which may affect life of people. More than a hundred countries are affected by this type of disease; dengue is one pandemic-prone viral disease that is fast spreading its tentacles in many parts of the world (Pe et al. 2011). Dengue is getting boom in areas like urban unhygienic places, suburbs, and the countryside but also in well-developed countries (Marras et al. 2007; Guzman et al. 2010). Early detection of this type of infectious disease is predicted by artificial intelligence (Laureano-rosario et al. n.d.; Story 2018; Diagnosis 2009; Guzman et al. 2010; Lee et al. 2017).

1.6.4 Non-infectious Disease

Non-infectious diseases are not caused by pathogens. Hence, it won’t spread from person to person. The main factors affecting non-infectious diseases are genetics, malnutrition, environment, and lifestyle. Examples of non-infectious diseases include cancer, Alzheimer’s disease, and epilepsy. Non-infectious diseases are considered as diseases which are not caused by pathogens. But non-infectious diseases are caused due to genetic or environmental factors, either by toxic environmental exposures or unhealthy lifestyles. Major non-infectious diseases have a multifactorial set of complex causes (Susilawati and McBride 2014), and a mix of genetic and environmental variables. Examples of non-infectious diseases include most cancers, cardiovascular diseases such as coronary artery disease, and diabetes mellitus (Dakappa et al. 2017 ; Dakappa et al. 2018).

The proper diagnosing technique adaptation will help to detect and diagnose these diseases in the early stage with consideration of these risk factors. One of the major fever symptomatic disease identification methods is a periodic temperature measuring device. This will provide a pictorial view of the temperature graph of the patients over the interval of time.

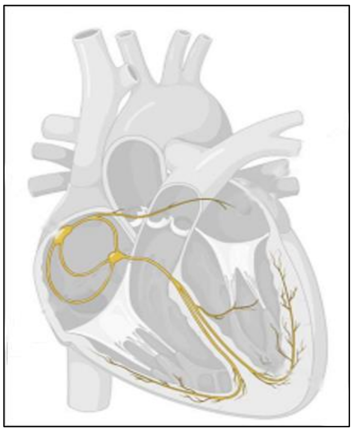
A sensor based (digital ear) thermometer is used to measure the gradient of temperature (i.e. degree of hotness or coldness of an object). A thermometer contains two essential components: a temperature sensor (Dakappa et al. 2018) (either the bulb of a mercuryin-glass thermometer or the pyrometric sensor in an infrared thermometer in a digital thermometer) and a display unit.

1.7 HEART DISEASES

The term ‘heart disease’ refers to a variety of heart disorders. The most frequent heart conditions are arrhythmia, Coronary artery disease (CAD), Atrial fibrillation (AF), and Myocardial infarction (MI). These heart diseases are commonly rectified with the help of heart tests like electrocardiogram (ECG), treadmill test - electrocardiogram (TMT-ECG), echocardiography, etc.

1.7.1 Arrhythmia

A cardiac arrhythmia is a deviation from the regular heartbeat's rhythm (Figure 1.4), i.e. is fast (tachycardia - greater than 100 beats a minute), slow (bradycardia- less than 60 beats a minute), and rapid changing (irregular). The rapid ventricular rates that interrupt atrial and ventricular output, may cause ventricular systolic dysfunction



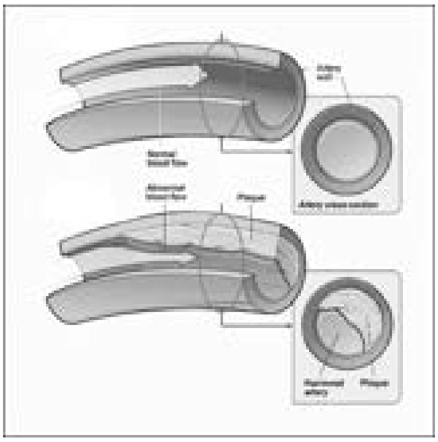


Figure 1.4 Arrhythmia heart condition **Figure 1.5** Plaque formation under CAD

1.7.2 Coronary Artery Disease (CAD)

The coronary arteries, are arteries which carry blood to heart, affected leads to coronary artery disease (CAD). i.e. Due to formation plaque in artery blocked or gradually decrease blood flow (as shown in Figure 1.5). The most frequent symptom is angina (chest discomfort), which causes chest pain and may lead to heart attack or other consequences like arrhythmia or heart failure.

1.7.3 Arterial Fibrillation

Atrial fibrillation is an irregular and rapid heart rhythm (arrhythmia) that may cause heart blood clots. Which is asymptomatic and difficult to rectify, due to this formation which intern leads to stroke, heart failure, and other heart-related issues. The heart's upper chambers (atria), which pulse asynchronous with the lower chambers (ventricles).

1.7.4 Myocardial infarction

Myocardial infarction popularly known as a heart attack, is a very serious condition that occurs due to the blockage of flow of blood to the heart muscle (severe CAD). However, the injured cardiac muscle will start deteriorating. If the treatment related to the restoration of blood flow fails, a heart attack can lead to permanent heart damage and patient death.

1.8 ELECTROCARDIOGRAM (ECG OR EKG)

An electrocardiogram (ECG) is a test that measures the electrical activity of the heartbeat. An electrocardiogram, commonly known as an ECG or EKG, is frequently performed by a health care professional, clinic, or hospital. An electrical impulse flows through the heart represented with each beat. The muscle contracts and pumps blood from the heart in response to this impulse. An ECG of a normal heartbeat will indicate the timing of the upper and lower chambers (Hong et al. 2020). An ECG is a depiction of the electrical activity of the heart muscle over time, commonly written on paper for ease of interpretation. Similar to skeletal muscle, cardiac muscle contracts in response to electrical depolarization of the muscle cells. An ECG is the sum of this electrical activity when amplified and recorded for the time interval.

Nowadays, ECG monitoring is available on several gadgets apart from health centres like smartwatches, mobile phones, smart bands, etc (Ebrahimi et al. 2020; Haverkamp et al. 2019). An ECG provides two types of information. First, based on time intervals on the ECG, the doctor can predict how long the electrical wave takes to pass through the heart. This indicates whether the electrical activity is normal or abnormal. In another method, by measuring the amount of electrical activity passing through the muscles of the heart, a cardiologist can effectively determine the proper working of the heart. The electrical depolarization pulse travels from the atria to the ventricles through the IVS (intact ventricular septum). This overall direction of travel of the electrical depolarization through the heart is known as the electrical axis.

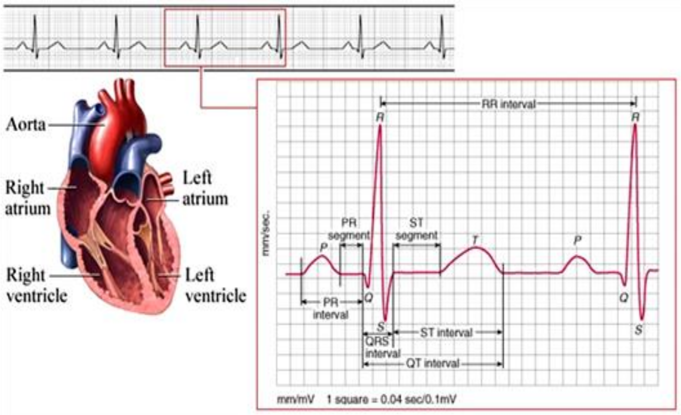


Figure 1.6 Structure of the heart and ECG (Al-ani 2014)

An ECG plot is given in Figure 1.6. An ECG plot has amplitude along the Y axis and the time along the X-axis. Amplitude refers to voltage and scale of Y-axis is 5 mm = 0.5 mV or 10 mm = 1 mV (Virgin and Baskar 2018). The X-axis refers to time, with 5 mm = 0.2 sec or 10 mm = 0.4 sec. An ECG plot is divided into the P wave, the PR segment ranges between 120 to 200ms, the QRS complex which ranges from 80 to 100ms, the ST segment, and the T wave which lasts 160ms. Furthermore, there is also the QT interval which is measured from the beginning of the QRS complex to the end of the T wave. Acceptable ranges vary with heart rate. So it must be corrected to the QTc by dividing by the square root of the RR interval, the time elapsed between two successive R waves of the QRS signal on the electrocardiogram.

1.9 TREADMILL TEST - ELECTROCARDIOGRAM (TMT-ECG)

A treadmill test (TMT) or cardiac stress test determines how far the heart can run before experiencing an irregular rhythm or a reduction in blood flow to the heart muscle. It allows doctors to learn how the heart reacts when it is pressurized. TMT will be performed in accordance with the Bruce protocol, (Akinpelu 2015; Viliane Vilcant; Roman Zeltser . 2022) i.e. a treadmill walk that will be gradually increased.

Various protocols are used, including Naughton and ramp protocols, but the most common is the Bruce protocol. A Modified Bruce technique is another adoptable method for patients. The standard Bruce procedure has three-minute phases. The gradient (Treadmill inclination) is 10% in stage I and increased by 2% each stage. The beginning speed is 1.7 mph and increases in 0.8 to 0.9 mph increments at every stage. Stage I of the Modified Bruce procedure has a gradient of zero, whereas stage II has a gradient of 5%. The first three phases of the Modified Bruce procedure have the same speed (1.7 mph). Stage 3 of the Modified Bruce procedure corresponds to Stage I of the regular Bruce protocol., Further stages are similar to Bruce's protocol (Akinpelu 2015).

The voltage values of ECG or TMT-ECG between an electrode as well as the muscle activity that measures from different prospects are stored as vector values. Behavior of these shows the overall beat of the heart as well as abnormalities in various regions of the heart muscle. Usually, ECG will be the greatest method for measuring and diagnosing irregular cardiac rhythms, specifically for abnormal rhythms produced by damage to the tissue that conducts electrical signals.

A computerised solution to interpret the ECG signal will involve digitization of the image for further processing. Investigation of the digitized ECG profile will be able to extract lot of useful information, indicative of the nature of heart condition. Since the data extracted is combinatorially complex in structure, a proper AI based analysis will be highly beneficial. Understanding the basic concepts of these algorithms leads to the development on specific application-based AI systems. Major applications (related to fever and ECG) which are correlated with this theory concepts are briefly discussed in the literature review, individually.

CHAPTER 2

LITERATURE REVIEW

The study is mainly focused on development of prediction system for two types of common health issues generally faced by the public. These two types of issues namely (i) Undifferentiated fever symptomatic diseases (like tuberculosis, non-tubercular bacterial infection, dengue fever, and non-infectious diseases) and (ii) cardiovascular disease (like coronary artery disease and heart attack). Monitoring these health issues periodically will help to prevent the major causes. These challenges effectively deal with intelligent system development. This intelligent system will help to predict the disease status based on input provided by user. These systems are train-based on features extracted related to diseased patients and normal patients’ conditions. The detailed literature study related to these diseases, data collection, prediction system, and outcomes is discussed as follows.

2.1 UNDIFFERENTIATED FEVER SYMPTOMATIC DISEASE

Healthcare sectors in the current decade are primarily focusing on intelligent systems. (Joshi, R; Kalantri 2014) discussed about the major reason for implementing intelligent systems in medical applications is the necessity of accurate way of prediction related to critical health issues. Primary studies extending the use of these systems under undifferentiated fever symptomatic disease conditions will reduce the risk and help to provide a proper diagnosis for the disease.

Machine learning algorithms have been applied in the field of medicine for predicting certain classification type of diseases. Among artificial neural networks (ANN), machine learning (ML) algorithms have been found to be the most efficient method for predicting certain diseases like heart diseases (Atkov et al. 2012; Baxt 1991), neurological diseases (Podgorelec 2012), cancer (Maclin and Dempsey 1994), etc., based on certain parameters like Electroencephalogram, clinical data (Amato et al. 2013), X-ray photographs, etc. In this study, research is mainly focused on fever symptomatic diseases and heart disease as shown in Figure 2.1.

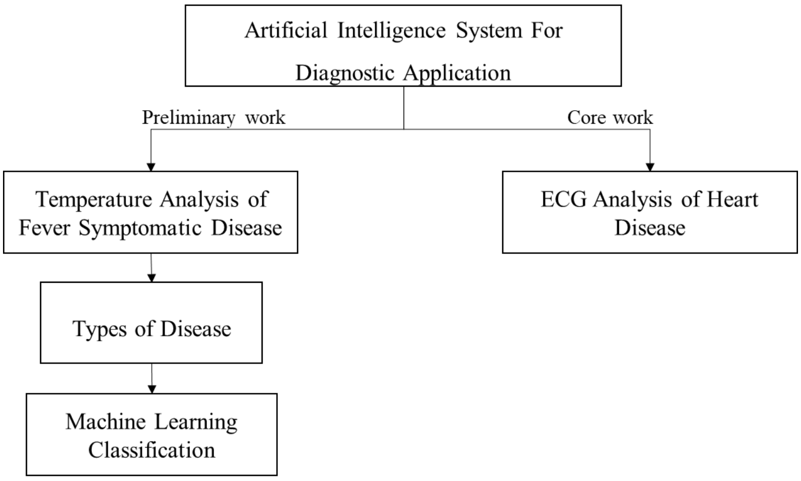

Artificial Intelligence System For Diagnostic Application
Preliminary work
Core work
Temperature Analysis of Fever Symptomatic Disease
ECG Analysis of Heart Disease
Types of Disease
Machine Learning Classification

Figure 2.1 Literature study layout related focused area of research

(Ogoina 2011) discussed related fevers can be arbitrarily classified into acute, sub-acute and chronic fevers based on duration and fever pattern observation. Undifferentiated fever symptomatic disease is the most common disease in developing countries like India, Nepal, Thailand, etc. Undifferentiated fever symptomatic disease is associated with nonspecific clinical symptoms, which are usually difficult to diagnose and treat at the earliest. Additionally, factors like age, blood pressure, and possessing diabetes may come as affecting parameters with symptoms. (Shaukat Dar and Ulya Azmeen 2015) stated that illness like tuberculosis and dengue are the most common death-causing diseases. Medication methods to rectify or classify these diseases will have a time and cost-consuming procedure. (Johnson and Odell. 2014) defined disease like non-tuberculosis myocardial bacterial infection affects the lungs and is very challenging to diagnose. The importance of recognition initial stage is highly essential before it becomes severe. (Anggraeni et al. 2017) shares the information about, on the basis of experience, it is reported that temperature based prediction is more power full tool to predict in the early stage of these typical types of diseases. Even during changes in the condition of health with respect to time, the present patient condition may lead to wrong decision making due to miscommunication between patient and health checker. This may disturb the accuracy of prediction system.

Moreover, it requires extensive invasive and noninvasive diagnostic tests for early differential diagnosis of undifferentiated fever symptomatic disease. Several researchers (Ittyachen and Ramachandran 2015; Joshi, R; Kalantri 2014; Mourad et al. 2003; Roth and Basello 2003) states the prediction system algorithm for clinical diagnostic problems will helps identify at early stage. These algorithms were developed based on results of basic hematological tests, x-rays, invasive tests, etc. (Roth and Basello 2003). The major drawback of traditional approach as discussed by (Joshi, R; Kalantri 2014) is the delay in arriving decision about diagnosis and the increased financial burden to the patients on account of the expensive tests involved. This can be improved by applying advanced prediction techniques relying on machine learning algorithms or artificial neural network models.

2.2 APPLICATION OF CLASSIFIERS IN DISEASE PREDICTION

(Dakappa et al. 2018) explains two major categories of fever symptomatic diseases, namely infectious and non-infectious diseases can be effectively classified using an artificial network algorithm. Similarly, a machine learning algorithm can also be applied to predict the undifferentiated fever symptomatic disease cases using temperature as the primary parameter. (Shaukat Dar and Ulya Azmeen 2015) discussed different types of classifier applications which are developed for the identification of dengue disease. Since there is no major work related to the classification of these fever symptomatic diseases, provide lead work with classifiers to classify the different diseases based on the same type of data. (Dakappa et al. 2017) states study used temperature and basic hematological data as parameters for differentiating undifferentiated fever into four categories of diseases (tuberculosis, dengue fever, intracellular bacterial infections, and noninfectious diseases) by applying a machine learning algorithm.

The Table lists several studies which deal with individual disease detection or prediction with a classifier-based AI approach (Table 2.1). These AI or machine learning approaches help to classify the disease with consideration of several symptoms of disease, temperature pattern of disease, Mosquito abundance (Lee et al. 2016; Young et al. 2016), etc. Datasets used for the prediction algorithm in the Table are derived from hospitals (Dakappa et al. 2017; Shaukat Dar and Ulya Azmeen 2015) and repository data (Mourad et al. 2003; Sarma et al. 2020). (Dakappa et al. 2018) study shows SVM and (Sarma et al. 2020) shows decision tree (DT)s as powerful applications in the classification of fever symptomatic disease. (Dyego et al. n.d.) states the application of classification further extends to deep neural networks to improve the accuracy of the system.

Table 2.1 Application of Machine learning and AI approach for classifications of Undifferentiated fever

Sl. No.	**Authors & Year**	**Disease**	**Dataset**	**Classification**	**Accuracy**
1	(Dakappa et al. 2018)	Infectious & Non-infectious disease	Local hospital data & features (Temperature data)	DNN	91.3%
2	(Young et al. 2016)	Mosquito abundance (malaria & dengue)	Digital mosquito ANN & monitoring	ANN & multilinear regression	Avg 90%
3	(Shaukat Dar and Ulya Azmeen 2015)	Dengue fever prediction	Hospital data (with additional features of disease)	Naïve basics, random tree, regression tree	Avg 92%
4	(Sarma et al. 2020)	Dengue	WHO-2019 Database	Decision tree (DT) & Random Forest (RF)	79%
5	(Salim et al. 2021)	Dengue	Local hospital data	SVM	70%
6	(Hooda et al. 2017)	Tuberculosis	Features and data	DNN	82%
7	(Dyego et al. n.d.)	Dengue	Revised observations	ML & DL (RF, SVM etc)	85%
8	(Xing et al. 2020)	NTM	Data of 116 patients & 103 patient	ML	85%

2.3 LITERATURE STUDY RELATED TO HEART DISEASES

The accountable major cardiac disease considered are Arrhythmia, CAD, Arterial Fibrillation, and Myocardial infarction. Arrhythmia is an irregular heartbeat (Chen et al. 2020), that occurs due to the improper functionality of a heart which is rectified by observation of electrical signals. In addition to this, another major common type of heart disease increasing rapidly in the United States and even in Asian countries is known as coronary artery disease (CAD), which affects the blood vessels through the formation of plaque (Cholesterol deposition) in arteries. This can lead to chest pain, stroke, or heart attack (Myocardial infarction), i.e., decreasing or stopping of blood flow to the coronary artery of the heart. Arterial fibrillation (AF) is another heart disease that occurs due to irregular or rapid variation of heart rhythm leading to clot of the blood in the heart. The existence of these types of diseases (Figure 2.2) can easily be identified using ECG, TMT-ECG, etc. (Martis et al. 2014) about ECG, which is a simple test to check the heart's rhythm (Repolarisation and Depolarisation) and electrical activity of the heart. Stressed ECG or TMT-ECG (Treadmill ECG) is an advanced ECG, helpful during the condition when normal ECG can’t provide sufficient information.

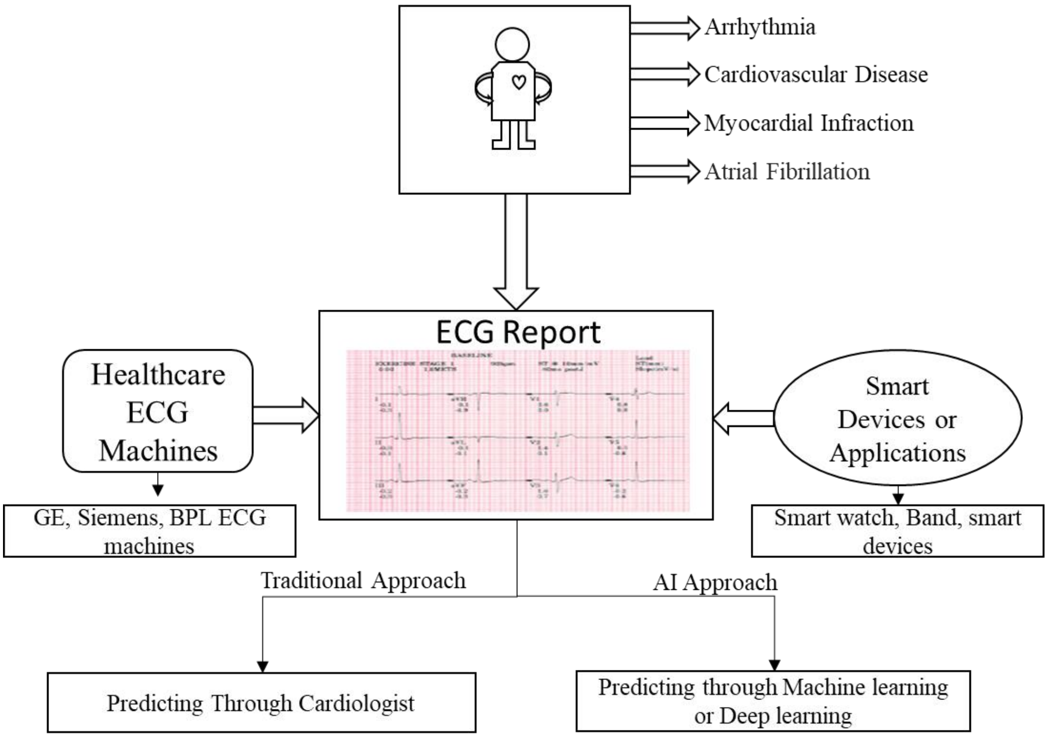

Arrhythmia
Cardiovascular Disease
Myocardial Infraction
Atrial Fibrillation
ECG Report
Healthcare ECG Machines
GE, Siemens, BPL ECG machines
Smart Devices or Applications
Smart watch, Band, smart devices
Traditional Approach
AI Approach
Predicting Through Cardiologist
Predicting through Machine learning or Deep learning

Figure 2.2 Overall literature study layout focused on heart related issues

Analysis of ECG/ TMT-ECG can be performed either based on the traditional approach or based on Artificial intelligence approach (AI). The traditional approach is based on time or frequency-domain. (Fariha et al. 2020; Pan and Tompkins 1985) explained regarding classification ECG rhythms as normal or abnormal offered a modified Pan-Tompkins based adaptive thresholding technique and (Ebrahimzadeh and Pooyan 2011; Gupta and Mittal 2021) explained discrete wavelet transform (DWT) is used to categorize ECG using Principle Component Analysis (PCA).

Classification of abnormality-based experts votes and handcrafted features related study as listed in Table 2.3. Various irregularities of ECG behavior which are not rectified through the traditional approach can be solved through AI approach. (Pławiak 2018; Sannino and De Pietro 2018) discussed related to modern deep learning models are designed to train from end to end, is difficult to combine with traditional expert feature-based approaches. There are two main approaches for addressing this issue. The first one is to develop DNN architectures using current expert knowledge (Khatibi and Rabinezhadsadatmahaleh 2019). The second approach is considering deep learning models as feature extractors and directly extract hidden encoded information. As a result, expert features and deep features may be simply combined, help to build classic machine learning methods on such features. The Table 2.4 describes machine learning approaches discussed in (Celin and Vasanth 2018; Haq et al. 2018; Khatibi and Rabinezhadsadatmahaleh 2019) like decision tree algorithm, K-nearest neighbours, artificial neural networks, etc. These results refer to the compatibility of the approach compare to previous traditional-based approaches. Periodical monitoring will be easier with the application of the AI approach using smart devices and mobile applications as discussed in Table 2.2 with respect to cardiologist analysis.

Table 2.2 Prediction of heart disease based on different feature parameters and smart approaches

Sl.No.	Author & Year	Consideration	Disease	Method	Features	Accuracy
1	(Marston et al. 2019)	ECG	Arrhythmia	CNN, LSTM, RNN, Gated RN, Deep belief network (Smart device)	P wave QRS complex	99.3%
2	(Haverkamp et al. 2019)	144 Subject Lead I ECG (30 sec) & 94 patient 12 ECG	Arterial fibrillation	Smartphone ECG (anonymized data)	Lead-I 12 Lead(Standard ECG & smartphone ECG)	97% & 94%
3	(McManus 2019)	121 participants, 2 minutes recordings	Arterial fibrillation	Recorded iPhone 4s (PULSE-Smart)	Root Mean Square of successive R-R difference, Shanon entropy, point care plot	95%
4	(Kruger et al. 2019)	M-health device - 68 MIT-BIH -dataset - 48	Arterial fibrillation	Frequency dispersion metric (FDM) R-R Interval variability	M-health ECG feature reading device	M.Health shows good accuracy
5	(Acharya et al. 2014)	BioPAC TM equipment (10 CAD & 10 Normal)	CAD- Heart rate	Heart rates from ECG	Time domain, frequency domain, Non-linear Technique (SVM)	75%

Table 2.3 Prediction of heart disease based on different parameters and traditional approaches

Sl. No.	**Author & Year**	**Disease**	**Dataset**	**Parameters**	**Accuracy**
1	(Pławiak 2018)	Cardiac disorder	MIT-BIH Arrhythmia at single lead 29 patients	Selection of expert votes & sum	98.94%
2	(Clifford et al. 2017)	Artificial fibrillation	Physio-net 2017	XG boost, CNN, RNN approach with or without LASSO & LASSO& additional feature algorithm	**Avg 82%** **With 85.57%**
3	(Arafat et al. 2005)	Coronary Artery Disease	(ECG & TMT-ECG) University of Missouri	Fuzzy, probability & combined	Combined better than fuzzy & probability
4	(Sannino and De Pietro 2018)	Arrhythmia	MIT-BIH Arrhythmia from 115 patients	Extracting handcrafted features with SVM	99.83%
5	(Khatibi and Rabinezhadsadatmahaleh 2019)	Arrhythmia	MIT-BIH Arrhythmia (MIT DB)47 patients 48 recording	Features extraction with CNN, handcrafted features & classification with DT, SVM, RF	99.77%

Table 2.4 Prediction of heart disease based on different machine learning and AI approaches

Sl. No.	**Author & Year**	**Disease**	**Data set**	**Classifier**	**Accuracy**
1	(Haq et al. 2018)	Heart disease	Clevel and heart disease dataset 2016 (303 patients & 76 feature considered) (297&13)	LK, KNN, ANN, SUM, NB, DT	Avg-80% Accuracy Max accuracy-Sum-86%
2	(Pławiak 2018)	Cardiac disorder	MIT-BIH Arrythmia database for one lead of 29 patients	SVM & Novel Sum + genetic tracking	98.99%
3	(Khatibi and Rabinezhadsadatmahaleh 2019)	Arrhythmia	MIT-BIH Arrythmia	DT, SVM {random Forest, KNN, DNN}	99%
4	(Celin and Vasanth 2018)	Normal or abnormal	MIT-BIH Arrythmia	SVM, Ada boost, ANN & Naïve bases.	Avg-94%
5	(Qibin Zhao and Liqing Zhang 2005)	Heart rhythm types	MIT-BIH Arrythmia	Wavelet transformation & sum	99%

In the importance of early detection of cardiovascular diseases, (Jahmunah et al. 2019) listed as ECG is the most cost-efficient and least invasive of all heart disease detection procedures. It gives comprehensive information about various kinds of diseases that affect humans and the need to detect them early in order to prevent the possibility of fatality later on.

The diagnostic procedures are faster while at the same time retaining the accuracy of the diagnosis. (Amato et al. 2013) according to this, accuracy of the ECG is of critical importance and inaccurate ECGs can lead to a wrong diagnosis by doctors, leading to wrong treatment. In spite of this, (Li and Boulanger 2020) explains the ECG monitoring system is one of the most significant monitoring systems to diagnose cardiovascular diseases.

During critical illness conditions like arrhythmia (Vilcant and Zeltser 2019) and coronary artery disease (Miller et al. 2016), etc, interpretation of disease by healthcare experts is done based on normal ECG and TMT ECG (which is recorded based on Bruce protocol as explained in (Badawy and Muaidi 2019; Shaw et al. 2015)). (Vasudeva et al. 2020) explained ECGs signals which are printed on thermal paper have shorter shelf life due to ink evaporation, mishandling, etc. Data in the signal are extracted through the data extraction method for future reference and these data are used in training the intelligent systems to classify the abnormality. The image processing technique is one of the best methods to extract data from the graph and store it.

There are a variety of diagnostic tools that can be used to detect CAD, like electrocardiogram (ECG), exercise stress test (Treadmill test (TMT)), echocardiogram, pharmacological test, and Cardiac catheterization (angiogram). Cardiac catheterization is a time-consuming and invasive procedure requiring experts for inserting a catheter into the arteries to remove plaque formation. Electrocardiogram and Tread Mill Test (TMT-ECG) are the usual techniques for the initial examination of the heart. These will help doctors to assess cardiovascular problems, abnormal heartbeat and identify the requirement for cardiac catheterization.

The electrocardiogram (ECG) is one of the most widely used procedures for detecting problems related to the electrical, vascular, and muscle systems of the heart. The voltage (millivolts) versus time (millisecond) graph generated from the Electrocardiography equipment is known as an electrocardiogram. (Jovic et al. 2019; Kumar et al. 2017; Wang et al. 2020) discussed about the automated analysis allows for the detection of electrocardiogram results associated with pathological problems without human involvement. To do this, digital electrocardiograms must be processed to quantify the amplitudes (mV) and durations (ms) of waves and intervals.

A modular feature extraction procedure will do away with the access rights issues which are likely to be encountered while using ECG machines with digitally recorded ECG data. It also enhances the utility value of ECG machines which otherwise just generate ECG graphs, by supplementing with high sampling rate data extraction. Most of the ECG machines available in developing or underdeveloped countries do not allow sharing of the digital form of ECG data. In the voltage versus time graph obtained from the Electrocardiography, automated analysis allows for the identification of ECG findings and assessment of pathological anomalies without the need for human involvement. (Badilini et al. 2003) findings, digitization through image processing is one of the methods to store ECG and TMT ECG data. The pre-processing of an image is primarily determined based on pixel information. The image's pixel dimension is determined by the size of the pixels array. The height and width of an image are determined by the array's number of rows (N) and columns (M). M X N is the size of the pixel array. The fundamentals of image processing for digitization are image segmentation, color correction, point, line, and edge detection, aid in the region-dependent threshold system, the removal of paper noise with a filter, morphological image processing operations for image dilation, erosion, and pixel indexing to store the data.

ECG analysis utilizing time-domain characteristics is challenging due to noise and the lack of knowledge of healthcare professionals. An intelligence system with effective signal pre-processing techniques before feature extraction is necessary to extract time-domain properties from ECG successfully as mentioned in (Ebrahimzadeh and Pooyan

2011). (Jian et al. 2021; Martis et al. 2014) expressed generally time-domain properties of ECG signals do not provide crucial information on non-linear interrelationships. A efficient computer-aided prediction system helps to recover non-linear features, which handle the inherent complexity of the time series and the non-linear behavior of ECG signals.

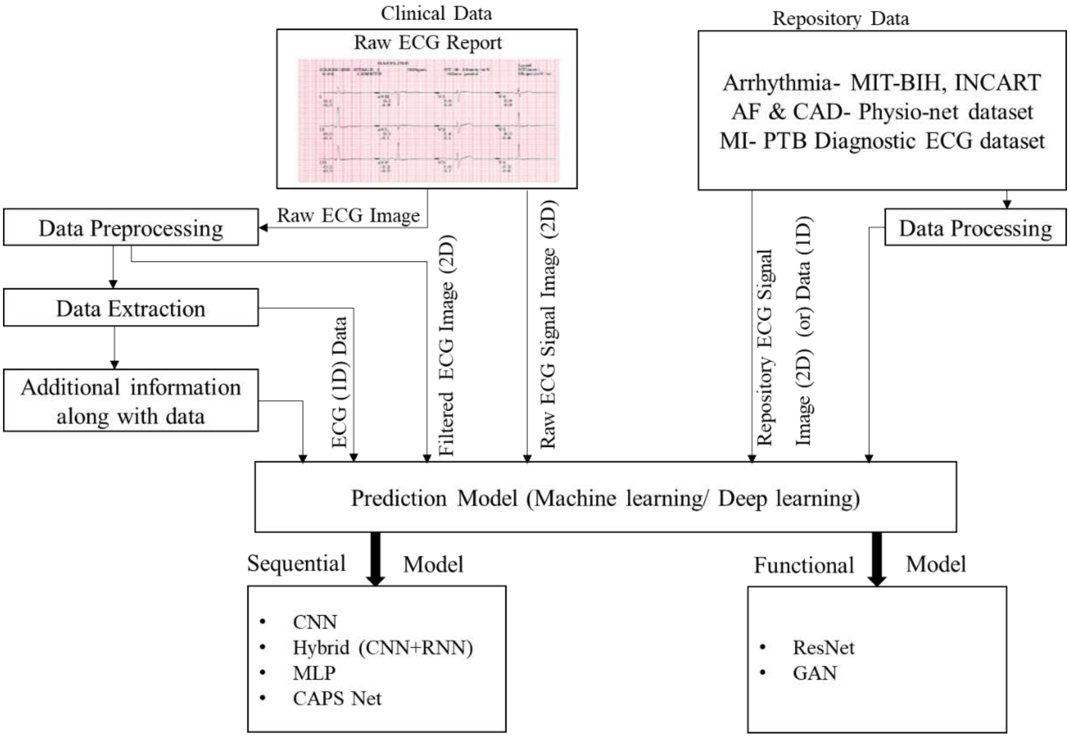

Clinical Data
Raw ECG Report
Repository Data
Arrhythmia- MIT-BIH, INCART
AF & CAD- Physio-net dataset
MI- PTB Diagnostic ECG dataset
Data Preprocessing
Raw ECG Image
Data Processing
Data Extraction
Additional information along with data
ECG (1D) Data
Filtered ECG Image (2D)
Raw ECG Signal Image (2D)
Repository ECG Signal Image (2D) (or) Data (1D)
Prediction Model (Machine learning/ Deep learning)
Sequential Model
Functional Model
CNN
Hybrid (CNN+RNN)
MLP
CAPS Net
ResNet
GAN

Figure 2.3 Flow chart of literature study focused on digitisation and prediction models

2.4 ECG DIGITIZATION

The computer vision method as explained by (Khleaf et al. 2015) to extract the ECG scan's paper is helpful to find data that used to predict arrhythmia disease with the help of the artificial intelligence system. Data extracting accuracy is good when compared with the manual extracting method. (Ravichandran et al. 2013) mentioned scanned image with OCR (Optical Character Recognition) digitization technique (Table 2.5). It uses the threshold method to eliminate gridlines and the median filter to remove noise before converting the data to a 1D vector. (Virgin and Baskar 2018) noted digital transformation index in the form of matrices is used to turn images into data using image processing techniques with the aid of Matlab. The matrix is then transformed into digital data, which is a copy of the ECG picture input.

Fabio Badilini (Badilini et al. 2005) carried out digitization by adopting digital ECG technology active contour modelling. ECG Scan was validated using a set of 60 ECG graphs. Validation is done based on the PQRS complex with the data scan from the scanner and the digitized data. Further, Deepak Kumar (Garg 2012) worked on digitization based on pixel interpolation. The scanned image is processed through various image processing tools. Then the pixel is interpolated through the column. The method involves a calculation of Heart rate, QRS Width, and Stability (variation in R-R peaks) from the extracted signal. The result is compared with the manually calculated parameter showing an accuracy of 96.4%. A fuzzy system was developed, from digital image processing, and color corrections, and the filter was used with the help of experience. Image segmentation is based on color, point, line, and edge detection. (Virgin and Baskar 2018) discuss on region-based thresholding, Removal of the noise with the help of the filter. Morphological image processing for the dilation of the image, erosion, and Hit-or-miss transformation. Matlab to convert images to (x, y) data by using image processing techniques. The fine details of the image are extracted and the equivalent digital information is indexed in the form of matrices. This matrix is converted into digital data which is a fine replica of the input ECG image. The digitized ECG data is saved as a (.dat) file.

The Tompkins Algorithm (Jambukia et al. 2015) was created in 1985-1986 to find features such as QRS complex using the low pass and high pass filters, squaring signal, integration, and adaptive thresholding with 95% accuracy. (Luukka and Lampinen 2010) adopt the common approach to obtain features of these ECG signals using Principal Component Analysis (PCA), while (Li et al. 2017) discussed about the wavelet packet decomposition (WPD) method combined with statistical methods can yield better outcomes. Digitization tool as explained (Badilini et al. 2005; Garg 2012; Sassi et al. 2017) is used to digitize ECG graphs and measure features such as heart rate, QRS distance, R-R peak variance, and PQRS complex with 90% accuracy.

Table 2.5 Parameters and techniques adopted for digitisation ECG.

Sl. No.	Author & Year	Disease Data	Consideration	Technique	Scanning	Correlation
1	(Tabassum and Ahmad 2020)	Arrhythmia Dataset	MIT-BIH scanned ECG strip	Matlab,12 Lead, ECG paper record	300 dpi	98%
2	(Ravichandran et al. 2013)	New data	GE health care. Finland (Muse cardiology system)	OCR(Matlab) {Optical Character Recognition}	300 dpi	85.90%
3	(Sharma et al. 2012)	Arrhythmia Dataset	MIT-BIH	Image threshold Technique	300 dpi	98%
4	(Baydoun et al. 2019)	Cardio Vascular disease	American University of Beirut Medical Center	Image processing	300 dpi	95%
5	(Badilini et al. 2005)	Heart disease	Paper printouts	Contour modeling	300 dpi	Good
6	(Sassi et al. 2017)	Heart disease	Paper printouts	Proof-of-concept test	300 dpi	Electronic health record and pdf-ECG have good correlation
7	(Khleaf et al. 2015)	Arrhythmia Dataset	MIT-BIH	Image processing technique	200dpi	96-98%

A real-time QRS detection algorithm explained (Hamilton and Tompkins 1986) uses a digital filtering approach, with a few modifications from earlier research by Pan and Tompkins (Pan and Tompkins 1985). The algorithm consists of linear, nonlinear filters and decision rules. It also includes low pass filter, high pass filter, derivative, and moving window integration with linear filter whereas squaring block is a nonlinear filter.

The ECG signal has to be interpreted in terms of specific quantitative features, i.e. P: arterial systole contraction Pulse, R is the peak of the ventricular contraction, S refers to downward deflection immediately after the ventricular contraction, Q is the downward deflection immediately preceding the ventricular contraction, and T defines the recovery of the ventricles. The method is defined in terms of locating the P, Q, R, S, and T waves in electrocardiograms, which are then measured with the R-peak (Fan et al. 2020; Gupta et al. 2021; Gupta and Mittal 2021), PR, and R-R segments (Akula and Mohamed 2019; Ghiasi et al. 2017). The ECG processing system discussed (Mathews et al. 2018) is capable of reading a wide range of pathological situations, including P, Q, R, S, T wave directions, T wave greater than R wave, and absence of Q wave in amplitude.

Since interpreting variations of ECG or TMT-ECG manually is difficult, a computer-aided diagnostic system may assist in cardiac health monitoring. Because of its nonlinear nature, (Cairns et al. 2016; Subbiah and Patro 2015) explains the nonlinear extraction approach which is ideally suited for extracting information from the ECG signal. Based on the wavelet transformation method (Al-ani 2014; Yochum et al. 2016), the QRS complex and QRS-T segments (30ms before 240ms after QRS position) are automatically detected. Thus, the selection of the best method delivers effective digitized ECG signal data by implementing the state-of-the-art technique.

Several researchers have focused on CAD to improve diagnostic tool efficiency by using either normal ECG or exercised ECG recordings. (OH et al. 2017; U.R., Acharya, sree, vinitha, swapna 2014) discussed on ECG signal analysis in normal and CAD people with AI, linear and non-linear approaches. But, The expert cardiologist analyses (Price et al. 2010; Yildirim et al. 2019) ECG signal patterns for medication.

Cardiologists examine time-domain elements(ECG features) such as T and Q wave amplitudes, and ST level segment (Li and Boulanger 2020) fluctuations as indicators of heart illness during traditional medical conditions.

In the past twenty years, a huge number of intelligent systems have evolved related to analyzing these ECG and TMT-ECG signals of the heart to assist health care experts. These intelligent systems developed to analyze ECG signals are primarily based on single-lead (Mathews et al. 2018; Rubin et al. 2018) and 12-lead ECG signals (Chen et al. 2020; Donida Labati et al. 2018; Fan et al. 2020; Liang et al. 2020; Ribeiro et al. 2020; Sassi et al. 2017). Several neural networks and machine learning algorithms are discussed to analyze the ECG or TMT-ECG. Extension of neural network called deep neural network helps to optimize the accuracy of the neural network. The major deep neural networks applied to obtain optimized solution are CNN (B. et al. 2019; Schmidhuber 2015), Recurrent neural network (RNN) (Ebrahimi et al. 2020; Swapna et al. 2018), Deep belief network (DBN) (Ebrahimi et al. 2020; Mathews et al. 2018), fully connected neural network (FC) (Fan et al. 2020) to predict the electrocardiogram (ECG) related issues like arrhythmia (Jiang and Kong 2007; Marston et al. 2019; Rahhal et al. 2016; Ullah et al. 2020; Wu et al. 2021) (Table 2.6), atrial fibrillation (AF) (Acharya et al. 2017b; Haverkamp et al. 2019; Li et al. 2019) (Table 2.7) myocardial infarction (MI) (Hao et al. 2020; jian, j.-z.;Ger, T.-R.;Lai, H.-.;Ku, C.-m.;Chen 2021) (Table 2.8), ST-elevation (Mishra et al. 2021), coronary artery disease (CAD) (Kurt et al. 2008; OH et al. 2017; Russo 2002; U.R., Acharya, sree, vinitha, swapna 2014) (Table 2.9), etc.

Table 2.6 Prediction of arrythmia disease based on different techniques and different parameters

Sl. No.	Author & Year	Arrhythmia Disease analysis	Data	Parameters	Technique	Accuracy
1	(Acharya et al. 2017c)	Arrhythmia	MIT-BIH	2’s & 5’s ECG data (without QRS)	CNN	94.9%
2	(Acharya et al. 2017d)	Classify heartbeats	MIT-BIH	Lead II ECG signal	DCNN	94.05% & 93.4%
3	(Oh et al. 2018)	Arrhythmia	MIT-BIH	Lead II ECG signal	CNN +LSTM	98.107%
4	(Ullah et al. 2020)	Arrhythmia	MIT-BIH	Converting 1D to 2D and used	Deep CNN	99.11%
5	(Xu et al. 2019)	Arrhythmia	MIT-BIH	ECG load II +handcraft ECG features	DNN	99%
6	(Khatibi and Rabinezhadsadatmahaleh 2019)	Arrhythmia	MIT-BIH	ECG head II handcraft + feature extraction algorithm	Feature engineering, comparison of deep learning and KNN	99.7%
7	(Sannino and De Pietro 2018)	Arrhythmia-heart beat classification	MIT-BIH	24-hour ECG from 32 to 89 year age group	DNP	99%
8	(Yildirim et al. 2019)	Arrhythmia-heart beat classification	Lead II ECG MIT-BIH	Auto encoder+ LSTM	CAD-LSTM	99%

Table 2.7 Prediction of coronary artery disease (CAD) based on different techniques and different parameters

Sl. No.	**Author & Year**	**CAD Analysis**	**Data**	**Parameters**	**Technique**	**Accuracy**
1	(Tan et al. 2018)	CAD	Physio-net database (7CAD,40 Normal)	Lead II, ECG Segments,5Sec data. Blindfold C/S validation	8 layered stacked (CNN-LSTM)	99.8%
2	(Kumar et al. 2017)	CAD	Physio-net database (7CAD,40 Normal)	(Data filtered) Flexible analytic wavelet transform to fold cross validation	Least square SVM	99.6%
3	(OH et al. 2017)	CAD	Physio-net database (7CAD,40 Normal)	Higher order statistics & Spectra (HOS)	Feature extracted PCA KNN & DT classifier	KNN-98.17% DT- 98.99%
4	(Acharya et al. 2017a)	CAD	Physio-net database (7 CAD,40 Normal)	2Sec-Data, 5 Sec- Data ECG + (age, sex, condition diabetes, blood pressure, mental state)	Deep CNN	94.95% (2 Sec) 95.11% (5 Sec)
5	(Acharya et al. 2014)	CAD- Heart rate	BioPAC TM equipment (10 CAD & 10 Normal)	Heart rates from ECG	Time domain, frequency domain, Non-linear Technique (SVM)	75%

6	(Babaoglu et al. 2009)	CAD & Lesion localization	330 patients (health care)	(Exercise stress Testing) TMT (EST)& coronary angiography	ANN (MLD)	91-65% (Left coronary, left anterior, left circumflex)
7	(Kurt et al. 2008)	CAD	1245 ECG	TMT or ECG, including age, sex, family history, smoking, diabetes, hypertension etc.	ML & ANN (LR, CART, MLD, RBF)	78%
8	(Fathima and Vimina 2022)	CAD	Statlog & Cleveland dataset UCI Data repository	ECG	DNN	98.77%-statlog 96.7-Cleveland

Table 2.8 Prediction of myocardial infarction (MI) disease based on different techniques and different parameters

Sl. No	**Author & Year**	**Disease**	**Data**	**Consideration**	**Technique**	**Accuracy**
1	(Jian et al. 2021)	MI	PTB-12 lead ECG database	12- Lead ECG	N-Net network (Feature Concentrate Network)	95.76%
2	(Han and Shi 2020)	MI	PTB Dataset (312 records of MI 80 normal)	12- Lead ECG	ML-Res Net	99.92%
3	(Hao et al. 2020)	MI	Zhejiang second people's hospital at China. (483 MI & 474 Non-MI)	12- Lead ECG	Multi-branch fusion network	94.72%
4	(Baloglu et al. 2019)	MI	PTB (483 MI & 474 Non-MI)	12- Lead ECG	Deep CNN	99%
5	(Mirza et al. 2022)	MI	PTB-ECG Physio-net	(148 MI & 52 normal)	1D – Deep Neural network	99.98%

Table 2.9 Prediction of atrial fibrillation (AF) disease based on different techniques and different parameters

Sl. No.	**Author & Year**	**Disease**	**Data**	**Consideration**	**Technique**	**Accuracy**
1	(Wang et al. 2021)	AF	MIT-BIH AF Database & FZU-FPH data	PDF of ECG waveform	Hybrid model, IQPSO-SVM algorithm	99.2%
2	(Kruger et al. 2019)	AF	M-health device -68 MIT-BIH -dataset -48	M-health ECG reading device	Frequency dispersion metric (FDM) R-R Interval variability	M.Health shows good accuracy
3	(Acharya et al. 2017c)	AF	MIT-BIH	2 Sec & 5 Sec ECG	II-layer deep CNN	92.5% for 2s 94.9% for 5s
4	(Wang et al. 2020)	AF	MIT-BIH	10 Sec ECG	ML Technology	98.8%
5	(Haverkamp et al. 2019)	AF	ECG for smart Phone (144 subjects)	Single Lead ECG (Lead I)	Comparing QRs +QT interval variation	97%

2.5 APPLICATION NEURAL NETWORK ON HEART DISEASE

An artificial neural network is a network which in a way simulates the functioning of human brain. The neural network model receives digitized data of ECG and TMT-ECG signals and generates proper recommendations as output. (Maniruzzaman et al. 2020; Sannino and De Pietro 2018; Xu et al. 2019) discussed about basic features (age, QRS complex, ST segment drop, etc.) in addition to input data lead to improved performance of the neural network. The deep neural network is an automatic learning process consisting of multiple layers and activation functions from the lower level to higher-level representation data. The automated convolutional neural network is a version of the deep neural network applied for the prediction of medical diseases. CNN models have been developed to analyze ECG signals to detect arrhythmia condition (Rahhal et al. 2016; Ullah et al. 2020), detects intervals of ECG segment (Acharya et al. 2017b), AF (atrial fibrillation) (Wang et al. 2020), and classes of ECG signals (Yildirim et al. 2018). With using 4 layers (Murat et al. 2020; Rajeswari et al. 2012), 11 layers architecture (Acharya et al. 2017b), 16 layers (Yildirim et al. 2018) architecture, and 34 layers (Fan et al. 2020) layer architecture respectively. The performance of any convolution neural network is assessed by the nature of data. The CNNs find wide applications in the area of image processing (Al-ani 2014), object recognition (Oquab et al. 2015), and handwriting data classification (Cireşan et al. 2011). In ECG-related analysis, the CNNs are mostly designed for arrhythmia, AF, and MI diseases under MIT-BIH (Alarsan and Younes 2019; Kruger et al. 2019; Liang et al. 2020) dataset.

CNN (Convolutional Neural network) is a popular form of deep neural network (DNN) that is commonly used in nonlinear problems. The CNN model effectively deals with the sound recordings of normal and abnormal heartbeats (82% accuracy). The use of CNN has been extended to the field of detection of arrhythmia (79% accuracy) (Andreotti et al. 2017) and atrial fibrillation (80% accuracy) (Ghiasi et al. 2017). CNNs are effectively applied for automatic CAD recognition, and (Alzubaidi et al. 2021) proposed that CNNs remain powerful despite moving and scaling invariance, making them advantageous.

Multi-perceptron neural network (Lehtinen et al. 1998) for coronary artery disease classification was evaluated on 347 people, 127 with confirmed CAD and 220 without CAD. The data was gathered using an exercise stress test that focused on ST segment depression, ST segment depression/Heart rate index, and ST/HR hysteresis of leads I, II, III, aVF, V2, V3, V4, V5, and V6 and have given great end result with a 91.5% accuracy. The (Babaoglu et al. 2010) objective of the research is to assess a radial basis function neural network as a tool for coronary artery disease diagnosis using the data of a standard ECG and exercise stress test (TMT-ECG). Each record describes the patient's condition and provided input data for the neural network, which included the level and slope of an ST segment of a 12-lead ECG signal recorded at rest and after exercise. Together with heart rate, blood pressure, load during the test, and the occurring of coronary pain, coronary arteriography results.

(Arafat et al. 2005) dealt on (Table 2.10) the diagnosis of coronary artery disease with fuzzy and probabilistic signals as mixed uncertainty. The results are computed using simply fuzzy or probabilistic ECG stress signals. The developed model with combined uncertainty achieves better results. A sample of 15 patients was chosen from a database of ECG stress test patient data obtained at the University of Missouri Department of Diagnostic Cardiology between 2002 and 2004. Researchers (Kaveh and Chung 2013) presented an automated technique for classifying subjects with atherosclerosis using a single lead of ECG sensors data. This approach not only optimizes sensor construction, but it also automates classification to deal with data overload. The system exhibits high accuracy and diagnostic performance using the MIT-BIH database, demonstrating the clinical utility of this unique classification method. Using machine learning, specifically support vector machines (SVM), researchers proposed a method for automating ECG categorization for atherosclerosis and early CHD identification, with a classification accuracy of 88 percent. In another study (Çolak et al. 2008), developed eight different learning algorithms for creating artificial neural network (ANN) models, for coronary artery disease (CAD) prediction. The data was collected from 237 patients (114 NOCAD, 124 CAD) considering 17 variables sex, age, hypertension, blood pressure, etc., and achieved an accuracy of 81% as the best out of eight algorithms.

(Davari Dolatabadi et al. 2017) developed an optimized SVM classifier for CAD using heart rate variability (HRV), extracted from ECG. discuss on features are extracted from HRV in time, frequency, and non-linear domains. The data set chosen from the long-term ST segment database having 80 human subjects achieved an accuracy of 79.1%. (Lee et al. 2007) tried out a methodology to develop the multi-parametric characteristics, including linear and nonlinear features, of HRV (Heart Rate Variability) for cardiovascular disease. The data set includes ninety-nine with CAD and ninety-four patients with normal and accomplished it an accuracy of 90 percent patients in the automatic detection of normal and Coronary Artery Disease conditions using heart rate signals. The heart rate signals are decomposed into frequency sub-bands using Discrete Wavelet Transform (DWT). (Giri et al. 2013) proposed model results showed that the ICA coupled with GMM classifier combination resulted in the highest accuracy of 96.8%, the sensitivity of 100%, and the specificity of 93.7%. The data set consists of 10 CAD and 15 normal subjects.

Table 2.10 ANN and DNN Techniques adopted to develop the prediction model for CAD.

Sl. No.	**Author & Year**	**Database**	**Technique**	**Accuracy**
1	(Lehtinen et al. 1998)	Lead: 9 leads Subjects: 127 CAD, 220 Normal	Artificial neural network (ANN) (ST segment, ST/HR index, ST/HR hysteresis)	91.5%
2	(Arafat et al. 2005)	Lead: lead v5 Subjects: 8CAD, 7 Normal	Combined uncertainty (Fuzzy uncertainty, Probabilistic uncertainty) (ST segment, R-wave)	84%
3	(Kaveh and Chung 2013)	Lead: Lead II Subjects:43 CAD,49 Normal	(ST –segment depression, T –wave inversion) Classifier SVM	88%
4	(Çolak et al. 2008)	Subjects: 124 CAD, 114 Normal	Artificial Neural Network (ANN) 17 input variables (sex, age, hypertension, blood pressure etc.,)	81%
5	(Lee et al. 2007)	Lead: Lead II, Subjects: 99 CAD, 94 Normal.	Classifier SVM (Linear and non-linear features).	90%
6	(Giri et al. 2013)	HRV signals, Subjects: 10 CAD, 15 Normal.	DWT, PCA, Classifier SVM	79.1%
7	(Poddar et al. 2015)	Lead: lead II Subjects: 64 CAD, 60 NOCAD.	Linear and non-linear features. (R-R interval) PCA, KNN, SVM	91.67%

2.6 TRANSFER LEARNING APPROACH

Transfer learning is a part of the machine learning method and artificial intelligence approach. Application of this approach helps to gain knowledge (weights) from the existing trained model (source data trained model) to predict different but similar problems (target data prediction similar to source data). A portion of the ImageNet database is used to train the majority of the pre-trained networks (transfer learning method).

The proposed work focused on dealing with both 12 lead and single lead signal images with the inclusion of image pre-processing and classification of image signals. The work extended to study the behavior in noise signal conditions and pre-processed signal conditions, and are compared. In the literature, there are several well-developed image categorization prediction algorithms. To determine the work's novelty, the suggested image classification prediction algorithm is compared with respect to the accuracy of a well-developed pre-trained algorithm (VGG16, Inception, MobileNetV2, ResNet, EfficientNet) generated by transfer learning methods. In terms of training loss and accuracy, the outcomes of the pre-trained algorithm and the proposed approach are compared. These outcomes are derived with consideration of raw(noisy) signal images and pre-processed images, and results are developed based on both single and multi-lead (12-lead) signal images.

2.7 MULTI-HEADED MODEL APPROACH

The input data are considered hybrid data types that deal with 1D time series data and signal images of clinical data of single-lead ECG signals. The developed multi-head CNN model's ability is also tested by considering the repository dataset of the physionet. The configuration of a (Rai and Mitra 2021; Suresh et al. 2020) proposed multi-head convolution neural network model consists of two or more convolution layers and pooling layers in parallel connection. Depending on the input, either a single input shared with parallel sub-layers or a number of conditional inputs linked to the same issue and shared with parallel sub-layers are taken into consideration. This sublayer’s output is concatenated in series to obtain a long vector (more parameters).

These vectors are fed into a fully connected layer to get the final classification. These models perform better than single-head or sequential training models because they have handled more input heads and can learn higher-order features, which allow for greater flexibility in processing input data.

2.8 SUMMARY OF THE LITERATURE

2.8.1 Summary of Undifferentiated fever diseases study

The summary of the Undifferentiated disease literature states the application of machine learning algorithms to classify undifferentiated fever symptoms of diseases including tuberculosis, nontuberculosis, dengue, and non-infectious disorders. The study displays numerous research that use classifier-based artificial intelligence to detect or forecast specific diseases. Most of the classification algorithms are discussed for specific disease detection or classification. This classification methodology is adaptable for both extracted hospital data and repository data with consideration of various disease parameters.

2.8.2 Summary of heart diseases study

The electrocardiogram (ECG/EKG) is One of the most popular non-invasive diagnostic tools for recording the physiological processes of the heart over time. Many cardiovascular diseases, including atrial fibrillation (AF), premature contractions of the atria or ventricles (PVC), myocardial infarction (MI), coronary artery disease (CAD), and congestive heart failure, can be diagnosed using ECG data (CHF). A summary of the literature discusses the source databases, collection of data, number of leads considered for observation, traditional approach, AI-based approach, and pre-trained model approach.

2.8.2.1 Sources of data: In recent years, study shows that there is a huge development of portable ECG monitors and ECG machine in the medical field, and wearable or smart devices in various healthcare areas. Therefore, detection and analyzing ECG data/image automatically with accurate results will become a trending research topic. Datasets were extracted and collected from hospital medical ECG devices, In addition to this, some were from mobile healthcare devices. The difference between such devices is that medical device data is a more valuable and complex collection of data. Healthcare ECG monitor gadgets, such as smart devices, applications, and smart bands, are widely available and quickly record ECG data, which are less accurate than medical equipment. Additionally, there are a lot of emerging applications, such as biometric human identification and sleep staging, can be implemented based on ECG data.

There are some of the open-source datasets are available in repositories for analysis. Some of the well-known datasets are as discussed over study, such that (i) The MIT-BIH Arrhythmia Database, which contains ECG recordings of 47 patients from Beth Israel Deaconess Medical Centre patient. (ii) The Physio-Net Computing in Cardiology Challenge 2017 dataset contains ECG recordings sampled at 300 Hz by an AliveCor healthcare device with durations ranging from 9 s to 60 s. (iii) The PTB Diagnostic ECG Database, which contains ECG data from 12 lead-related 290 subjects. 216 of these people have one of the 8 major types of cardiac disorders, 52 have health control, and 22 are unknown. The MIT-BIH Atrial Fibrillation Database contains AF-related ECG recordings. Which were obtained from the Beth Israel Hospital in Boston.

2.8.2.2 Number of leads: A 12-lead ECG system can detect more abnormalities than a single-lead ECG system (like lead I of a 12-lead ECG). For example, posterior wall MI can only be detected by chest leads (V1 to V4), and a single lead will not detect this type of abnormality. AI-based approaches with a single lead and twelve lead ECG classification are helpful to detect and classify more abnormalities easily.

2.8.2.3 Duration: Short-term ECG data and long-term ECG data can be used for analysis. Short-term ECG data is less complex and may detect many heart disorders and the development of fast detection AI-based approach with moderate accuracy. Long-term ECG, on the other hand, can aid in the detection of disorders with intermittent symptoms, such as paroxysmal ventricular fibrillation (VF), and atrial fibrillation (AF) disease which are not predictable through short-term ECG.

2.8.2.4 Annotations: detection of ECG measurement annotations (peak R and end markers for P-, QRS-, T-, and U-waves), beat level annotations (PVC, etc.), and rhythm-level annotations (combination of both beat-level annotations and other diseases such as AF and VF) are difficult to predict orally. Derivation of Annotations using the oral approach requires a specialized field expert. This limitation can effectively be overcome through the AI-based approach.

2.8.2.5 Pre-processing: The authors deal with various types of pre-processing which can be done on the sample data plots available in the repository data set as well as clinical data types in order to filter the unwanted signal and also decompose the signal to get the exact ECG signal. That can be extracted to get the time series data and desired classified feature characteristics.

2.8.2.6 Prediction algorithms: These plays major role in the early detection of cardiovascular diseases. ECG is the most cost-efficient and least invasive of all the detection procedures. It gives comprehensive information about various kinds of diseases that affect humans and the need to detect them early in order to prevent the possibility of fatality later on. The main problem with the manual analysis of ECG signals, similar to many other time-series data, lies in the difficulty of detecting and categorizing different waveforms and morphologies in the signal. This is easily overcome with automatic prediction algorithms.

This can be achieved as two-step process, in which cardiology professionals derive the features based on raw ECG data, known as "expert features," and based on these, decision rules or other machine learning approaches to generate final findings are framed. Expert characteristics are classified as statistical features (such as heart rate variability (Ebrahimzadeh and Pooyan 2011), sample entropy (Acharya et al. 2014)), time-domain, and frequency domine features. In practice, expert characteristics are retrieved automatically by utilizing computer-based techniques. However, these are still insufficient because they are limited by data quality and human expert knowledge (Cairns et al. 2016; Hong et al. 2020). A literature study discussed different machine learning and Deep learning approaches that have yielded promising results in a wide range of applications in the identification of ECG diseases. The fundamental benefit of deep learning approaches is that no explicit feature extraction stage involves related to human expertise. Deep learning models are capable to do feature extraction automatically, implicitly based on their extensive data learning capabilities and flexible processing architectures.

2.9 RESEARCH GAP

Initial study of undifferentiated fever analysis, based on the temperature data is a unique study, which provided motivation for further research on coronary artery disease detection.

In India, heart disease has also recently become the leading factor of death in all parts of India. The cause for this is the growing use of cigarettes and inadequate fruit and vegetable diet. Symptoms of coronary heart disease include chest pain and breathing trouble, during physical activity. However, even people not exhibiting symptoms could still have a chance of heart disease. Screening helps to provide timely treatment in the early stages of the disease. This may reduce the risk of death and other complications later. Although screening and treatment assist persons against higher risk of cardiovascular disease, a routine examination might aid in the early detection of heart disorders in children and adults. Hence, scans like ECG/TMT-ECG are very crucial for the early detection of the disease. However, TMT-ECG scans are typically unaffordable to those living in rural areas. Moreover, the equipment for conducting it may not be available in hospitals in rural areas. This is one of the motivations behind this project.

Majority of ECG research use data from a publicly accessible archive. Archived data sets from the repository (MIT-BIH, Physio-net) are readily available with noise cancelation, features extracted, and with proper data format, unlike clinical lab data. For example, one of the CAD repository datasets available is the Physio-net dataset which is derived from 8 CAD patients and 40 Normal patients. But this is a very limited dataset and not sufficient to develop an end-to-end generalized CAD prediction model. Likewise, there are other research which concentrated ECGs either in time domain based or image-based analysis. These analyses are carried out based on consideration of single lead, specific leads, specific parameters and twelve leads. Major drawback of algorithms which are developed based on repository dataset are limited number of data set and failed in generalization ability. One-dimensional data analysis over limited number of datasets will end with either lower accuracy or the overfitting error. This is overcome with either larger clinical data or additional clinical parameter with available repository data consideration. The literature study discussed about work done on one dimensional and two dimensional individually, clinical data with limited number of trails and repository data. Major work carried out over literatures for detection of coronary artery disease based on clinical data with an average accuracy of 90% over limited dataset trails. There is no work related to CAD which carried out with large number of clinical trials, under consideration of one dimensional, two dimensional and hybrid analysis. This is one of the motivation to develop a novel study of automated hybrid multi headed model with consideration of both types of data with leading accuracy will helps to monitor CAD by healthcare experts.

The purpose of the present study is to develop a novel, accurate AI-based prediction system for CAD detection. The well-suited end to end CAD disease prediction with consideration of both time domain based (automatic feature extraction and traditional feature extraction) and image analysis based (Fully automatic) prediction model development for clinical data. Initially these clinical data are preprocessed and converted to one dimensional for time domine model analysis. The development neural model is unique in terms of exhaustive live patients' dataset, higher frequency of sampling of feature data and unique combination of activation functions across layers of the neural networks to derive optimized time domine based network, reflecting novelty. In parallel to this the other analysis which deals the images of ECG and TMT-ECG to obtain good prediction model for CAD diagnosis. There is a lag in research for deriving TMT-ECG based prediction model, stands as novel approach for development. Validate the model with pretrained (transfer learned) network provides a proof to the suitability. Further the study can extend to derive multi-headed model which can deal simultaneously both time domine signal data (1D) and image data (2D) analysis stands as a very unique study related to this type of disease.

Since there is no systematic reviews/study focusing on proper deep learning methods (for both 1D and 2D), the promising methods for ECG data mining or collection. Provides the motivation that it is crucial to conduct a systematic review of existing deep learning methods with single lead and multi lead for ECG/TMT-ECG data from the perspectives of CAD model architectures and application. The overcome of these challenges and problems related to the current research helps to develop a system which will assist the healthcare expert in accurately predicting the presence or absence of CAD among patients during periodic tests.

RESEARCH OBJECTIVES

1. Development of a Neural Network based prediction system for the identification or classification of unidentified fever symptomatic diseases.
2. Development of a novel strategy for automatic data extraction from ECG and TMT-ECG graph.
3. Developing a recommendation system for the identification, classification, and diagnostic assistance tool for coronary artery disease (CAD).
4. Validation of CAD prediction system results against angiography results.
5. Modular integration of different phases of DNN models into a reliable comprehensive diagnostic system.

CHAPTER 3

RESEARCH METHODOLOGY

Vast knowledge of domain experts can be embedded in the knowledge base of an intelligent system. Such knowledge can be used in future prediction of disease through well-established algorithms, so that medical experts can be supported with infusion of reliable diagnostic inputs, Of course the model's development. This regard needs to be data used are to be carefully preprocessed and results are thoroughly validated.

Accordingly, a neural network (NN) model is developed in the first stage, taking the case of fever for analysis for medical diagnosis. In the next stage, an intelligent prediction model (DNN) is developed for the detection of cardiac disease.

3.1 DEVELOPED ECG SYSTEM – AN EXPERIMENTAL SETUP OF ECG

ECG is a test helps to identify the cardiac abnormalities through electrical activity which is measured from heart. There are plenty of ECG measuring devices are available commercially. The device identified as either a single lead ECG or twelve lead ECG, based on the number of electrodes used for sensing cardiac signals while mounting on the human body. This can be achieved based on the multiple ways one is with Arduino Uno microcontroller, in addition AD8232 sensor with three Ag/AgCl electrodes (Kanani and Padole 2018), Agatsa (Sanket Life) ECG Sensor (Davalagi et al. 2020) which is commercially available, Healthcare center ECG machine and GE healthcare diagnostic ECG MAC 5500 HD machine. With consideration of all, the optimized method of ECG extraction was used for further process.

3.1.1 Arduino Uno microcontroller

The Arduino Uno is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino. Arduino Uno is used to control the M7112 sensor and also theAD8232 sensors. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 digital I/O pins (six capable of

PWM output), 6 analog I/O pins, and is programmable with the Arduino IDE (Integrated Development Environment), via a type B USB cable. It can be powered by the USB cable or by an external 9-volt battery, though it accepts voltages between 7 and 20 volts. The ATmega328 on the board comes pre-programmed with a bootloader that allows uploading new code to it without the use of an external hardware programmer. It uses the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.

3.1.2 AD8232 Sensor (3 lead ECG sensor)

The AD8232 ECG sensor is used to determine the electrical activity of the human heart. The ECG sensor's works similarly to an operational amplifier. Three AD8232 sensors in parallel were connected together to obtain ECG data. That is Ag/ AgCl electrodes consist of non-invasive stickers which can be attached to the body to extract the signals. The code is written in Arduino and the data is plotted using a serial monitor. This will helps replicate ECG and record the time interval data directly from the model. The signals are recorded from the sensor once the signal becomes stable. The three sensors (Figure 3.1 and Figure 3.2) are connected in parallel using a breadboard and jumper wires.

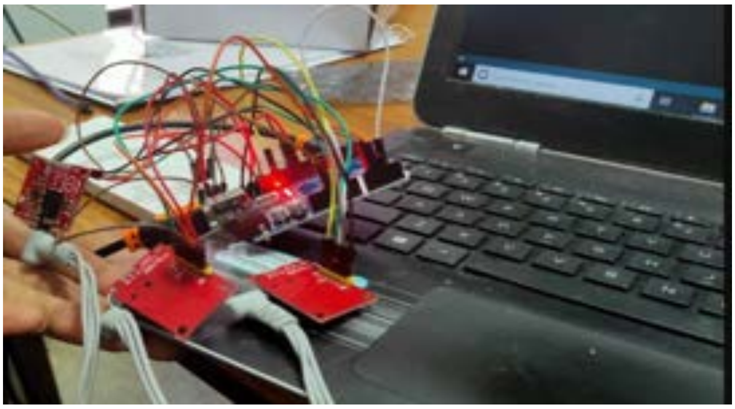


Figure 3.1 The three sensors connected in parallel

The script written in python extracts the data from the sensor and writes it into a .csv file which can be visualized using a plotting software like MATLAB and also through Python functions. The circuit design of the above setup as shown in Figure 3.2.

In the above circuit, the 3.3-volt connections of the Arduino and the three sensors are connected in parallel with the help of breadboard. The same is done for the ground connections. The outputs of the three sensors are connected to ports A1, A2 and A3 of the Arduino. The L0- is connected to ports 13,11 and 9 of the Arduino whereas L0+ is connected to ports 12,10 and 8.

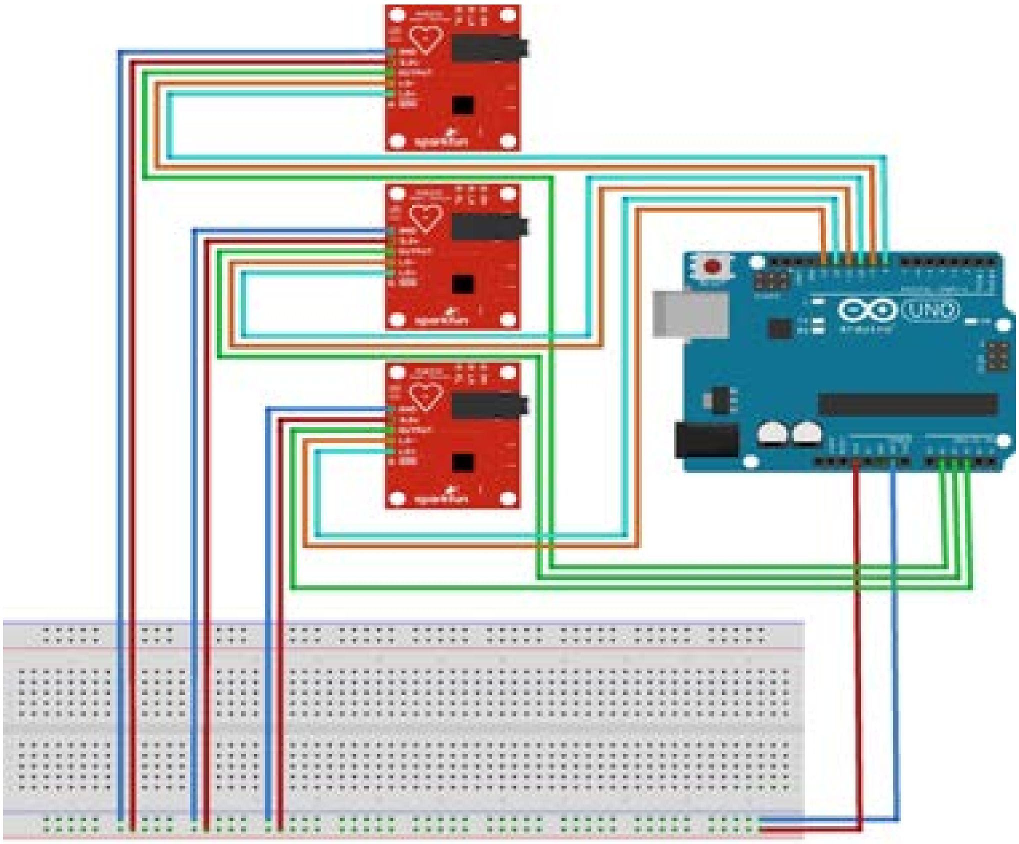


Figure 3.2 Circuit design for AD8232's in parallel using Arduino and breadboard

3.1.3 Plot obtained using single AD8232 sensor

The plot is obtained by connecting a single AD8232 sensor (Figure 3.3) to the setup containing an Arduino as microcontroller and the plot is obtained with the help of a python script which reads the data serially and plots it.

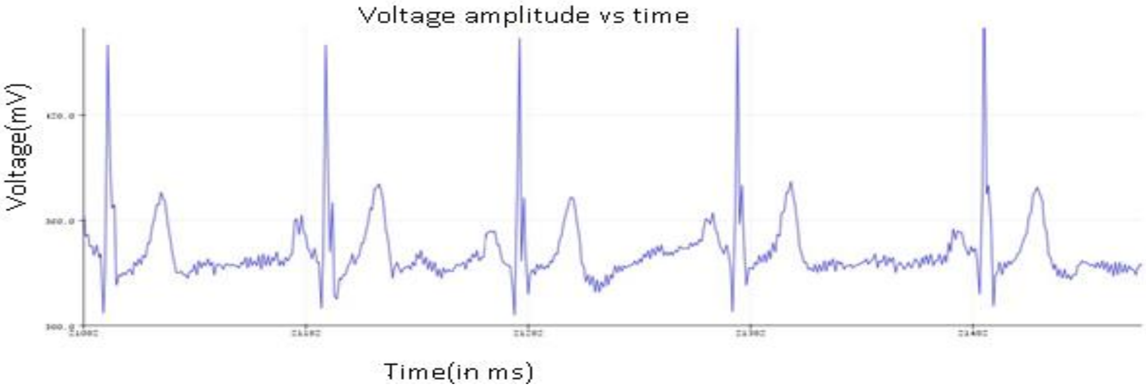


Figure 3.3 AD8232 single senor output

Here, three AD8232 sensors are connected in parallel. The set up consists of Arduino, breadboards, jumper wires and three sensors. The readings are obtained using the python script and plotted as shown in Figure 3.4. Further, this process extends to twelve lead ECG machine (GE MAC 5500HD) to gather the entire information related to heart.

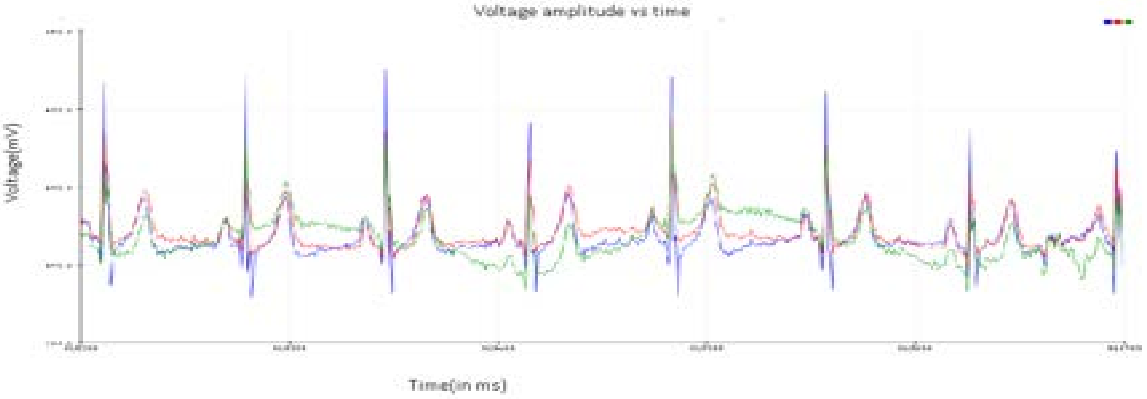


Figure 3.4 Output from three AD8232 sensors in parallel

3.2 AGATSA (SANKET LIFE) ECG SENSOR

The 12 lead ECG sensor by Agatsa can be used to obtain ECG signals through the built-in app and generate customized reports (Figure 3.5). Based on the reports, it is possible to obtain ECG data points. These ECG data points were used to validate the data in the customized Arduino based ECG sensor. The device doesn't provide information on disease screening.

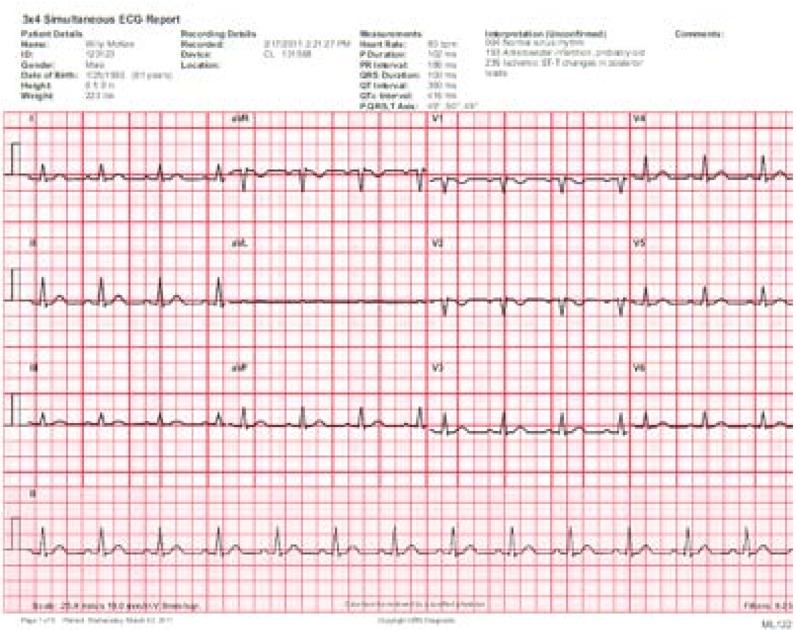


Figure 3.5 Agatsa ECG sensor report

3.3 GE MAC 5500HD Machine

The MAC 5500 HD Resting ECG is a high-end ECG system that provides advanced disease management capabilities via an industry-leading collection of algorithms and advanced networking. This is well known for Acute Coronary Syndrome (ACS) detection, accurate pacemaker detection for advanced interpretation of paced rhythms, and improved efficiency. Connected to treadmill will help to access the accurate TMT-ECG signals.

3.4 CLINICAL DATASET

Accurate and reliable clinical data is very vital to the medical team in timely diagnosis and treatment for patients. In many of the cases, number of health parameter and complexity of data relation may make the task of doctors and medical experts really challenging. The decision and effectiveness of the treatment depends purely on the expertise and depth of knowledge of the concerned team of doctors. So, an assisting tool having historic data for diagnosis, especially in critical situations will be highly beneficial, both medical expert as well as the patients. AI based prediction models fed with reliable clinical data can accurately give inferences.

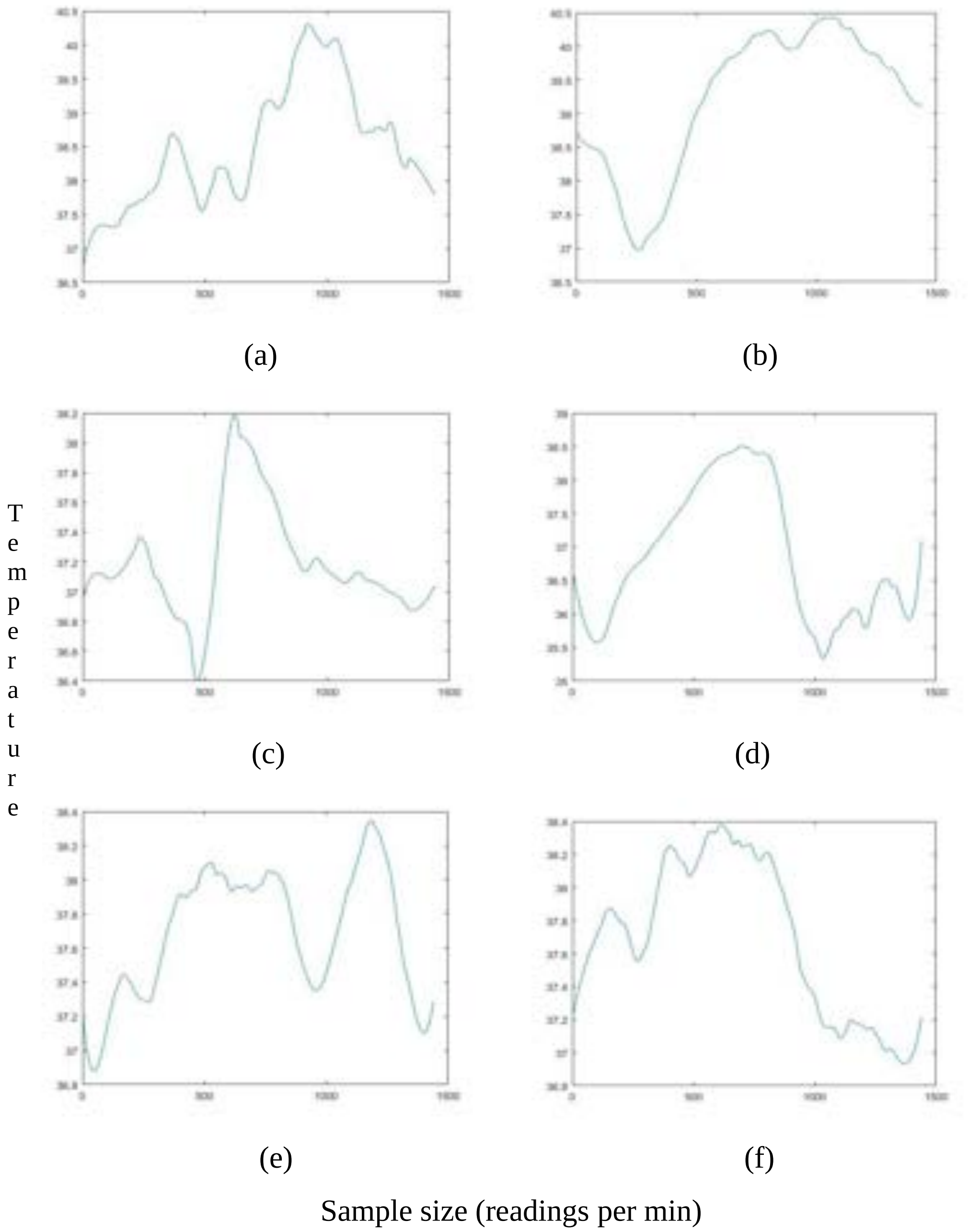
This work primarily deals with temperature data for classification using machine learning algorithm and heart related data (ECG, TMT-ECG and Angiogram) for rectify and predict the coronary artery disease.

3.4.1 Temperature data

To study the efficacy of AI models in disease prediction, symptomatic fever data is takes up for investigations. The relevant data are collected from a reputed neighboring hospital (KMC Mangalore). These data are extracted through temperature sensors over equal interval of time (every minute) for twenty-four hours. 103 patients suffering from dengue fever, tuberculosis, non-tuberculosis bacterial infection, and non-infectious disease are considered for the observation. Out of 103 patients, 28 patients belong to tuberculosis, 31 patients are non-tuberculosis bacterial infection, 16 are dengue, and 28 are Non-infectious diseases. Hence there are 1440 temperature data extracted from each patient (samples as shown in Figure 3.6 (a-f), 3.7 (a-f), 3.8 (a-f) and 3.9 (a-f) respectively) and in total 1440*103 data matrices are used in the prediction algorithm. These algorithms are further optimized with consideration of 30 minutes and 60 minutes of temperature data with feature data (ESR (Erythrocyte sedimentation rate), WBC (White blood cell count), Neutrophils, Basophils, Eosinophils, Monocytes, Lymphocytes, Platelets, Age, Body mass index (BMI), Spontaneous bacterial peritonitis (SBP), Diastolic blood pressure (DBP), Pulse) and without features data.

The fever temperature variation of all the four types of disease is plotted with respect to the time interval over a 24-hour duration and sample plots of all the diseases are shown in the following Figure:

3.4.1.1 Tuberculosis Diseases



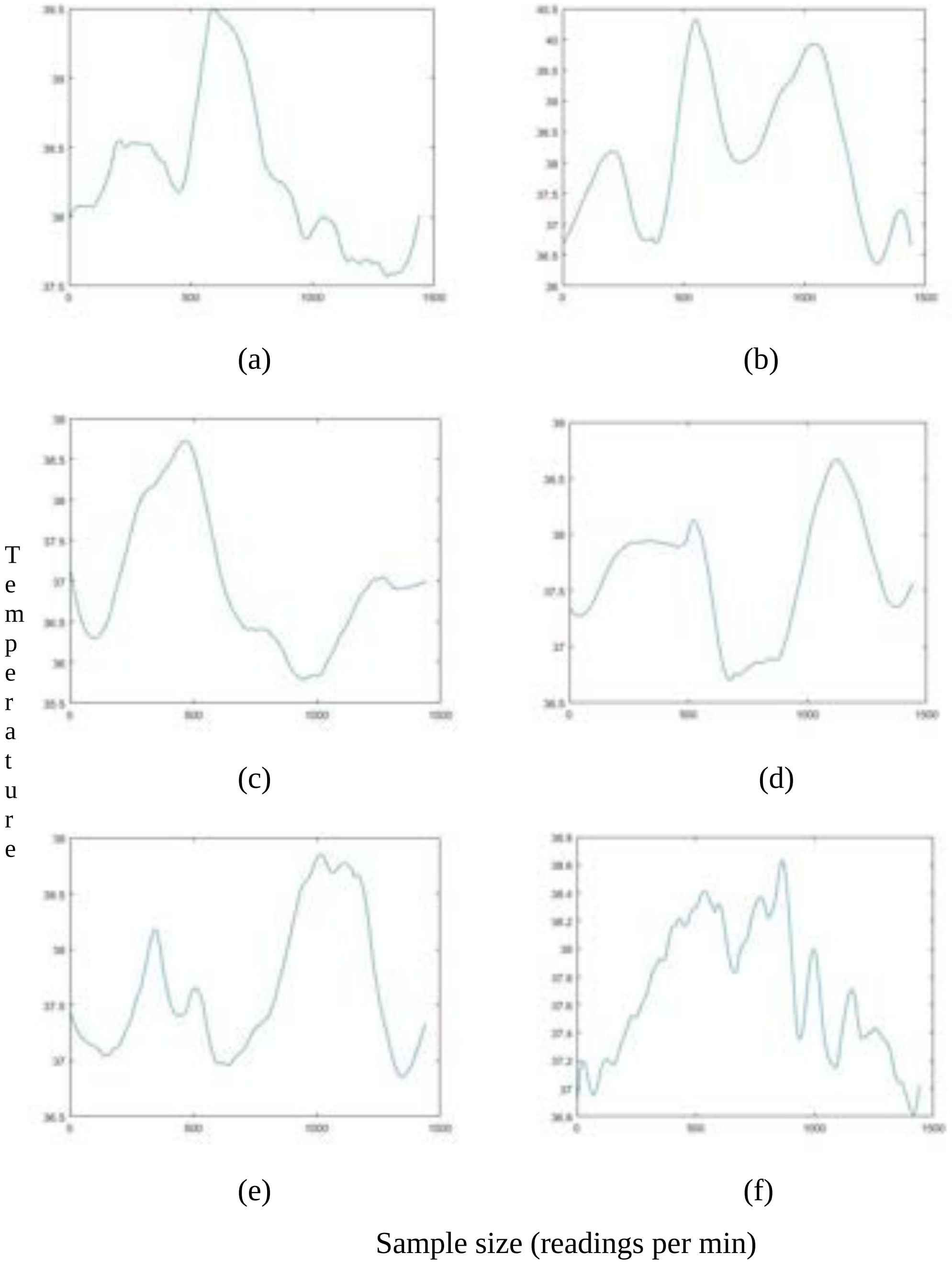
(a) (b)

(c) (d)

(e) (f)

Figure 3.6 (a-f) sample datasets temperature distribution pattern for Tuberculosis disease

3.4.1.2 Non-tubercular bacterial infection



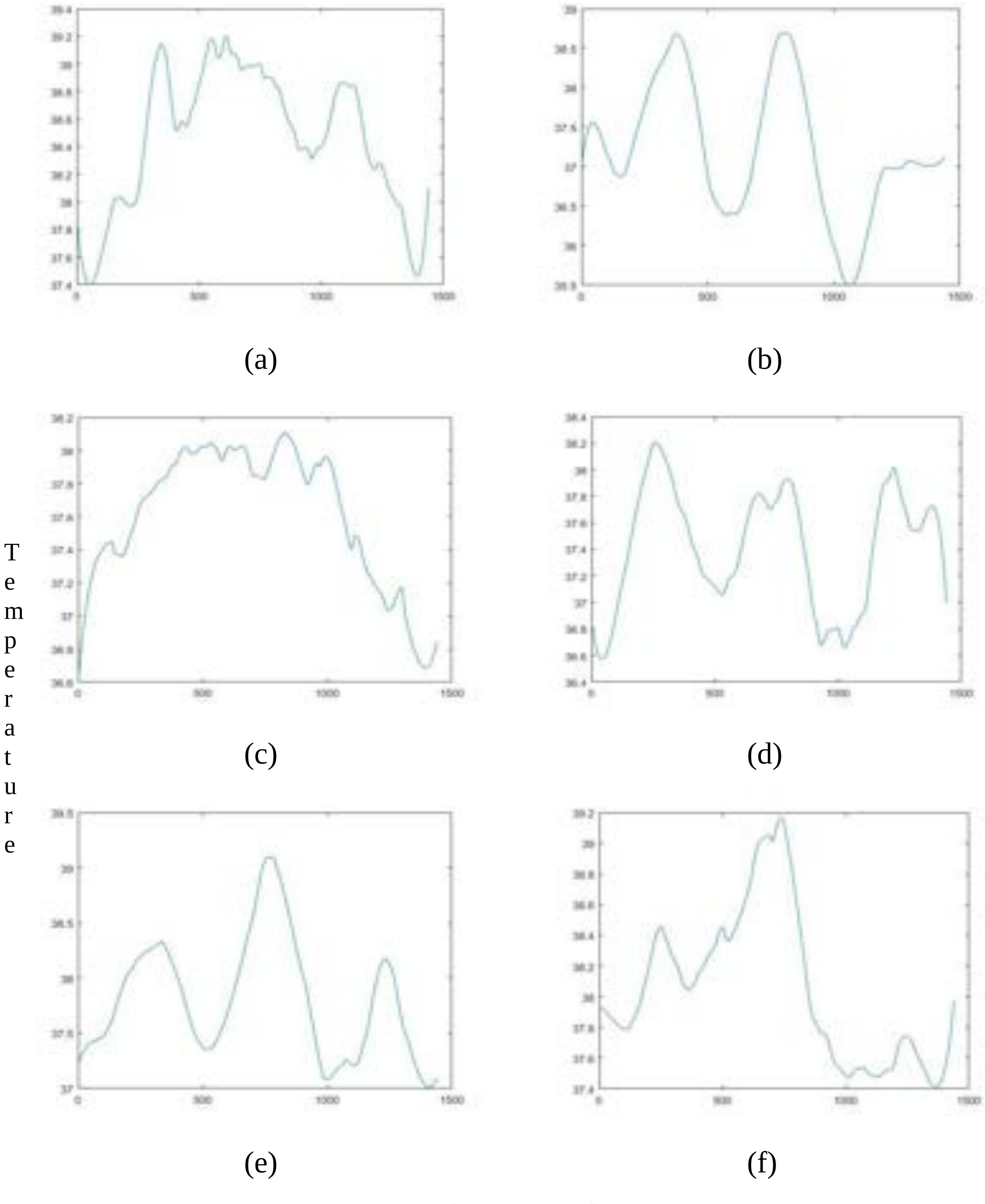
(a) (b)

(c) (d)

(e) (f)

Figure 3.7 (a-f) sample datasets temperature distribution pattern for Non-tubercular bacterial infections diseases

3.4.1.3 Dengue fever case



(a) (b)

(c) (d)

(e) (f)

Figure 3.8 (a-f) sample datasets temperature distribution pattern for dengue fever diseases.

3.4.1.4 Non-infectious diseases

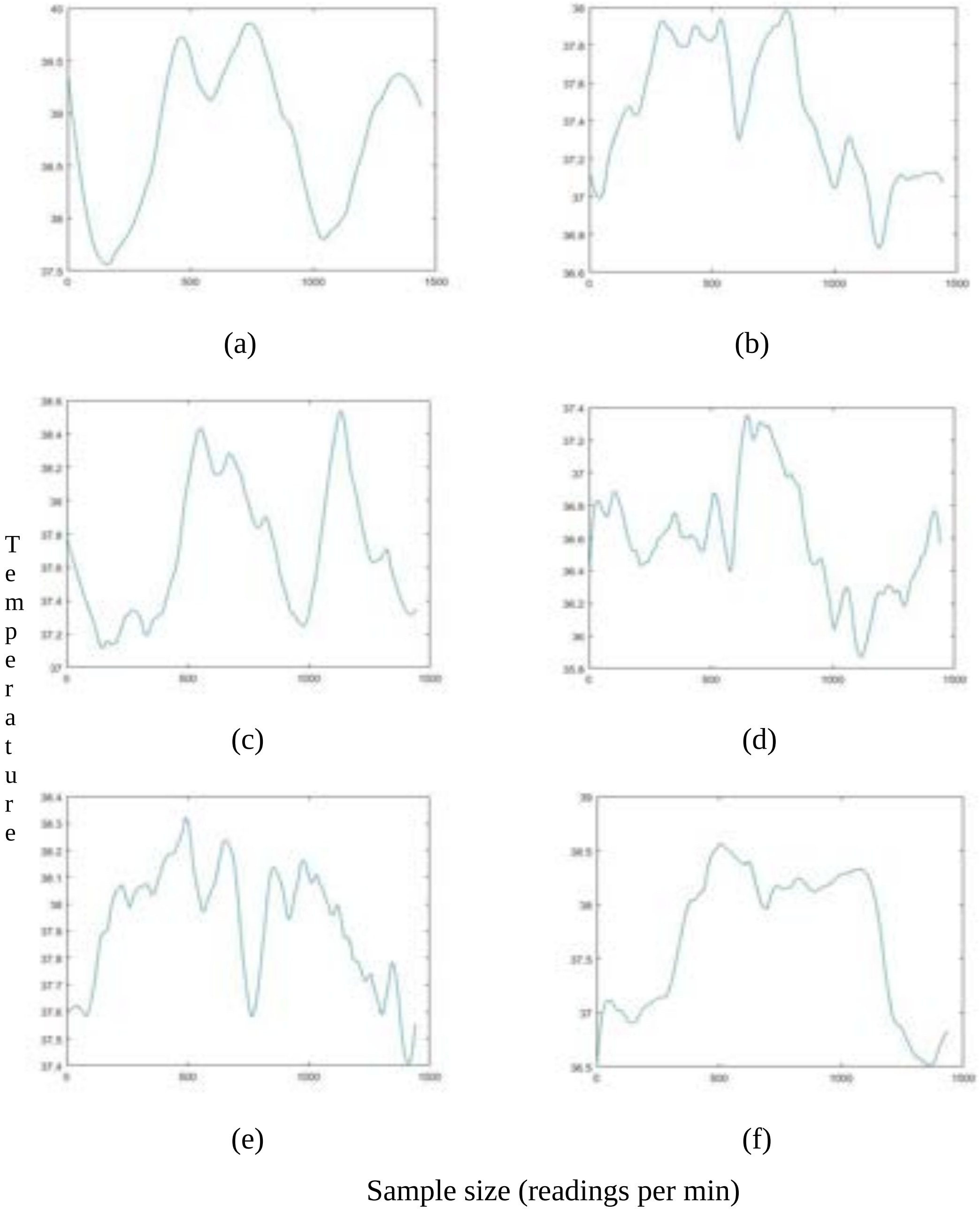


Figure 3.9 (a-f) sample datasets temperature distribution pattern for non-infectious diseases

3.4.2 ECG and TMT-ECG data

The research focused on the development of a prediction system, involving twelve lead ECG and TMT ECG signal images and single lead ECG and TMT ECG signal images. The study concentrated on people who lived in remote areas, and without considering any other parameters except ECG and TMT ECG signal images. The ECG information is gathered from a well-known nearby hospital that specializes in cardiac problems (KMC Mangalore, Karnataka, India). The data set includes horizontally scanned signal images of 236 healthy volunteers and 316 CAD patients (considering male and female patients) with angiography confirmation. A 12-lead ECG in resting state and a TMT ECG image (Figure 3.10) for normal and CAD participants are shown in Figure 3.11. The collected datasets are 1sec duration signal of each lead, will help to identify the disease in a short time. The analysis is carried out with consideration of raw signal images and filtered signal images; the raw signal images are the images that are directly obtained from scanned images without being subjected to any pre-processing methods. The pre-processed image signals are derived from a raw signal image with the application of the following pre-processing procedure.

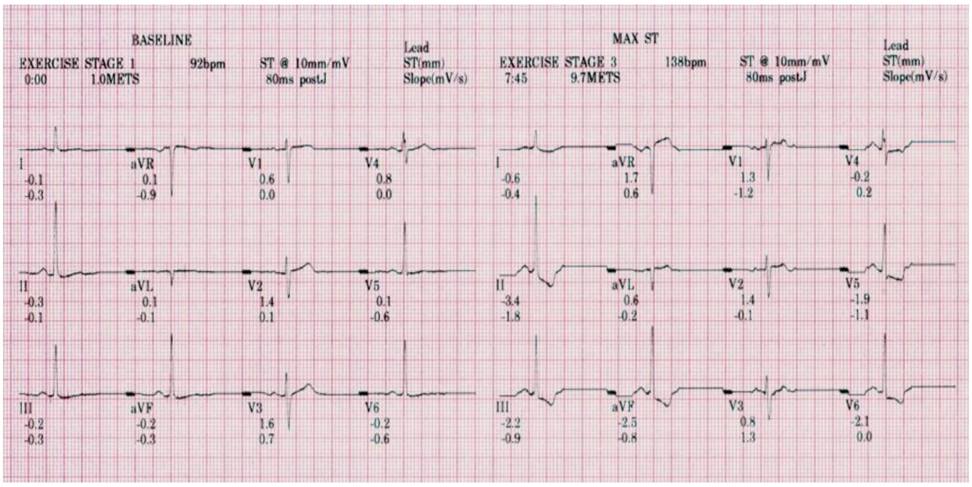


Figure 3.10 The scanned ECG signal of 12 lead signals

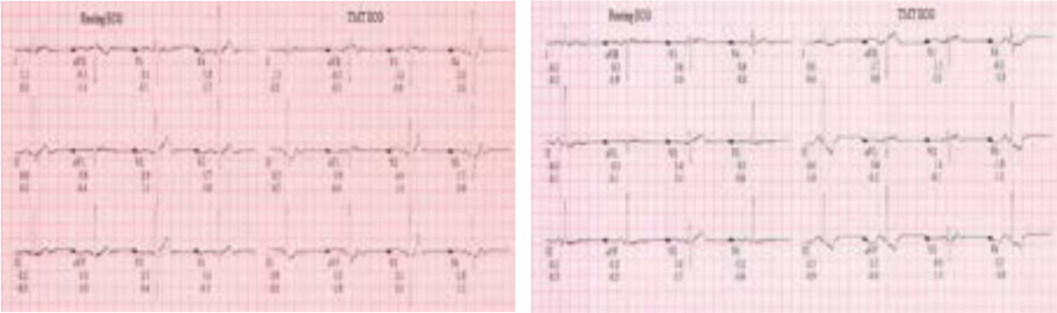


Figure 3.11 The scanned paper ECG of 12 lead signals for normal (Left) and confirmed CAD (Right) which have resting and treadmill test ECG signal images

3.4.3 Repository data

There are large repository datasets available related to heart diseases such as arrhythmia (MIT-BIH (Moody and Mark 2001), Arterial fibrillation (Moody 2004), and cardiovascular diseases such as CAD (Alizadehsani et al. 2019), but compared to the first two, coronary artery disease-related datasets are rare. The CAD dataset which is available in repository contains limited trails of 7 CAD patients and 40 normal patients reading (Arafat et al. 2005).

CAD repository datasets available is the Physio-net dataset are derived from 7 CAD patients and 40 Normal patients (sample of data set is plotted as shown in Figure 3.12). The public domain data Physio-Net 2017 is inbuilt in the MATLAB interface (Clifford et al. 2017; Ghiasi et al. 2017; Tan et al. 2018). These data sets are available in the digital signal format. This data set consists of 5788 ECG signal data samples. Data points are with 1000 sampling rates which consists of a minimum of four heartbeats. Using this data, an image file of ECG was created in MATLAB and the image was saved in a format suitable for the application of the developed data extraction method.

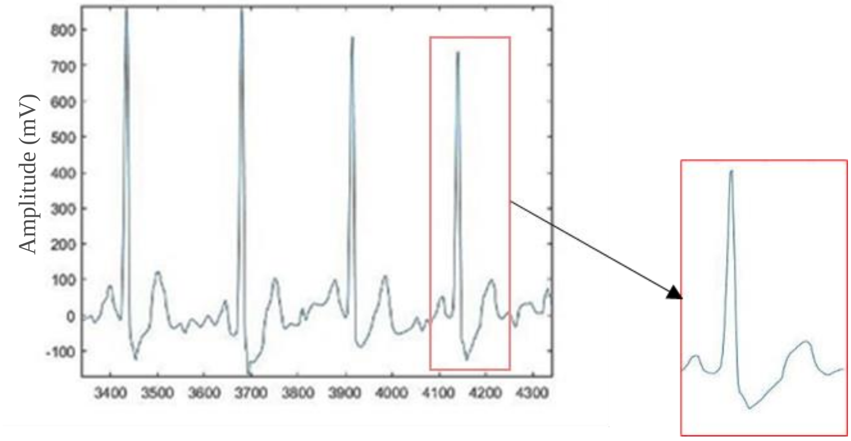


Figure 3.12 ECG distribution plot over a time of period for the repository dataset

3.5 UNDIFFERENTIATED FEVER ANALYSIS

The fever symptomatic analysis which is carried out as an initial study with limited amount of data. This study is concentrated on one dimensional machine learning analysis of 24 hours temperature data in its first stage. Further this analysis is extended to carry over the dataset, which is increased by splitting 30 minutes and 60 minutes temperature data, with and without consideration of additional features. On the basis of results, the best suitable machine learning classifier is opted for future data classification.

3.5.1 Analysis of 24-hour temperature data pattern

Intelligent systems are emerging as an important component in the area of medical science, owing to its significant assistance to doctors in the identification of crucial diseases. Major diseases like Tuberculosis, Non-tubercular bacterial infections, Dengue fever, and Non-infectious diseases have fever as common symptoms due to the variation in white blood cells level. The common way to find the fever symptomatic disease is, by observing the temperature variation along with other symptoms. Because of this, identifying the nature of fever may be a challenging task. The present work deals with the development of a prediction model for such type of diseases with the help of a single parameter i.e. temperature data, without consideration of any other symptoms. The temperature was recorded every single minute over 24-hours as a total of 1440 data sets. The fever diseases considered for prediction are

1. Tuberculosis
2. Non-tubercular bacterial infections
3. Dengue fever
4. Non-infectious diseases

In neural networks, the adjustment of the network's weights and biases will be done automatically through mathematical procedures like training and learning functions. The training function is defined by a general algorithm that affects all the weights and biases of a defined network. In pattern recognition problems, the data are represented as input vectors.

In this study, prediction of fever symptomatic disease was done with help of temperature data as input. The temperature data of 103 patients captured over 1440 intervals is enriched with statistical data like mean, standard deviation, etc. so as to reduce the problem of overfit. In the output layer, the target vector consists of 4 elements (namely, tuberculosis, non-tuberculosis, dengue, and non-influencing diseases). These targets are defined like (1 0 0 0), (0 1 0 0), (0 0 1 0) and (0 0 0 1) corresponding to four types of fever as named earlier. For deciphering such information in the output layer, the manual classification of temperature data for four different fevers provided by medical case history records is used. Hence, these data are preprocessed for vector classification.

The ANN model developed is trained with MATLAB neural network pattern recognition program. In pattern recognition, sigmoid hidden and softmax (transfer function to calculate layer's output through net input) output neuron (patternnet) layers are used in the feed-forward network (Dr.N.Kannathasan 2018). Training of this network is done with a scaled conjugate gradient backpropagation method. General Flow chart of undifferentiated fever prediction analysis is as shown in Figure 3.13.

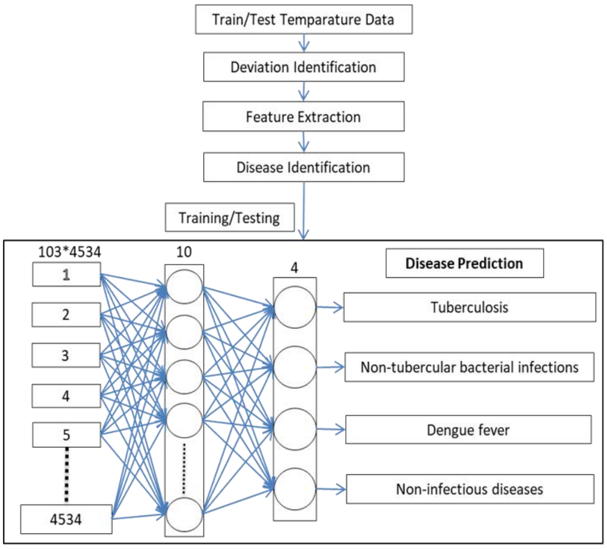


Figure 3.13 Flow chart of undifferentiated fever prediction analysis

In preprocessing stage, input data are derived using available temperature data. For the given set of data, statistical data like mean, standard deviation, square roots of means, and square roots of standard deviations of incremental 30min data set are calculated. In addition to this, the results of kurtosis, skewness, above mean value, and below mean value of the parent dataset and incremental dataset are also determined and added to parent data for effective prediction. This results in a total of 4534 input neurons in the network. Output results are identified based on probability score which is related to defined disease. Out of the values obtained in the output neurons, the output neuron having maximum numeric value is selected. The disease pertaining to the selected neuron is the right prediction.

Thus, there are 4534 neurons containing time-dependent temperature data as input. The hidden layer and classification layer are represented as shown in the Figure 3.14.

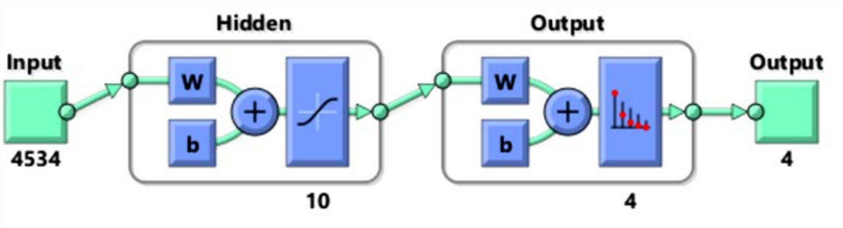


Figure 3.14 Simple neural network

Initially, input and target column matrices are imported to the neural network tool of MATLAB. 75% of data are considered for training, 5% of data are used for validation, and remaining 20% of data are used for testing. Patternnet layer consists of 10 hidden neurons. The Training network process depends entirely on the scaled conjugate gradient backpropagation method (Sankar et al. 2013). At the end of training, the error is analyzed using a performance plot and confusion plot.

3.5.2 Refinement of undifferentiated fever analysis

A known medical classified disease data used for development of model in order to identify new data which lie under pre-known category. A supervised learning method (Kusuma and J 2018) is used to develop the prediction model for these types of diseases classification. Steps involved in the proposed method are data collection, preprocessing and feature extraction, training with different classifier algorithms, comparing results of various algorithms, identification of best model and real-time data analysis with the selected model (Figure 3.15).

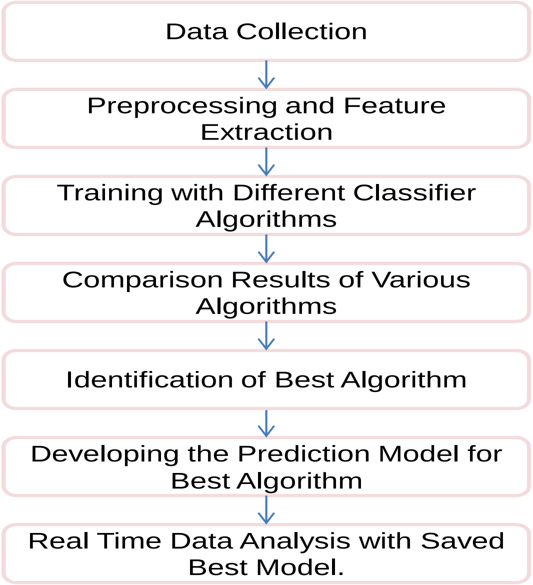
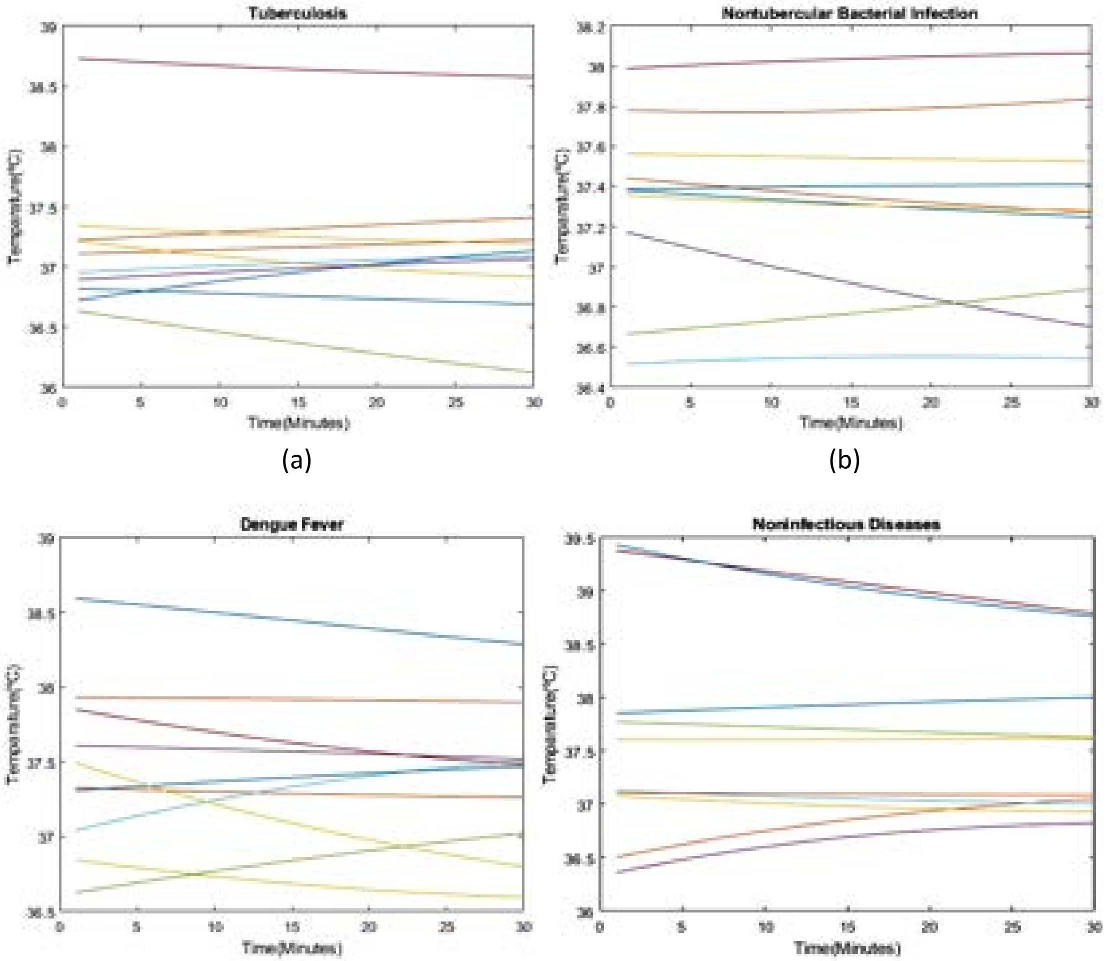

Data Collection
Preprocessing and Feature Extraction
Training with Different Classifier Algorithms
Comparison Results of Various Algorithms
Identification of Best Algorithm
Developing the Prediction Model for Best Algorithm
Real Time Data Analysis with Saved Best Model.

Figure 3.15 Generalized methodology of fever symptomatic disease analysis

Data collection is the process of recording temperature with the help of temperature measuring devices (sensors). Every subject or patient has undergone observation for about 30 minutes or 60 minutes. Temperature data has been collected for 30 minutes with an interval of 1 minute without considering any other risk factors or symptoms (Figure 3.16).



(a) (b)

(c) (d)

Figure 3.16 Temperature variation versus time for ten random samples of (a).Tuberculosis, (b).Non-tubercular bacterial infection, (c).Dengue fever and (d).Non-infectious diseases (Different Colors are defined in the graph which indicates the ten individual patents temperature data for 30min under different disease conditions).

The collection of 30-minute temperature data of a patient is a real-time challenge. Placing a patient in observation for a long period, noting down changes in symptoms, temperature, and decisions based on these changes in symptoms is a long process (Ogoina 2011). To overcome this issue, the temperature measuring device has been placed over recommended locations on the body, and temperature variation data has been collected continuously for 30 minutes.

Pre-filtered patient data are collected with help of the laboratory facility of Kasturba Medical College in Mangaluru, Karnataka. To avoid the noise contained in data collection process, the Savitzky-Golay filter has been used. These data have directly undergone a training process to understand the nature of training accuracy for the dissimilar temperature behavior of the patient.

The above-mentioned filtered data set is considered for training the neural network model without considering mean, median, standard deviation, and other statistical conditions with respect to set of array data. These data were classified into two categories; training (80% of data) and testing & validation (remaining 20% of data). Once a set of training and testing data are ready, the different required classifier models are defined with their respective parameters and limitations.

3.5.3 Prediction system development based on machine learning technique

The prediction system (Figure 3.17) developed is based on a single variable i.e. temperature. The major classifiers which are helpful to classify the type of disease with a supervised learning method are used for the development of a network based on data considered. The major classifiers recommended to study were generative and discriminative classifiers. These are named logistic regression, decision tree classifier, k-nearest neighbor’s classifier, linear decrement analysis, Gaussian Naive Bayes classifier, and Support Vector Machine. With the help of Python IDE, the analyses were done for adopting the best classifier based on the accuracy of these classifier models.

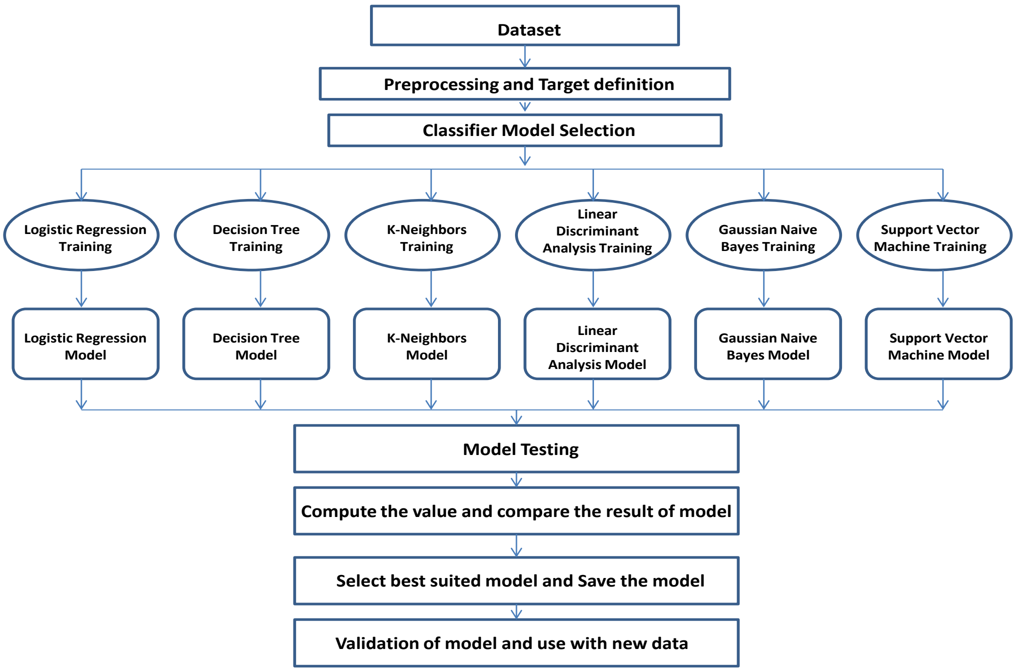

Dataset
Preprocessing and Target definition
Classifier Model Selection
Logistic Regression Training
Decision Tree Training
K-Neighbors Training
Linear Discriminant Analysis Training
Gaussian Naive Bayes Training
Support Vector Machine Training
Logistic Regression Model
Decision Tree Model
K-Neighbors Model
Linear Discriminant Analysis Model
Gaussian Naive Bayes Model
Support Vector Machine Model
Model Testing
Compute the value and compare the result of model
Select best suited model and Save the model
Validation of model and use with new data

Figure 3.17 Prediction model methodology.

Accuracy is stated based on the obtained output of a defined model. The performance of these models is analyzed with help of a confusion matrix. It defines the entire model behavior under the classification of categories. The comparisons (Steinmetzer et al. 2019) of the above-mentioned algorithms are initially made based on the accuracy score of the model. The best-suited model for initial data is then taken into account for further processing in the classification of diseases.

The analysis of temperature data is purely a one-dimensional data analysis. Which are cannot directly applicable for on the image type data (ECG data). During the image type data analysis, either using direct image (ECG image) or by convert the two-dimensional image (ECG image) into a one-dimensional data (ECG time series data) for analysis. Digital image processing (used for ECG digitization) is the method to convert two-dimensional image into one dimensional data. The step-by-step procedure followed for ECG digitization with a help of digital image processing are explained in subsequent section.

CHAPTER 4

DIGITAL IMAGE PROCESSING

Image processing is the refinement of the image with a help of suitable algorithm. The processing of an image depends on the pixel details. The pixel is the smallest element of the digital image and is also named a picture element. The selection of pixel array size is one of the important steps in digital image processing (Badilini et al. 2005). The dimension of the picture is decided by the dimension of the pixels array. It may be rectangular or square. The width of image is decided based on the number of columns (M) and the image height is decided by the number of rows (N) (Shrivastava et al. 2014). The pixel array is represented in the form of M*N. The coordinate system of the image is different from the coordinate system in mathematics. In the Coordinate system in the image, x increases left to right, and y increases from top to bottom. The resolution factor helps to match the size of the real-world image (Pratt 2003) and digital image.

For example, an image of 4500 x 3000 pixels with a resolution of 300 pixels (Tabassum and Ahmad 2020) per inch (PPI) would be a real-world image size of 15" x 10". In resolution terms, “ppi” is pixels per inch and “dpi” is dots per inch. ppi and dpi are interchangeable (1dpi = 1ppi). “dpi” is used in the printer and the “ppi” is used in the digital image. Another effective parameter considered is, “intensity”, which is needed to truly define an image. Each pixel has its own intensity value or brightness. If all the pixels have the same value, the image will be a uniform shade; all black, white, gray, or some other shade. It is in the type of intensity used for each pixel that image types vary. Black and white images only have intensity from the darkest gray (black) to the lightest gray (white) (Debasish Biswas. et. al. 2011). Color images, on the other hand, have intensity from the darkest and lightest of three different colors, Red, Green, and Blue (Figure 4.1). The various mixtures of these color intensities produce a color image. Thus, two most basic types of digital images, BW and Color, are known as gray-scale and "RGB" images respectively. In addition to the intensity type of each pixel, the range of intensity values also varies. Intensity values in digital images are defined by bits. A bit is binary and has only two possible values, 0 or 1. An 8-bit intensity range has 256 possible values, 0 to 255. (Mishra et al. 2021) This can be seen mathematically by 2(x of bits). For 1-bit or binary image: 2^1= 2 possible values and for an 8-bit image: 2^8= 256 possible values.

The standard digital photo uses an 8-bit range of values; RGB images (Figure 4.1) use 8-bit intensity ranges for each color and BW images have a single 8-bit intensity range. Since RGB images contain 3 x 8-bit intensities they are also referred to as 24-bit color images. When considering of interval between the values: Theoretically, an 8-bit range could occupy values from 0 to 1 using 1/256th increment, but practically, 8-bit images are defined to use only integer values from 0 to 255.

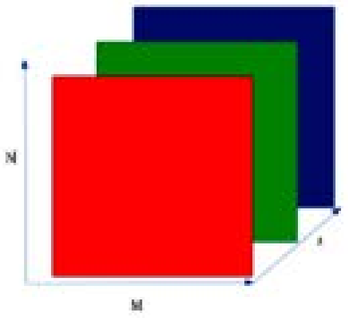


Figure 4.1 RGB color distribution in M*N*3 array

4.1 ECG IMAGE PRE-PROCESSING AND DIGITIZATION

The nonlinear dynamic behavior of the electrocardiogram (ECG) signal is well known, and it was a key feature that is used in this study. Since the CNN system needs feedback in the form of digitized signals, the digitization process was implemented, wherein the data were extracted from ECG and TMT-ECG graphs. The digital image processing will help to analyze the ECG graph in 1D format. i.e conversion of scanned ECG/ TMT-ECG into one-dimensional vector data. Digitalization process applied on a scanned image of TMT ECG graph. The method of digitization was also recommended to overcome various paper degradation problems, in addition to the reasons cited in the introduction. The graph sheet includes two sets of the graph, one which belongs to resting ECG and another one noted as the excited state of stressed ECG (Excited TMT condition).

According to literature analysis, the digitization of ECG graphs was applied only for single-lead ECGs, and the majority of cases were treated with only resting condition ECGs. However, TMT-ECG was measured while the patient was running on the treadmill, based on the Bruce protocol. This plays a key role in predicting arrhythmia, CAD, and other heart-related conditions. The proposed approach was designed to digitize both the resting and TMT-ECG conditions (TMT-ECG). This novel approach will include digitized ECG data and features for each of the 12 leads individually.

The electrical signal of the heart was printed as a waveform (TMT report), which was collected through 10 electrodes and printed as 12-lead heart signals. The ECG and TMT-ECG reports used for data extraction were of resting (ECG) and exercised (TMT-ECG) conditions, aided in the study of the entire pattern of heart disease activity. Different stages of data extraction in the study are discussed as shown in Figure 4.2. Digitization methodology is adopted by consideration of single-lead ECG and twelve lead ECG image processing techniques and adaptation of the best suitable approach.

The raw twelve lead ECG image was filtered out during the pre-processing stage in order to obtain a clear signal image. This pre-process includes skew detection and correction (scanned image orientation correction), color-based image segmentation for background removal (grid removal) i.e. making a threshold to darken the main data region and lighten the background for removal of the grid line and helps to remove all the color content and make the image black and white with the help of threshold values. This binarized image is filtered with the help of black and white area open method or application of a median filter to remove salt and pepper noise, printed text detection using easy OCR (Optical Character Recognition), and application of OpenCV to erase the text or masking which is applied to an image to remove all the string elements which are included in the area of the digitalization field, image dilation or morphological operation to improve pixel quality, and multi-cropping based on single lead image position by giving height and width. These images are further processed by pixel indexing method to extract the digitalization value (1D) of that particular image and medical features related images are extracted using hybrid method feature extraction process. The final processed ECG/TMT-ECG signal is obtained by the pixel values

(1D) and the signal is validated by computing specific pixel per column and application correlation relationship.

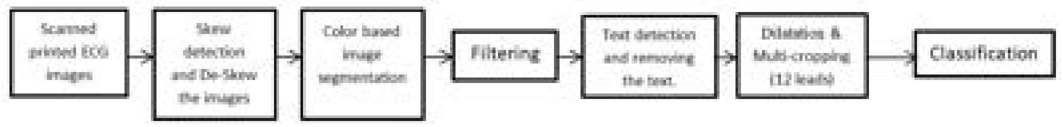

Scanned printed ECG images
Skew detection and De-Skew the images
Color based image segmentation
Filtering
Text detection and removing the text.
Dilatation & Multi-cropping (12 leads)
Classification

Figure 4.2 Overview of the block diagram related to pre-processing

The pre-processing step assists in obtaining the whole dataset (i.e., a single lead raw image to extract 1D data and pre-processed (filtered) image and 12 lead filtered images for image analysis) that is needed to develop the prediction algorithm. The detailed procedure of pre-processing methods is stated as follows in Figure 4.3.

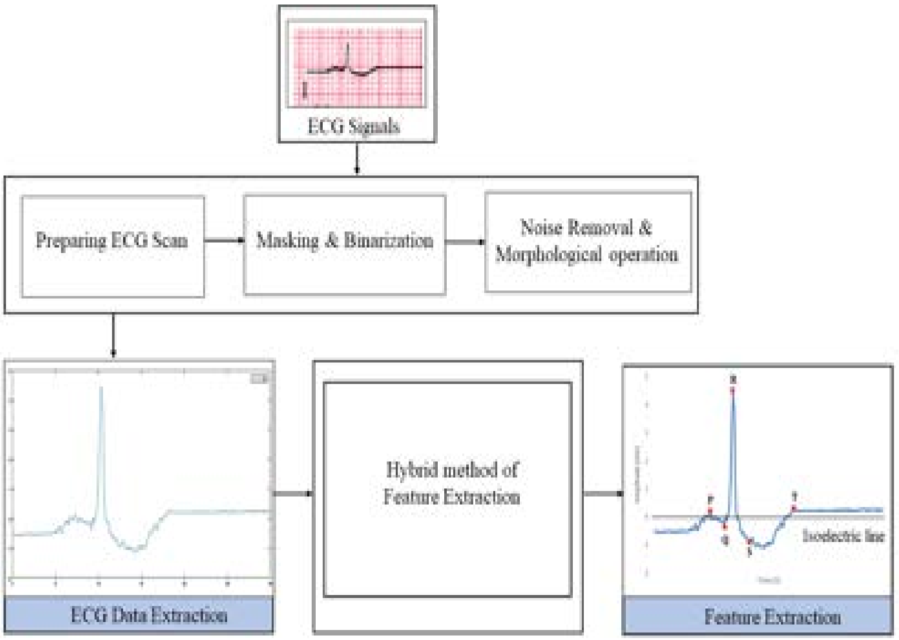


Figure 4.3 Single-lead TMT-ECG digitization

Based on the above two procedures, the best optimal method is selected for data extraction from scanned paper (Figure 4.4). The detailed explanation of the method is explained as follows.

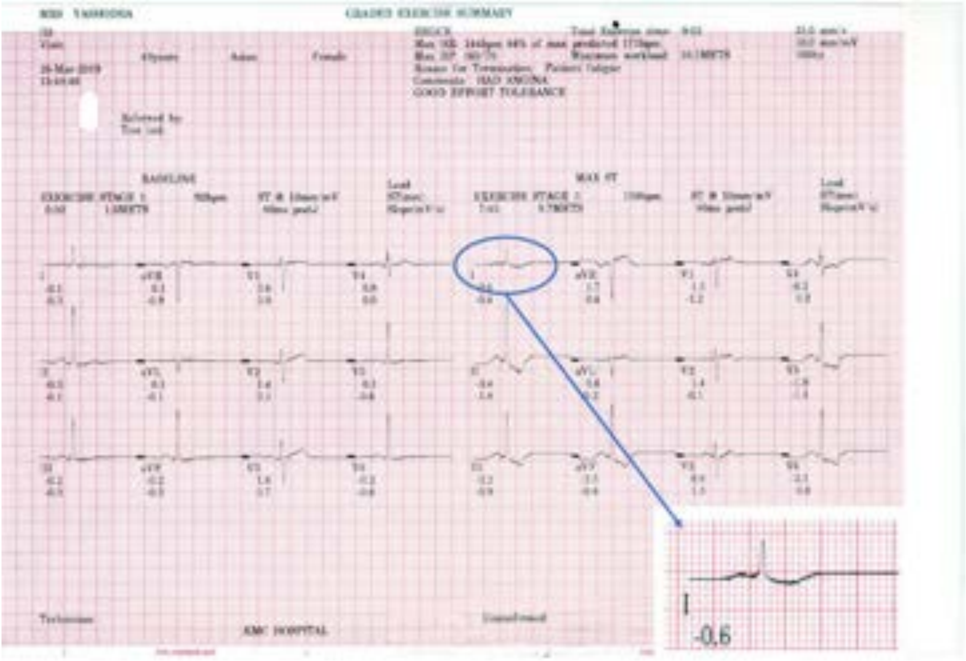


Figure 4.4 Twelve-lead TMT-ECG ECG report

4.1.1. Image Scanning

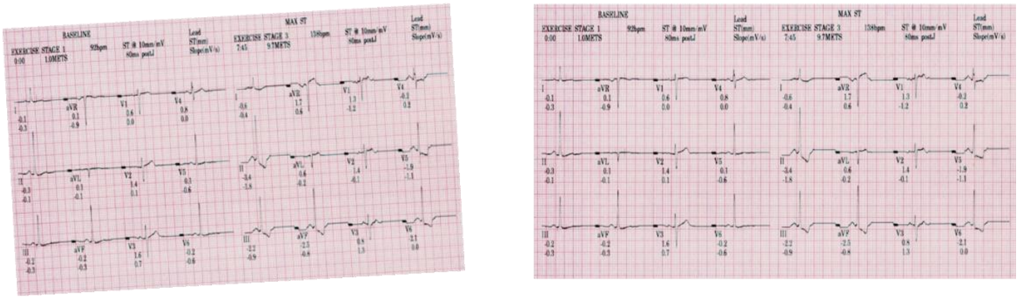
The combined ECG and TMT-ECG report was taken from the GE healthcare diagnostic ECG MAC 5500 HD machine at the neighboring hospital. The thermal paper-printed ECG data were scanned using a flatbed HP A4/A5 scanner with a 600 dpi scanner resolution (Jayaraman et al. 2012) (Figure 4.4). Any errors in inclination while placing the report on the scanning bed is identified as skew.

Skewed images are slanted images (Figure 4.5a) that are typically found in scanned/captured photos. The slanted images could have a negative impact on accuracy. One of the finest methods for de-skewing an image (Figure 4.5b) is to use the Hough transformation (which is a basic linear transformation function) (Mukhopadhyay and Chaudhuri 2014). The background grid lines of the scanned ECG image are used to compute the skew rotation angle of the image. A straight line can be detected using the Hough transform (Al-Khatatneh et al. 2015). In image space, the slope intercept model is defined by the straight-line equation 4.1 shown below.

$$Z = bx + c \tag{4.1}$$

The slope of the line is defined by the parameter b, while the intercept (Z-intercept) point is defined by the parameter c. The Hough transform is used to express the properties of straight lines in terms of slope-intercept model parameters, rather than discrete points $(x_0,\ y_0)$, $(x_1,\ y_1)$. As a result, the straight-line Z = bx + c can be represented in the Hough space as a point (c, b). Using the transform function as in equation 4.2, every point (x, y) in cartesian space is mapped into sine curve in $\rho - \theta$ Hough space. The coordinates of a point when shifted by an angle around becomes and is defined as

$$
\begin{gathered}
\mathrm{x} = \mathrm{Cos}(\theta) \times (\mathrm{x}_1 - \mathrm{x}_0) + \mathrm{Sin}(\theta) \times (\mathrm{y}_1 - \mathrm{y}_0) \\
\mathrm{y} = -\mathrm{Sin}(\theta) \times (x_1 - x_0) + \mathrm{Cos}(\theta) \times (\mathrm{y}_1 - \mathrm{y}_0) \qquad (4.2) \\
\rho = xcos\theta + \mathrm{y}\sin\theta
\end{gathered}
$$



(a) (b)

Figure 4.5 (a) skewed Image of ECG (b) De-skewed image of ECG

4.1.2. Image Masking

The scanned report contains string values pertaining to electrode locations and general information about the patient. Such information and the electrode position are printed which need not be digitized. These string values needed to be removed from the image as redundant data. This can be removed manually but it will be time-consuming. After adding a mask to the image (Figure 4.6), such information can be removed. The image masking helps to remove such redundancies from the images (Figure 4.7). This process can be automated with the help of a text detection and removal process.

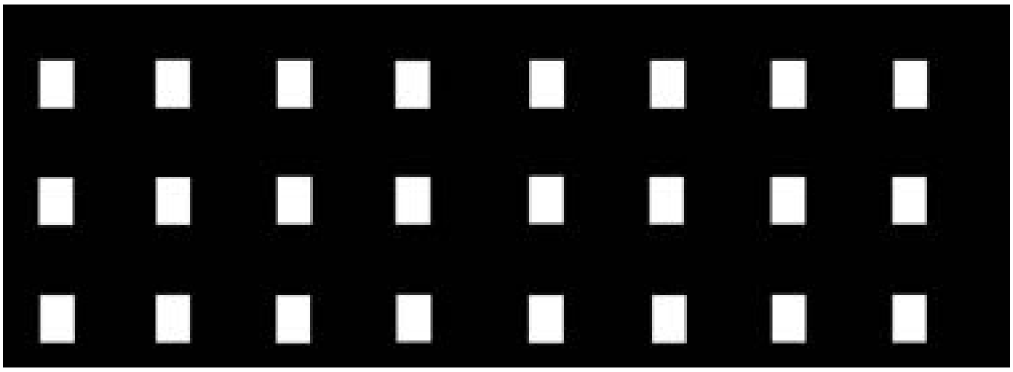


Figure 4.6 Masking image

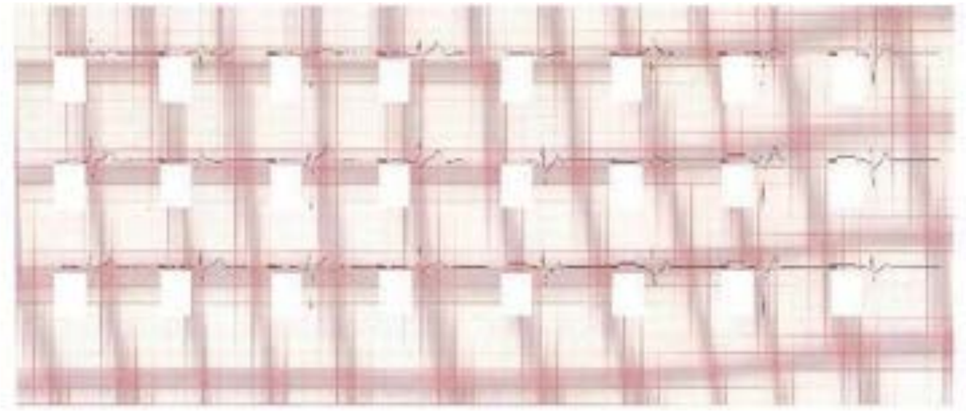


Figure 4.7 ECG graph after adding the mask

4.2 THRESHOLD SETTING OR COLOR IMAGE SEGMENTATION

The background grid in the graph image should be removed in order to extract the signal image from the scanned image. When it comes to removing the backgrounds from graph images, a color histogram comes is to be referred. The color distribution over an image is represented by a histogram. In the ECG graph, the intensity of the red pixel is shown in Figure 4.8. The histogram (Figure 4.8) was generated with the help of MATLAB R2020b module, namely "Color Threshold GUI" (Carolina Sparavigna 2015). Every digital image has three colors that are used a color image, RED, GREEN, and BLUE (RGB).

- RED

Will lighten very dark color while also changing the tone.

- GREEN

Will darken light color while also changing the tone.

- BLUE

Will darken light color while also changing the tone.

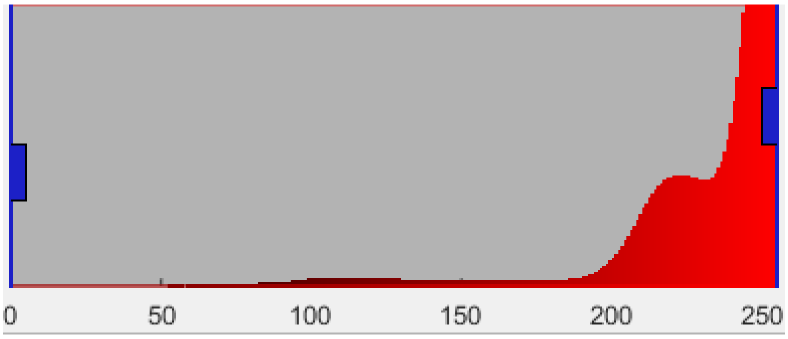


Figure 4.8 Histogram of RED color of ECG

A color histogram represents the number of pixels that have colors in each of a fixed list of color ranges. The histogram of the image gives the idea about which color should be removed so that it will be clean and dark. Red color erosion makes black darker. Ten different ECG/TMT-ECG scanned images were selected randomly to achieve proper red pixel erosion. In the above color threshold module, the threshold value for the red pixel intensity was determined to be 145–255 by the trial-and-error method i.e. removing 0-145 intensity pixels of the red, black line becomes darker. Removing the red pixel threshold for the red color starts from 145 to 255.

Define thresholds for channel 1 based on histogram settings
channel1Min = 145.000;
channel1Max = 255.000;
Define thresholds for channel 2 based on histogram settings
channel2Min = 0.000;
channel2Max = 255.000;
Define thresholds for channel 3 based on histogram settings
channel3Min = 0.000;
channel3Max = 255.000;

Once the threshold was identified, the system automatically removes the lesser range of red pixels from the image. Finally, a black line ECG graph was retained at the end of red pixel intensity filtering. The same is also achieved based on color-based image segmentation method, which gives combined results of thresholding and binarization.

This process is optimized with consideration of background grids, which are often lighter in shade compared to ECG signal images. The image is processed column by column, with each column's darkest pixels being replaced with black pixels, while the remainder pixels being kept as white pixels. This results in binary image formation (Figure 4.9). During the process, there is a possibility of obtaining unwanted printed characters which can be removed (kaisJameel and R. Manza 2012).

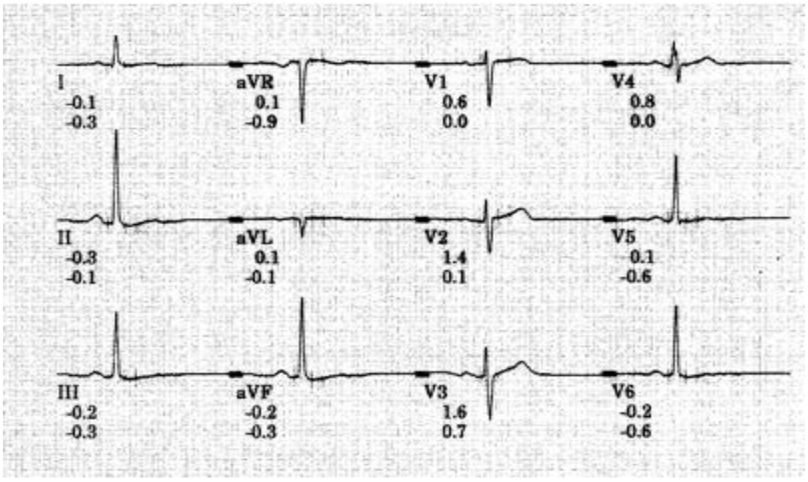


Figure 4.9 Binary ECG image with noise

4.3 BINARIZATION

Binary images are images that have been quantized to two values, usually 0 and 1. But, often color image belongs to 0 to 255 color range (8 bits image). Before converting images to Black and White, those have to be converted into the grayscale image. The GIMP image software has three algorithms (4.3, 4.4 and 4.5) which are used to convert images into black and white. These are;

1. The lightness method averages the most prominent and leads to prominent colors:

$$(\max(R, G, B) + \min(R, G, B)) /2 \quad (4.3)$$

2. The average method simply averages the values:

$$(R + G + B)/3. \quad (4.4)$$

3. The luminosity method

$$\text{Luminosity} = 0.21R + 0.72G + 0.07B \quad (4.5)$$

The luminosity method (4.5) is a more sophisticated version of the average method (Debasish Biswas. et. al. 2011). It also averages the values, but it forms a weighted average to account for human perception. It is more sensitive to green when compared with other colors. So green is weighted most heavily. After converting all layers of the image in the grayscale, it is available for converting into the binary image (Figure 4.10). In binary, if pixel intensity is less than threshold value, it is set to 0 (black) and else set to 1 (white). Equation (4.5) shows the luminosity formula, where *R* refers to red, *G* refers to green, and *B* represents the blue pixel values.

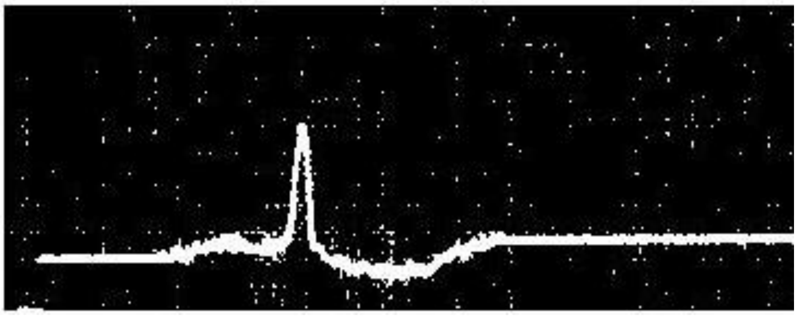


Figure 4.10 Binary image of ECG

4.4 OPEN AREA DELETION

The use of a binarization Gaussian filter can reduce the clarity of the image. To prevent this, the open areas of the image were deleted by a process that combined the removal of noise and image dilation. Such an open area deletion feature eliminates the pixel cluster of lesser than fifty pixels for binary images and leads to better clarity of the image (Figure 4.11).

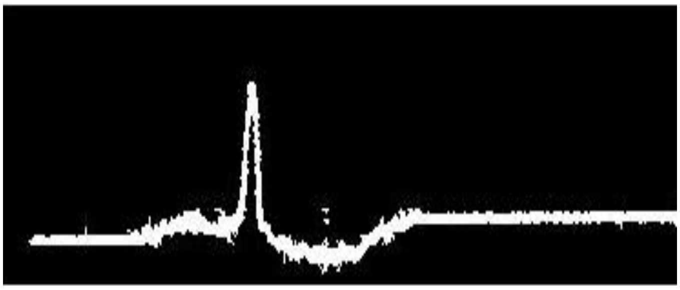


Figure 4.11 Clean binary image of ECG

4.5 MORPHOLOGICAL OPERATION

The ECG signal, after the elimination of open areas is represented by a thick line. A morphological operation was used to retrieve the typical data segment out of this. The morphological operation algorithm chooses the averaged pixels in a row of pixels. Consequently, the morphological operation's output will be a thin binarized image, resulting in a clean, thin graph (Figure 4.12). Mathematically, it is written as (equation 4.6) (Pratt 2003).

$$\mathrm{thin}(X, Y) = X - (X \otimes Y) \tag{4.6}$$

This operation is a form of hit-and-miss transform, where the thinning of a set X by structuring elements (SE) Y is denoted by X ⊗ Y. The thinning by a sequence of SEs is as in (equation 4.7)

$$(X \otimes Y) = [X - ((X \ominus Y_1) \cap (X^C \ominus Y_2) \tag{4.7}$$

Where z in universe X belongs to the output Y_1 fits in X (hit) and Y_2 misses X, and the erosion of an image by dilation of background is referred by X^C.

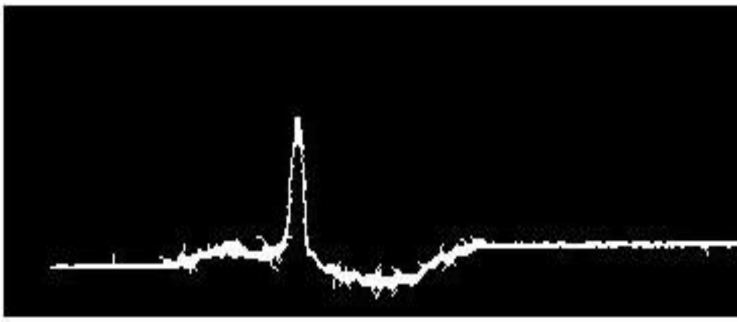


Figure 4.12 After a morphological operation.

After the application of color-based segmentation, the resultant images are used for filtering. This is an automatic optimized method to obtain a clear image. This will provide the same result of open area deletion and morphological operation.

4.6 FILTERING

The removal of background degrades the actual ECG signals due to the presence of noise called as salt and pepper noise. Image noise is defined as the random variation of small pixel values of an image. Because noisy signals might lead to misdiagnosis, it's important to minimize the noise effectively using filters. Visual inspection (Zhu and Wang 2012) yields the best results when a median filter is used to remove the majority of the noise. This image filtering technique scans the image with a tiny matrix and recalculates the center pixel value by taking the median of all pixel values within the matrix. The resulting image is shown in Figure 4.13.

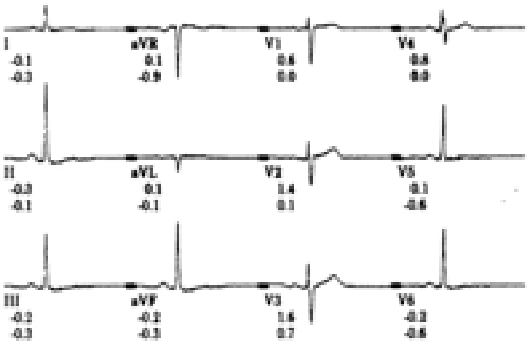


Figure 4.13 Filtered ECG image

This procedure helps to optimize the methodology of digitization. Once this process is completed, image can be considered for text detection and removal, if it is not removed in the masking methodology. Text detection and removal will be one of the sophisticated automated methods for removing all string values compared to the image masking procedure.

4.7 TEXT DETECTION AND REMOVAL

This procedure aids in the removal of undesired string element pixels that do not correspond to the signal image (Printed characteristics not considered for analysis). Optical Character Recognition (OCR) is a useful technique for converting text in photographs into computer-readable text (Figure 4.14). Easy OCR algorithm is developed in python with the help of Pytorch to detect readable text. In addition to this, CRAFT (Baek et al. 2019) algorithm is used for detecting phase with the help of CRNN Recognizing model. It consists of three main phases i.e. Feature extraction, sequence labeling, and decoding. The character location score and character affinity score which are generated by the easy OCR method will help to fully cover various text shapes over the image in a bottom-up approach.

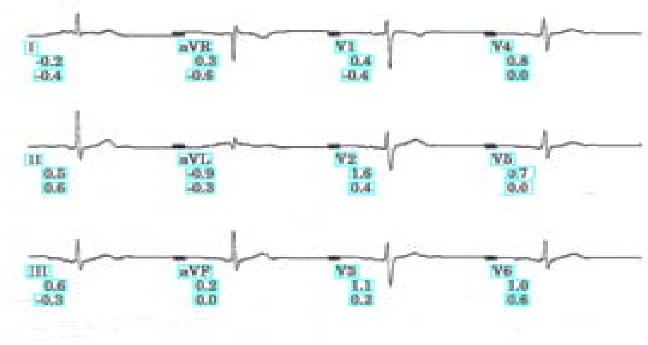


Figure 4.14 Text detected image

Inpainting operation developed using OpenCV helps to remove the recognized text. Deletion of the text leads to masking the original image where the text has to be erased. To achieve this, the dimensions of the mask image and the input image must be the same. The mask will show non-zero pixels (detected text) in parts of the input image that contain text and would be in-painted (using Navier-Strokes algorithm) (Bertalmio et al. 2001), while zero pixels(signal image) will remain unchanged(Figure 4.15).

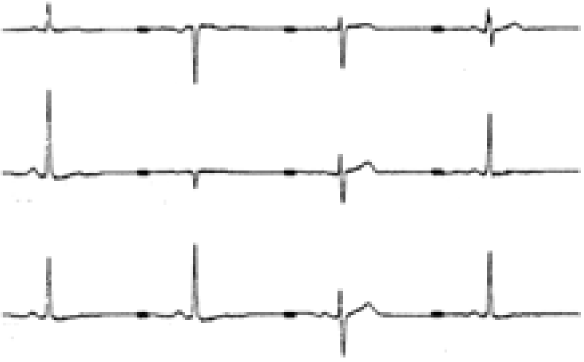


Figure 4.15 Text erased ECG image

4.8 DILATION AND MULTI-CROPPING.

The dilation is an operation that helps to improve the image features with consideration of 2 inputs, one which is related to dilated image and the other one corresponding to a two-dimensional structuring segment. The dilation also helps to amplify features in the signal image (Raid et al. 2014). The multi-cropping method makes it possible to extract a single lead ECG and TMT ECG signal image from a 12-lead ECG and TMT ECG signal images. The OpenCV is used to crop each individual lead by determining the lead image's height and width (Figure 4.16). The input signals for the prediction model developed based on a single lead are derived based on this multi-cropping method. This method is also extended to derive raw single-lead ECGs and TMT ECGs from raw twelve lead signal images.

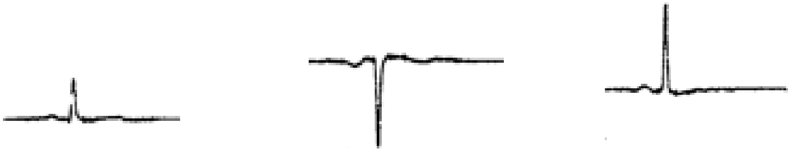


Figure 4.16 Cropped single lead images

These cropped signals are used to extract one-dimensional data with the help of pixel indexing methodology.

4.9 SIGNAL DATA STORAGE USING PIXEL INDEXING

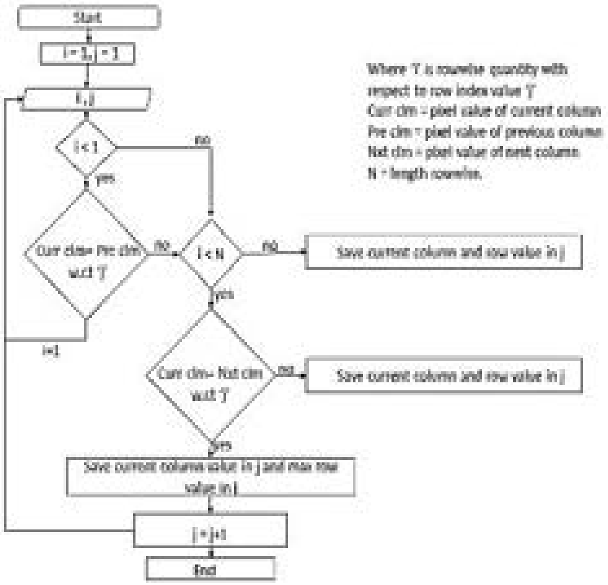

Start
i = 1, j = 1
i, j
i < 1
no
yes
Curr clm= Pre clm w.r.t 'j'
no
i < N
no
Save current column and row value in j
yes
Curr clm= Nxt clm w.r.t 'j'
no
Save current column and row value in j
yes
Save current column value in j and max row value in j
i+1
j = j+1
End
Where 'i' is rowwise quantity with respect to row index value 'j'
Curr clm = pixel value of current column
Pre clm = pixel value of previous column
Nxt clm = pixel value of next column
N = length rowwise.

Figure 4.17 Flow chart for pixel indexing

The binary signal (ECG) graph is turned into an array sequence using the pixel indexing approach (Jufriadif Naama, Catur Suharintob 2017), which corresponds to the amplitude (mV) and time series (ms) on the ECG graph. The 2-D array contains the results of all non-zero column and row items. The row value (time series) of the pixel is stored in the first column of the array. Similarly, the second column of the array contains the associated row pixels of a particular column value (amplitude). If a single column contains more than one number of pixels for a given period, the pixel with the highest pixel index value is stored. The flow chart of the algorithm is represented as shown in Figure (Figure 4.17).

Pixel indexing data is saved in a readable file format (.dat and. excel) and converted from dots per inch to millimeters using the equation below for future uses (equation 4.8).

$$l = ld \times 25.4/600 \tag{4.8}$$

where l is the length in millimeters and ld is the length in dots per inch (1 inch = 25.4mm) at 600 dots per inch (dpi), with each dot equaling to 0.0423 mm. The grid unit along the amplitude axis equates to 0.1mV, and the same along the time series axis (i.e a grid unit) corresponds to 0.02s in the ECG graph (Tabassum and Ahmad 2020). The extracted data from all 12 lead plots are compared with the original 12 lead image's ECG graph, as seen below (Figure 4.18).

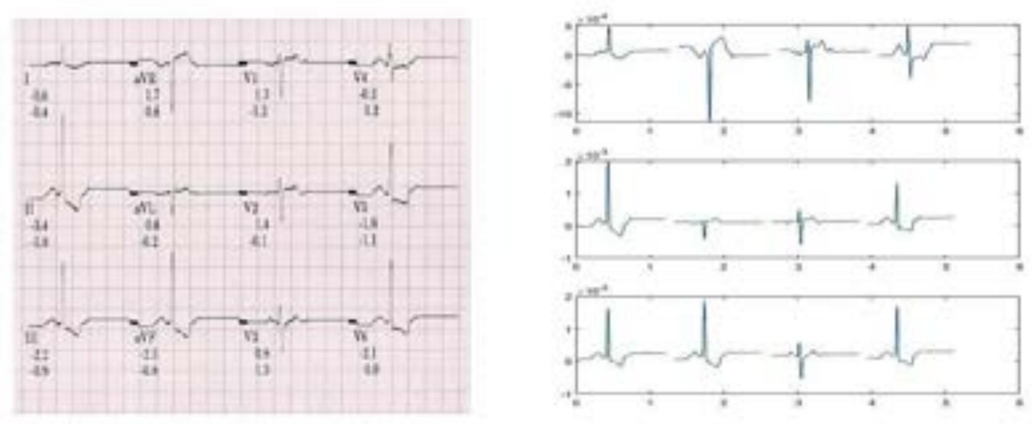


Figure 4.18 12 Lead ECG data comparison between original data and extracted data

4.10 ECG FEATURE PARAMETER EXTRACTION

The combination of several methods is used to detect the amplitude values of P, Q, R, S, T, and related segment interval values (QT interval, QRS complex interval, and ST interval segment). The method used to derive the QRS interval is Pan-Tompkins's method and is validated with slope-based R-value detection method. P and T values are derived based on the statistical method.

The Pan-Tompkins is one of the finest methods for extracting the QRS feature from ECG data with a very good accuracy (Pan and Tompkins 1985). The breadth, slope, and amplitude of the QRS complex are used to analyze the ECG signal during diagnosis. The following filter stages are included in the Pan- Tompkins algorithm to find QRS complex: (i) low pass filter and high pass filter, (ii) derivative (differentiator), (iii) squaring, (iv) integration, (v) adaptive thresholding.

4.10.1 Low Pass and High Pass Filters

A bandpass filter is used to boost the signal-to-noise ratio. A filter bandwidth of 5-15 Hz is ideal for maximizing QRS contribution while simultaneously reducing muscle noise, baseline drift, powerline interference, and the frequency content of the P and T waves. The second-order low pass filter's (Figure 4.19) transfer function is given by equation 4.9

$$\mathrm{H(z)} = \frac{(1-\mathrm{z}^{-6})^2}{(1-\mathrm{z}^{-1})^2} \tag{4.9}$$

The differential equation (equation 4.10) is given by

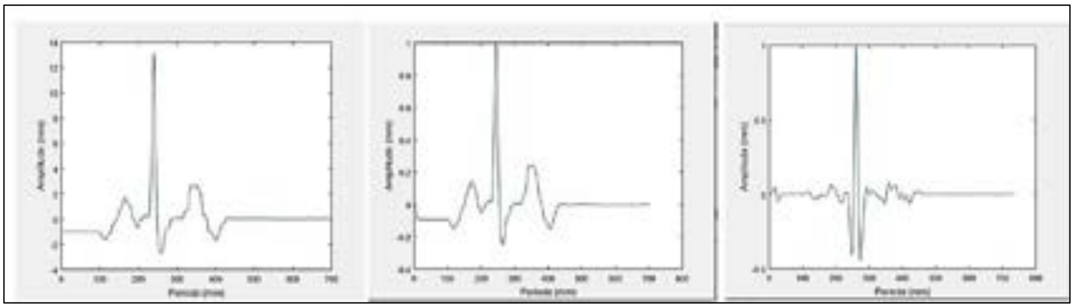
$$\mathrm{y(nT)} = 32\mathrm{x(nT}-16) - [\mathrm{y(nT}-\mathrm{T}) + \mathrm{x(nT)} - \mathrm{x(nT}-32\mathrm{T})] \tag{4.10}$$

where T is the sampling period and the cut-off frequency and gain of the filter are 11 Hz and 36 respectively. The delay of the filter is given by 6 samples. The transfer function (equation 4.11) of second-order high pass filter is given by

$$\mathrm{H(z)} = \frac{-1+32\mathrm{z}^{-16}+\mathrm{z}^{1}-32}{1+\mathrm{z}^{-1}} \tag{4.11}$$

The differential equation (equation 4.12) is given by

$$\mathrm{y(nT)} = \frac{1}{8\mathrm{T}}\{-\mathrm{x(nT}-2\mathrm{T}) - 2\mathrm{x(nT}+\mathrm{T}) + \mathrm{x(nT}+2\mathrm{T}) \tag{4.12}$$



(a) (b) (c)

Figure 4.19 (a) Before application of low pass filter, (b) After application of low pass filter, and (c) After application of high pass filter

4.10.2 Squaring

When a differentiated output signal is squared (which is achieved by equation 4.13), the result is a positive signal with a larger QRS complex than other signal waves. The maximum amplitude value will provide the R-value.

$$y(nT) = [x(nT)]^2 \qquad (4.13)$$

4.10.3 Moving Window Integration

In addition to the R wave, moving window integration (Figure 4.20) collects information about the QRS complex. The window size must be considered carefully. If it is too large, the QRS complex and T wave will combine with integration, if it is too small, a single QRS complex creates several peaks.

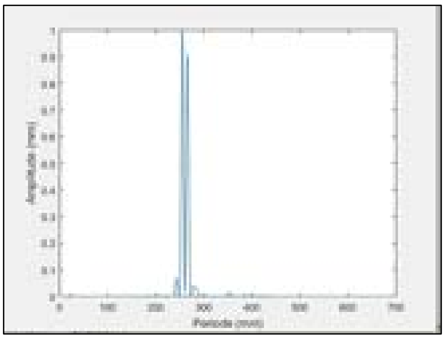


Figure 4.20 Application Moving Window Integration

This can effectively be implemented based on the differential equation (equation 4.14)

$$\mathrm{y(nT)} = \frac{1}{\mathrm{N}}[\mathrm{x(nT - (N-1)T) + x(nT - (N-2)T) + x(nT)}] \quad (4.14)$$

A window width n=30 is suitable for sampling frequency fs=100Hz

4.10.4 Adaptive Thresholding

Once window width is fixed, then the value of the QRS complex is determined by the thresholding method adopting. This method will determine two threshold values based on signal peak and noise peak values. Based on these threshold values, the QRS (Figure 4.21) complex is determined.

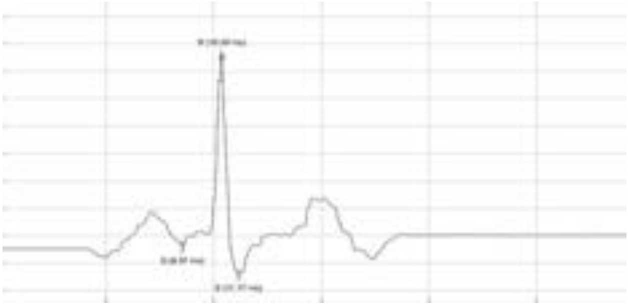


Figure 4.21 QRS Feature parameters

The R-value obtained from the Pan-Tomkins method is validated with the help of the slope-based R-value detection method. Once the QRS interval is fixed, Then R-value is calculated by applying the slope-based method to find the slope of the QRS wave. Where the slope of the QRS wave will become zero that point is considered a peak value i.e. maximum amplitude value (R-value). Based on the QRS interval and slope-based method, the statistical method is adopted to find the values of P and T by choosing the closest maximum amplitude value before Q and the closest maximum amplitude value after the S value.

4.10.5 Feature Extraction

The literature studies have highlighted several algorithms developed to extract ECG features. In the majority of cases, the data used in these algorithms were QRS complex and ST sections of the ECG graph. In this work, The Pan–Tompkins method (Pan and Tompkins 1985; Sedghamiz 2014), with the combination of a slope-based and statistical method (Li et al. 2017), was used in the feature extraction analysis. The Tompkins extraction method is ideally suited for extracting from the QRS complex.

The accuracy of an intelligent system in the prediction of the disease depends on lead data (1D) and key features extracted from ECG graph data. The lead data with the QRS complex and the ST section were inadequate to create an efficient framework. Additional segmentation of lead data (time series data) helps to improve the efficiency of the system. The segments were extracted using a statistical procedure that selects, neighboring maximum and minimum values of the known segment data. This was achieved by knowing the QRS complex (QRS segment) position and the P and T amplitudes (Figure 4.22). Using these amplitude values, the other segments (PQ, QT, ST) could easily be determined.

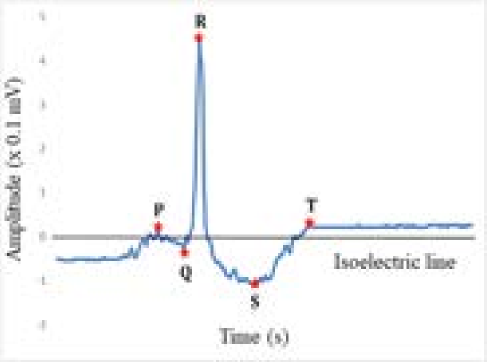


Figure 4.22 Extracted feature data of the ECG.

one-dimensional data and features of ECG/TMT-ECG extracted through image processing are used for development of coronary artery disease prediction model. These one-dimensional data are saved as array type which is helpful for dealing of one-dimensional prediction model. In following section, a detailed discussion related to development of prediction model based on these one-dimensional data and even related to two-dimensional and hybrid models.

CHAPTER 5

PREDICTION MODEL FOR CORONARY ARTERY DISEASE

The feature data extracted from the ECG and TMT-ECG signals were used to develop the prediction system for the CAD, using convolution neural network (CNN) modeling. The prediction model to detect the CAD are developed based consideration of both types of data i.e structured and unstructured data. Structured data type analysis by considering time line data extracted from ECG/ TMT-ECG graph for both single and twelve lead ECG/TMT-ECGs with a help of one-dimensional convolutional prediction system. Unstructured data type analysis is carried out with consideration of image data ECG/ TMT-ECG graph for both single and twelve lead ECG/TMT-ECGs with a help of two-dimensional convolutional prediction system. Further an optimized hybrid multi-headed prediction model is developed based on consideration of both types of data with respect to single and twelve lead ECG/ TMT-ECGs. The detailed discussion related to these concepts are as discussed below.

5.1 ONE-DIMENSIONAL ECG PREDICTION ANALYSIS

The one-dimensional prediction system was developed on two considerations. The first one was based on single-lead signals and the other on 12-lead signals (I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6). In the design of the CNN model, array of the time series voltage datasets obtained through the data extraction procedure, including the ECG feature data, were used as input. The amplitude values of the signal, QRS complex, ST segment, P, Q, R, S, and T values are also part of the input.

Initially, in the single-lead CNN model, a total of 815 time-series data extracted from individual ECG graphs were normalized and fed into the input layer to generate an output during the training cycles. Iterations were repeated for the designated number of training data in a similar way. These data were obtained as a one-dimensional dataset and were used in the development of the CNN model. CNN architectures, each with different number of layers and different combinations of activation functions were analyzed to arrive at an optimal CNN network. Three subcategories were made based on the number of layers used for the optimization procedure, i.e., (1) a one-layer CNN architecture, (2) two-layer CNN architecture, and (3) three-layer CNN architecture (Figure 5.1) (Sannino and De Pietro 2018).

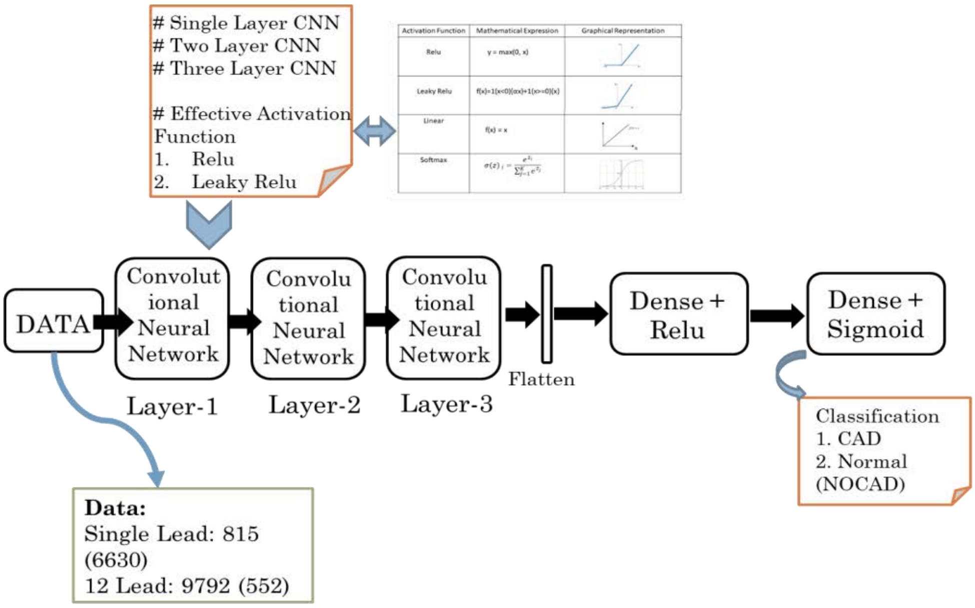


Figure 5.1 Generalized three-layer convolutional neural network architecture

5.1.1 Single-Lead ECG/TMT-ECG-Based CNN Architecture

Single lead ECG / TMT-ECG architectures is, consideration individual leads (limb leads, augmented leads or precordial leads) for analysis which are extracted from the twelve lead ECG/ TMT-ECG with help of preprocessing. The preprocessed data are the input to the development of optimized convolutional neural network. The mathematical representation of the generalized convolution operation is given by:

$$x_n = \sum_{k=0}^{n-1} y_k f_{n-k} \tag{5.1}$$

where the variable y corresponds to the input signal, f denotes the filter, and n represents the number of data/feature elements in y, respectively. The variable ‘x’ represents the output vector. These layers were convoluted with a kernel size of 1 and a stride of 1 using equation 5.1 (Acharya et al. 2017d). Four activation functions adopted in the CNN model across the different layers were: ReLU, SoftMax, LeakyReLU, and linear.

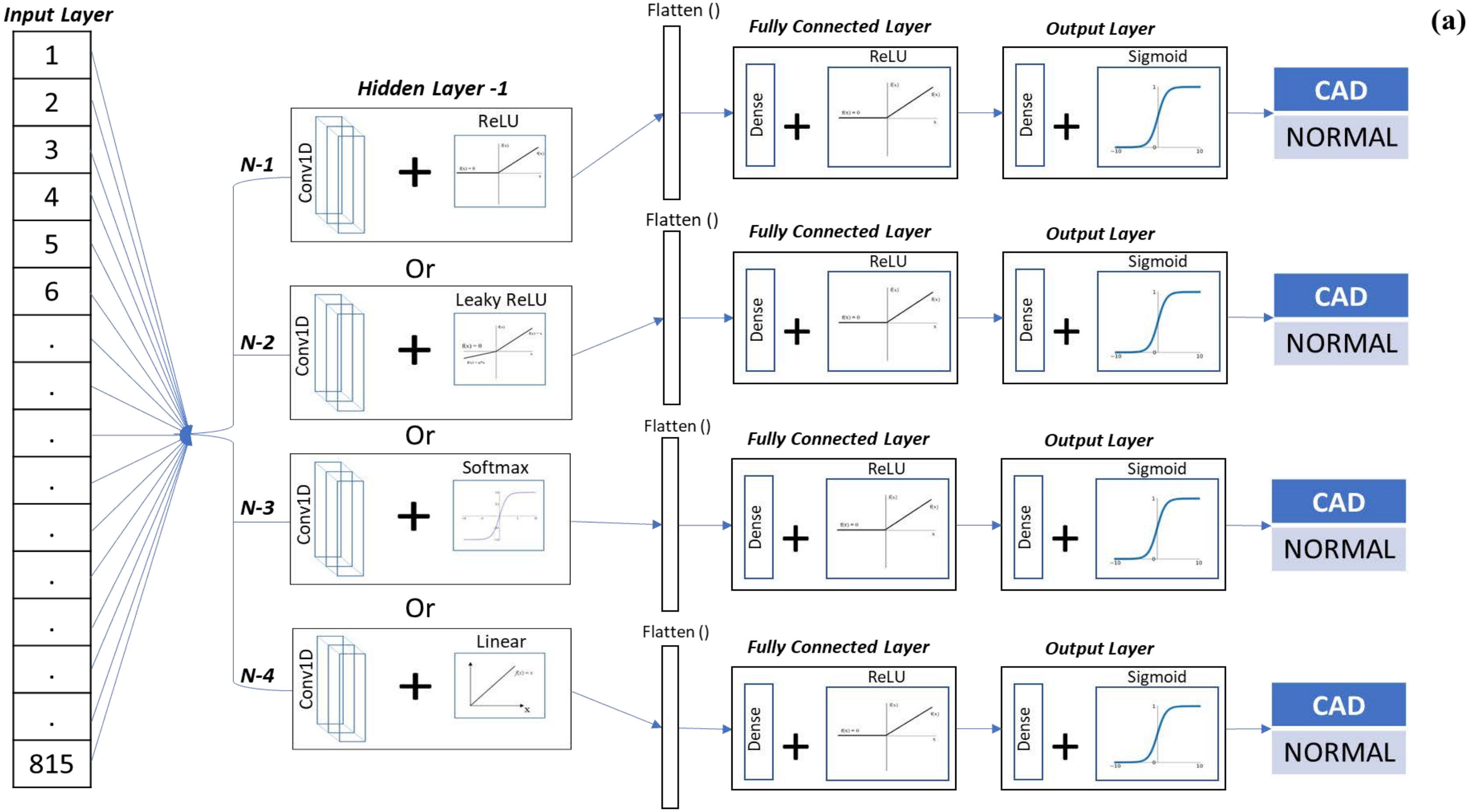
Table 5.1 Detailed CNN architecture for a single-lead ECG with one-convolutional, two-convolutional, and three-convolutional layers respectively

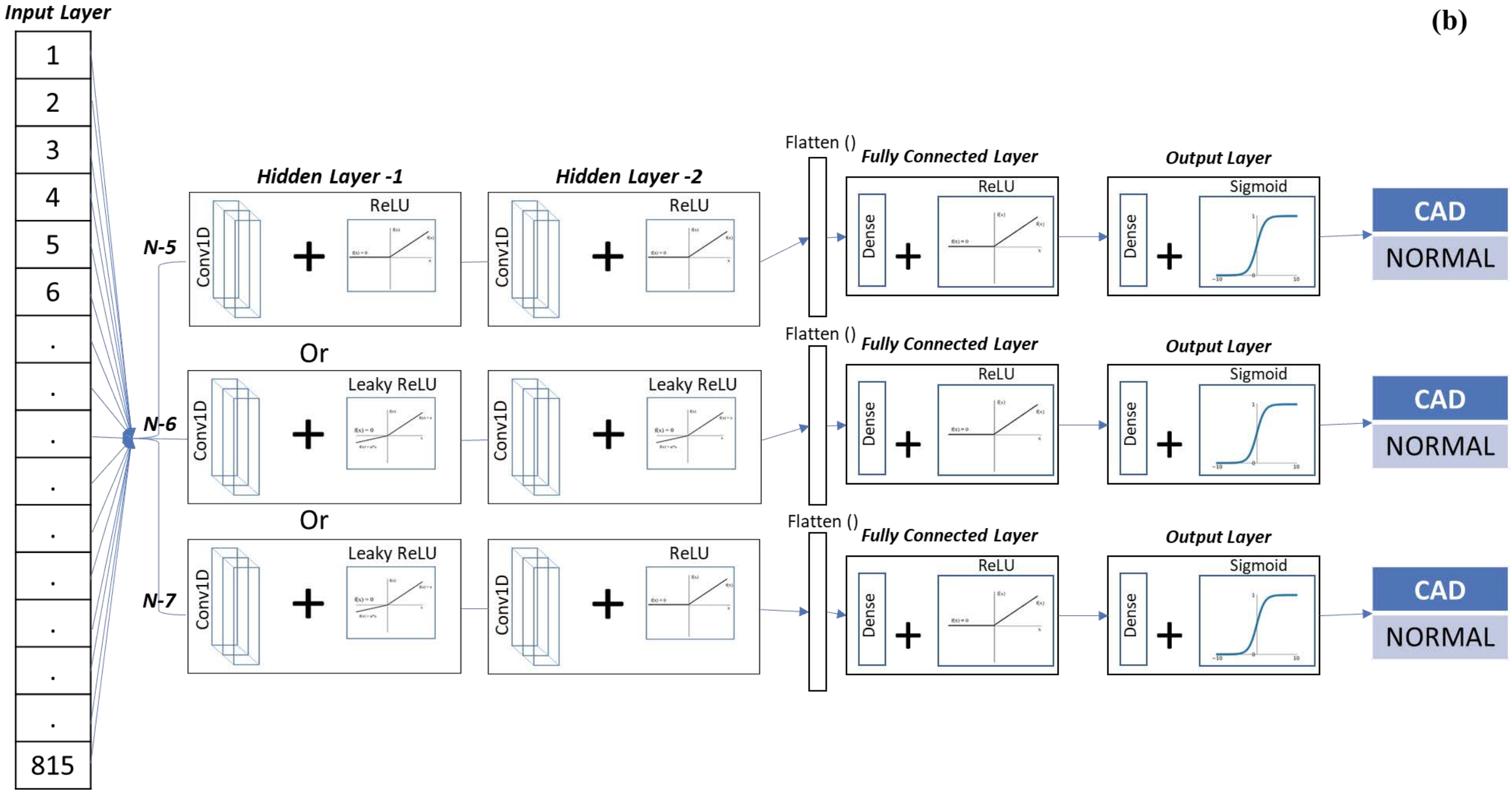
	Single Layer CNN				**Two Layer CNN**			**Three Layer CNN**		
	Network (N)									
	N-1	N-2	N-3	N-4	N-5	N-6	N-7	N-8	N-9	N-10
Input	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)	(815,1)
Hidden_1 Layer	Conv1D + RELU (814,64)	Conv1D + LeakyRELU (814,64)	Conv1D + Softmax (814,64)	Conv1D + Linear (814,64)	Conv1D + RELU (814,64)	Conv1D + LeakyRELU (814,64)	Conv1D + LeakyRELU (814,64)	Conv1D + RELU (814,64)	Conv1D + LeakyRELU (814,64)	Conv1D + RELU (814,64)
Hidden_2 Layer	None	None	None	None	Conv1D + RELU (813,32)	Conv1D + LeakyRELU (813,32)	Conv1D + RELU (813,32)	Conv1D + RELU (813,32)	Conv1D + RELU (813,32)	Conv1D + LeakyRELU (813,32)
Hidden_3 Layer	None	None	None	None	None	None	None	Conv1D + LeakyRELU (812,16)	Conv1D + RELU (812,16)	Conv1D + RELU (812,16)
Flatten	[52096]	[52096]	[52096]	[52096]	[26016]	[26016]	[26016]	[156624]	[156624]	[156624]
Fully Connected	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)
Output	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)

The following approach was adopted to arrive at a proper CNN architecture for effective prediction. Initially, four independent single-layer CNN models (N-1 to N-4), each with different activation functions, were developed, as shown in Table 5.1. All of the developed CNN models have inputs of sequential models (sequential model is a stack of plain layers, each contain one input and one output tensor) (Zhang et al. 2019b) (Figure 5.2a). As shown in Figure 5.2a, the output of the independent CNN architecture passes through a flattened layer and a fully-connected dense layer with sixteen filters. The sigmoid activation function was used in the output layer to classify the CAD and normal data. By this stage, the effective activation functions for the next step were identified.

Table 5.1 shows the layer-wise details of the output shape with the number of filters under one-dimensional data conditions. For example, the output shape of Conv1D + ReLU (814, 64) implies 814 numbers of the one-dimensional data flow from a particular layer and 64 channel filters to analyze these data to generate the corresponding output. The system was trained with 5304 (out of 6630) patients' data -a single-lead dataset, i.e., 80% over -an interval of time, and the data had 815 elements recorded, with respect to time. The dataset included all of the features (P, Q, R, S, T, QRS complex, and ST segment for training and testing purposes. The convolutional neural network model, which is trained for a standard ratio of 80% and 20% for training and testing, was validated based on CNN methodology.

Out of the four activation functions, the best two (ReLU and Leaky ReLU) were identified in the previous step. The two-layer CNN architecture with different combinations of these activation functions was developed (ReLU–ReLU, Leaky ReLU–Leaky ReLU, Leaky ReLU–ReLU) as shown in Figure 5.2b to obtain the best result. In the next step, the three-layer CNN architecture (Figure 5.2c) was developed from the knowledge of these results, leading to better results. Overfitting problem was observed with a higher number of layers after this and, hence, the three-layer CNN architecture (Figure 5.3) was finalized as the best one.


(a)
Input Layer
1
2
3
4
5
6
.
.
.
.
.
.
.
.
.
815
Hidden Layer -1
N-1
N-2
N-3
N-4
Conv1D
ReLU
Or
Leaky ReLU
Or
Softmax
Or
Linear
Flatten ()
Fully Connected Layer
Dense
ReLU
Output Layer
Dense
Sigmoid
CAD
NORMAL


(b)
Input Layer
1
2
3
4
5
6
.
.
.
.
.
.
.
.
.
815
N-5
N-6
N-7
Hidden Layer -1
Hidden Layer -2
Conv1D
ReLU
Or
Leaky ReLU
Flatten ()
Fully Connected Layer
Dense
ReLU
Output Layer
Dense
Sigmoid
CAD
NORMAL

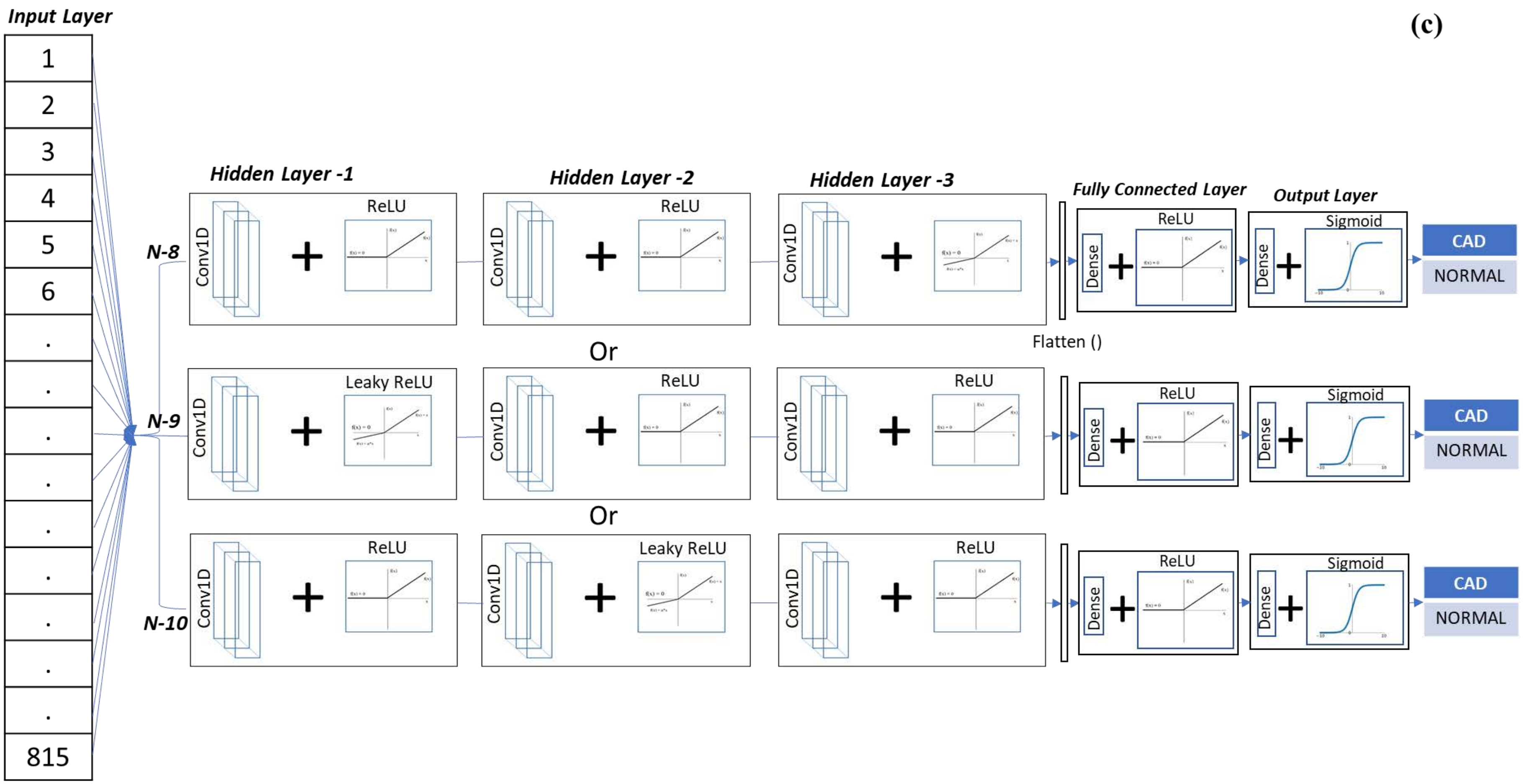


Figure 5.2 (a) Single CNN layer, (b) double CNN layer, and (c) triple CNN layer for the single-lead ECG/TMT-ECG

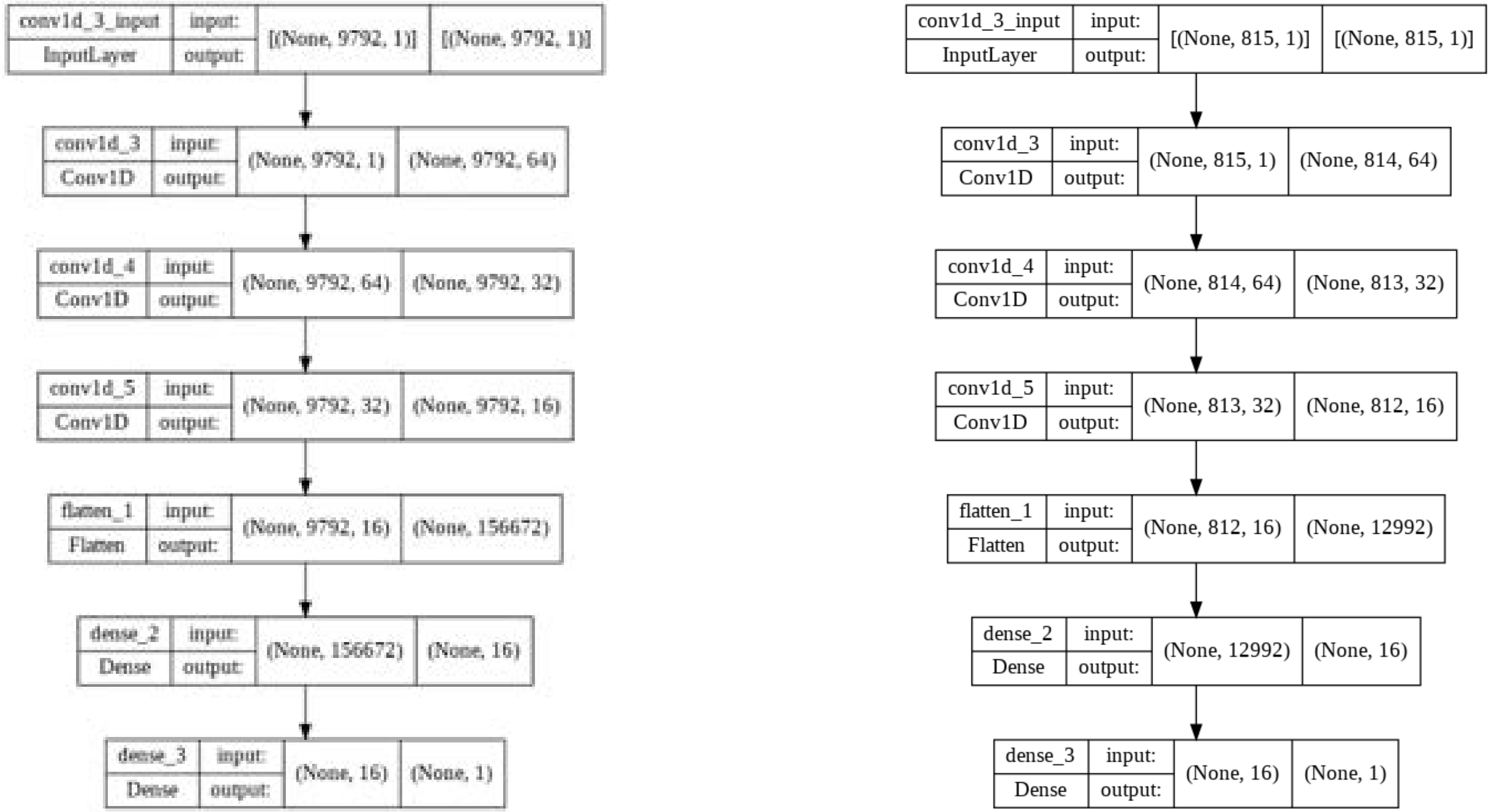


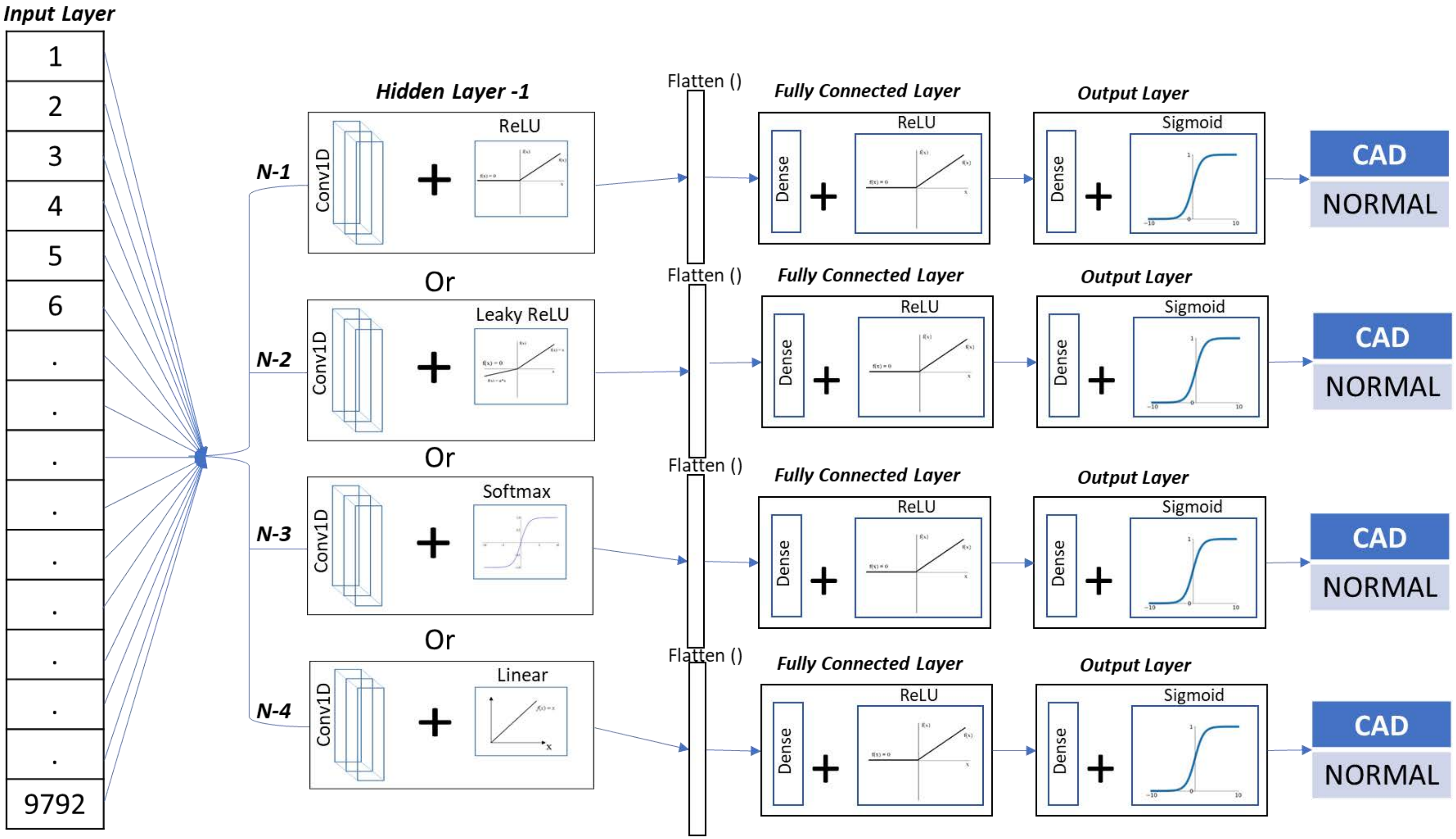
Figure 5.3 The proposed CNN architecture for single-lead and multi-lead (twelve lead) ECG/TMT-ECG respectively

5.1.2 Multi (12)-Lead ECG/TMT-ECG Based CNN Architecture

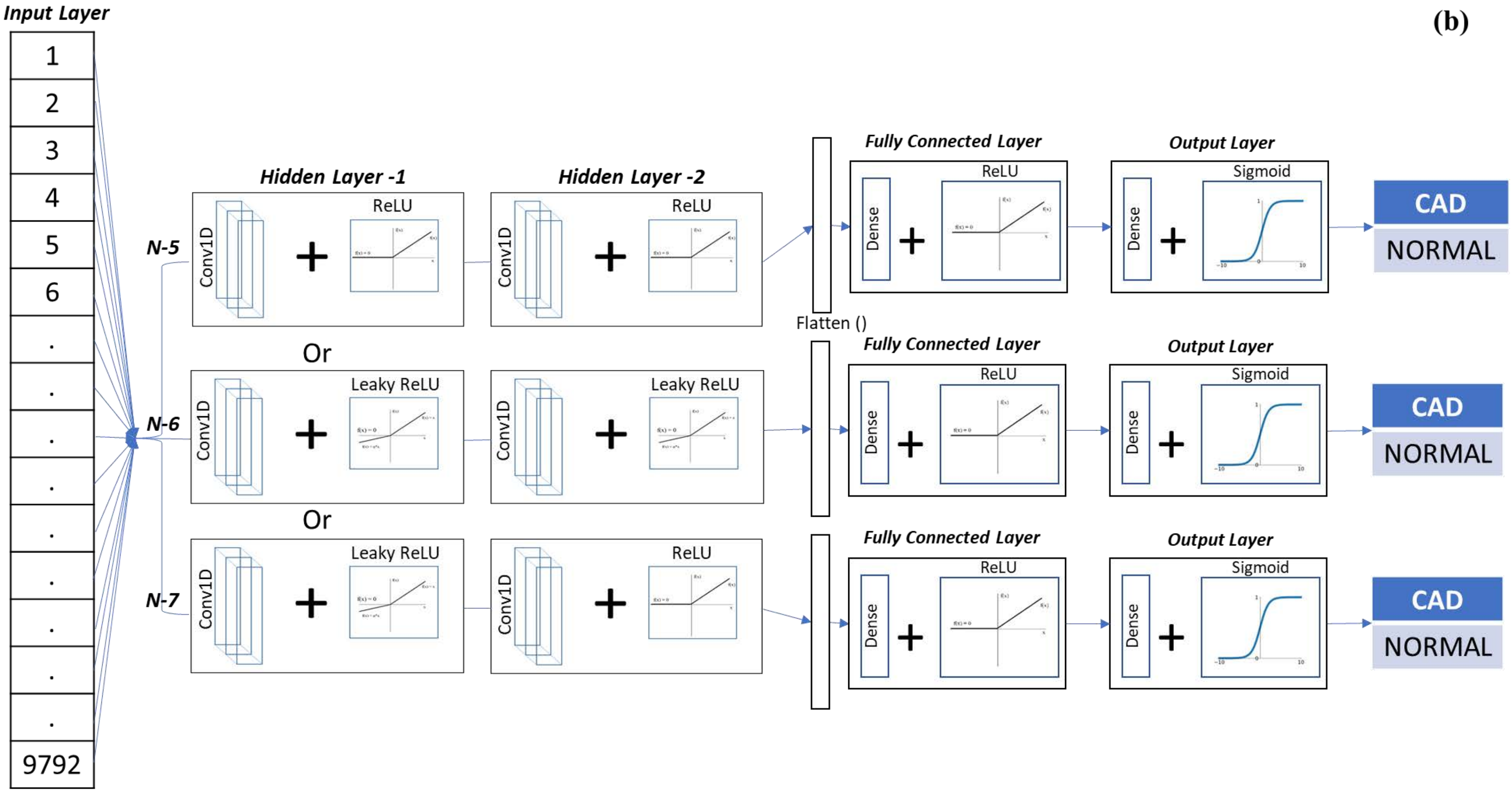
ECG machines in general will provide 12-lead ECG/TMT-ECG signals as output, i.e., I, II, III, aVR, aVF, aVL, V1, V2, V3, V4, V5, and V6, which correlate to the entire behavior of the heart, and each lead has different amplitude curves. The study of 12-lead ECG signals consists of all 12-lead ECG signal data, which are collected by the same procedure as with the single-lead ECG. These are in the form of a one-dimensional dataset. The resulting combined 12-lead one-dimensional dataset was used in the development of CNN architecture.

Since CNNs perform more effectively on nonlinear datasets (Acharya et al. 2017b), similar architectures that are used in single-lead ECG datasets are applied to 12-lead ECGs. The defined CNN architecture worked with the input of data gathered from 552 patients. Here, each sample had 9792 parameters instead of 815 as was the case in single-lead architecture as shown in Figure 5.4a, which was recorded over time intervals. Based on a standard protocol, 80% of the data were used for training and 20% for testing conditions.

(a)


Input Layer
1
2
3
4
5
6
.
.
.
.
.
.
.
.
.
9792
N-1
N-2
N-3
N-4
Hidden Layer -1
Conv1D
+
ReLU
Or
Conv1D
+
Leaky ReLU
Or
Conv1D
+
Softmax
Or
Conv1D
+
Linear
Flatten ()
Fully Connected Layer
Dense
+
ReLU
Output Layer
Dense
+
Sigmoid
CAD
NORMAL
Flatten ()
Fully Connected Layer
Dense
+
ReLU
Output Layer
Dense
+
Sigmoid
CAD
NORMAL
Flatten ()
Fully Connected Layer
Dense
+
ReLU
Output Layer
Dense
+
Sigmoid
CAD
NORMAL
Flatten ()
Fully Connected Layer
Dense
+
ReLU
Output Layer
Dense
+
Sigmoid
CAD
NORMAL

(b)


Input Layer
1
2
3
4
5
6
.
.
.
.
.
.
.
.
.
9792
N-5
N-6
N-7
Hidden Layer -1
Hidden Layer -2
Conv1D
ReLU
Or
Leaky ReLU
Or
Leaky ReLU
ReLU
Flatten ()
Fully Connected Layer
Dense
ReLU
Output Layer
Sigmoid
CAD
NORMAL

(c)

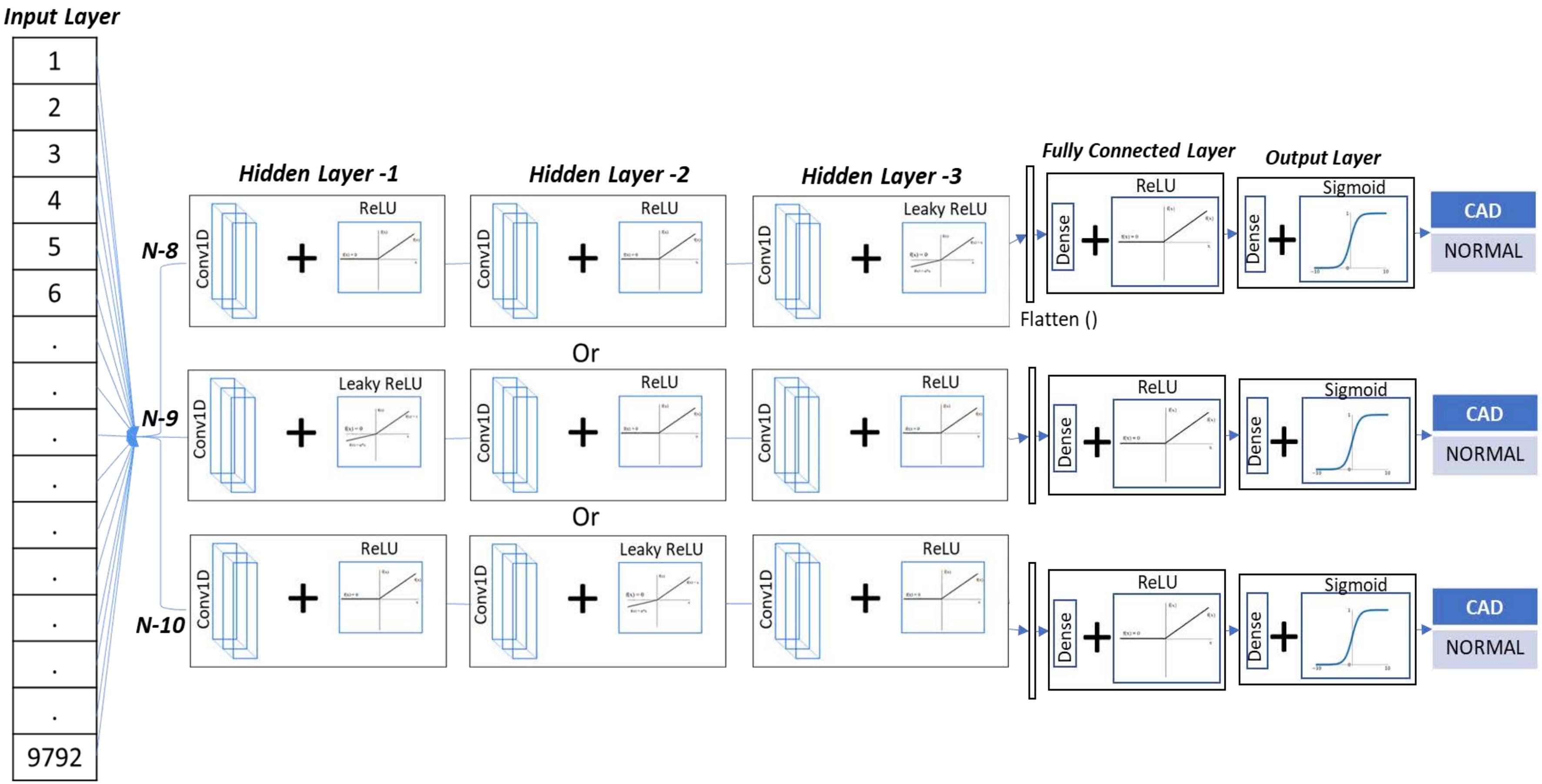


Figure 5.4 (a) Single CNN layer, (b) double CNN layer, and (c) triple CNN layer for the single-lead ECG/TMT-ECG

As mentioned in Table 5.2, four different independent single-layer convolutional networks were developed, each using different activation functions (ReLU, leaky ReLU, SoftMax, and linear, respectively) having a kernel size of 1 and stride of 1. The output in each of the CNN models (with a shape of (9791, 64)) was flattened and flown to the output layer through a fully-connected dense layer as illustrated in Figure 5.4a. The output function defined with a dense layer consisted of the sigmoid activation function (Rahhal et al. 2016). The sequential model is considered as input for all of the CNN models and the output classifications of these models were carried out through the binary classification process. Hence, all of the models were compared using the binary cross-entropy loss function (Rajeswari et al. 2012) with the Adam optimizer (The Theano Development Team et al. 2016). This process was most suitable to classify the data for ‘CAD’ or ‘normal data’.

The same procedure was extended to the two-layer CNN architecture, i.e., the sequential model followed by two CNN layers. The first-layer CNN architecture output shape of (9791, 64) was input to the second-layer CNN. This two-layer CNN architecture was developed using the best two of the effective activation function combinations (Table 5.2). Its output shape (9790, 32) was flattened (313280) and flown to the final dense layer (through a fully-connected layer) with a sigmoid activation function, as illustrated in Figure 5.4b.

In the next step, the three-layer CNN architecture was designed based on the knowledge gained from the development of the two-layer CNN architecture. The three-layer CNN architecture has a sequential model of three CNN layers with output shapes of (9791, 64), (9790, 32), and (9789, 16), respectively, and with the best combination of two activation functions. The flattened output of the last CNN layer was fed to the fully-connected layer. The final output function consists of a dense layer with a sigmoid activation function, with an output of ‘0’ or ‘1’. A value of ‘0’ indicates ‘normal health’ with respect to the ECG and a value of ‘1’ indicates the presence of ‘CAD’ (Figure 5.4c). Since it is a binary classification, the predefined binary cross-entropy loss function and Adam optimizer were used to analyze the 12-lead ECG data. To avoid the overfitting error and to obtain the best-optimized classification model, all convolutional neural network models (10 networks) were analyzed using a constant number of epoch values (25 epochs). The results of these combinational will help to derive an effective and suitable CNN system that is valid over both types of datasets (single-lead and multi-lead datasets) (Figure 5.3).

Table 5.2 Detailed CNN architecture for multi (12)-lead ECG with one-convolutional, two-convolutional, and three-convolutional layers, respectively

	Single Layer CNN				Two Layer CNN			Three Layer CNN		
	Network (N)									
	N-1	N-2	N-3	N-4	N-5	N-6	N-7	N-8	N-9	N-10
Input	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)	(9792,1)
Hidden_1 Layer	Conv1D + RELU (9791,64)	Conv1D + LeakyRELU (9791,64)	Conv1D + Softmax (9791,64)	Conv1D + Linear (9791,64)	Conv1D + RELU (9791,64)	Conv1D + LeakyRELU (9791,64)	Conv1D + LeakyRELU (9791,64)	Conv1D + RELU (9791,64)	Conv1D + LeakyRELU (9791,64)	Conv1D + RELU (9791,64)
Hidden_2 Layer	None	None	None	None	Conv1D + RELU (9790,32)	Conv1D + LeakyRELU (9790,32)	Conv1D + RELU (9790,32)	Conv1D + RELU (9790,32)	Conv1D + RELU (9790,32)	Conv1D + LeakyRELU (9790,32)
Hidden_3 Layer	None	None	None	None	None	None	None	Conv1D + LeakyRELU (9789,16)	Conv1D + RELU (9789,16)	Conv1D + RELU (9789,16)
Flatten	[626624]	[626624]	[626624]	[626624]	[313280]	[313280]	[313280]	[156624]	[156624]	[156624]
Fully Connected	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)	Dense + RELU (16)
Output	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)	Dense + Sigmoid (1)

5.2 TWO-DIMENSIONAL CAD ECG IMAGE PREDICTION ANALYSIS

A prediction algorithm developed with the help of a deep learning algorithm, to predict the CAD disease. Since Convolutional neural networks are more suitable to deal image-based dataset (two dimensional), a deep learning model is uses convolutional neural networks to develop prediction algorithm for ECG and TMT ECG signal images. This prediction algorithm is primarily developed in Python utilizing Keras with TensorFlow back-end and Sci-Kit learn modules. In comparison to the central processing unit (CPU), these algorithms can effectively perform in a Graphical Processing Unit (GPU). Because the GPU considerably reduces the image-based (two dimensional) algorithm's computation time. The proposed network model (Figure 5.5) is developed with accounting twelve lead ECG and TMT ECG signal images (raw and filtered) as well as single lead ECG and TMT ECG signal images extracted from the twelve lead signal images (raw, filtered).

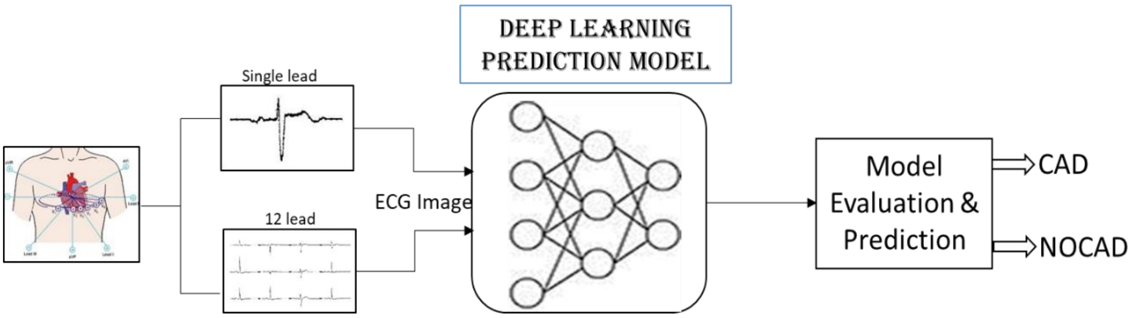


Figure 5.5 Generalized architecture of developed deep neural network for single lead and twelve lead ECG/TMT-ECG signal images

The prediction model here includes ECG and TMT ECG signal images of CAD and normal patients input. A total of 552 twelve lead signal images and 6,624 extracted single lead signal images comprise input. 85% of the total dataset for training, 10% are used validation, and 5% are used testing. The classification technique is developed based on two concepts: one with a consideration of the raw signal images immediately derived from the scanned ECG signal images, and the other one is considered as a filtered image, i.e., the preprocessed image.

5.2.1 Two-dimensional ECG prediction model

Two-dimensional prediction model, here is a deep learning model which is developed based on single lead ECG and twelve lead ECG signals. The model which is designed for CAD prediction is based on the effective factors that are considered in pre-trained models (transfer learned) to improve model accuracy (i.e. include effective depth, effective convolutional layer for feature extraction, and effective feature parameter selections in layer by layer). Based on this information, the designed architecture uses 5 convolutional layer blocks (64, 32, 32, 32 and 128 neurons) with a kernel size of 5x5 along with a ReLU activation function. The output of each convolution layer is processed through a max pooling layer with a size of 2x2. The layer-by-layer optimization is achieved by avoiding over fitting, with the application of dropout layer after the convolution layer and followed by max pooling layer. The flattened output of convolutions is considered as input to fully connected layers with 128, 128 and 64 neurons with in between 10% and 30% dropout respectively (Figure 5.6).

The fully connected layer designed with ReLU activation function is connected to output layer which consists of single neuron to represent one of the classes using sigmoid activation function. The model architecture was trained using the Adam optimizer with a learning rate of 0.001 with consideration of small parameter values (batch size = 128, epochs, callbacks, etc.). The developed model architecture is as shown in Figure 5.7. The designed model and pre-trained model are supported with two types of datasets (with image size 150 x 150 pixels). One is twelve lead dataset which consists of 512 patients' signal image data including normal and CAD patients corresponding to state of rest and exercised condition. The other one is similar dataset but it is related to single lead data (1 sec durational data) which are in total 6,624 signal images.

The performance of the proposed convolutional based deep neural network is compared with pre-trained networks. Pre-trained models use the weights learned from previous datasets (ImageNet) to apply on the current data set to achieve the desired accuracy. During the development of designed deep neural network condition, weights are generated over the dataset which are imported as current data. The standard protocol of data segregation is considered for both single and multi-lead data, i.e 85% for training, 10% for validation, and 5% for testing.

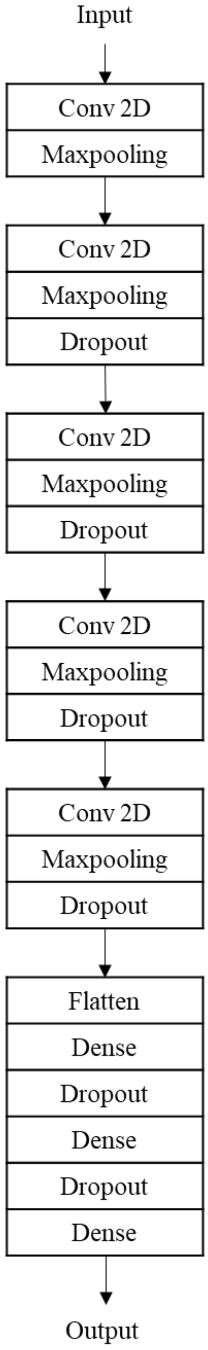


Figure 5.6 Layer details of deep convolutional neural network

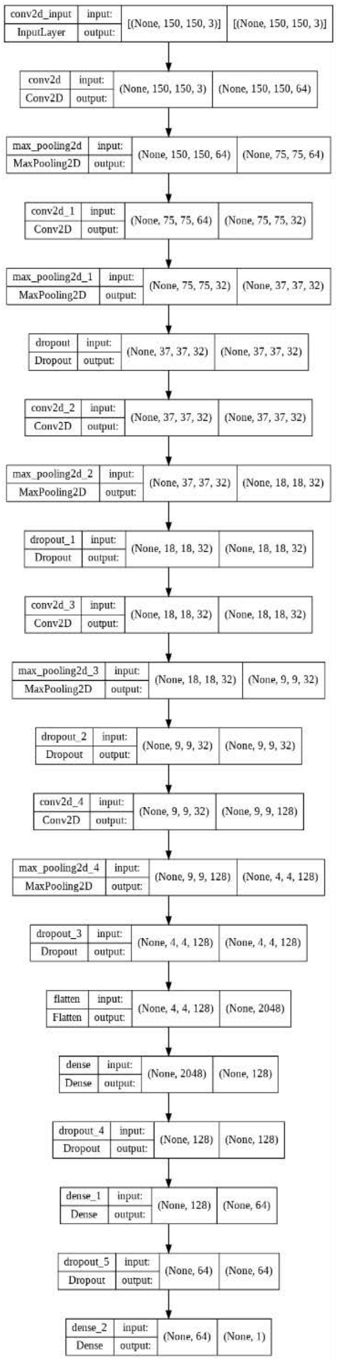

conv2d_input | InputLayer | input: | output: | [(None, 150, 150, 3)] | [(None, 150, 150, 3)]
conv2d | Conv2D | input: | output: | (None, 150, 150, 3) | (None, 150, 150, 64)
max_pooling2d | MaxPooling2D | input: | output: | (None, 150, 150, 64) | (None, 75, 75, 64)
conv2d_1 | Conv2D | input: | output: | (None, 75, 75, 64) | (None, 75, 75, 32)
max_pooling2d_1 | MaxPooling2D | input: | output: | (None, 75, 75, 32) | (None, 37, 37, 32)
dropout | Dropout | input: | output: | (None, 37, 37, 32) | (None, 37, 37, 32)
conv2d_2 | Conv2D | input: | output: | (None, 37, 37, 32) | (None, 37, 37, 32)
max_pooling2d_2 | MaxPooling2D | input: | output: | (None, 37, 37, 32) | (None, 18, 18, 32)
dropout_1 | Dropout | input: | output: | (None, 18, 18, 32) | (None, 18, 18, 32)
conv2d_3 | Conv2D | input: | output: | (None, 18, 18, 32) | (None, 18, 18, 32)
max_pooling2d_3 | MaxPooling2D | input: | output: | (None, 18, 18, 32) | (None, 9, 9, 32)
dropout_2 | Dropout | input: | output: | (None, 9, 9, 32) | (None, 9, 9, 32)
conv2d_4 | Conv2D | input: | output: | (None, 9, 9, 32) | (None, 9, 9, 128)
max_pooling2d_4 | MaxPooling2D | input: | output: | (None, 9, 9, 128) | (None, 4, 4, 128)
dropout_3 | Dropout | input: | output: | (None, 4, 4, 128) | (None, 4, 4, 128)
flatten | Flatten | input: | output: | (None, 4, 4, 128) | (None, 2048)
dense | Dense | input: | output: | (None, 2048) | (None, 128)
dropout_4 | Dropout | input: | output: | (None, 128) | (None, 128)
dense_1 | Dense | input: | output: | (None, 128) | (None, 64)
dropout_5 | Dropout | input: | output: | (None, 64) | (None, 64)
dense_2 | Dense | input: | output: | (None, 64) | (None, 1)

Figure 5.7 Developed deep convolutional neural network layer wise input and output parameter details

5.2.2 Transfer Learning approach

Transfer learning algorithm is implemented to analyze the full behavior of cardiac disease under single lead and twelve lead conditions for comparative purpose. The accuracy and computation loss of the proposed model were compared to a well-known pre-trained algorithm (VGG16, MobileNetV2, Inception, ResNet, and EfficientNet) in the field of image classification using the transfer learning methods (Figure 5.8). The implementation of these pre-trained algorithms with the help of transfer learning method for present data are discussed as follows and compared with proposed algorithm.

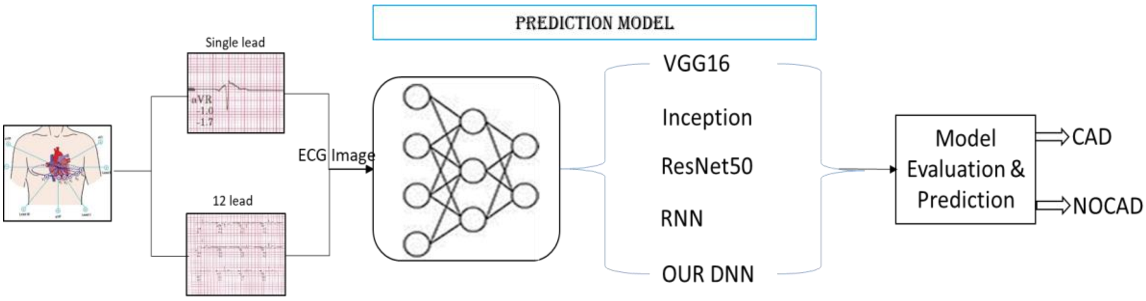


Figure 5.8 Generalized architecture of pre-trained neural network adopted for transfer learning approach

5.2.2.1 VGG16 transfer learning

VGG16 is made up of five convolution layer blocks which are then linked to a multi-layer perceptron (MLP) classifier. The first three convolution layer blocks of VGG16 are not altered during transfer learning adaptation, as they were previously trained on ImageNet dataset. The fourth and fifth block of the convolution are trained based on current dataset. This process is known as fine-tuning. This is because, the model architecture allows the model to adjust the weights which is more relevant to the current dataset. Fourth and fifth layers of the convolutional block are encoded with generic and reusable features, whereas the first three layers are encoded with specialized, trained dataset features. The weight updates in the fourth and fifth convolutional blocks will help to learn the changes of features present in the current dataset.

The last convolution block is connected to three layers of a multi-layer perceptron, the first two of which are hidden layers and the third one is the output layer. The first two hidden layers each have 128, 64 nodes, using a rectified linear unit (ReLU) as the activation function. The hidden layers are followed by dropout layer, which removes randomly 40% of the parameters. The output layer is made up of a single node with a sigmoid activation function that classifies data into any one out of two classes.

The Adam optimizer was used to train the model architecture, with a learning rate of 0.001. With standard split methodology, all datasets (training, validation, and testing) are flowing to model as per defined batch-size. For the lacking performance of the validation loss for two consecutive epochs, the model architecture is programmed to reduce learning rate by a factor of 0.5, and to cease training the model for future epochs if the validation loss does not improve for next 5 epochs.

5.2.2.2 MobileNetV2 transfer learning

During the implementation of the MobileNetV2 model, the classification layer parameter is determined by the very last layer before the flattening process. This layer response will alter based on changes in upper layer parameter. Since it keeps feature parameters related to importance of image rather than the generality features of the image, it is not crucial alterable as compared to bottom layer. The frozen convolutional layer is considered as a base model used for feature extraction process. These feature vector values are converted into actual prediction with application of global average pooling method. At the end of the process, this will transform feature vectors into 1280 element vectors.

Then, to forecast the existence of disease, these vectors are connected to dense layers consisting of 512 neurons with ReLU activation function and output layer consisting of single neuron with sigmoid activation function respectively. The model's performance is assessed by stacking it into a binary cross entropy loss function and an Adam optimizer, and the outcomes measured in terms of accuracy and loss function values with the help of previously defined trainable parameters and conditions.

5.2.2.3 Inception transfer learning

The 42 layers in the Inception model are known to consider this as deeper neural network for classification, and the performance of a class in Inception is primarily determined by the layer before the flatten layer. The initial learning technique is identical to MobileNetV2 and both networks are varied only in terms of considering the parameter values. During feature extraction, the weight of the convolution layer is frozen and are used to transform into real prediction, with the help of element vector (18432) parameter is extracted from the feature vector using global average pooling method. These element vectors are connected to output layer consisting of one neuron with sigmoid activation function followed by dense layer (512 neurons) having ReLU activation function with a 40% dropout. The trainable parameters (batch-size, learning rate etc.) considered are the as same as proposed network condition.

5.2.2.4 ResNet transfer learning

The ResNet model is built using dimensional convolutional layers with three different types of filters. The first and third filter have 1x1 kernel sizes, while the second filter has a 3x3 kernel size. All the three filters have the same stride. The residual blocks have three main stages. Except for the first stage, which has a 2x2 maximum pooling layer after the initial Conv2d layer, batch normalization is applied between two subsequent Conv2d layers having ReLU activation function. Output of each stage is combined with original data and is passed through ReLU activation function before moving to next stage.

A 2X2 average pooling layer’s output is connected to two dense layers together with output of third stage, the first of which has 4096 neurons and the second of which has 2048 neurons. During transfer learning, another three layers are mounted on ResNet's final layer. The first two layers of the three dense layers with 1024 and 512 neurons respectively are coupled to the ReLU activation function. The third one in the output layer which has single neuron with sigmoid activation function, followed by dropout layer of 40%, will helps to identify and classify the disease. To find the optimal model during classification, the model performance evaluation is carried out using the Adam optimizer and binary cross entropy loss function with the help of defined trainable parameter values considerations.

5.2.2.5 EfficientNet transfer learning

In general, any deep neural network is constructed with the goal of improving model performance, which leads to an increase in the number of units or layers. However, this method may or may not improve the performance. The EfficientNet aids in the provision of an effective compound scaling strategy (which scales all dimensions, i.e., depth/width/resolution) for increasing model size in order to obtain maximum accuracy.

The study covers only the optimum method (EfficientNet-B0) and its application to the current data. EfficientNet model is a pre-trained model that uses an image with a size of (224, 224). The distribution value of feature in the image are depends on all color channels (RGB) which can be avoided by normalizing. The image with dividing 255 (max pixel value) to each color segment along with pixels resulting rescaled image have features in the range of 0 to1. The EfficientNet pre-trained model is originally trained with ImageNet. These weights are used in training present data to extract current features. Then these features are fed into fully connected layers having 512 neurons with ReLU activation function. These are then connected to output layer with one neuron having sigmoid activation function to classify the CAD and normal patient ECG’s signals. The binary cross entropy loss function and adam optimizers are used to evaluate the performance of two classes with the help of standard trainable parameters.

5.3 MULTI-HEADED PREDICTION MODEL

The present chapter focuses on the development of a multi-headed, hybrid input CNN model for single-lead ECG classification. The literature (Mahmud et al. 2022) discusses utilizing 1D input data to for categorization of disease using a 1D multi-headed model. The developed multi-headed CNN model can process hybrid ECG data types concurrently, including ECG image data (2D) and time series data (1D). These input ECG data correspond to normal & coronary artery disease patients, which are collected as scanned signal images. A total of 6624 single-lead ECG readings from 552 people were collected and used as input. The input data are pre-processed in the previous stage to obtain noise-free signal images (Figure 5.9). The inputs for the multiheaded hybrid CNN model are divided into two categories: one category is one-dimensional time series data obtained from corresponding pre-processed image, and the other category is pre-processed single-lead ECG signal images of one-second duration (amplitude value of one second, ie. 815). These data are input into the model in the respective heads and are analyzed in parallel to create the prediction model (Multi-headed CNN model) (Figure 5.10).

The prediction model is concentrated mainly on two concepts, feature extraction and classification. Hence the model was built with the help of the convolutional neural network (CNN) because CNN is the most popular feature extractor model, but one drawback of CNN is that of losing spatial information in the training process. This can be compensated by using the Maxpooling layer, which holds only major information while training. Hence, it prevents the loss of some valuable feature parameters (Serhani et al. 2020; Wang et al. 2020). The dropout layer followed by the Maxpooling layer is implemented under necessary conditions to avoid the effect of overfitting problems during the training stage. The output of these models is merged and connected to fully connected layers to classify signals.

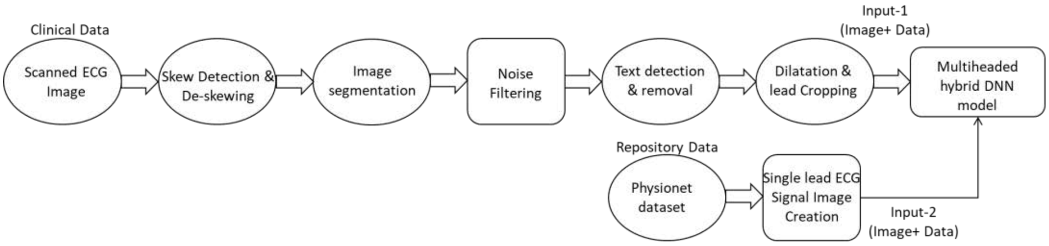


Figure 5.9 Pre-processed noise-free ECG/ TMT-ECG signal image extraction for multi-headed hybrid model

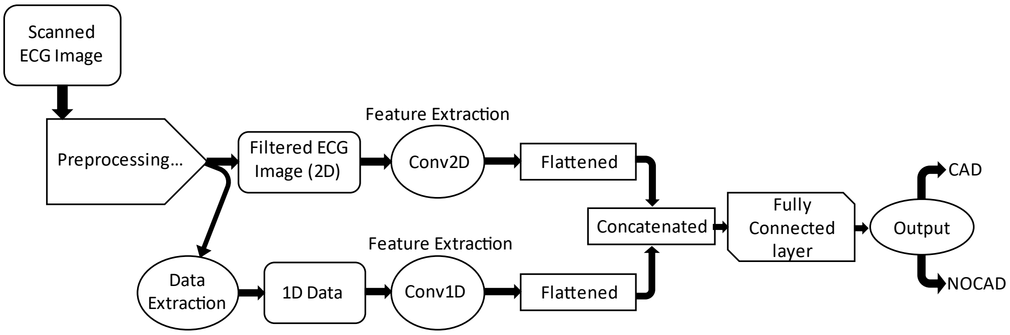


Figure 5.10 Generalized architecture of multi-headed hybrid ECG/ TMT-ECG prediction model

The multi-headed model is commonly defined as a two or more prediction models system processing simultaneously. This type of model is also known as the functional model. That models deal with coronary artery disease and normal data, i.e., a person's correlated pre-processed signal images and time series data. The coronary artery disease images are labeled as 'One' and the normal ones are labeled as 'Zero'. Hence it is evaluated as a binary problem by converting the problem's structure to binary.

The derivation best hybrid multiheaded model to predict coronary artery disease based on ECG alone and combined ECG & TMT-ECG data with a consideration of three ways,

1) ECG signal image (2D data) +Extracted ECG time series Data (1D data)
2) ECG signal image (2D data) + ECG flattened image (1D data)
3) ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data

All these three combinations rely on two types of datasets – single lead filtered data and twelve lead filtered data. The single lead data consist of 3312 ECG signal images and 3312 TMT ECG signal images. The twelve-lead data consist of 256 ECG signal images & 256 TMT ECG signal images. The first analysis is carried out (as discussed above) based on the consideration of ECG & ECG +TMT-ECG signal images together with corresponding ECG and ECG +TMT- ECG one-dimensional time series data. This study is carried out to identify the behavior of the multiheaded models under ECG alone and ECG with TMT-ECG data condition. This will help to identify the importance of the requirement of ECG and TMT- ECG in order to rectify the disease. The methodology also helps to study the behaviour of the multiheaded models under different data conditions. The different combinations of data being tried out here are: Single lead ECG, Single lead ECG + TMT-ECG, twelve lead ECG, twelve lead ECG + TMT-ECG.

Of the three conditions listed above, the first two corresponds to Two headed prediction model (Figure 5.11 and 5.12) and the last one will act as a three-headed model (Figure 5.13) with consideration of all types of data.

In development of first hybrid multiheaded model, the input of the pre-processed ECG signal images is scaled and transferred to feature extraction (encoded features of image) using a convolution neural network. The 2D convolution neural network (conv2D) with 6 layers is used as a feature extractor from input ECG signal images. Out of the six conv2D layers, every even layer is coupled to the Maxpooling and dropout layers. These convolutional layers extract the features of the ECG signal image. These convolution layers are made up of 32, 32, 64, 64, 128 and 128 filters, respectively. These convolutional layers are activated using the kernel size (2,2) & Rectified linear unit (ReLu) activation function. The encoded output of these convolutional layers is flattened and connected to fully connected layers with a filter size of 128, and 32 respectively having a ReLu activation function. The output parameters of these fully connected layers are combined with feature parameters obtained from a 1-dimensional convolutional network-based second functional head of the prediction model. Since the one-dimensional time series ECG data is used as input to the second head, extraction of features is done easily using a one-dimensional convolutional network (Conv1D). These features are extracted from 1-dimensional data using six conv1d layers.

Since convolution loses the important feature parameters, the max-pooling layers together with Conv1D are used to retain important features. The dropout layer is followed by the max-pooling layer to reduce the overfitting error. These Conv1D layers with filter dimensions of 32, 32, 64, 64, 128 and 128 are activated using the ReLu activation function and a kernel size of 2, respectively. Convolutional layer output is flattened and coupled to three fully connected layers with filters of 256, and 128 correspondingly. These layers are triggered with the ReLU activation function and the overfitting of results is controlled with the dropout layer. In order to connect the final output layer for classification, the output features of the fully connected layer are concatenated with the output features of the final layer of the first functional prediction model. The final output layer has a single filter with a sigmoidal activation function that indicates whether the findings belong to 1 (CAD) or 0 since the analysis is considered as a binary classification. The flow of the process is indicated in the Figure 5.11. The binary cross-entropy loss function and Adam optimizer are used to effectively categorize the outcomes of concatenated two sub-models. This analysis is carried out with consideration of the initial call-back function, i.e. reducing learning rate (lr.0.0001). If validation loss doesn’t improve in five successful epochs, the best model is saved in terms of validation loss and with early stop condition, iteration is stopped.

The developed prediction model is also examined using the CAD data set obtained from the repository (Physio net dataset). These 1D time series data sets were collected from 40 healthy participants and 7 patients with CAD. As explained in the data preparation, two-dimensional images are created from these one-dimensional data types, due to the requirement of one-dimensional and two-dimensional data types as input to the multiheaded model. The initial pre-processing stage is bypassed since the two-dimensional data ECG image of a single lead is created using the one-dimensional time series noise-free data that has already been recorded.

Under the current input conditions, the same multiheaded prediction model architecture is taken into account. This analysis is mainly carried out to find the behavior of the model under generalization conditions. The multiheaded hybrid prediction model has 2 types of input, the first one being the two-dimensional single lead signal image which is constructed from physio net data, and another one being the one-dimensional time series dataset of the corresponding single lead signal image which is provided in the parallel head. A prediction model was created by using 810 samples as input (where 360 are related to CAD and the remaining 450 are related to NOCAD), apportioning 85%

for training, 5% for validation, and 10% for testing. The model architecture, activation function, and hyper-parameters remain the same like in the earlier study. The trained multiheaded model with the Physio net dataset is also validated by adopting 10 fold cross-validation function. This helps to find a state of generalized model for the prediction of coronary artery diseases. Hence, model will capable analyzing any type of ECG signal image which are related to coronary artery disease classification.

The prediction algorithm model discussed above with the combination of ECG or ECG & TMT ECG's image and one-dimensional time series data is maintained same further all the three data type conditions. i.e model architecture which is mentioned in ECG / TMT-ECG data analysis is unaltered to remaining data type conditions. The condition maintained over above model are unaltered during analysis of other two multi headed models. This will help to maintain uniformity during the analysis of the model under all four types of data conditions over different networks.

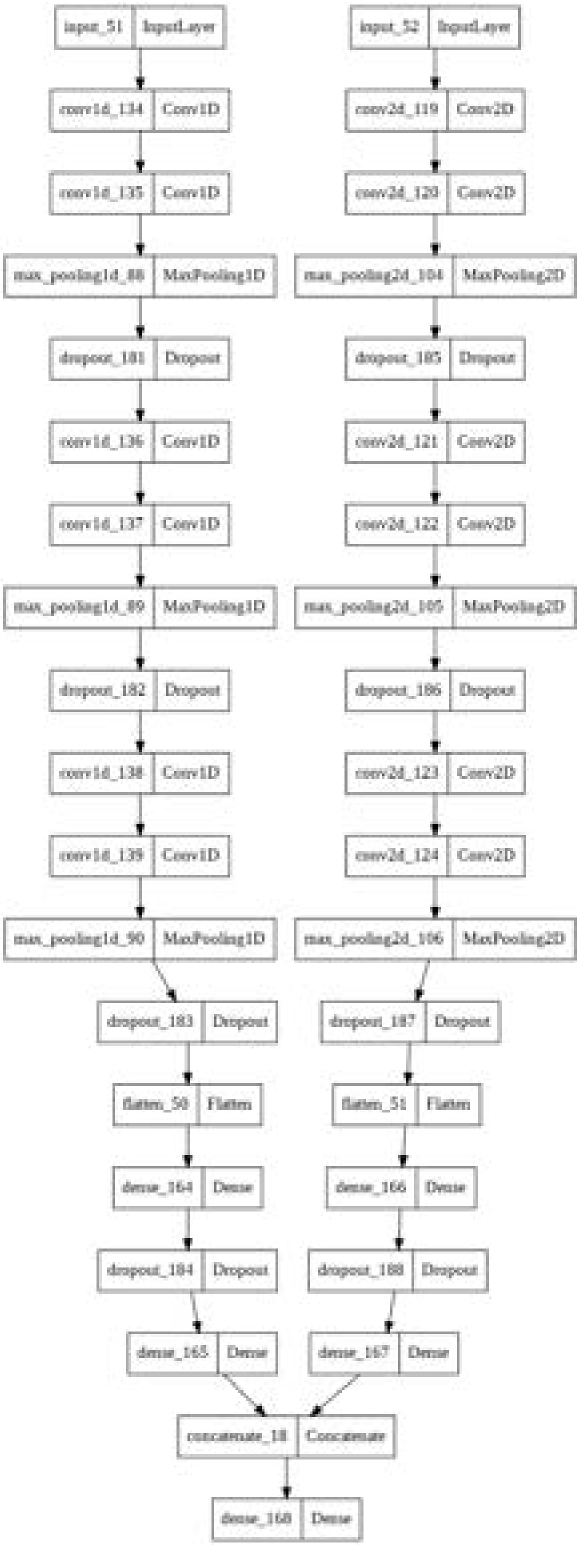


Figure 5.11 Multi headed deep neural network model architecture for (a) ECG signal image (2D data) +Extracted ECG time series Data (1D data) (b) ECG signal image (2D data) + ECG flattened image (1D data)

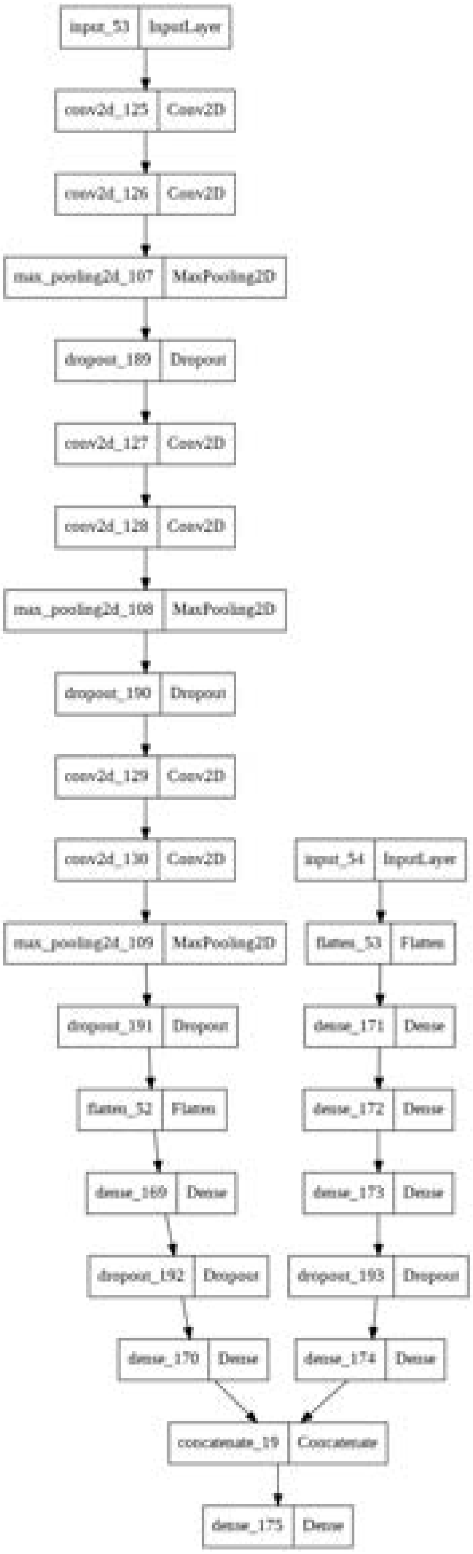


Figure 5.12 Multi headed deep neural network model architecture for ECG signal image (2D data) + ECG flattened image (1D data)

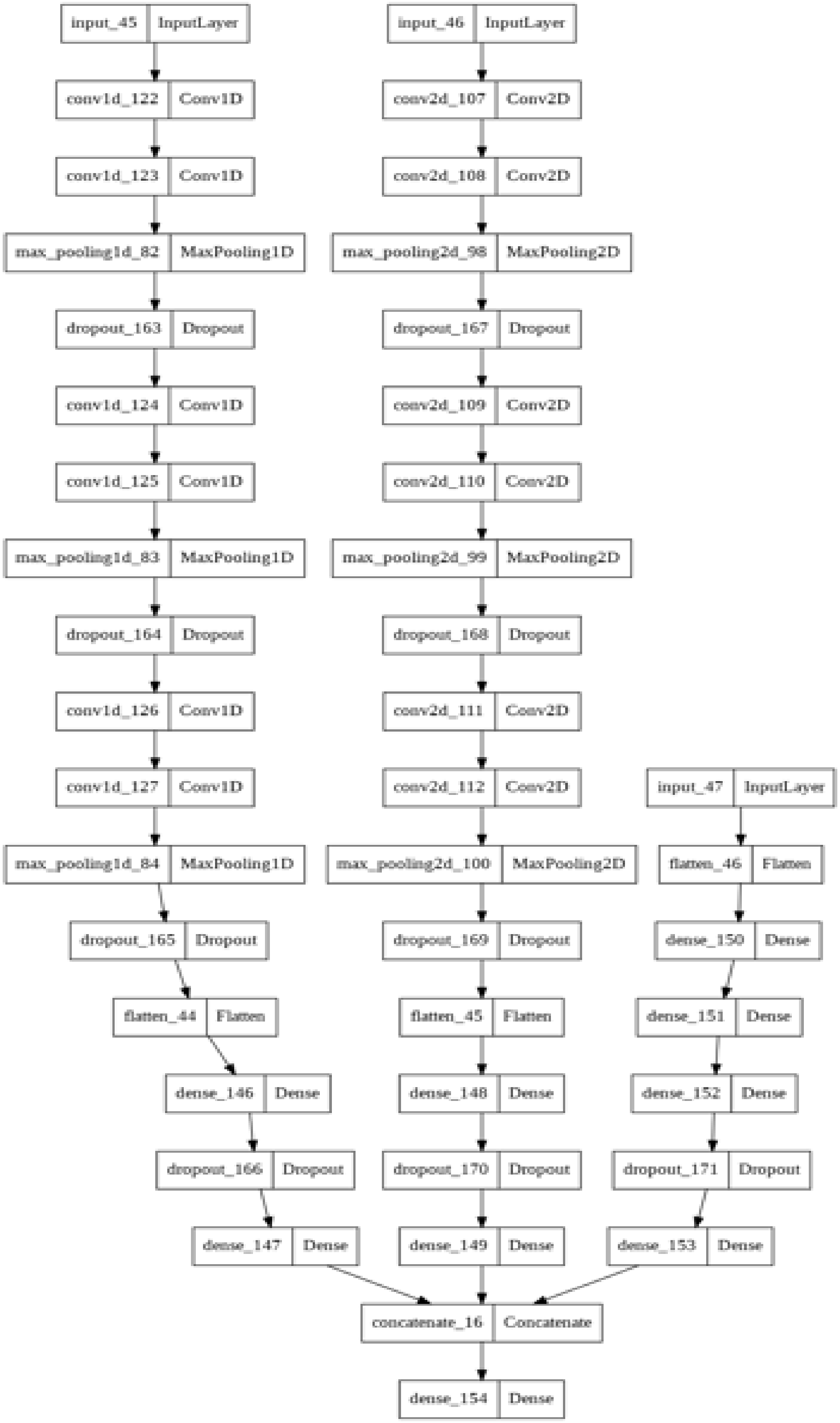


Figure 5.13 Multi headed deep neural network model architecture for ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data

Figure 5.12 shows the methodology of data flow during the second multi headed algorithm which deals input with ECG signal image (2D image Data) and ECG flattened image (one-dimentional image data) in corresponding heads of the multiheaded model. As explained in the first multi headed algorithm, the model has two headed which are handled with two different types of data & data feature handling algorithms. The end results of these are concatenated and connected to a fully connected & output layer to obtain the desired output. Here the one head of multiheaded algorithm input with ECG or ECG & TMT-ECG image (2D data) will be processed the same as explained in the first algorithm.

Another corresponding head which deals with the same ECG image type which is used in the previous head, but instead of extracting a feature of ECG. The image is flattened with the help of a flattening layer and imported to the model. This will helps to convert the two-dimensional image to a single dimensional image data and these data are fed to the fully connected network to study end-to-end features. These fully connected layers are made up of 16, 64, 256, & 32 filters respectively. The major unwanted features which effecting the output of the model is removed with the help of the dropout layer, which is inserted after 256 filtered layers. The resulting output parameters which are obtained over both heads are concatenated and connected to the output layer with a filter size of one. This will help to collect important information from both heads, which treat data in different ways and are combined to make end accuracy results improvement. Which may further optimize by using three headed model.

The third multi headed algorithm is one of the additional approaches which is considered. This is also a multi-headed model which deals with all the conditions discussed earlier in simultaneously. Unlike the previous two approaches of CAD, the prediction algorithm deals with all three conditions of data handling in three corresponding heads. The study investigates the behaviour of the model over different types of data parallelly. The study is conducted over both types of data, that is single lead and twelve lead with consideration of ECG or ECG& TMT-ECG signals. This multiheaded model has three heads to deal with three different data types. The algorithm used for these three data types is similar to the algorithm which is used for the two-headed model. This will helps to study uniformity related to the particular data handling method validation approach.

In the three-headed model (Figure 5.13), the first head of the algorithm deals with two-dimensional ECG image data. Another parallel head which handles 1D ECG time series data is input to the one-dimensional convolution network. It also combines results of the third head which is processed through flattened image analysis. The result of all these three heads is concatenated to combine all feature parameters and passed through the output layer to classify the disease.

The adoption of all three types of hybrid prediction models is focused on all the datatypes which include both single lead and multi-lead with ECG alone and combined ECG and TMT-ECG data, results are compared and stated in the results and discussion section. Based on these results best suitable hybrid algorithm is adopted. The following section is discussed about the detailed explanation related to results of one-dimensional datasets, two-dimensional dataset and multiheaded models for the all types of datasets (i.e. small, large).

CHAPTER 6

RESULTS AND DISCUSSIONS

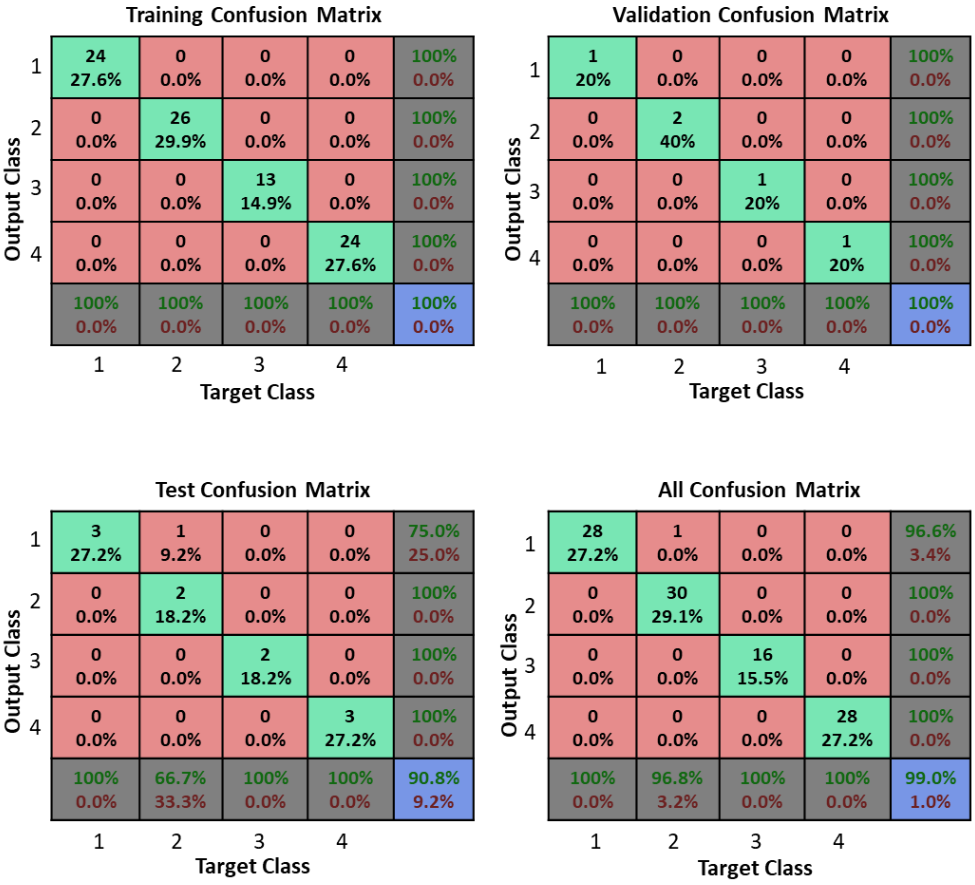
As already mentioned earlier, the development of AI system is carried out in two phases. In the first phase, an AI system is implemented for classifying symptomatic fever based on time dependent data. Based on the success of this AI system, a full-fledged AI based diagnostic prediction model for ECG analysis is implemented in the second phase.

6.1 FEVER SYMPTOMATIC DISEASE PREDICTION SYSTEM

The fever symptomatic disease prediction system was developed under MATLAB neural net pattern recognition environment based on the analysis of 24 hours of temperature data. The network was finalized for the least value of error at the end of training and validation. During this stage, results of several parameters like confusion plot, performance plot, and error histogram were taken into consideration. The confusion plot defines the relation between the output of trained network values and predetermined target values. The first four rows, columns of Table 6.1a - Table 6.1d represent the four types of disease which were initially defined. The final row and column define the percentage of correctness in identification. Diagonal values in the matrix define the number of patients suffering from the pre-defined disease of a particular category identified by the row number in the matrix.

Table 6.1a refers to the training performance of the given dataset for four types of diseases. This trained network was initially validated (Table 6.1b) and tested (Table 6.1c) through the initially defined percentage of the dataset from an overall data set. The acceptability of the results was verified by the overall Table 6.1d. The combination of the all three processes is defined in this overall Table. The bottom-right corner cell of this Table represents the overall accuracy of the trained network. A total of 99% accuracy was obtained from the trained network in predicting disease.

Table 6.1 Confusion matrix of 24-hour temperature prediction model during (a) training, (b) validation, (c) testing, and (d) overall, respectively [**Note:** Row ‘1’ represents Tuberculosis, Row ‘2’ represents Non-Tuberculosis, Row ‘3’ represents Dengue fever and Row ‘4’ represents Non-infectious disease]



Training Confusion Matrix

	1	2	3	4	
1	24 27.6%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	26 29.9%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	13 14.9%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	24 27.6%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Validation Confusion Matrix

	1	2	3	4	
1	1 20%	0 0.0%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	2 40%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	1 20%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	1 20%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%

Test Confusion Matrix

	1	2	3	4	
1	3 27.2%	1 9.2%	0 0.0%	0 0.0%	75.0% 25.0%
2	0 0.0%	2 18.2%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	2 18.2%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	3 27.2%	100% 0.0%
	100% 0.0%	66.7% 33.3%	100% 0.0%	100% 0.0%	90.8% 9.2%

All Confusion Matrix

	1	2	3	4	
1	28 27.2%	1 0.0%	0 0.0%	0 0.0%	96.6% 3.4%
2	0 0.0%	30 29.1%	0 0.0%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	16 15.5%	0 0.0%	100% 0.0%
4	0 0.0%	0 0.0%	0 0.0%	28 27.2%	100% 0.0%
	100% 0.0%	96.8% 3.2%	100% 0.0%	100% 0.0%	99.0% 1.0%

The plotting of Receiver Operating Characteristics (ROC) (Azar and El-Metwally 2013; Vijayashree and Sultana 2018) is a technique to visualize, organize and classify diseases based on the performance of data. It represents the specificity and sensitivity of the data. In the ROC curve represented in Figure 6.1, the curve tends to the left and top edges of the plot indicating better classification. ROC graphs are commonly used in medical decision-making.

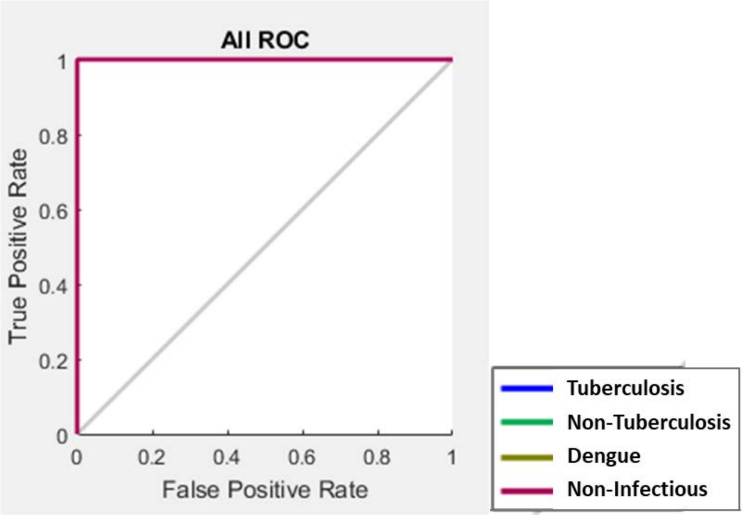


Figure 6.1 ROC plot of 24-hour temperature prediction model during overall condition

Figure 6.2. Represents the performance plot of the trained network, defining the nature of error reduction. Generally, the errors reduce after several epochs (46 epochs) of training. An increasing trend of the validation plot signifies the onset of overfitting.

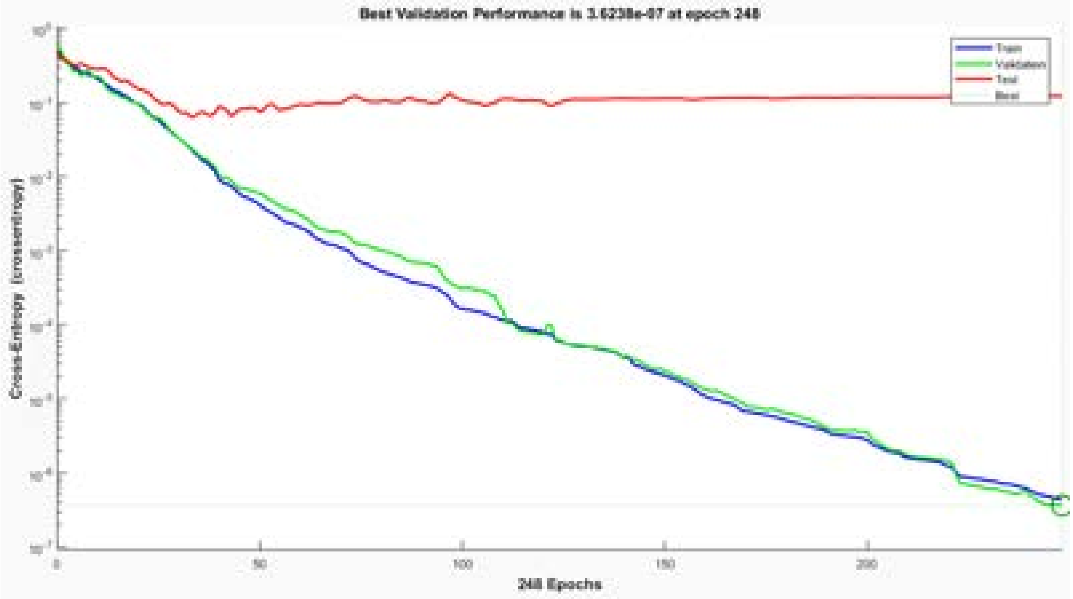


Figure 6.2 Performance plot of 24-hour temperature prediction model

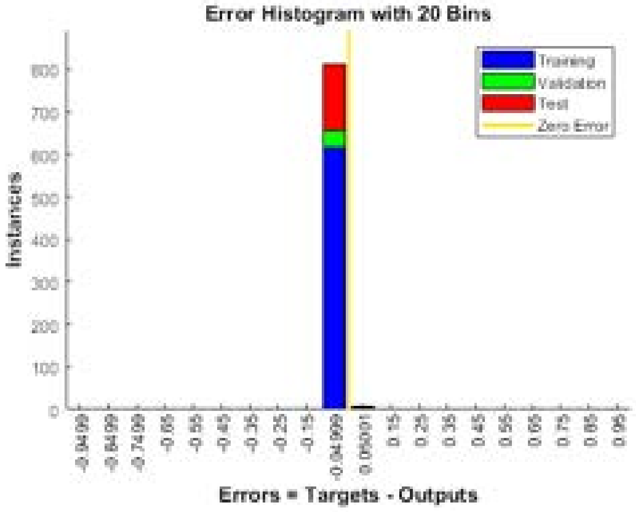


Figure 6.3 Error histogram plot of 24-hour temperature prediction model

Because of the default setting, the training process is shut down once six consecutive increases in the validation error are observed. The epoch with the lowest validation error indicates good performance as shown in Figure 6.3. Frequency histogram of error distribution (Jacobé de Naurois et al. 2019) (Figure 6.3) also represents the error variations between target (real) values and predicted (estimated) values in a better way. 99% of training errors are below 0.1, and validation and testing errors are below 1, resulting in good prediction.

Matlab mainly deals with support vector machine classifiers. Hence to understand the behavior of all classifiers over a defined data, it is further analyzed with major generative and discriminative classifiers with the application of machine learning technique using Python workbench (Jupiter Notebook). For most of the practical data, discriminative classifiers (kNN and decision tree) have shown better accuracy than the generative classifier. Learning the parameters of a prediction function for a data set and testing it on the same data is a methodological mistake. The model would just try to repeat labels of sample data, which automatically results in a perfect score. But it will fail to predict anything useful on untried data, where the situation is called overfitting. To avoid this, evaluating estimator performance (cross-validation) is a common practice while performing supervised machine learning, by using a part of the test data set.

Table 6.2 Comparison of results between discriminative type classifier and generative type classifier-based model

Approached Model	**Accuracy Score**	**Error**	**F1-Score**
Logistic Regression	0.2461	0.7539 (75.3%)	0.2
Decision Tree Classifier	0.996	0.0051 (0.5%)	1
K-Neighbors Classifier	0.79	0.21(21%)	0.78
Linear Discriminant Analysis	0.281	0.719 (71.9%)	0.3
Gaussian Naive Bayes classifier	0.3007	0.6993 (69.93%)	0.28
Support Vector Machine	0.366	0.6464 (0.67%)	0.39

Table 6.2 explains the classification prediction in terms of ‘Medically classified’ and ‘Model predicted’ for the total number of four disease cases among classifiers such as logistic regression, decision tree classifier, k-nearest neighbors classifier, linear decrement analysis, Gaussian Naive Bayes and Support Vector Machine.

From the Table 6.2, it is observed that the results of the decision tree model provide the best behavior for such type of data. In addition to the accuracy score of all the models, F1 score provides proof of the suitability of the model. The weighted average of precision (Figure 6.4a) and recall (Figure 6.4b) is known as F1 score (equation 6.1). It takes both false negative and false positive data into account. In case of an uneven class of distribution, F1 score will be more accountable when compared to the accuracy score. F1 score is calculated as,

$$F1\ Score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{6.1}$$

Models which have good precision value and recall ability will end with better results of F1 score. Results of other models have failed to provide good results of data prediction due to lack of common pattern behavior, dissimilarity in nature and absence of preprocessing methods to improve the accuracy of the model. These abnormal behavioral data are predicted in a better way in case of discriminative classifiers (Zhang et al. 2019a).

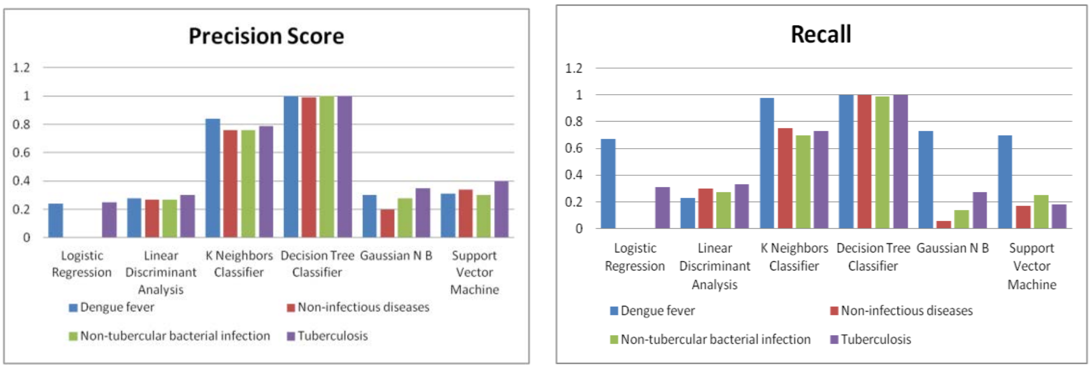


Figure 6.4 (a) Precision score and (b) Recall ability of diseases with different classifier conditions.

The accuracy of results varies based on the training methods adopted by the classifier. The errors depend on this variation and F1 score. The accuracy value (close to 1) defines the best-suited method for these types of data (Table 6.2). Upon observation of the above comparison results, the decision tree classifier (Azar & El-Metwally, 2013) and kNN provided good classification accuracy and F1 score results.

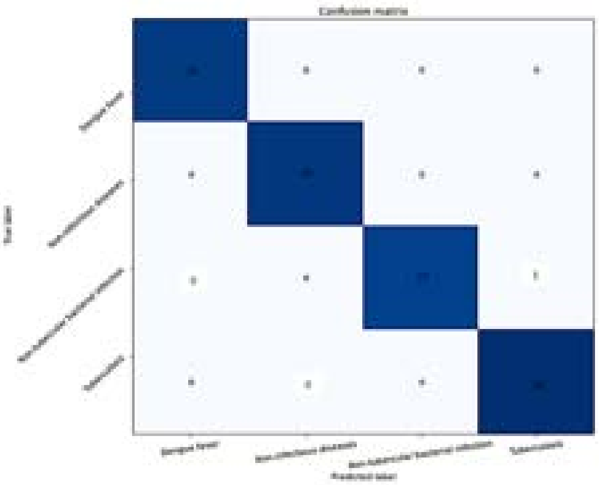
Two sets of confusion matrix have been constructed using the kNN as well as decision tree classifier results. The category specific data distribution in the confusion matrix cells gives an indication of the accuracy of the model. Overall percentage (Table 6.3) is obtained by summing the diagonal cell values of the confusion matrix. The excellent model will have distribution of prediction values in the diagonal matrix only. If it does not show a diagonal matrix, then it is an indication of the error quantity. This may be due to improper handling of equipment (temperature measurement), improper observation, error due to wrong labeling, etc…

Table 6.3a Confusion matrix results for best classifier algorithm.

	k- Nearest neighbor Algorithm (%)	Decision Tree Algorithm (%)
Training	82%	100%
Validation	79%	100%
Testing	77%	99%
Overall	79%	99.6%

As can be seen above, kNN and decision tree classifier provided significantly very good results. Model from the decision tree classifier is saved for future, for a similar type of data prediction application. Classification metrics of decision tree classifier is showed in the table 6.3b. By saving these trained models for the future, need for further training is avoided.

Table 6.3b Confusion matrix of Decision tree model



6.2 REFINEMENT OF FEVER PREDICTION SYSTEM

Continuous collection of patient data for 24 hours is a tedious process. To overcome this problem, short-duration data are considered for the same algorithms. Short-duration data sets are formed based on the method of splitting. Analysis is done with and without additional features. The additional features considered for refinement are ESR (Erythrocyte sedimentation rate), WBC (White blood cell count), Neutrophils, Basophils, Eosinophils, Monocytes, Lymphocytes, Platelets, Age, Body mass index (BMI), Spontaneous bacterial peritonitis (SBP), Diastolic blood pressure (DBP) and Pulse.

Specifically, the two conditions are:

(i) Diagnostic prediction system with temperature data alone

(ii) Diagnostic prediction system with temperature data and additional features.

6.2.1 Diagnostic prediction system with temperature data alone

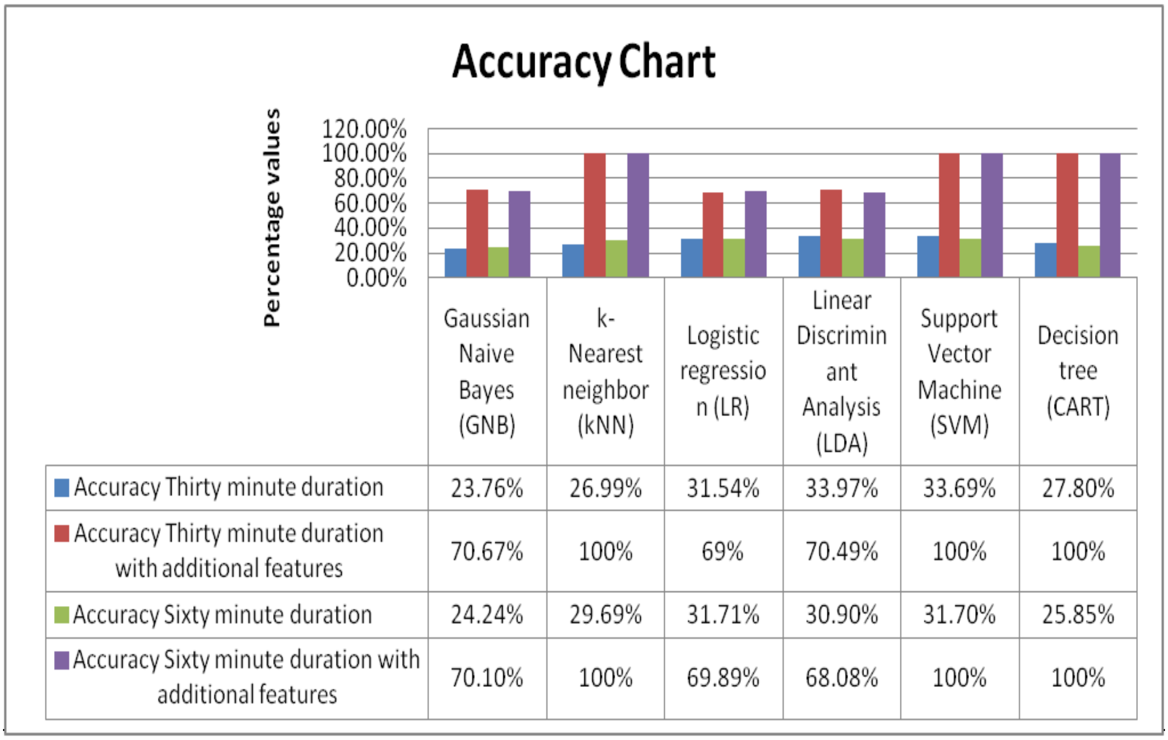
Prediction system is developed based on temperature data of the patient. Prediction system training and testing are fully on the single data set called temperature data. The model is tried out for 30min and 60min of temperature data, for final comparison with the 24 hours data-based model. 30 minutes and 60 minutes data of 103 patients results in a total of 4944 and 2472 sets of data respectively.

Case-1: In this case, a system with 30min data with one-minute interval from 103 patients has been developed. Out of 4944 data sets, 20% data is selected for testing (i.e 989 sample data sets collected each having 30 minutes of temperature variation information). Out of this, 132 were cases of Dengue fever, 273 belong to Non-infectious diseases, and 314 correspond to non-tubercular bacterial infection and the remaining 270 belong to Tuberculosis. For most of the practical unsymmetrical data, discriminative classifiers (kNN, decision tree) have shown better results than the generative classifier. Based on the literature, major types of discriminative classifiers with one generative classifier have been considered for study.

Accuracy is defined based on the required output for the predefined model. The confusion matrix is defined for the entire model to understand the behavior of the model. The comparisons of the above-mentioned algorithms are initially made based on the accuracy score of the model (Table 6.4). The best-suited model for initial data is then taken into account for further processing on the classification of diseases.

In addition to the accuracy score of all the models, the F1 score will provide proof of the suitability of the model. F1 score results varied based on the training methods for all the classifiers. The F1 score value (close to 1) defines the best-suited method for these types of data (Figure 6.4). Upon observation of the above comparison results, decision tree and kNN classifiers provided good classification accuracy and F1 score (Figure 6.5), with the consideration of additional features.

Table 6.4 Accuracy Score obtained for all the classifiers under all four prediction system categories.



	Gaussian Naive Bayes (GNB)	k-Nearest neighbor (kNN)	Logistic regression (LR)	Linear Discriminant Analysis (LDA)	Support Vector Machine (SVM)	Decision tree (CART)
🟦 Accuracy Thirty minute duration	23.76%	26.99%	31.54%	33.97%	33.69%	27.80%
🟥 Accuracy Thirty minute duration with additional features	70.67%	100%	69%	70.49%	100%	100%
🟩 Accuracy Sixty minute duration	24.24%	29.69%	31.71%	30.90%	31.70%	25.85%
🟪 Accuracy Sixty minute duration with additional features	70.10%	100%	69.89%	68.08%	100%	100%

Results of other models have failed to provide good results in prediction due to lack of common pattern behavior, dissimilarity in nature, insufficient features, and no preprocessing methods which are adopted to improve the accuracy of the model. These abnormal behavioral data are predicted in a better way in the case of discriminative classifiers.

Case-2: In this case, a system with data for a duration of 60min with one-minute interval for 103 number of patients has been developed. Out of 2472 data sets, 20% data is selected for testing (i.e 495 sample data sets collected each having 60 minutes of temperature variation information). Out of this, 69 were cases of Dengue fever, 126 belong to Non-infectious diseases, and 159 correspond to Non-tubercular bacterial infection and the remaining 141 belong to Tuberculosis. Even in this case, discriminative classifiers (kNN, decision tree) have shown better results than the generative classifier. The accuracy score of different classifiers is as listed in the Table 6.4. To improve the accuracy of prediction, the dataset is enriched with a set of additional parameters corroborating the temperature data.

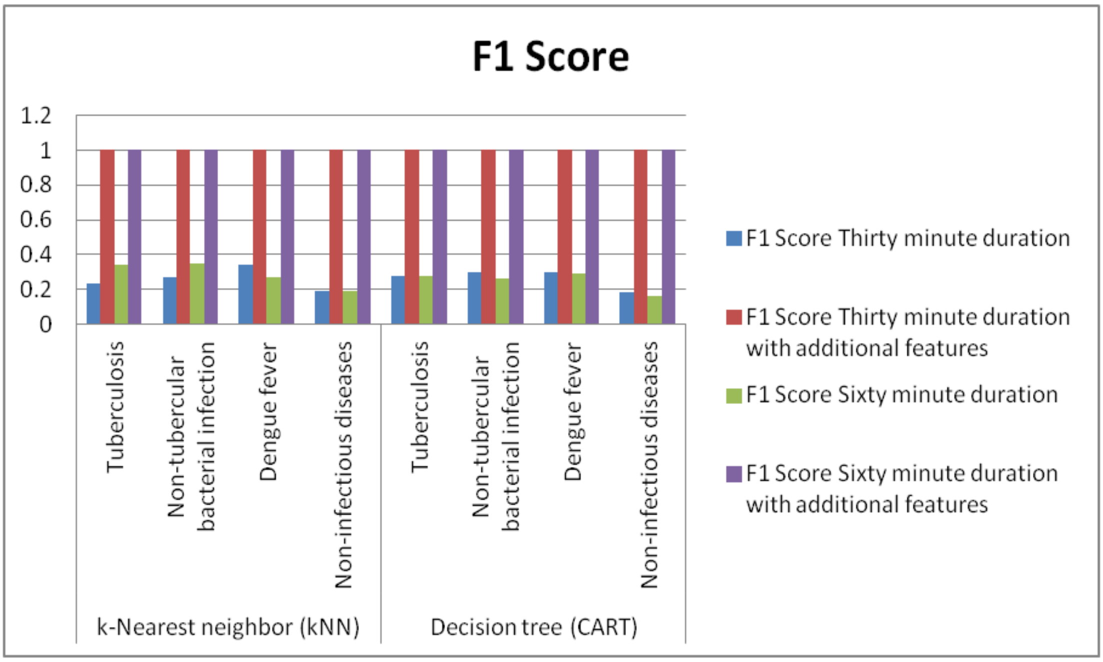


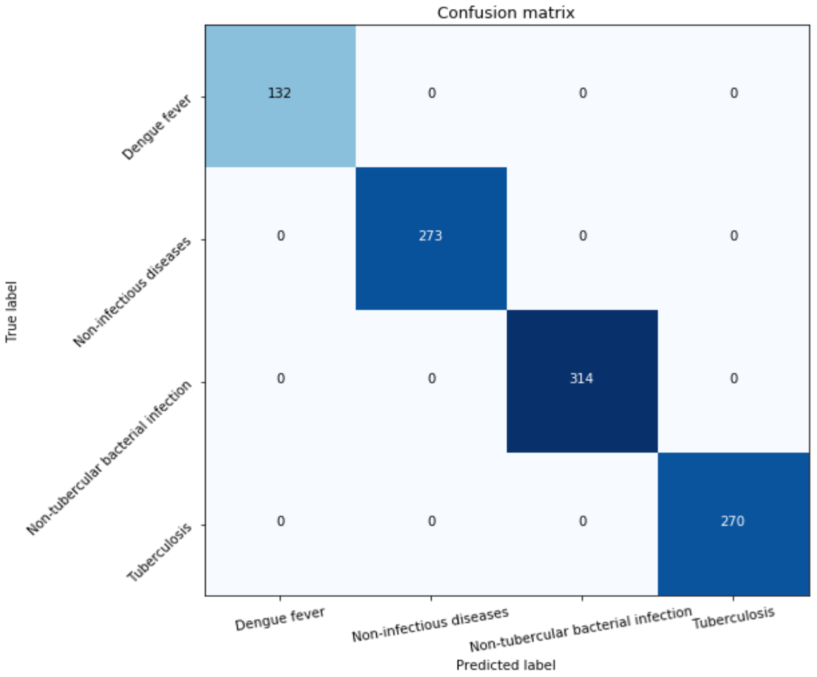
Figure 6.5 Evaluation of F1 scores for Dengue fever, Non-infectious diseases, Non-tubercular bacterial infection, and tuberculosis disease under the different prediction systems.

6.2.2 Diagnostic Prediction System with Temperature Data and Additional Features

As observed from Table 6.4, the single parameter dependent prediction system has not yielded accurate results for most of the cases in testing. Accuracy drops (around 28%

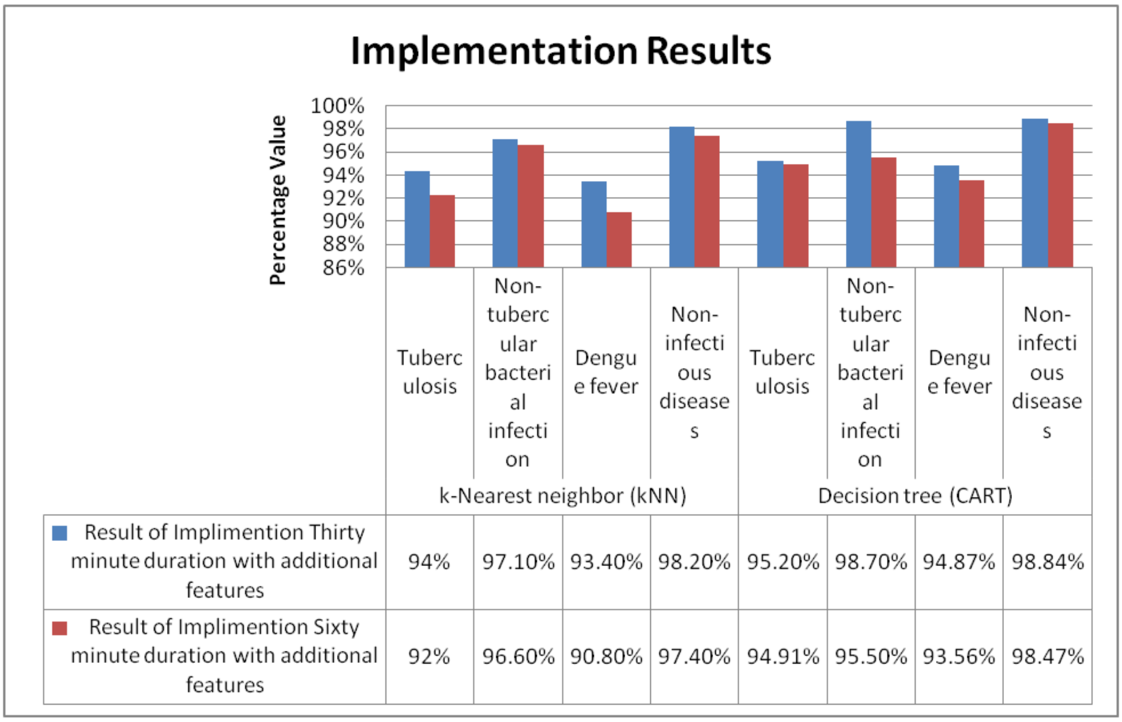
under 30min data with decision tree classifier and around 27% under 60min data with decision tree classifier) with untried data indicates the data samples used for training are not sufficient to make a proper classification, thus hampering the ethical standards. An error occurred in a defined system is reduced by either increasing the data set quantity or consideration of known compulsory medical test details as additional features. Added additional features are helping the system to learn effectively and accurately predict for unseen data with trained experience. Additional features which are considered for refinement of the temperature prediction system are ESR (Erythrocyte sedimentation rate), WBC (White blood cell count), Neutrophils, Basophils, Eosinophils, Monocytes, Lymphocytes, Platelets, Age, BMI, SBP, DBP, Pulse. A combination of additional features and temperature data is good enough to produce better results under training and implementation over unseen data.

Table 6.5 Confusion Matrix of Decision Tree (CART), kNN and Support vector machine for Dengue fever, Non-infectious diseases, Non-tubercular bacterial infection, and tuberculosis disease under thirty minutes with additional features.



Results of the trained system with additional features vastly improves the results of all the classifiers. Results are improved due to higher volume of information during training like unique features in addition to previous temperature data. In both the cases (30min and 60 min) as listed in the Table 6.4, major classifiers show 100% accuracy in prediction. Prediction systems developed based on these classifiers are also showing a good fit for unseen data classification.

Table 6.6 Behavior of kNN and CART-based prediction system over unseen data with consideration of additional features.



	k-Nearest neighbor (kNN)				Decision tree (CART)			
	Tuberculosis	Non-tubercular bacterial infection	Dengue fever	Non-infectious diseases	Tuberculosis	Non-tubercular bacterial infection	Dengue fever	Non-infectious diseases
🟦 Result of Implimention Thirty minute duration with additional features	94%	97.10%	93.40%	98.20%	95.20%	98.70%	94.87%	98.84%
🟥 Result of Implimention Sixty minute duration with additional features	92%	96.60%	90.80%	97.40%	94.91%	95.50%	93.56%	98.47%

Here, in the training phase the decision tree, kNN and SVM are showing 100% result. But, SVM classifier consumes more time for training in both the cases and achieved zero F1 score for dengue in case of only temperature parameter. Hence SVM classifier is discarded. Remaining methods (kNN, CART) show similar good results and hence considered for future study of testing. Data classification represented in confusion matrix for both of these cases for 30min and 60min duration, with additional features is showing 100% result (Table 6.4). But during testing, 30min data with additional features shows better results (Table 6.6) when compared with 60min data with additional features.

The confusion matrix (Table 6.5) shows the classification of undifferentiated fevers under kNN and Decision Tree (CART) prediction classifier for 30min duration with additional features. Implementation results reveal how well the models perform with respect to unseen data. Unseen data classification accuracy is found to be about 97% (Table 6.6) for the decision tree classifier trained for 30-minute temperature data with additional features namely ESR (Erythrocyte sedimentation rate), WBC (White blood cell count), Neutrophils, Basophils, Eosinophils, Monocytes, Lymphocytes, Platelets, Age, BMI, SBP, DBP, Pulse.

6.3 ONE DIMENSIONAL DATA BASED AI SYSTEM FOR LARGE DATA HANDLING

Once the fever classification system has been successfully developed based on AI techniques, investigation is carried out on coronary artery disease for the purpose of developing a reliable and accurate diagnostic prediction model, where the data size is huge. This data is extracted from the ECG graphs of more than 500 patients.

6.3.1 Correlation study on ECG signals

The signal data of 552 patients are considered as input to the digitization procedure with a sampling rate of 700 over a single lead. To validate these results, randomly selected 100 patient samples are considered. The combined study of Pearson's correlation coefficient and least square fit analysis is carried out, with the original data extracted from the manual method as the reference data. The results of the mean (mean Δ) and median (median Δ) of the sample by sample difference (derived ECG minus original ECG) of a single lead of 100 samples are presented in Table 6.7. It is observed from the Table that the mean difference (mean Δ) and maximum value (Δ) difference are very less and the other two are comparatively better than the IM2GRAPH method (Siegel et al. 2016). The obtained data of both methods are further compared with the results of the manual data extraction method, based on protocols established by healthcare experts and the WebPlotDigitizer tool (Figure 6.6 and Figure 6.7). The correlation between the data extracted from the present method, software tool, and manual method is as shown in Figure 6.8. Results show a good fit and good correlation between the developed method and original data in terms of mean, standard deviation, and maximum values.

The efficiency of correlation between IM2GRAPH data results and original data drops is due to the failure of the software to extract the data if there is a small discontinuity present in the TMT-ECG graph.

(a)Normal condition

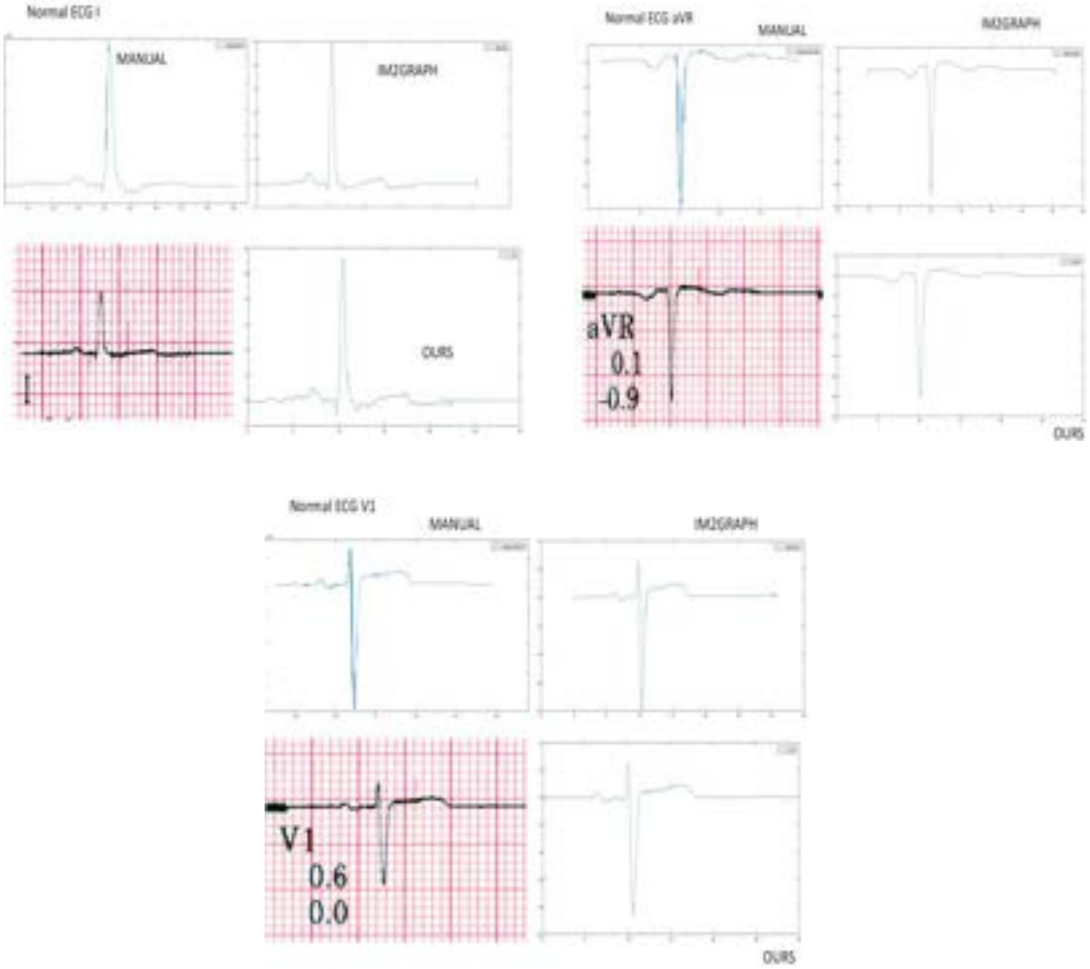


Figure 6.6 Comparison ECG plot obtained different methods for all limb leads (I, II, III), augmented leads (aVR, aVL, aVF) and Precordial leads (V1, V2, V3, V4, V5, V6)

(b) Stressed condition:

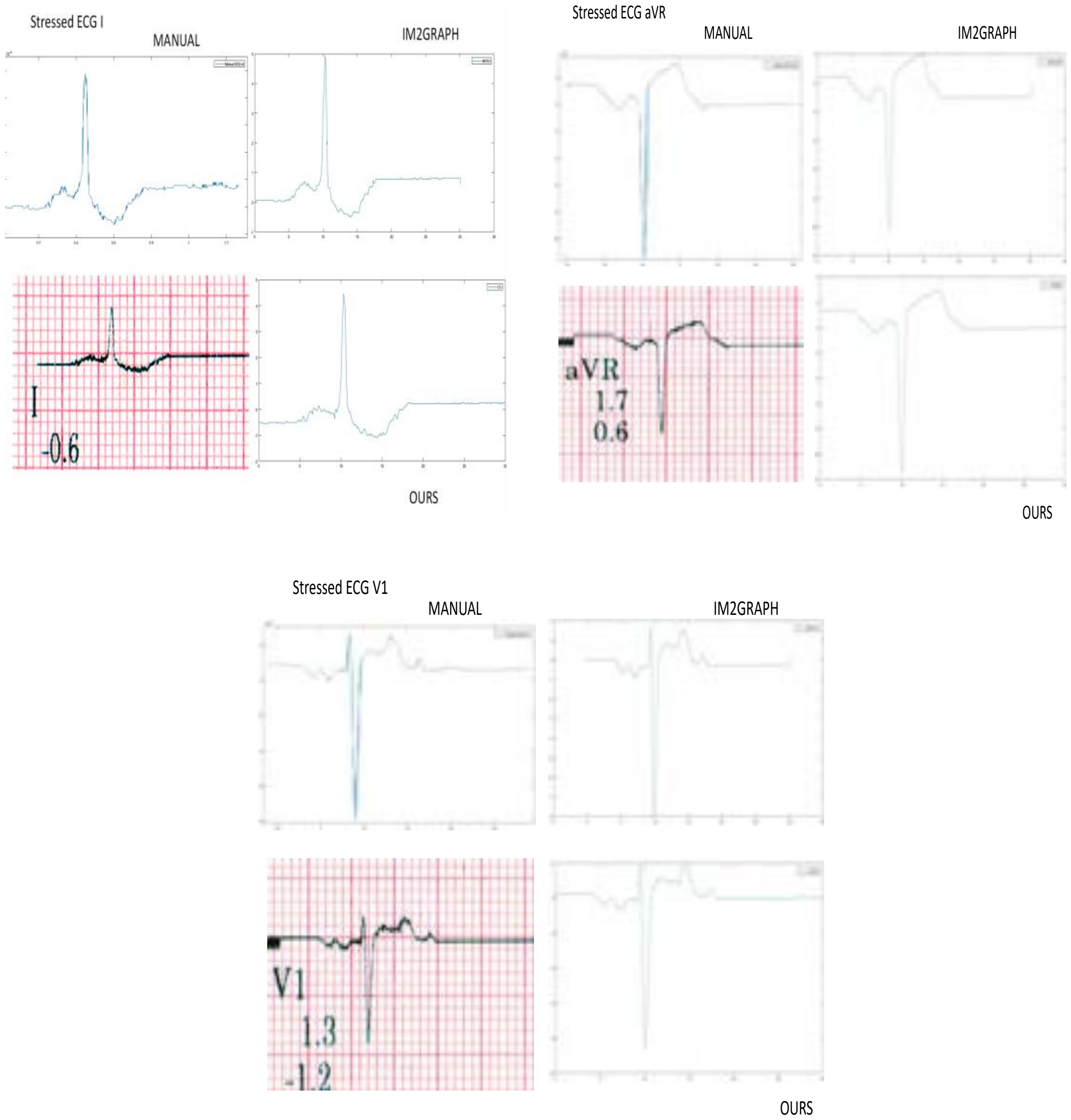
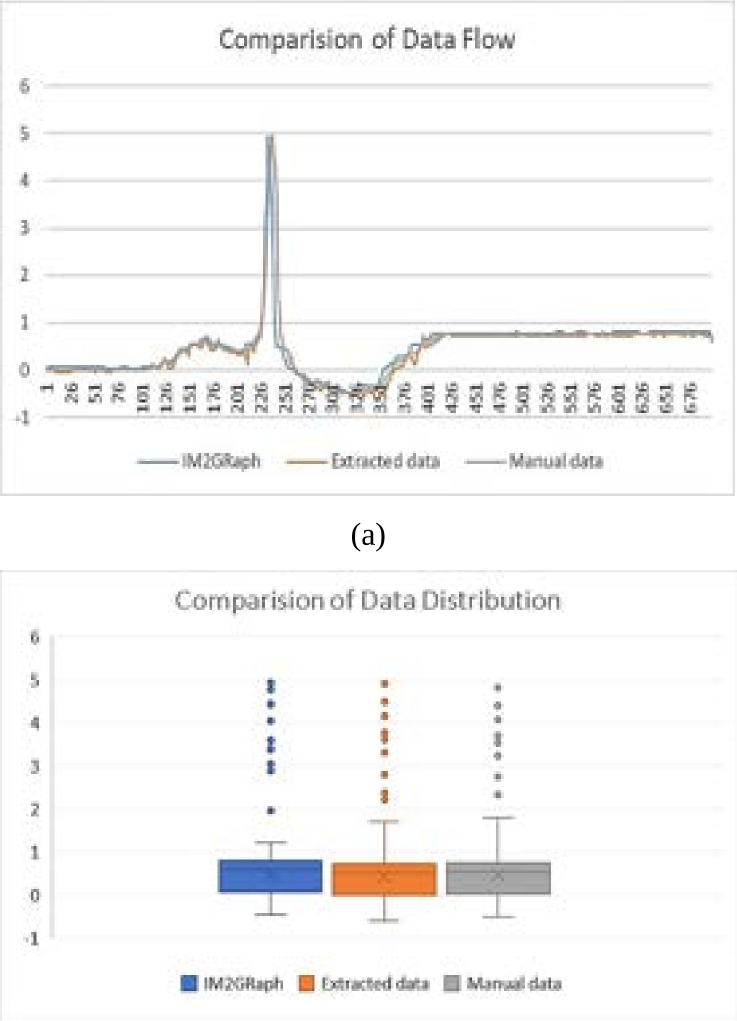


Figure 6.7 Comparison Stressed ECG (TMT-ECG) plot obtained different methods for all limb leads (I, II, III), augmented leads (aVR, aVL, aVF) and Precordial leads (V1, V2, V3, V4, V5, V6)



(a)

(b)

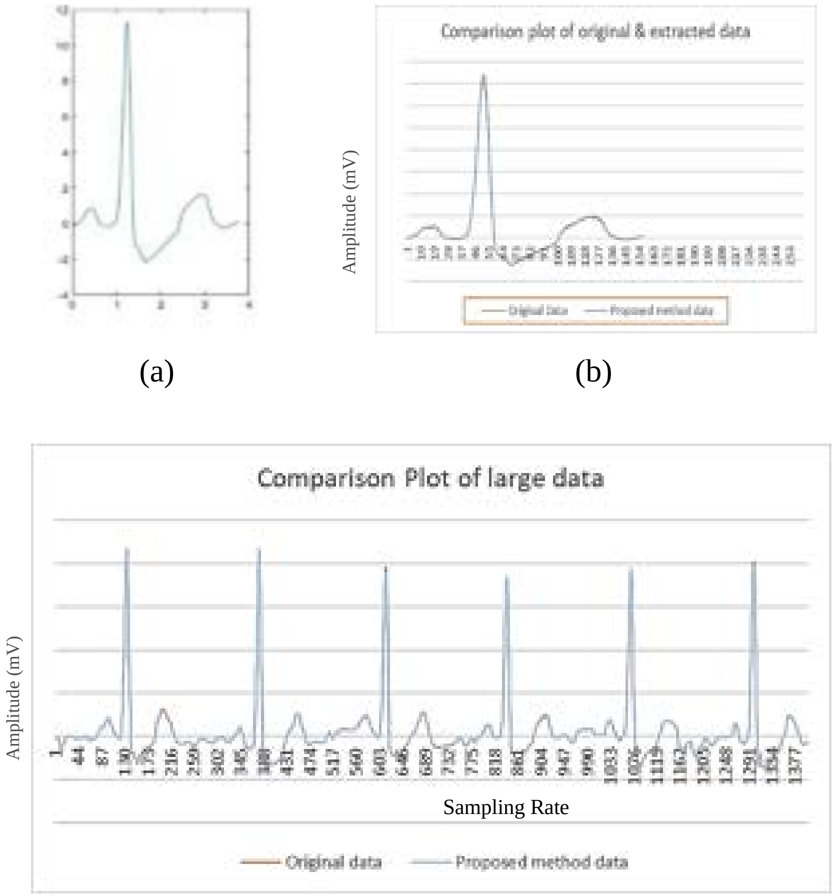
Figure 6.8 Comparison (a) data flow and (b) data distribution for all the three types of data extraction method

Table 6.7 Correlational results of the developed method and IM2GRAPH method with respect to original data

Extracted data vs original data	Mean (Δ)		Median (Δ)		Correlation (Δ)	
	Proposed	IM2GRAPH	Proposed	IM2GRAPH	Proposed	IM2GRAPH
Mean	-0.0311	-0.0628	0.02102	0.06902	0.98	0.92
Standard Deviation	0.02423	0.06456	0.03541	0.1243	0.9	0.66
Minimum	-0.06378	-0.1048	0.06784	0.1676	0.89	0.86
Maximum	0.02785	0.2014	0.02516	0.3447	0.97	0.91

The developed system plays major role in the case of TMT-ECG data extraction due to its uniqueness of (i) no requirement of the definition of graph axis for data extraction (ii) Compensation for small data discontinuity in the graph with its data learning algorithm. Very close pattern-matching proved that data extracted through the image process showed good behavior when compared with the original graph (some example Figures of comparison is as showed in Figure 6.6 and 6.7, which explain cases of both the condition normal and stressed with all three types of leads i.e. limb, augmented and precordial). The observed error is very less. Hence, the method which is adopted for data extraction is considered as one of the best methods and these extracted data further flow to the algorithm as features for the prediction of ECG/TMT-ECG. Based on these algorithms, decision can be taken whether the patent suffers from CAD (coronary artery disease) or not.

The results are also validated by adopting the proposed method of data extraction to the public domain data PhysioNet 2017 inbuilt in the Matlab interface (Baydoun et al. 2019; Clifford et al. 2017). The data extraction is studied on two concepts, one based on single-lead correlation and the other one based on the multi-lead concept. The digital image of this public domain dataset is shown in Figure 6.9c. Figure 6.9a shows an area of interest selected from Figure 6.9c, for applying the developed method to extract data. The correlation of data points of PhysioNet and extracted data set for a single lead are as shown in Figure 6.9b. The correlation of data points of PhysioNet and extracted data set for repetitive-lead ECG is as shown in Figure 6.9c. Here, repetitive lead refers to a continuous sequence of leads for a wider duration of time.



(a) (b)

(c)

Figure 6.9 (a) Plot of Area of interest selected from Physio-net data set, (b) Comparison plot of extracted result and original data related to area of interest, (c) Comparison plot of extracted result and original data for large data.

During extraction of features, initially features are extracted through time interval basis and to achieve this, input is obtained through field experts. These results are compared with derived results to find the correlation.

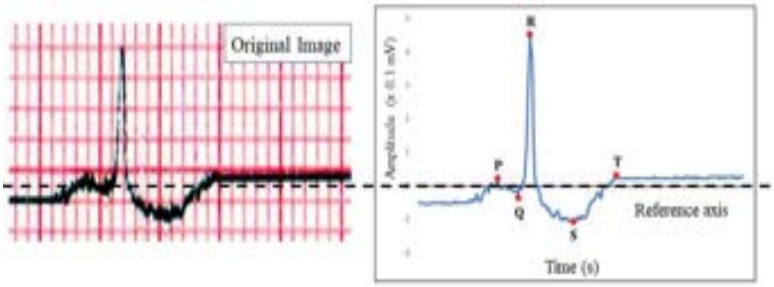


Figure 6.10 Comparison of features of the ECG sample and digitized ECG for the sample data.

Moreover, Pearson’s correlation coefficient (Schober et al. 2018) was also adopted to find the similarity between normalized original features of data and extracted data features for the selected leads. These were one-dimensional datasets, such as in the earlier case. The result of the averaged correlation was found to be 93.6%. Thus, the developed prediction model for CAD incorporated a better approach for feature extraction in terms of PQRST and data extraction from the ECG signals as shown in Figure 6.10 and Table 6.9.

The hybrid method of feature parameter extraction (Pan-Tompkins, slope-based and statistical method) is more effective. This method is evaluated based on feature parameters obtained for the same hundred samples utilized in data extraction, with the help of health care experts. The resultant values of feature parameters for Figure 6.11 are presented in Table 6.8.

Table 6.8 Feature parameter of Extracted data and Manual data for single lead TMT-ECG.

Features	Amplitude Values(mV) (Present Method)	Amplitude Values(mV) (Manual Method)
P	0.01271	0.01
Q	-0.02542	-0.025
R	0.42487	0.42
S	-0.11015	-0.12
T	0.02118	0.02

Table 6.9 Correlation of feature parameter values of hybrid method.

Feature Parameter	Mean	Standard Deviation	Correlation Value
P value (mV)	0.0124	0.0568	0.872
Q value (mV)	-0.1542	0.0244	0.935
R value (mV)	2.8767	0.2213	0.981
S value (mV)	-0.1121	0.0322	0.913
T value (mV)	0.06218	0.0483	0.898
PR interval (ms)	0.1385	0.0255	0.946
QRS interval (ms)	0.8643	0.1224	0.988
ST interval (ms)	0.03792	0.0268	0.958

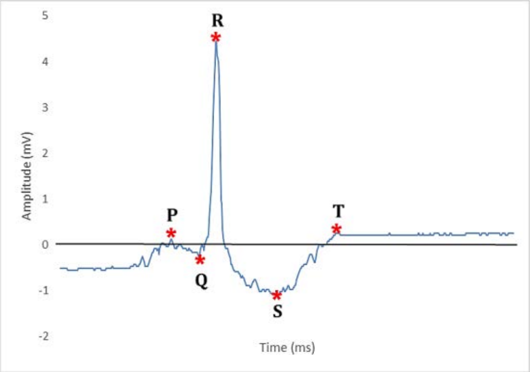


Figure 6.11 Extracted feature data of TMT-ECG

Table 6.9 shows the average of mean, standard deviation, and correlation values of feature parameters i.e. of P, Q, R, S, T, and segmentation values (PR, QRS, ST) of 100 samples. The results are analyzed using Pearson's correlation coefficient method. Table 6.9, shows the good correlation values between the proposed method of feature parameter and the original feature parameter. This proves the effectiveness of the proposed method to extract feature parameters. The resulting values of feature parameters with respect to isoelectric reference line of a single lead which is considered as an example are shown in Figure 6.11 and Table 6.8.

There are a good number of studies of the ECG data and feature parameter value extractions (Baydoun et al. 2019; Ravichandran et al. 2013; Sassi et al. 2017), which show similar results. Similar validation has been done by one of the studies (Baydoun et al. 2019) using the same Physionet dataset. But the limitation of the study by Baydoun is that it is applicable only if the single lead exists on the particular row of the voltage (mV) vs Time (ms) graph and this limitation is easily overcome with the proposed method with improvement results as shown in Table 6.10.

Table 6.10 Correlation coefficient values of developed method for physio-net data and results developed by Baydoun.

Feature Parameter of Physio-net dataset	Correlation Value	
	Proposed method	Results of (Baydoun et al. 2019)
Overall data	0.975	0.952
P value (mV)	0.847	0.833
Q value (mV)	0.953	0.927
R value (mV)	0.982	0.838
S value (mV)	0.893	0.822
T value (mV)	0.918	0.781
PR interval (ms)	0.986	0.984
QRS interval (ms)	0.981	1

As observed by the literature survey, few studies (Badilini et al. 2005; Ravichandran et al. 2013; Sassi et al. 2017; Virgin and Baskar 2018) have been done on data and feature extraction for ECG graphs but the results are lagging in accuracy. The current study has focused on a limited region of the graph (area of interest) with a large sampling size that helps to improve the accuracy of the method over other methods. In the present study, the correlation accuracy of data extraction is about 98% (based on repository data) and which is better than other methods.

The limited area of interest for high accuracy is one of the drawbacks to the study for consideration of large-size ECG images. Large-sized ECG images may consume more time for long durational ECG data extraction, with a high sampling rate. But it can be fine-tuned by the end user for desired accuracy by proper customization. Since the present system has been developed on Matlab and Python platform, the same can be accessed for use by other users also. The current method will be more helpful, since it is adaptable for all types of ECG and TMTECG graph images because it is developed and validated with all 12 leads of 552 patients' data. Therefore, the application of this data extraction tool will be more powerful to develop a good intelligent system based on ECG and TMTECG data. This may act as an assisting tool for healthcare experts and doctors to predict disease based on ECG and TMTECG very effectively.

6.3.2 Performance metrics evaluation of one-dimensional data analysis

The convolutional neural network (CNN) is an extended part of artificial neural networks (ANN) and machine learning. Which is more helpful over image recognition and classification tasks. Usually, convolutions are the process helps to represent the images as a combination of numbers and identify the key features within the image. The convolution networks are also applicable for one-dimensional dataset, this is one of the advantages over other networks (i.e. applicable to both one-dimensional and two-dimensional data). These importance leads to selection of convolutional neural networks over analysis of ECG/ TMT-ECG datasets.

The CNN model over this application was developed on a system that was configured with the AMD Ryzen processor consisting of SSD and 24 GB RAM with a 2GB graphics card (Cireşan et al. 2011; Tripathy et al. 2017). The training algorithm ran with TensorFlow and Keras backend (Abadi et al. 2016; The Theano Development Team et al. 2016). Each epoch approximately took an average of 5 to 10 s. Tables 6.11 and 6.12 show the results of the single-lead and multi-lead ECG/TMT-ECG for the single-, two-, and three-layer convolutional neural networks, respectively (precision, recall, F1-score, accuracy, and validation loss), depending on the classification of CAD and normal ECG segments based on the single-lead ECG and 12-lead ECG data (one-dimensional). In Tables 6.11 and 6.12, one could see that very high diagnostic accuracy was obtained with a defined number of training samples.

Table 6.11 Results of the single-lead ECG for the single-, two-, and three-convolutional layer networks.

	Single-Layer CNN				**Two-Layer CNN**			**Three-Layer CNN**		
	Network (N)									
	N-1	**N-2**	**N-3**	**N-4**	**N-5**	**N-6**	**N-7**	**N-8**	**N-9**	**N-10**
Precision	72.50%	84%	32%	97%	97.50%	98%	99%	100%	96.50%	99%
Recall	74.50%	81.50%	50%	95.50%	98%	98%	99%	99%	98%	99%
F1-Score	72.50%	82%	39%	96%	97.50%	97.50%	98%	100%	97.50%	99%
Accuracy	73%	84%	64%	96%	98%	98%	99%	100%	97%	100%
Validation Loss.	0.4825	0.37	0.65	0.1269	0.0773	0.069	0.022	0.00075	0.0098	0.00412

*N= Network

Table 6.12 Results of the multi (12)-lead ECG for the single-, two-, and three-convolutional layer networks.

	Single-Layer CNN				**Two-Layer CNN**			**Three-Layer CNN**		
	Network (N)									
	N-1	**N-2**	**N-3**	**N-4**	**N-5**	**N-6**	**N-7**	**N-8**	**N-9**	**N-10**
Precision	29.50%	98%	29.50%	97%	29.50%	97.00%	98%	99%	99%	99%
Recall	50.00%	97.50%	50.00%	96.50%	50.00%	97.50%	97%	99%	99%	99%
F1-Score	37.00%	96.50%	37.00%	97%	37.00%	96.50%	97%	100%	100%	100%
Accuracy	59%	98%	59%	97%	59%	97%	98.30%	99%	99%	99%
Validation Loss	0.68	0.054	0.6878	0.079	0.68	0.052	0.025	0.00022	0.00017	0.00028

*N= Network

The data were of two types (i) single-lead signal and (ii) 12-lead signal data. These data were used for system training; the detailed evaluation (Sedghamiz 2014) of the model was performed based on testing and validation data using a standard protocol. The commonly used performance metrics were accuracy (equation 6.2), sensitivity (equation 6.3), f1 score (equation 6.1), specificity (equation 6.4), and precision (equation 6.5). (Liang et al. 2020; Ribeiro et al. 2020)

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{6.2}$$

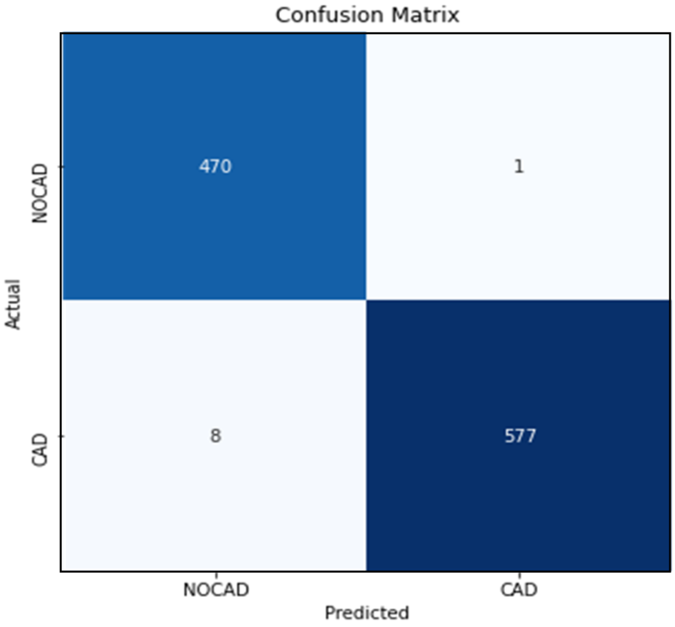
$$\text{Sensitivity/Recall} = \frac{TP}{TP + FN} \tag{6.3}$$

$$Specificity = \frac{TN}{FP + TN} \tag{6.4}$$

$$Precision = \frac{TP}{TP + FP} \tag{6.5}$$

These are related to the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates of the confusion matrix (Figures 6.13 and 6.14).

Table 6.13 Confusion matrix of developed model for the single-lead ECG.

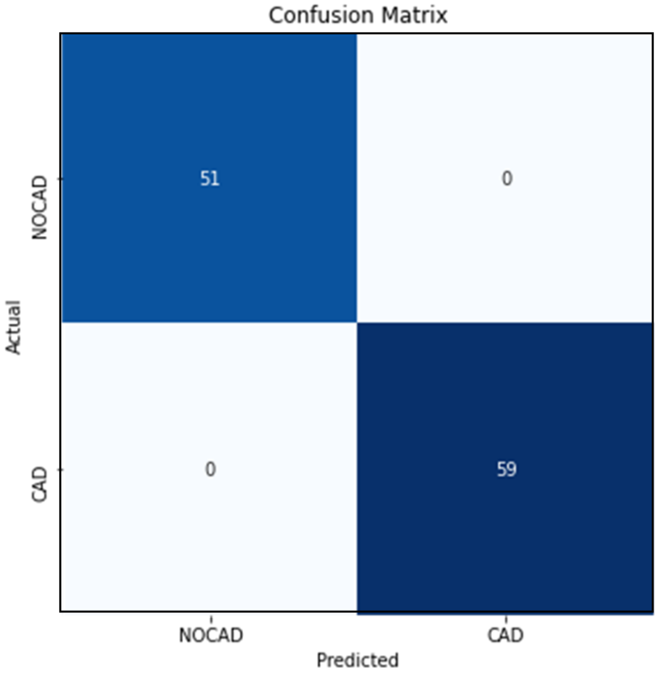


Confusion Matrix

Actual	NOCAD	CAD
NOCAD	470	1
CAD	8	577

Predicted

Table 6.14 Confusion matrix of developed model for the 12-lead ECG.



Confusion Matrix

Actual	NOCAD	CAD
NOCAD	51	0
CAD	0	59

Predicted

6.3.3 Identification of the Optimal Model for one-dimensional ECG data

The CNN system learns from the empirical set of data features automatically with multiple levels of abstraction. Hence, it allows learning the complex functions of the input data with features accessed automatically. The results of the different CNN layers and activation function scenarios, which were used for experimentations, are compared in Tables 6.11 and 6.12, considering accuracy, precision, validation loss, F1 score, and recall. Network-8 exhibited good accuracy during the training and testing phases (99% and 100%), with less validation loss (0.00022 and 0.00075) for both single-lead and 12-lead ECGs. The results are acceptable, owing to the good precision and recall rates. The model by (Tan et al. 2018), of the long short-term memory (LSTM) network with a convolutional neural network (CNN), provided good results in automatically detecting CAD ECG signals with the inclusion of only specific lead II and ECG segments. The specific lead limitation was easily overcome with the application of the present method (consideration of all leads) by retaining the same and better accuracy. The prediction method of single-lead and twelve-lead includes data extraction and data classification. Since it is a complex methodology, the processing time might be higher compared to a normal repository dataset analysis. This model can further be tested for patients located in geographically-far locations, so that generality can be ensured for a wider diaspora. A technically experienced person can effectively manage these types of prediction systems. The three-layer CNN network model (Network-8) (Figure 6.12) is preferred for the classification of CAD due to its superiority in performance.

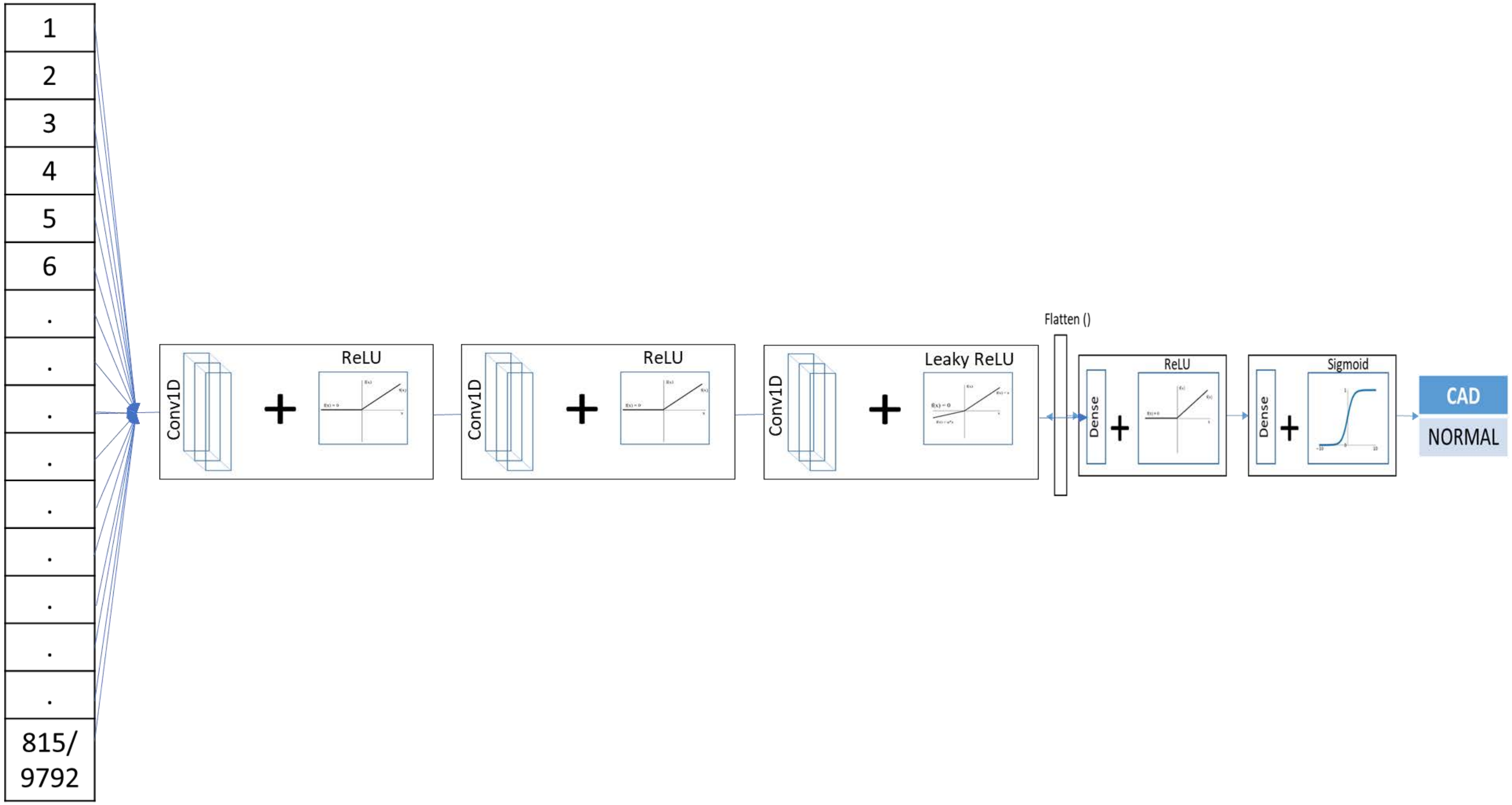


Figure 6.12 The best CNN model for the single-lead ECG and the multi (12)-lead ECG.

The results of Tables 6.11 and 6.12 indicate that a three-layered network with ReLU, ReLu, and LeakyReLU activation functions assigned to the first, second, and third hidden layers, was the best-suited architecture for the model, which was not attempted earlier. The developed three-layer CNN model worked equally well for both the single-lead and 12-lead ECG signal data, exhibiting generality. The performances of the developed model in terms of training and validation accuracies are plotted in Figures 6.13a, and 6.13b for the single-lead. Figure 6.14a, and 6.14b show good convergence of validation loss; it is an ideal fit for twelve-lead ECG/TMT-ECG data, which were extracted from the ECG/TMT-ECG graph. The model achieved a training accuracy of 99% and validation about 98.8% with a very small validation loss of less than 0.0008% on the 25th epoch.

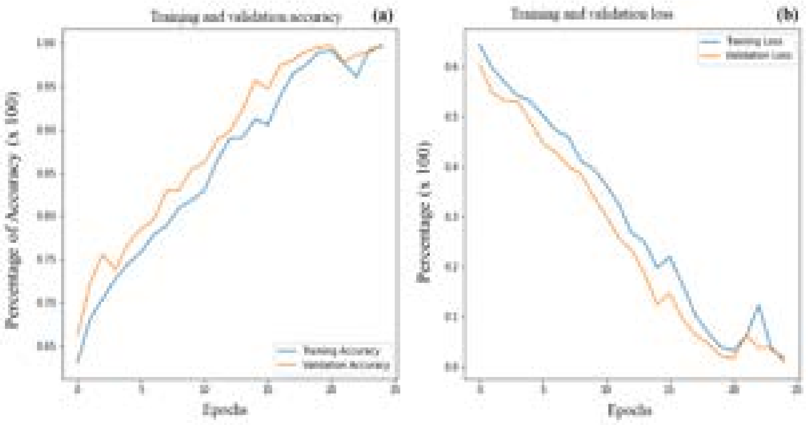


Figure 6.13 (a) Accuracy and (b) validation loss of developed model for single-lead ECG and TMT-ECG.

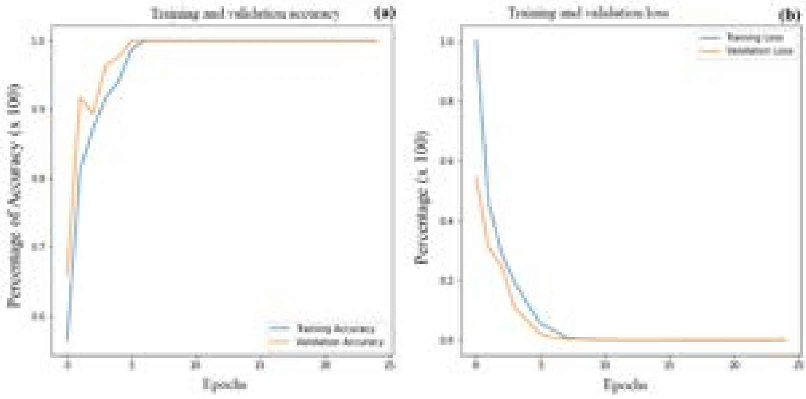


Figure 6.14 (a) Accuracy and (b) validation loss of developed model for twelve-lead ECG and TMT-ECG.

6.3.4 k-Fold Cross Validation technique

This is a standardized method of estimating the performance of the evaluation of the prediction model. This is usually applied during a smaller number of data samples that are present as input. The k-fold cross-validation process contains a single parameter, k, which designates how many groups should be created from a given data sample. Here, the k-fold parameters were considered as 5, i.e., k = 5, which indicates that splitting the given data samples into 5 groups is known as the 5-fold cross-validation, as explained in Figure 6.15.

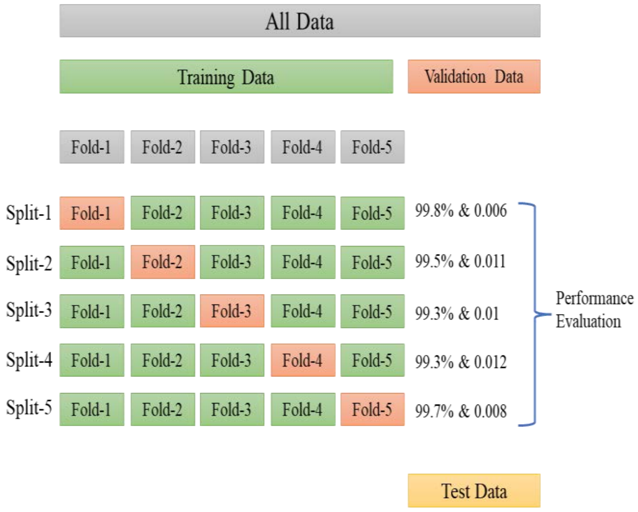


Figure 6.15 Five-fold cross-validation technique adopted for single lead ECG and TMT-ECG.

Each split contains five-fold values, out of that, four are used for training and the remaining one is used for validation.

Table 6.15 shows the detailed information about accuracy and standard deviation scores obtained during each split for the single-lead ECG data and twelve-lead ECG data. Results show the validation scores of Network-8, considering single-lead and 12-lead data. The average accuracies were obtained as 99.5% and 99.1%, respectively, for single-lead and 12-lead data. Hence, this is one of the best methods to deal with clinical lab datasets. The model shows good accuracy during the validation phase for both single and 12-lead ECG data.

Table 6.15 Accuracies of the k-fold cross validation over single lead and twelve lead ECG and TMT-ECG data.

	Single-Lead ECG Data		**Twelve-Lead ECG Data**	
	Accuracy (%)	**Standard Deviation**	**Accuracy (%)**	**Standard Deviation**
Split-1	99.8	0.006	98.7	0.021
Split-2	99.5	0.011	99.4	0.013
Split-3	99.3	0.01	99.3	0.011
Split-4	99.3	0.012	98.8	0.019
Split-5	99.7	0.008	99.1	0.017
Average	99.5%	0.009	99.1%	0.016

6.4 ANALYSIS OF THE TWO-DIMENSIONAL PREDICTION MODEL

The analysis of two-dimensional data (image) is carried out for developed and pre-trained algorithm by using single lead and twelve lead data with consideration of normal ECG and TMT ECG under raw signal image conditions and filtered signal images which are obtained through image pre-processing. These algorithms are implement through python by exploiting the graphical processing unit (GPU) to reduce the computation time required to obtain the trained model. The developed algorithm is back-ended with TensorFlow and helps to optimize the network. The data which are considered for analysis are twelve lead data, of 552 image samples and single lead data, of 6,624 image samples of patients which includes normal ECGs and exercised ECGs. These data are fed into 6 types of algorithms, five are pre-trained (transfer learned) and one is developed. The performance of algorithm is evaluated and compared based on the accuracy, loss function, roc value, and confusion matrix.

The trainable parameters (batch size, learning rate, optimizers, and loss functions) are considered for all six similar algorithms. That is batch size considered for the training algorithm is 128 and the learning rate is defined as 0.001 with consideration of Adam optimizer and binary cross entropy loss function. These algorithms are trained with an early stop condition if the performance does not improve (Validation loss) after 5 successful epochs. Tables 6.13 and 6.14 show the accuracy of the model throughout training, validation, and testing for raw and filtered single lead ECG signal images.

6.4.1 Analysis of Single Lead

The Table 6.16 indicates performance evaluation of the single lead raw ECG signal image in terms of accuracy and loss function. The results indicate that the comparison of all pre-trained models and developed networks shows that the developed network with raw ECG signal image conditions shows less accuracy due to an insufficient feature parameters availability for prediction because of a lot of noise. The accuracy drop under raw ECG signal condition implies that the prediction model is not good enough to predict the CAD disease under raw signal image condition which can be overcome with filtered ECG signal image (Figure 6.16).

Table 6.16 Comparison of transfer learned and developed models' accuracies of raw data for single lead signal images

Model	Training		Validation		Testing	
	accuracy	loss	accuracy	loss	accuracy	loss
VGG16	90.20%	0.25	86.10%	0.30	89.16%	0.24
Inception	76.79%	0.45	77.05%	0.47	81.02%	0.40
Mobile net	89.57%	0.23	85.62%	0.31	89.16%	0.27
ResNet	57.38%	0.64	56.63%	0.64	57.53%	0.66
Efficient Net	73.85%	0.52	75.69%	0.49	79.82%	0.45
Developed CNN	87.74%	0.38	86.95%	0.38	88.13%	0.37

Table 6.17 Comparison of Transfer learned and developed models' accuracies of filtered data for single lead signal images

Model	Training		Validation		Testing	
	accuracy	loss	accuracy	loss	accuracy	loss
VGG16	88.16%	0.27	85.94%	0.33	84.34%	0.34
Inception	86.73%	0.31	84.75%	0.37	87.05%	0.35
Mobile net	90.58%	0.23	86.10%	0.32	87.95%	0.27
ResNet	84.5%	0.35	82.4%	0.39	83.1%	0.41
Efficient Net	84.32%	0.34	86.18%	0.32	87.95%	0.30
Developed CNN	93.22%	0.17	92.7%	0.21	93.9%	0.21

The filtered signal images are obtained by the pre-processing method, and are used to train all six models which help to compare the results under the condition of raw ECG signal image and filtered one. When these pretrained models' result is compared to the developed prediction model, shows that it performs far better in prediction than the other pre-trained models and raw single lead ECG signal condition. The developed model's accuracy increased to 93.2 percent during training and 93.9 percent during testing when filtered images were used instead of the raw image, and making it more suitable for the implementation of the prediction system.

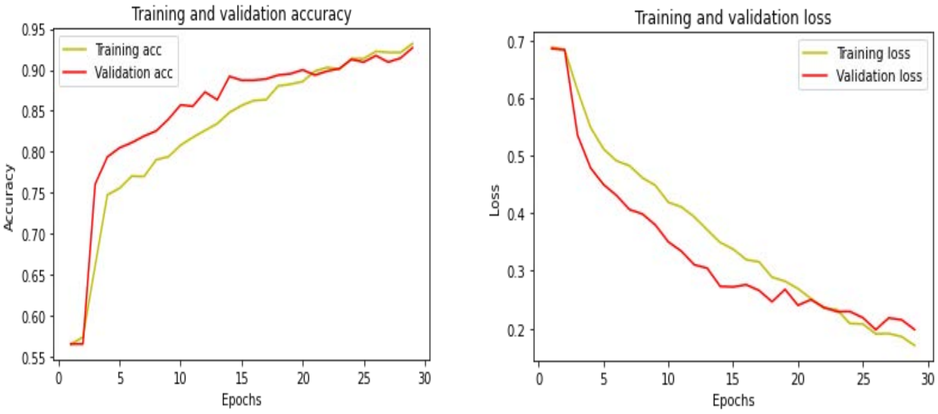
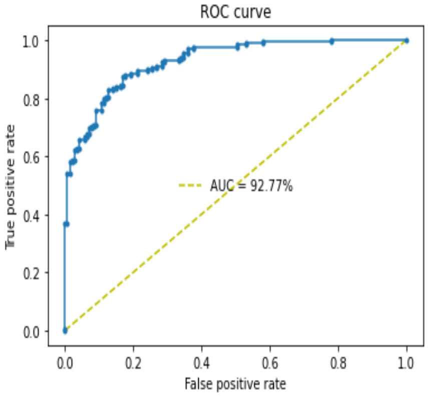


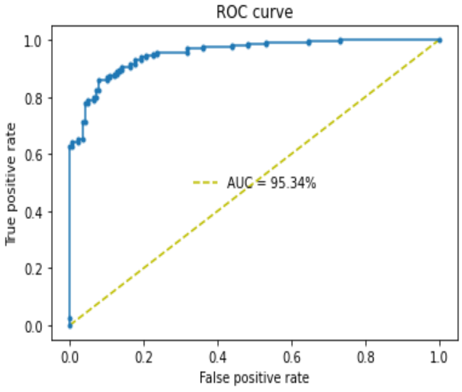
Figure 6.16 Accuracy plot and Validation loss of developed network for filtered single lead two-dimensional image data

The results are nearly similar to those of Mobile Net and VGG16. However, these two pre-trained models are complex in nature compared to the developed model (layers of Mobile Net= 53 and VGG16=16) and require more computational time for analysis. With the developed network, it is possible to overcome these issues effectively.

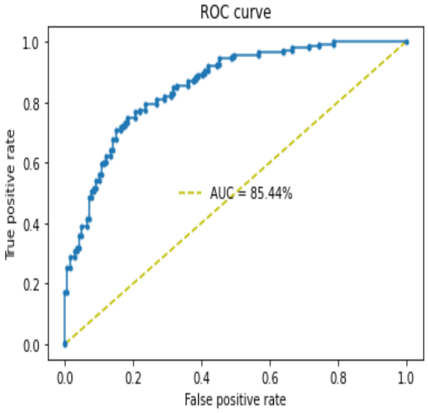
Even the models are evaluated by taking account of the testing (unseen) samples i.e., 5% (190 CAD and 142 NOCAD) in terms of the confusion matrix and ROC (Receiver Operating Characteristics). The ROC curve for filtered single-lead ECGs and TMT-ECG with all conditions (six algorithms) that are considered for prediction analysis is shown in Figure 6.17a to 6.17f. The ROC plot curve is supposed to be in the left upper corner which reflects its effective performance in True positive classification (Effective prediction of CAD and NOCAD images). The developed network has a very good ROC plot, indicating that it has excellent binary classification diagnosis capabilities.



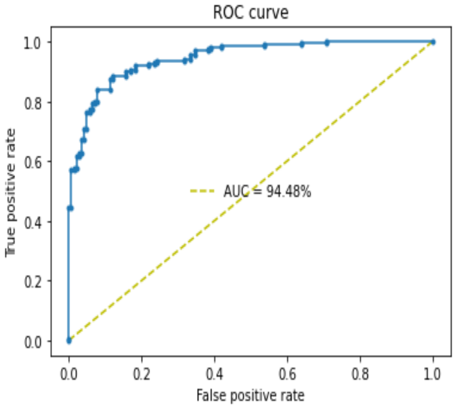
(a) ROC of VGG16



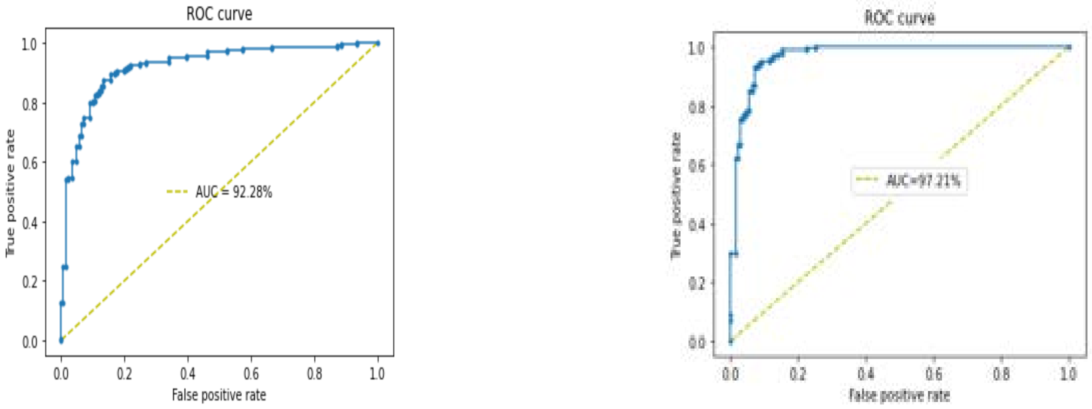
(b) ROC of MobileNetV2



(c) ROC of ResNet



(d) ROC of EfficientNet



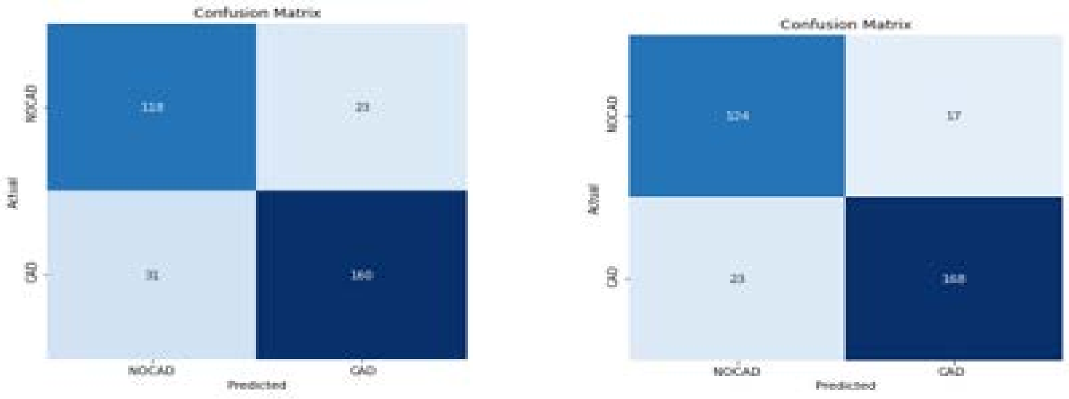
(e) ROC of Inception

(f) ROC of developed model

Figure 6.17 (a-f): Receiver Operating Characteristics (ROC) plot of transferred learning model and developed model for single lead filtered two-dimensional data.

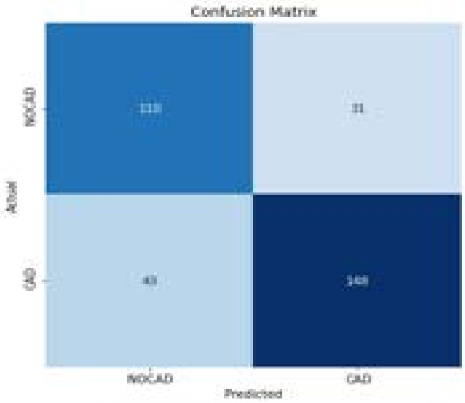
Along with ROC, the confusion matrix (Table 6.18a to 6.18f) demonstrate a number of the image data lies on the True positive, True negative, False positive, and False negative columns. Table 6.18a to 6.18f represent the percentage of filtered single lead ECG signal images that were successfully classified in the confusion matrix. Out of 142 NOCAD images, the developed method achieved a high true positive classification number of 132 images successfully classified, while 177 CAD images are correctly classified against 190 images. The computational and analysis duration is significantly reduced because the network is not as deep as compared to other pre-trained algorithms.

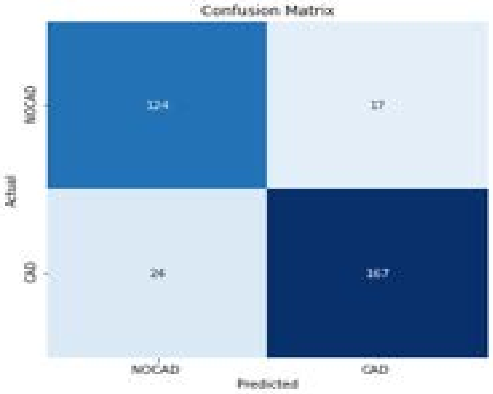
Table 6.18 (a-f): Confusion matrix plot of transferred learning model and developed model for single lead filtered two-dimensional data



(a) Confusion matrix of VGG16

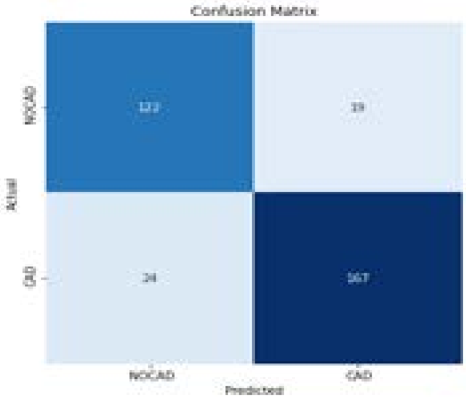
(b) Confusion matrix of MobileNetV2

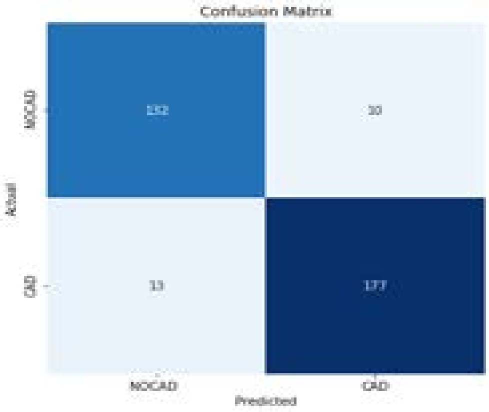




(c) Confusion matrix of ResNet

(d) Confusion matrix of EfficientNet





(e) confusion matrix of Inception

(f) confusion matrix of Proposed model

For each epoch, the model classification accuracy score and loss score for testing and validation sets are examined. The basic idea behind having a loss score in deep network learning is, to use it as a feedback signal to alter the model's weights in order to reduce the loss score in the following train epoch. The test set (5%) are essential to provide to know the model's generalizability to new and unknown data. Over-fitting occurs when a model achieves a high accuracy score in the training set but performs poorly in the validation set. This occurs when the model over-optimizes in its learning phase of the training image samples. This can be mitigated by tuning a network that improves model accuracy during training and validation by reducing loss function.

The mathematical representation of classification accuracy equation 6.6 is as follows:

$$\text{Classification accuracy} = \frac{\text{correctly predicted samples}}{\text{total number of samples}} \qquad (6.6)$$

Additional to classification accuracy, the model performance was evaluated in terms of precision, recall, and F1 score. The precision, recall, and F1-score are derived using the confusion matrix. These parameter values provide a visual representation of the algorithms' evaluation. These parameters are calculated from all the pre-trained and developed models. Equations 6.1 to 6.5 shows the mathematical definition of F1 score, accuracy, sensitivity/recall, specificity, and precision respectively.

The precision score reveals how many samples the model correctly predicted as positive class and the recall score provides the information related to how many truly positive samples were correctly predicted as positive. The F1 measure score is frequently employed in conjunction with accuracy and recall scores since it functions as a harmonic mean and provides a more accurate and balanced score indicator.

The accuracy score and confusion matrix performance are used to determine whether a prediction model is effective for classifying CAD disease based on ECG or TMT-ECG signal images. The Table 6.19 shows the filtered single lead ECG images precision, recall, and F1- Score values for CAD and NOCAD conditions.

Table 6.19 Precision, Recall and F1 Score of transfer learned and developed network for single lead signal image data

Model	Precision		Recall		F1-score	
	NOCAD	CAD	NOCAD	CAD	NOCAD	CAD
VGG16	79	87	84	84	81	86
MobileNetV2	84	91	88	88	86	89
Inception	84	90	87	87	85	89
ResNet	72	83	78	77	75	80
Developed CNN	91	95	93	93	92	94

The precision, which specifies how many samples of images that the model predicts belong to the real class, was around 95 percent for CAD and 91 percent for NOCAD. The recall reflects how many real samples of images were properly predicted by the algorithm (model), and the score obtained for CAD and NOCAD images is 93 percent. Similarly, the F1-Score functions as a harmonic mean of precision and recall, and the developed model achieves a very good and acceptable score when compared to other pre-trained models (i.e., 94 for CAD and 92 for NOCAD images).

6.4.2 Analysis of Twelve Lead

The same methodology is adopted for 12 lead ECG signal images, as is used for single lead signal images, with consideration of raw and filtered signal images including both normal ECG and TMT-ECG signal images. Since it is a comparison study, the same parameters are used to evaluate the pre-trained and developed algorithm results like single lead signal images. The parameters taken into account to deal with the twelve lead signal images are accuracy, loss, ROC, confusion matrix, precision, recall, and f1-score.

There are 552 images in 12 lead signal images, including CAD 316 and NO-CAD 236 in which both resting and TMT ECG (exercised ECG) are taken into account. The analysis is done in two ways: one using raw ECG signal images, and the other one filtered signal images. Different algorithms (such as pre-trained (5) and developed (1) algorithms) are used to analyze ECG and TMT ECG signal conditions. The results of training, validation, and testing accuracy are taken into account while evaluating pre-trained and developed algorithms. The accuracy and validation loss of VGG16, Inception, MobileNetV2, ResNet, EfficientNet, and developed networks are compared in Table 6.20 and Table 6.21 with respect to raw ECG and TMT-ECG signal images and Filtered ECG and TMT-ECG signal images respectively.

The Table 6.20 shows the accuracy and validation loss results of the developed algorithm against the transferred learned algorithms for raw data. But, the developed algorithm for raw images does not show good performance. The filtered images are derived from the raw image using a defined image pre-processing methodology. These filtered signal images are considered as input to the developed algorithm (Table 6.21). The accuracy of the result for the developed algorithm is better than the pre-trained algorithm (94.03 percent).

Table 6.20 Comparison of Transfer learned and developed models' accuracies of raw data for twelve lead signal images

Model	Training		Validation		Testing	
	accuracy	loss	accuracy	loss	accuracy	loss
VGG16	92.99%	0.17	86.79%	0.35	78.87%	0.40
Inception	97.03%	0.07	88.68%	0.33	82.14%	0.53
Mobile netv2	97.66%	0.08	90.57%	0.28	85.71%	0.29
ResNet	57.11%	0.68	60.38%	0.67	53.57%	0.72
Efficient Net	91.08%	0.23	84.91%	0.29	85.71%	0.31
Developed CNN	56.90%	0.68	60.38%	0.68	53.57%	0.69

Table 6.21 Comparison of Transfer learned and developed models' accuracies of filtered data for twelve lead signal images

Model	Training		Validation		Testing	
	accuracy	loss	accuracy	loss	accuracy	loss
VGG16	90.87%	0.20	88.68%	0.21	89.2%	0.36
Inception	85.77%	0.46	88.68%	0.5496	92.86%	0.37
Mobile netv2	95.97%	0.12	96.23%	0.22	89.28%	0.32
ResNet	57.11%	0.69	60.38%	0.69	53.57%	0.67
Efficient Net	86.62%	0.37	90.57%	0.23	85.7%	0.37
Developed CNN	94.03%	0.17	94.34%	0.21	100%	0.04

The filtered image (ECG& TMT ECG) condition, the accuracy plot, and the validation loss plot generated for the developed algorithm demonstrate good accuracy compared to pre-trained or transfer learning models. The correlation between training and validation accuracy as shown in Figure 6.18 (a) with respect to epochs. It also shows the relationship between validation loss, during the time of training and validation as shown in Figure 6.18 (b) with respect to epochs.

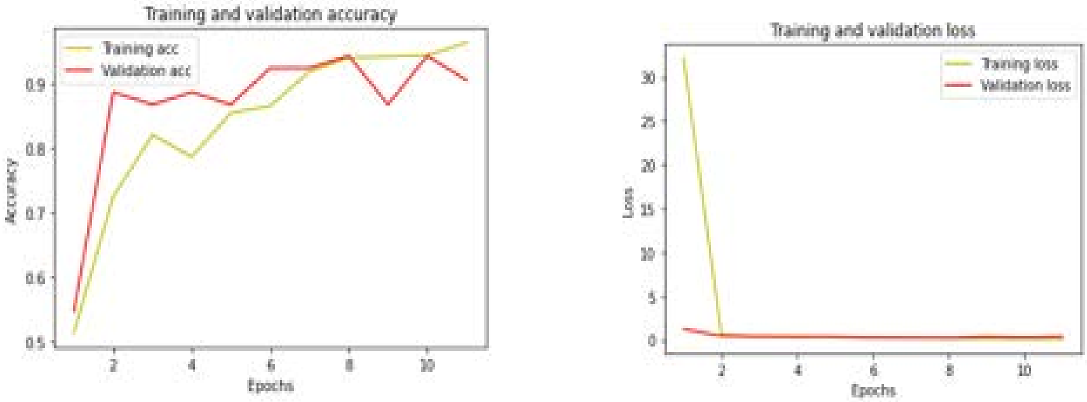
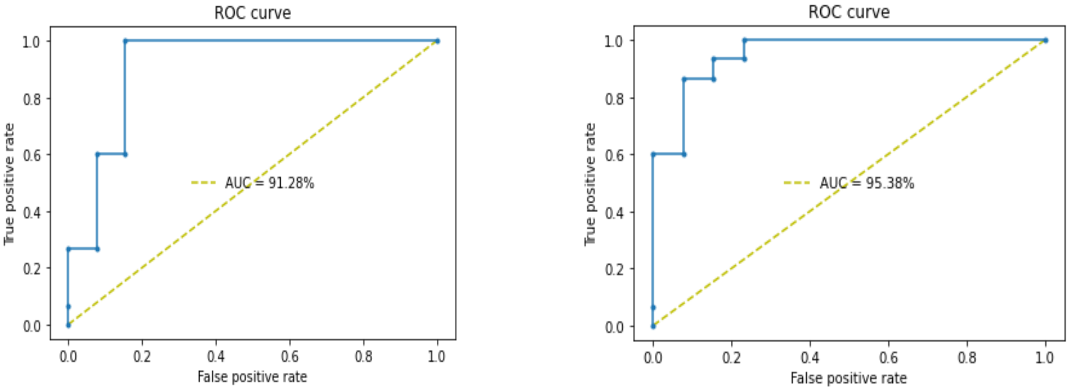


Figure 6.18 (a) Accuracy plot and (b) Validation loss of developed network for twelve lead two dimensional image data

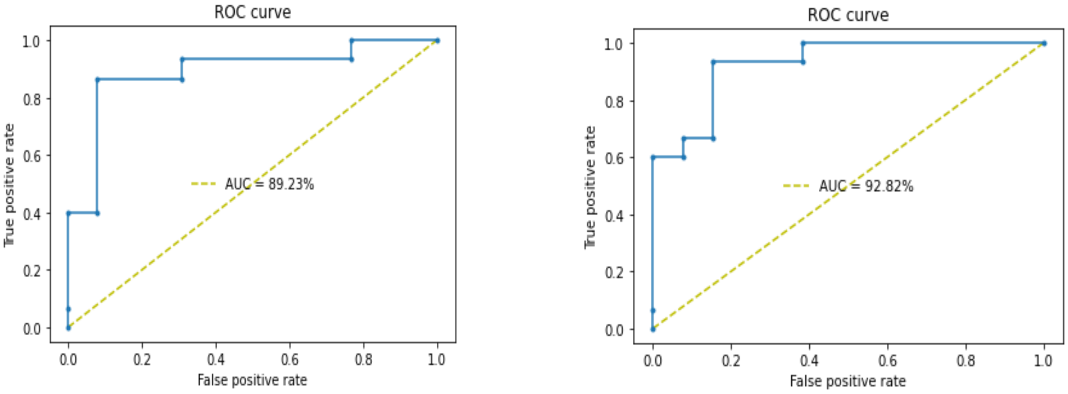
The algorithms which are developed and transfer learned are further evaluated based on the ROC plot and confusion matrix as similar to the analysis of single lead image condition. The ROC is a plot which is used to evaluate the performance of binary classification. Since the study is focused on the differentiation of CAD and normal patients based on signal images, the ROC plot will help to evaluate the performance of the algorithms effectively.

The Figures 6.19a to 6.19f show the performance of six algorithms (five pre-trained transfer learned and one developed) in terms of the ROC plot. The curve that tends to the left most top corner relates to the true positive rate of prediction. The ROC plot of the developed algorithm demonstrates a good True positive rate of 100 percent for the categorization of CAD and NO-CAD images.



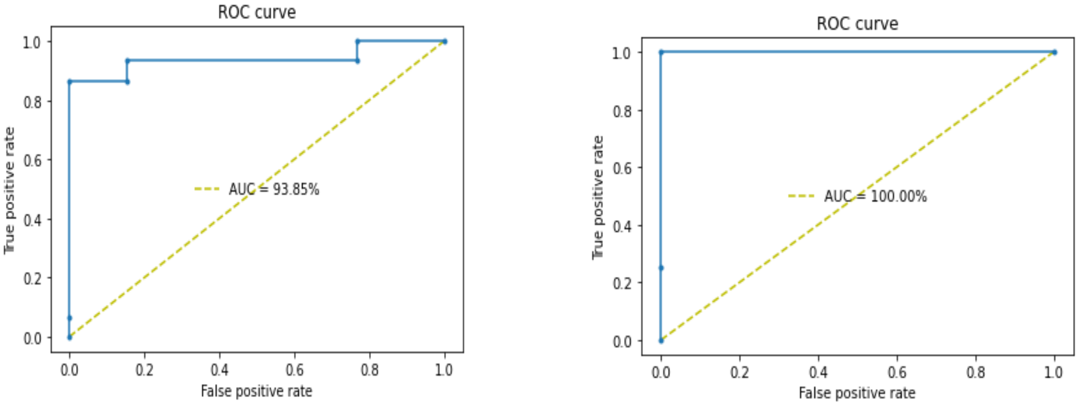
(a) ROC of VGG16

(b) ROC of MobileNetV2



(c) ROC of ResNet

(d) ROC of EfficientNet



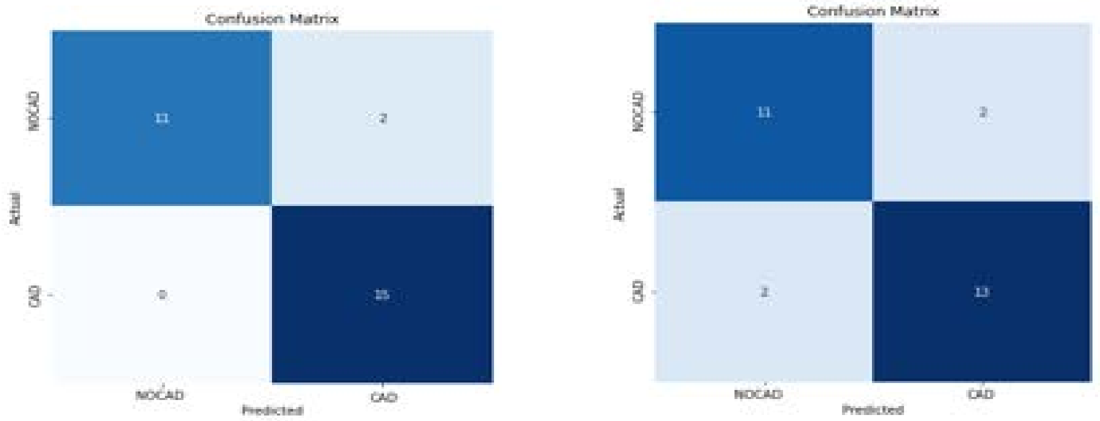
(e) ROC of Inception

(f) ROC of proposed model

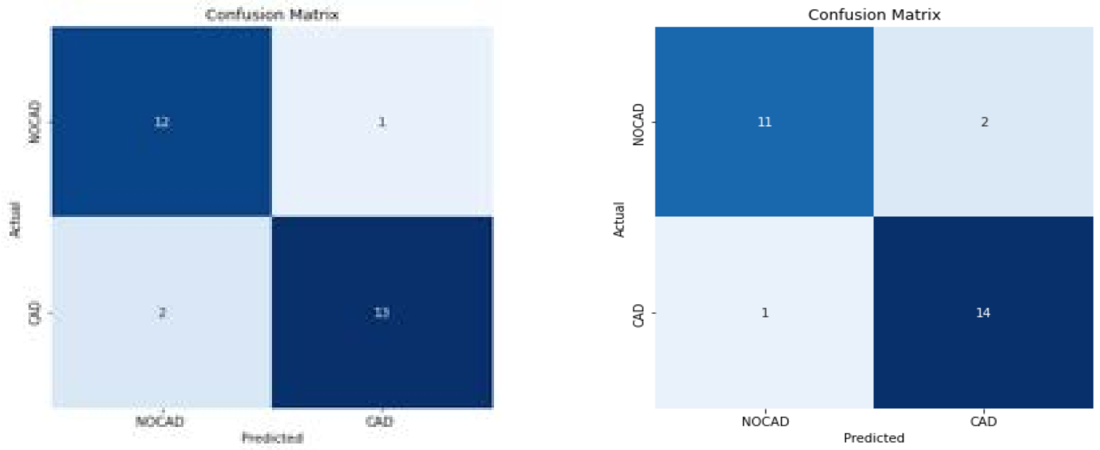
Figure 6.19 (a-f): Receiver Operating Characteristics (ROC) plot of transferred learning model and developed model for twelve lead filtered two-dimensional signal images

The confusion matrix helps to visualize the quantity of data which are classified for the same class and opponent class. Table 6.22a to 6.22f show the confusion matrix for the transfer learned algorithms and developed algorithm for test images of 12 lead ECG signals. The confusion matrix parameters (True Positive, True Negative, False Positive, and False Negative), indicate the efficiency of the developed algorithm to classify CAD and NOCAD. Under the filtered 12 lead signal image condition, the developed algorithm shows the clear classification of test images for CAD and NOCAD classes (100 percent accuracy of test images).

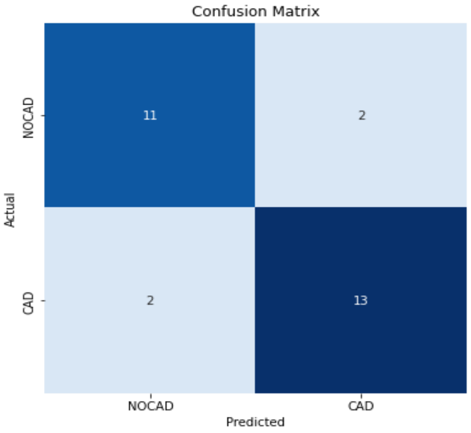
Table 6.22 (a-f): Confusion matrix plot of transferred learning model and developed model for twelve lead filtered two-dimensional signal images



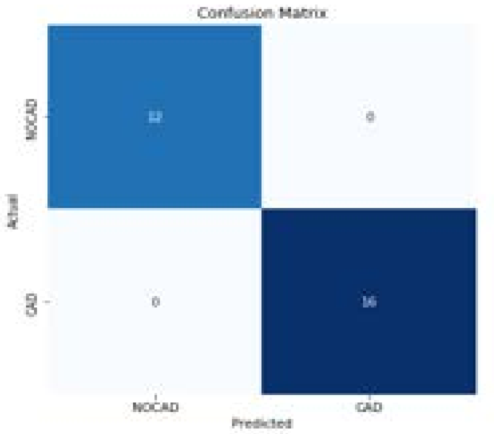
(a) Confusion matrix of VGG16 (b) Confusion matrix of MobileNetV2



(c) Confusion matrix of ResNet (d) Confusion matrix of Efficient Net



(e) Confusion matrix of Inception



(f) Confusion matrix of developed model

The parameters precision, recall, and f1-score which are derived from the confusion matrix are used to further evaluate all six algorithms. Table 6.23 represents the precision, recall, and f1-score values of the developed algorithm and pre-trained algorithms for filtered 12 lead ECG and TMT-ECG signal image conditions.

Table 6.23 Precision, Recall and F1 Score of transfer learned and developed network for 12 lead signal image data

Model	Precision		Recall		F1-score	
	CAD	NOCAD	CAD	NOCAD	CAD	NOCAD
VGG16	88%	100%	100%	85%	94%	92%
MobileNetV2	87%	85%	87%	85%	87%	85%
Inception	87%	85%	87%	85%	87%	85%
ResNet	93%	86%	87%	92%	90%	89%
Efficient Net	88%	92%	93%	85%	90%	88%
Developed CNN	100%	100%	100%	100%	100%	100%

The developed algorithm shows very good and acceptable results for all the parameters under test image conditions. Hence, the developed algorithm can be implemented as a prediction tool in medication conditions to identify the existence of coronary artery disease and normal (NOCAD) patients. These may extend to include an examination of a large medical ECG/TMT ECG image dataset in order to consider a full-fledged prediction tool. However, this will aid doctors operating in remote places to identify the severity of the disease. The algorithm can also help to take a second opinion for doctors when they are dealing with crucial CAD patients.

6.5 MULTI-HEADED MODEL

Every model consists of as a backbone plus a head. If the backbone trained for an added random head results for the better performance, then the model is known as multiheaded model. Majority of the literature studies deal with one-dimensional ECG signal data or ECG signal images (both the repository and are clinical data of ECG which anticipated from the ECG signal) (Nejedly et al. 2022; Suresh et al. 2020). These literatures focuses on signal image categorization using one-dimensional data and two-dimensional handling algorithms such as sequential models or pre-trained models. The major disadvantage of these studies is that they deal with only one data type and optimization will be limited. To overcome this, the current study concatenates images of single-lead ECG signals and extracted data from corresponding signals to build a multiheaded hybrid prediction model using functional modeling approach.

With the use of the supervised learning approach, this prediction system is deployed in a real-time application according to ethical standards and uses data from a reputed hospital [KMC Cardiology dept]. This helps to segregate CAD and normal patients effectively with respect to angiographic report.

To check the versatality of the model, the repository data set is also used to validate a model. The multiheaded hybrid prediction model is built with data extracted from single-lead ECG signal image(1D) and pre-processed single-lead ECG signal image (2D). The extracted time series data from single-lead ECG signal images are validated by reconstructing the image and finding Pearson’s correlation as described in the dataset preparation method.

The initial multiheaded prediction system is developed based on pre-processed single-lead ECG signal image which collected from the health care center. A total of 3312 ECG single-lead signals are taken into account, of which 1896 are related to CAD and 1416 are normal patient data. These are collected over the duration of one second. The related time series data for the images are utilized as 1D data in another head in parallel as per standard protocol. Out of 3312 data sets, 85% are used for training, 5% are used for validation and 10% are used for testing. The complete multi-headed hybrid prediction model setup is constructed, trained, and tested on a Collaboratory platform, known as Google collab developed by Google Research. This is well suited for Python-based Machine learning & deep learning algorithm as well as helpful for data analysis performance.

The developed hybrid model with hyper parameters, with a fixed batch size of 34 samples, and the maximum number of 50 epochs are provided using an early stopping pre-defined callback function. The model is trained with the Adam optimization algorithm with a learning rate of 0.0001. Since it's a two-label classification, binary cross-entropy is adopted as loss function. Binary cross-entropy is expressed mathematically as in equation 6.7 (Rai and Mitra 2021).

$$\mathrm{L}(\mathrm{P},\mathrm{T}) = -\mathrm{T_c}\ \ln \mathrm{P_c} - \mathrm{T_n}\ \ln \mathrm{P_n} \qquad (6.7)$$

Where L is the loss function, $\mathrm{P_c}$ and $\mathrm{P_n}$ are predicted the probability of CAD and Normal cases. $\mathrm{T_c}$ & $\mathrm{T_n}$ are target or expected probability of CAD & Normal case respectively.

The multiple call-back functions are used to monitor the data flow process and accuracy improvement of the model. Based on callback function response the reducing the learning rate command is induced over no improvement in validation loss with two successive epochs. The goal of the model checkpoint is to preserve the best, and optimized model.

Table 6.24 Performance accuracy of multi headed model with all the data types during different combinational approach

Model **Dataset**		ECG signal image (2D) + Extracted ECG time series data (1D) (%)	ECG signal image (2D) + ECG flattened image (1D) (%)	ECG signal image (2D) + Extracted ECG time series data (1D) + ECG flattened image (1D) (%)
Single Lead	Only ECG (3320)	91	82	90
	ECG +TMT-ECG (6640)	93	87	93
Multi-Lead	Only ECG (276)	93	89	96
	ECG +TMT-ECG (552)	93	93	96
Repository Data	ECG	---	---	88

The Table 6.24 shows the average performance accuracy of the model during training. The multiheaded hybrid deep neural network model is built with clinical data and its generalization ability is tested by considering 810 samples collected from the repository dataset where 360 are related to CAD and the remaining 450 are related to NOCAD patients. The results of repository data are drawn from the same model which is developed for clinical data. The classification ability (accuracy) of the model during training and testing are good and acceptable for both data types (as shown in Figure 6.20 and Figure 6.21).

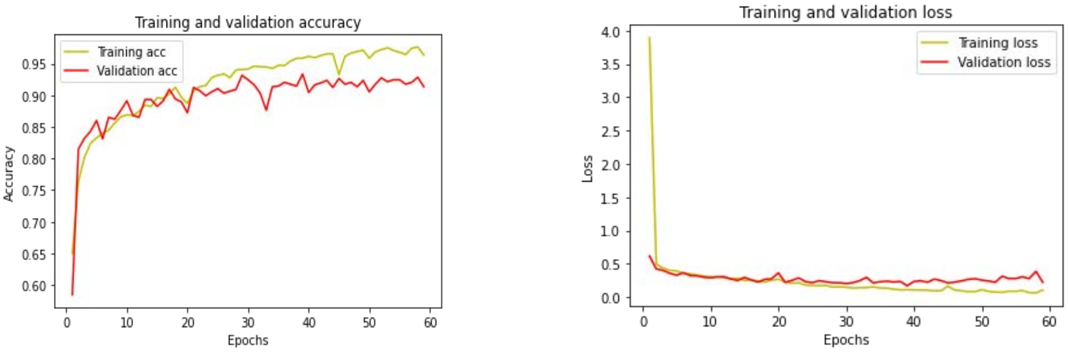


Figure 6.20 Clinical data multi headed model performance based on single lead with consideration of ECG and TMT signal image (2D) + Extracted ECG and TMT time series data (1D) +ECG flattened image (1D).

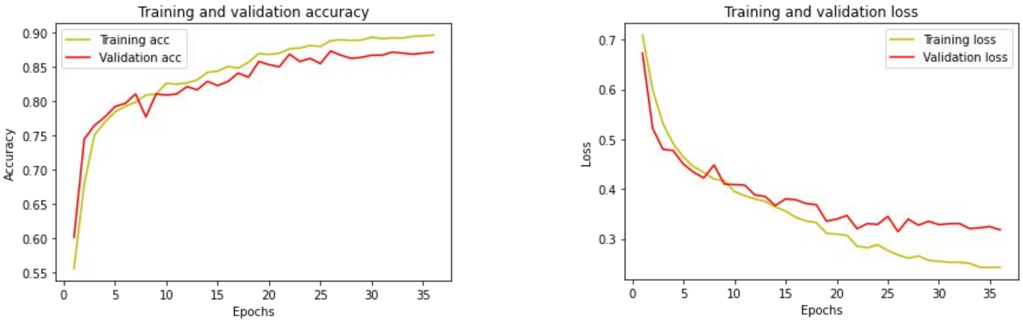


Figure 6.21 Repository data multi headed model performance based on single lead with consideration of ECG signal image (2D) + Extracted ECG time series data (1D) + ECG flattened image (1D).

This model is evaluated by applying the k-fold validation methodology. Usually, this methodology is used to assess the model when there is a limited data set condition, i.e. when there aren't enough data sets to segregate (train, validation, and test). The K-fold methodology is adopted for a model to understand the behavior of the model over an entire dataset. The K-fold approach divides the data into K subsets and repeats the process, where each time K-subset is used as a test set, and others are used as training data. In this study K-value is considered as ten, i.e. ten-fold cross-validation approach is used to evaluate the model reliability. The data set is divided into 10 subsets, of which 9 are utilized for training and 1 for testing. This 9:1 ratio fluctuates depending on the subset chosen for testing, indicating that all subsets other than the test subsets are used for training. The average accuracy drawn for this method is 96.1 for clinical data and 88.1 for repository data respectively, which indicates that the model is good and stable for the study of hybrid datatypes.

Table 6.25 Accuracies of the k-fold cross validation over multi headed model

Model Dataset		ECG signal image (2D) + Extracted ECG time series data (1D) (%)		ECG signal image (2D) + ECG flattened image (1D) (%)		ECG signal image (2D) + Extracted ECG time series data (1D) + ECG flattened image (1D) (%)	
Average K fold results		Accuracy %	Standard Deviation	Accuracy	Standard Deviation	Accuracy	Standard Deviation
Single Lead	Only ECG (3320)	91.7	0.064	82.2	0.2268	90.3	0.076
	ECG +TMT-ECG (6640)	92.8	0.056	87.4	0.138	93.1	0.053
Multi-Lead	Only ECG (276)	93.5	0.051	88.9	0.095	96.1	0.037
	ECG +TMT-ECG (552)	92.6	0.057	93.1	0.051	96.2	0.032
Repository Data	ECG	---		---		88.1	0.104

The ROC curve used as performance metrics states that how True-positive and False-positive values are predicted by the model (equation 6.8 and equation 6.9).

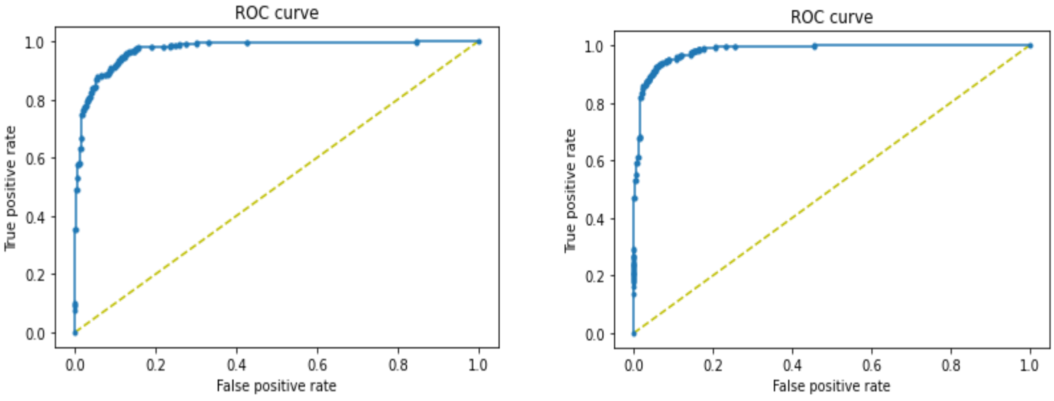
$$\text{True positive rate (TPR)} = \frac{TP}{(TP+FN)} \qquad (6.8)$$

$$\text{False positive rate (FPR)} = \frac{FP}{(FP+TN)} \qquad (6.9)$$

Where TP is True Positive, Tn is True Negative, FP is False Positive and FN is False Negative. ROC is the curve of TPR Vs FPR. For the model's best result outcomes, the plot of the curve should be lying on the left top corner. The Figures 6.22, 6.23 and 6.24

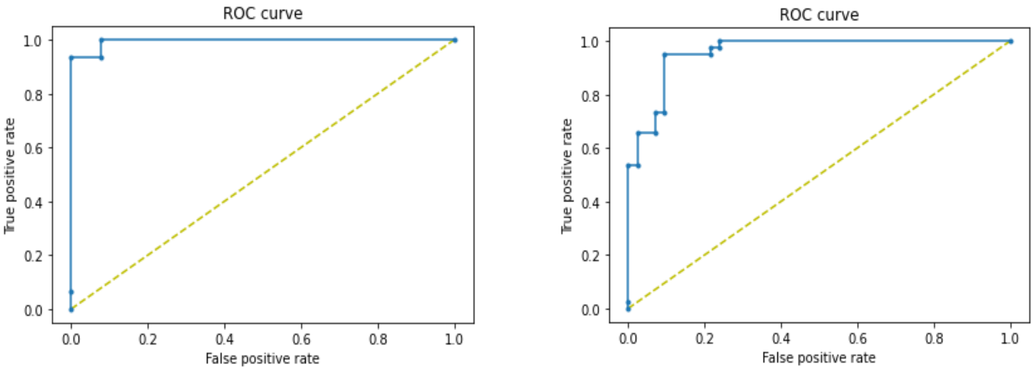
show the ROC plot of the model considering clinical data for all the conditions and 6.25 shows the ROC plot of the model considering repository data respectively.

(a) Single lead



(ECG) (TMT- ECG)

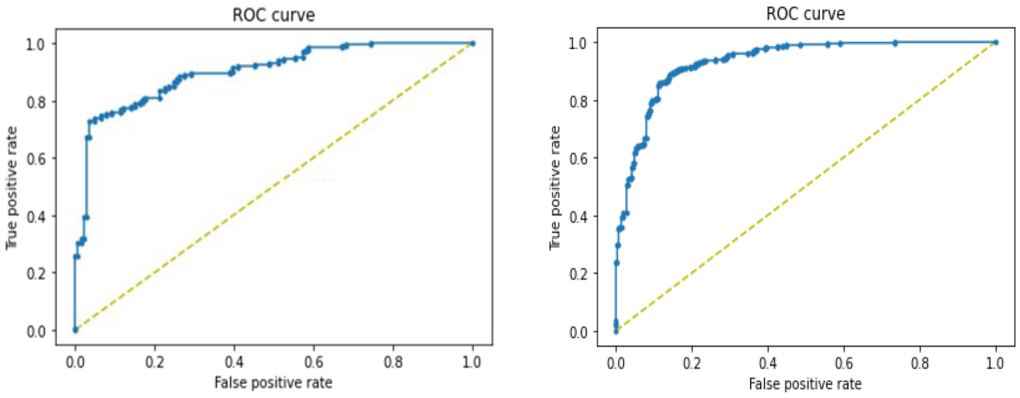
(b) Multi-Lead



(ECG) (TMT- ECG)

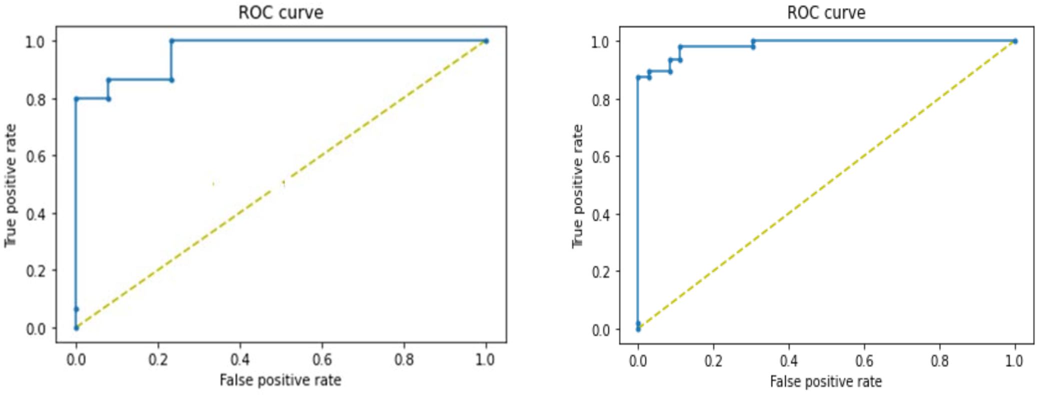
Figure 6.22 ROC plot for ECG signal image (2D data) +Extracted ECG time series Data (1D data) of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

(a) Single lead



(ECG) (TMT- ECG)

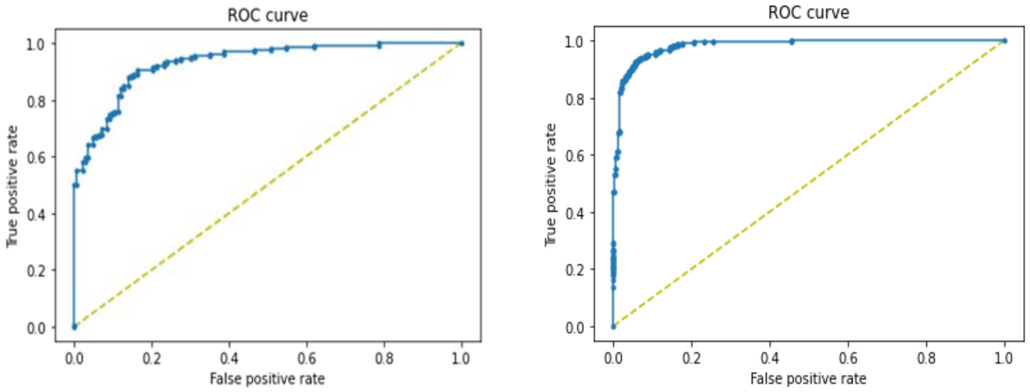
(b) Multi-Lead



(ECG) (TMT- ECG)

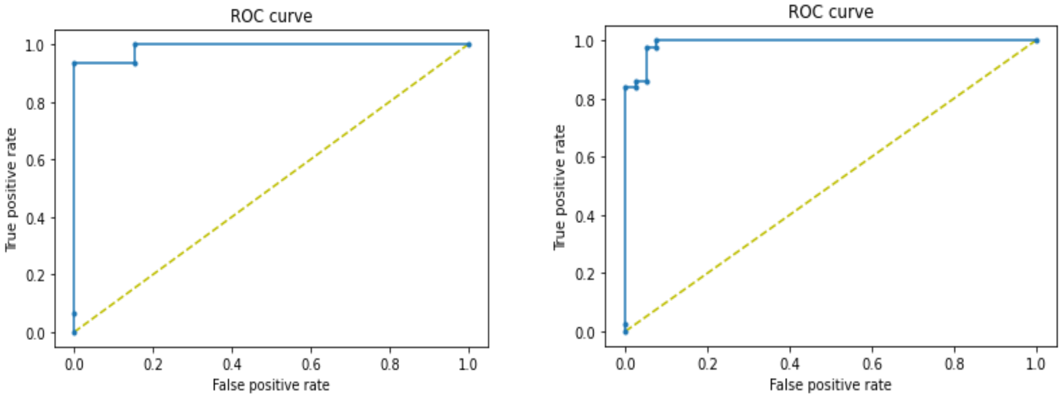
Figure 6.23 ROC plot for ECG signal image (2D data) + ECG flattened image (1D data) of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

(a) Single lead



(ECG) (TMT- ECG)

(b) Multi-Lead



(ECG) (TMT- ECG)

Figure 6.24 ROC plot for ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data of (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

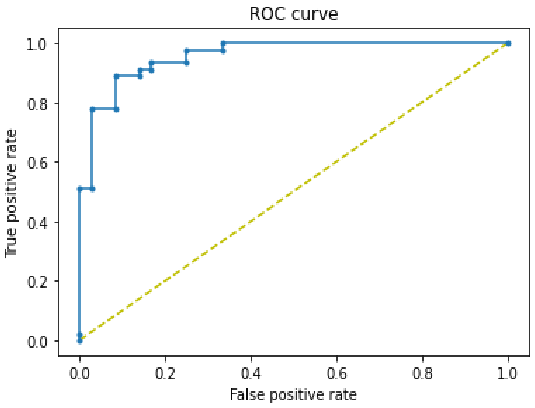


Figure 6.25 ROC of Repository data for single lead hybrid multi headed model

The AUC (Area under the ROC Curve) measures the overall effectiveness of every classification that could be used. This measures the model's accuracy and quality of prediction. Performance metric (Hameed et al. 2020; Seera and Lim 2014) measurement is most suitable for multi-class imbalanced data, and the macro-F1 score is the most important metric for measuring the performance of the classification model.

Table 6.26 Precision, Recall and F1 Score of multi headed model with all the data types during different combinational approach

Model **Dataset**		ECG signal image (2D) + Extracted ECG time series data (1D) (%)			ECG signal image (2D) + ECG flattened image (1D) (%)			ECG signal image (2D) + Extracted ECG time series data (1D) + ECG flattened image (1D) (%)		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Single Lead	Only ECG (3320)	92	91	91	81	82	82	89	90	90
	ECG +TMT-ECG (6640)	93	93	93	86	87	87	93	93	93
Multi- Lead	Only ECG (276)	93	93	93	90	90	90	97	97	97
	ECG +TMT-ECG (552)	93	93	93	93	93	93	97	97	97
Repository Data	ECG	--	--	--	--	--	--	88	88	88

The other performance metrics which are considered for analysis are

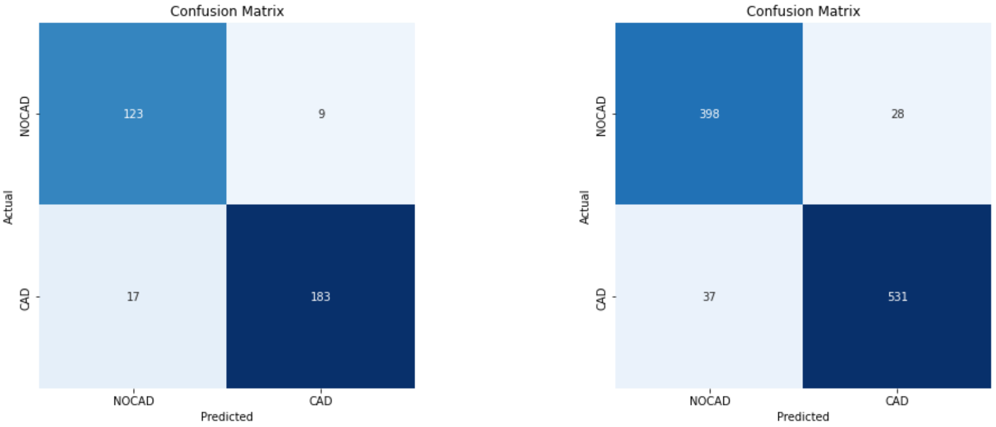
1) Precision
2) Recall (Recall is a sensitivity of the model which is defined as a true positive rate)
3) F1-score

The Table 6.26 shows the values of all these performance metrics for all the three cases over ECG and ECG+TMT-ECG of clinical data and repository data sets.

The classification performance of the testing data set is defined with the help of a confusion matrix. The Table 6.27, 6.28, 6.29 and 6.30 show performance of the confusion matrix for clinical data with consideration of all the conditions and repository data respectively under multiheaded hybrid model conditions. Here TN and TP corresponds to binary classified records (0,1) respectively, and FP and FN corresponds to wrongly classified records.

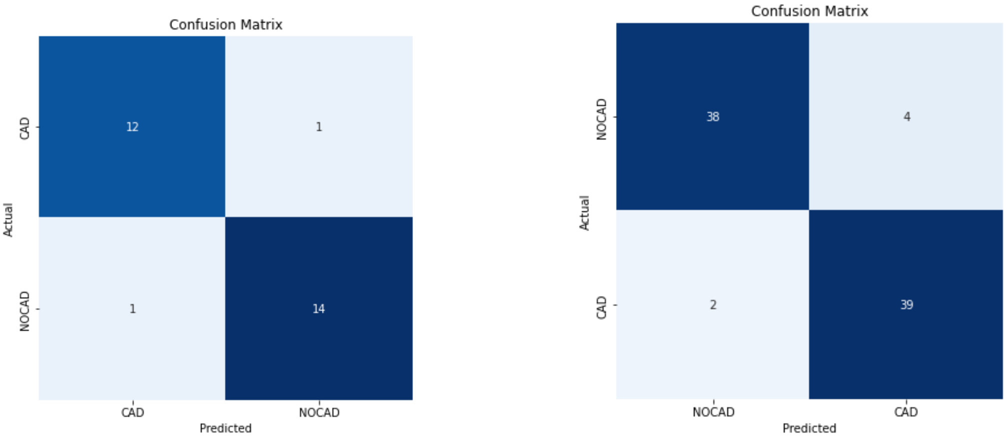
Table 6.27 Confusion matrix of ECG signal image (2D data) +Extracted ECG time series Data (1D data) for (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

(a) Single lead



(ECG) (TMT- ECG)

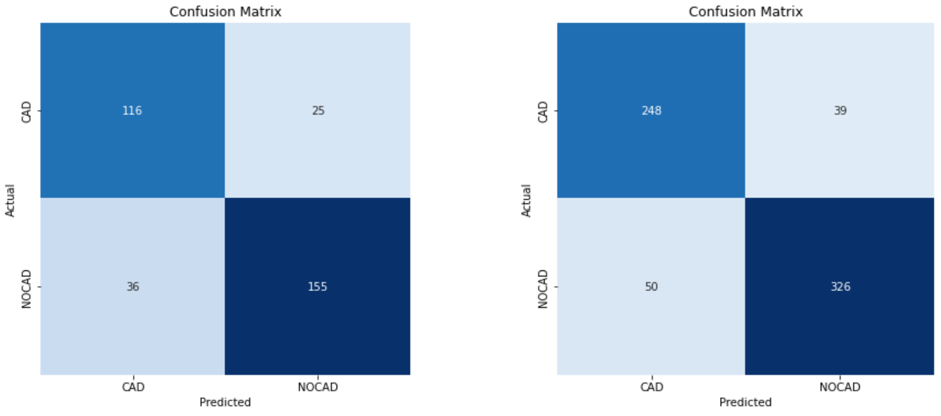
(b) Multi-Lead



(ECG) (TMT- ECG)

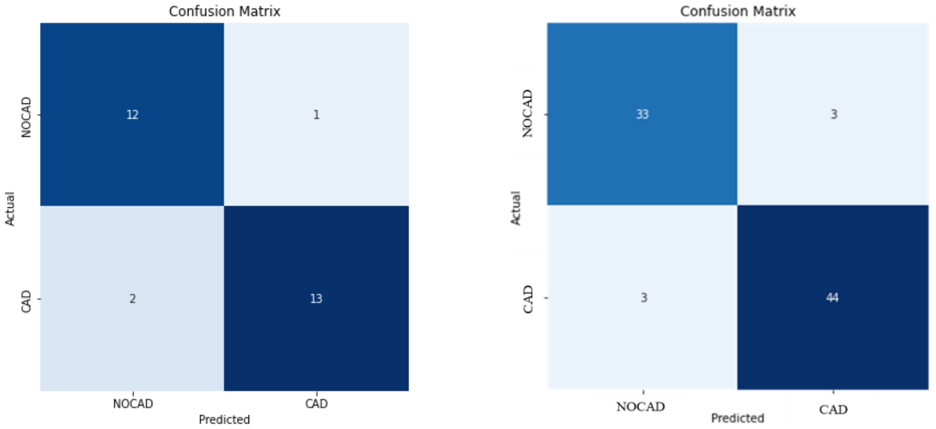
Table 6.28 Confusion matrix of ECG signal image (2D data) + ECG flattened image (1D data) for (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

(a) Single lead



(ECG) (TMT- ECG)

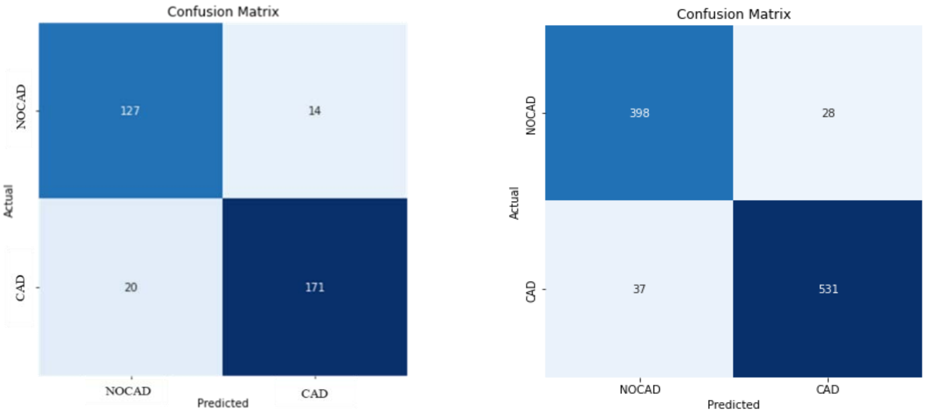
(b) Multi-Lead



(ECG) (TMT- ECG)

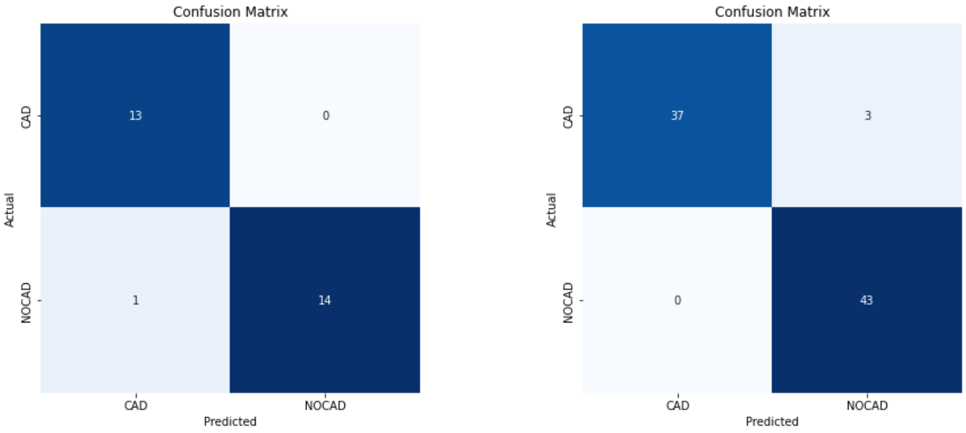
Table 6.29 Confusion matrix of ECG signal image (2D data) + ECG flattened image in (1D) + ECG 1D time series data for (a) single lead hybrid multi headed model and (b) Multi-lead hybrid multi headed model

(a) Single lead



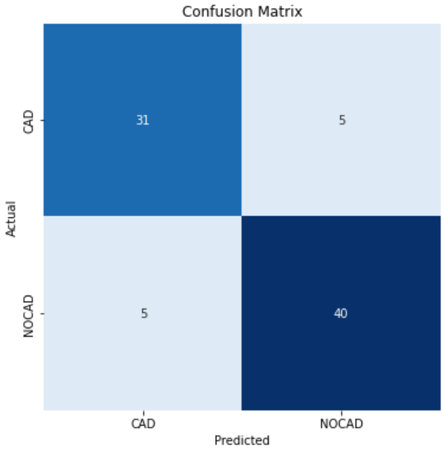
(ECG) (TMT- ECG)

(b) Multi-Lead



(ECG) (TMT- ECG)

Table 6.30 Confusion matrix of repository data for single lead hybrid multi headed model



Here, the data lying on TN, TP, FP & FN indexes can easily visualize. By studying these performance metrics, it is observed that multiheaded hybrid DNN model is the most suitable model for the classification of single lead ECG signals and twelve lead ECG signals, towards detecting coronary artery disease. But, the computation time in this model is higher when compared to normal sequential CNN model due to complex nature of hybrid data. The system is suitable for real-time clinical implementation with a suitable GUI and doctors can access it as the assisting tool for continuous evaluation of periodic diagnosis of patients during medication.

6.6 GRAPHICAL USER INTERFACE (GUI)

A simple graphical user interface (GUI) (Cairns et al. 2016) (Figure 6.26 home page) is developed for image prediction model, taking into account the trained weights, with consideration of single lead and twelve lead ECG and TMT ECG signal images. The GUI is developed using Visual Studio software. It provides an option to select whether the prediction is to be performed with the help of single lead ECG images (ECG and TMT ECG) or 12 lead ECG images (ECG & TMT ECG).

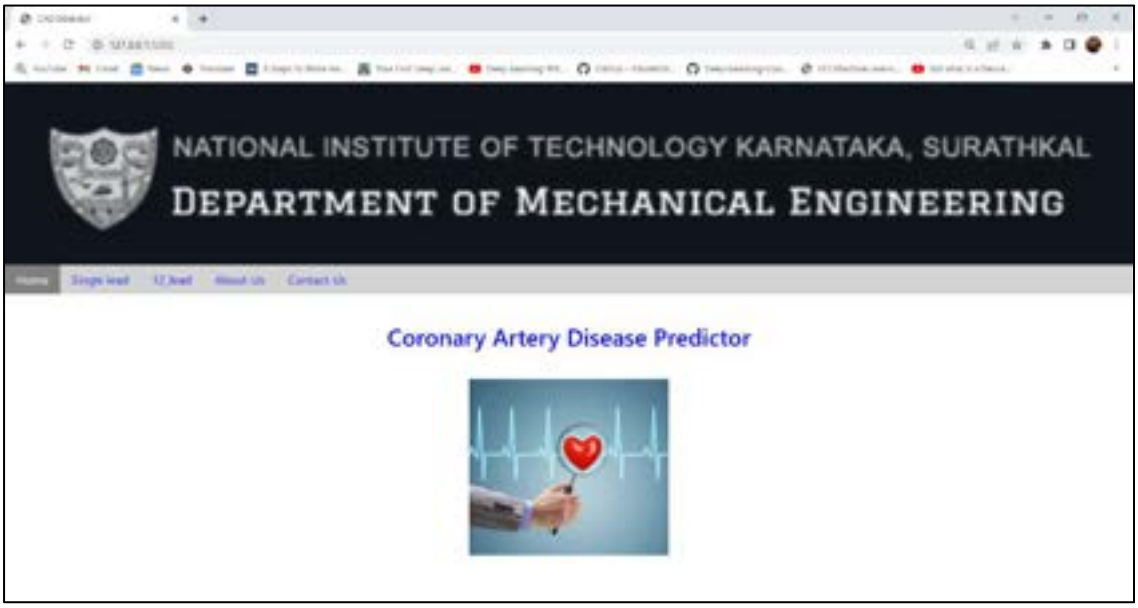

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL
DEPARTMENT OF MECHANICAL ENGINEERING
Home
Single lead
12 lead
About Us
Contact Us
Coronary Artery Disease Predictor

Figure 6.26 GUI home page of prediction model

The GUI incorporates a module for preprocessing imported raw ECG and TMT (single lead or twelve lead) signals. Once the raw image (either single lead or twelve lead signals) has been imported into the GUI, the image is automatically filtered by pre-processing step insertion. These filtered images are analyzed based on the weights which corresponds to trained single lead or twelve lead signal images algorithms.

Figure 6.27 and Figure 6.28 depicts the selection of single lead and twelve lead raw ECG signal images respectively. The "predict" button invokes will generation of results through the proper selection of image preprocessing. The results are in terms of binary prediction type, i.e either "predicted positive" or "predicted negative".

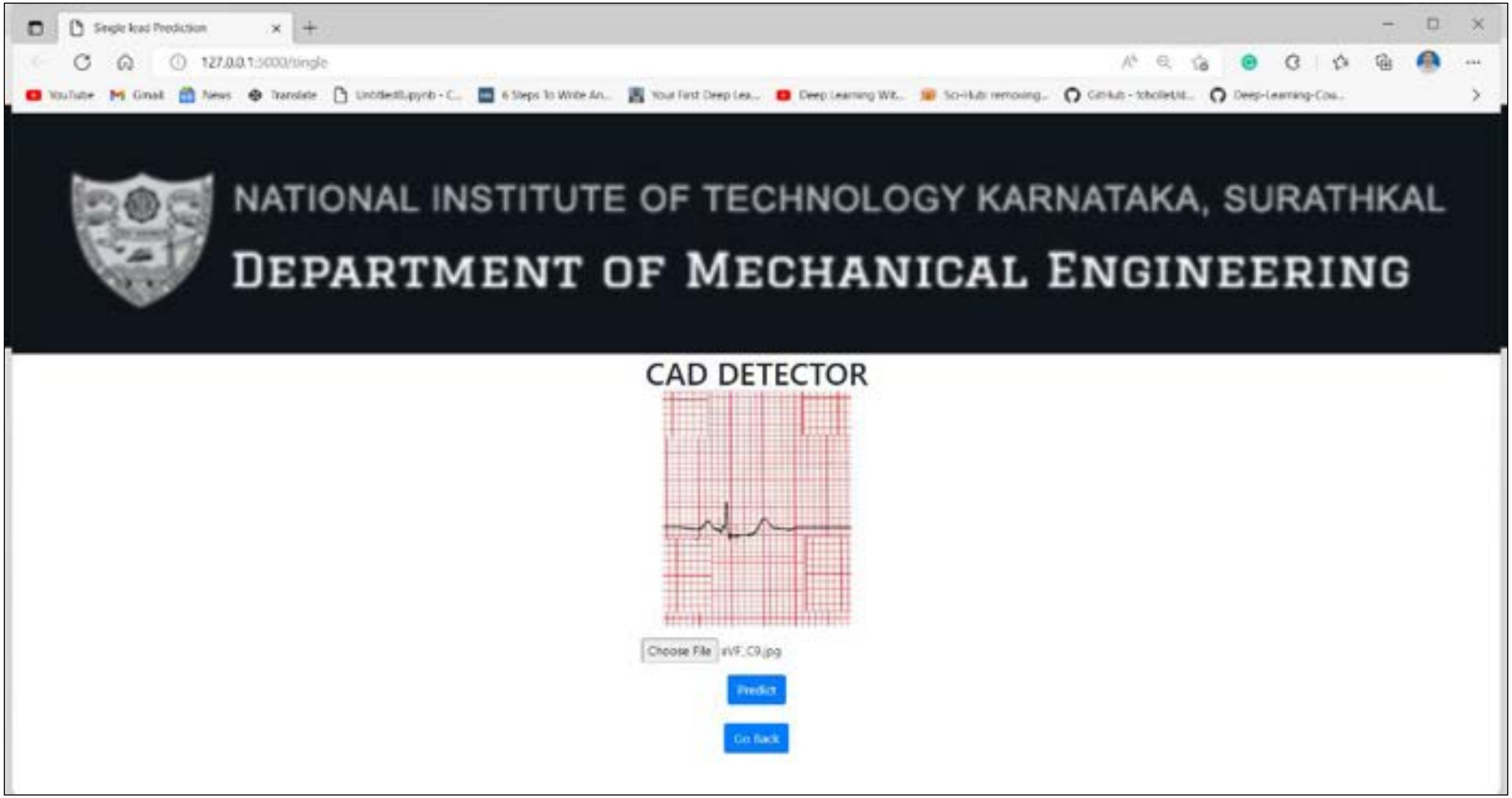
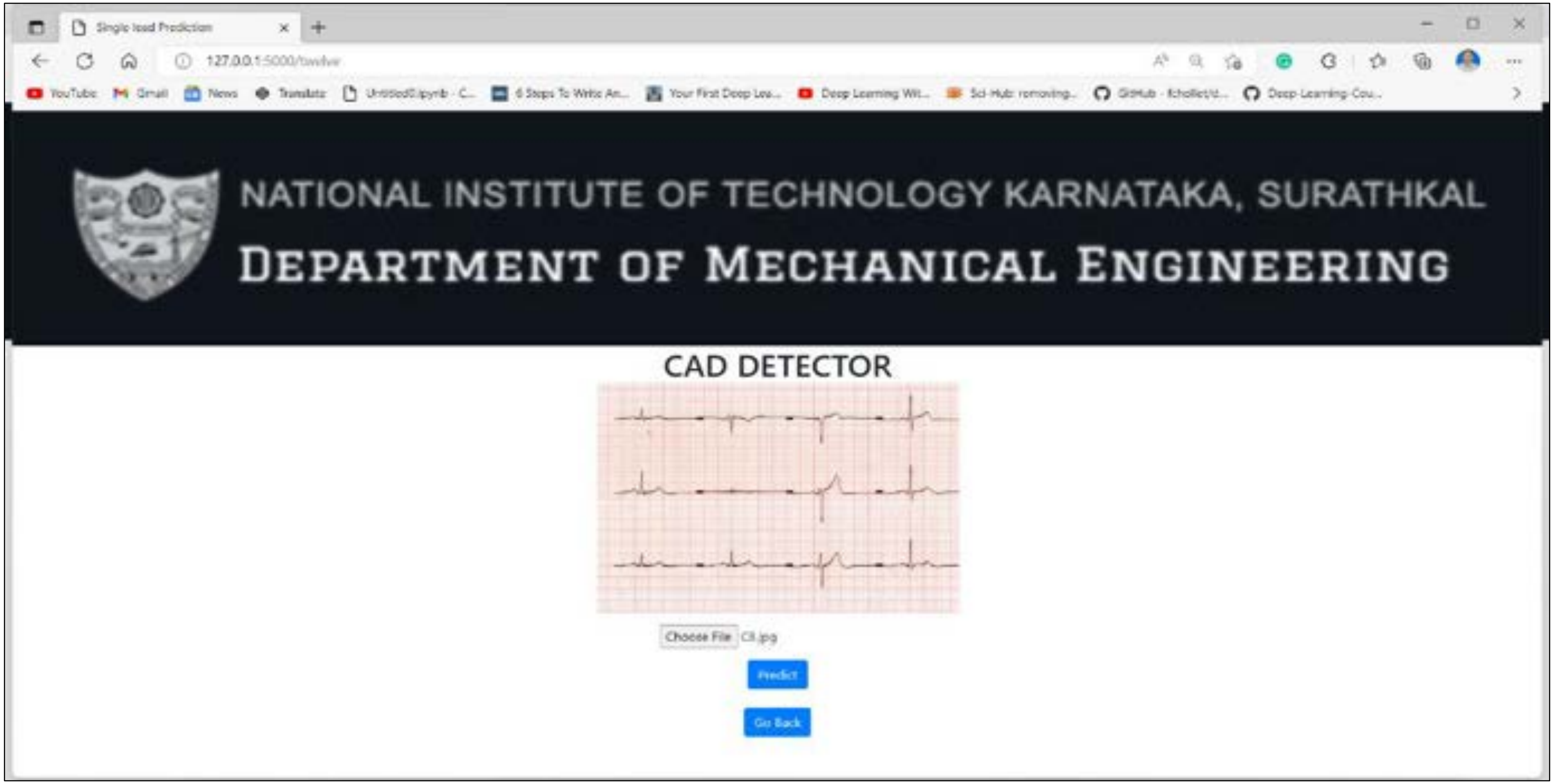
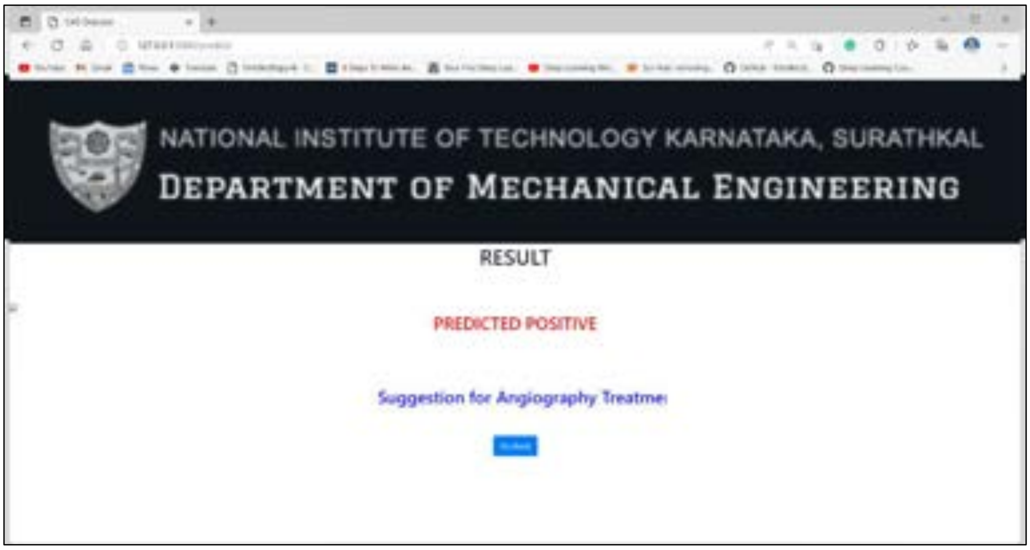


Figure 6.27 Single lead ECG image selection page


Single lead Prediction
127.0.0.1:5000/twelve
NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL
DEPARTMENT OF MECHANICAL ENGINEERING
CAD DETECTOR
Choose File
Predict
Go Back

C8.jpg

Figure 6.28 Twelve lead ECG image selection page



NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

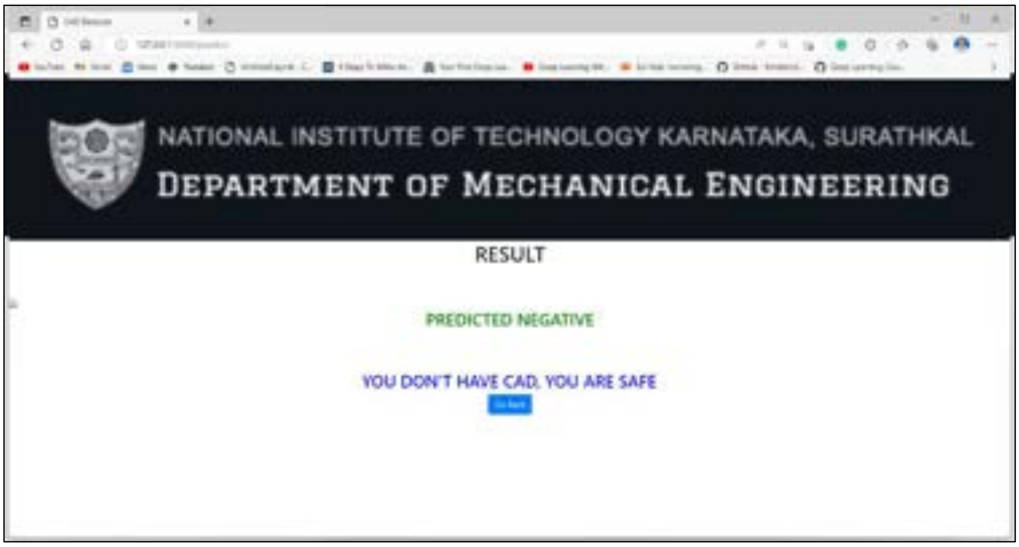
DEPARTMENT OF MECHANICAL ENGINEERING

RESULT

PREDICTED POSITIVE

Suggestion for Angiography Treatme

Figure 6.29 Indication model results in terms of "Predicted Positive" for Single lead and multi lead ECG/ TMT-ECG selected image



NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA, SURATHKAL

DEPARTMENT OF MECHANICAL ENGINEERING

RESULT

PREDICTED NEGATIVE

YOU DON'T HAVE CAD, YOU ARE SAFE

Figure 6.30 Indication model results in terms of "Predicted Negative" for Single lead and multi lead ECG/ TMT-ECG selected image

The protocol suggestions are defined as either "suggestion for angiography treatment" or "you don't have CAD, you are safe" (as shown in Figure 6.37 and Figure 6.38). This will help to sort out the categories and even helps doctors in rural area to take preliminary actions as well as judge the need to consult cardiology experts for possible treatment related to CAD.

RESEARCH SUMMARY

Prediction systems prove to be powerful tool for doctors in correlating disease behavioral cases. The study involved the development of two types of prediction systems, one for the undifferentiated fever cases and the other one for the coronary artery disease (CAD).

The presented method of analysis and prediction can reduce diagnostic complications and aid in the early diagnosis of diseases. Using only the temperature data of disease as an input to the system, and with the help of a noninvasive model, the prediction of major diseases like Tuberculosis, Non-tubercular bacterial infections, Dengue fever, and Non-infectious diseases can be done accurately without any other clinical parameters.

The prediction model is initially developed with single variable data (temperature) captured for 24 hours duration. Accuracy of this system is found to be 99%. In the next step, to reduce the size of data being handled and refine the present system, 30 minutes' and 60 minutes' duration data (i.e. temperature) are considered with and without additional features, which are commonly adopted in medical practice. Various classifiers are tried out for the prediction system.

The results have shown good outcome characteristics in a prediction system with decision tree classifiers over other classifiers. Decision tree and k-Nearest neighbor classifiers provided a good resultant accuracy is 100% and the F1 score is 1 and has dominated using its effective way of classification of Tuberculosis, Non-tubercular bacterial infection, Dengue fever, Non-infectious diseases over 30 minutes of temperature data with additional feature consideration. Thus, this diagnostic system can be of great assistance to doctors treating patients with symptomatic fever diseases and saving lives in crucial instances.

In the next stage, an intelligent prediction system is developed for the binary classification of coronary artery disease. The basic data used are the live data in the form of ECG and TMT-ECG signals gathered from a reputed hospital. A novel image processing approach is implemented for relevant feature extraction, optimally.

Data extraction is necessary to store thermal paper data for future data analysis securely. The developed image processing and data extraction method are the reliable and the best fit for all the cases of ECG and TMT-ECG data extraction. The data extracted are compared and validated with commercially available software tool and manual method, and the accuracy is found to be 98%. The proposed data extraction method is also validated with the digitally available dataset (Physio-net data set) and the results here are promising with a correlation coefficient of 0.975. The algorithm further enriched with support for additional features is capable to extract the major important features with 93.6% accuracy. A hybrid feature extraction method is used here. These extracted data and features help to build a good intelligent prediction model to support doctors and healthcare experts to diagnose cardiovascular diseases.

Thus, prediction model based on CNN architecture is developed for both single-lead and 12-lead ECG datasets compiled from clinical data. While 6630 datasets are used for the single-lead prediction model, 552 datasets are used for the 12-lead model. A hybrid method combining Pan–Tompkins, slope-based and a statistical method involving feature extraction is implemented, which resulted in extracting all of the features related to the ECG. During the modeling, the CNN model with a single layer, two layers, and three layers are experimented with different activation functions in different layers. Later, the three-layer CNN model was found to be the best architecture, with ReLU, ReLU, and LeakyReLU activation functions in the first, second, and third layers, respectively. This model showed an accuracy of 99% during the training phase, while it exhibited an accuracy of 98.6% in the validation trials. Thus, the three-layer CNN model, with ReLU, ReLU, and LeakyReLU activation functions works well for single-lead as well as 12-lead ECG datasets and is able to predict the CAD much more reliably than any other models. Hence, the present model has proved to be the best fit for the classification of CAD.

The diagnosis of CAD at an early stage will help the doctors during medication stage. The sequential deep learning model developed based on CNN optimization effectively classifies the disease over the single lead and multi-lead (12 lead) ECG and TMT ECG signal image dataset. The image dataset considered are raw image and filtered image dataset (which are achieved from pre-processing methodology). The developed algorithm results are compared in terms of accuracy, confusion matrix, and ROC with the well-known transfer learned algorithms (VGG16, MobileNetV2, Inception, ResNet, EfficientNet) in the field of signal image classification. The developed solution is dominating with 93.2% accuracy over a single lead and 94% for twelve lead ECG, TMT ECG signal images.

To analyze the combined effects of ECG and TMT-ECG extracted time series data (1D) and image dataset (2D), multi headed model is opted. These multi-headed models are analyzed based on different conditions like extracted time series dataset (1D) + image dataset (2D), flattened image dataset (1D) + image dataset (2D) and extracted time series dataset (1D) + flattened image dataset (1D) + image dataset (2D) with ECG alone and combined ECG and TMT-ECG datasets respectively. The three headed model deals with all (extracted time series dataset (1D) + flattened image dataset (1D) + image dataset (2D)) and shows an acceptable result (96%) which is further validated with repository dataset. This confirms the generalization ability of the model. Thus, the developed multi-headed model can reliably predict either using ECG data alone or ECG combined with TMT-ECG also, once again proving its generalization.

This prediction system, with good graphical user interface (GUI) when used with a portable ECG unit can be of immense use in primary health care centers. Thus, all possibilities of combination of data, relevance to classify CAD have been investigated, with proper integration of signal input, image processing, analysis and detection of CAD, with graphical user interface.

Conclusion

1. The study revealed that 24 hours temperature data is sufficient to predict fever symptomatic disease (namely Tuberculosis, Non-tubercular bacterial infections, Dengue fever, Non-infectious diseases) in a patient. Refinement of the study for 30 min temperature data, with medical protocol provide good results.
2. The data extracted through the novel hybrid feature extraction method (i.e. P, Q, R, S, T, PQ-segments, QRS complex and ST segments) closely resemble to the actual ECG feature values.
3. Prediction model dealing with 1D data provides 98.6%, the 2D data provides 94% and the novel scheme of multidimensional approach provides 96% accuracy in classification of CAD and NoCAD.
4. K-fold (5-fold) validation is adopted to each model and achieved overall validation accuracy more than 98%.
5. An end-to-end modular integration system is developed by combining all phases of algorithm. This integrated GUI helps the doctors to interact with the model through data effectively and conveniently.

FUTURE WORK

The developed prediction models (machine learning algorithm for fever symptomatic disease classification, deep neural networks deals ECG and TMT-ECG data and ECG and TMT-ECG signal images) is tried to cover all the possibility of area used for the analysis. The analysis of work also deals with multi headed models which are parallelly deals both the data and images to at time with different algorithms to better identification and classification of disease. Which may further improve with

1. Including more clinical data with irrespective to ECG machines,
2. Generalization ability by analyzing more repository data or clinical data,
3. Development of full-fledged application helps to access the model through smartphone, and
4. The web based graphical user interface helps to access the model with respective of places.

These are major improvement can implement and even the comparative deep neural network for already classified fever symptomatic diseases is one of the studies to analyze the behavior of data. This improvement will helps overcome especially like over rural areas health issues.

REERENCES

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z, Citro C., Corrado G. S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mane D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viegas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., and Zheng X. (2016). "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems." *Distributed, Parallel, and Cluster Computing, Machine Learning*, 2, 1–19.

Acharya U. R., Faust O., Sree V., Swapna G., Martis R. J., Kadri N. A., and Suri J. S. (2014). "Linear and nonlinear analysis of normal and CAD-affected heart rate signals." *Computer Methods and Programs in Biomedicine*, 113(1), 55–68.

Acharya U. R., Fujita H., Lih O. S., Adam M., Tan J. H., and Chua C. K. (2017). "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network." *Knowledge-Based Systems*, 132, 62–71.

Acharya U. R., Fujita H., Lih O. S., Hagiwara Y., Tan J. H., and Adam M. (2017). "Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network." *Information Sciences*, 405, 81–90.

Acharya U. R., Oh S. L., Hagiwara Y., Tan J. H., Adam M., Gertych A., and Tan R. S. (2017). "A deep convolutional neural network model to classify heartbeats." *Computers in Biology and Medicine*, 89(July), 389–396.

Adeniyi D. A., Wei Z., and Yongquan Y. (2016). "Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method." Applied Computing and Informatics, 12(1), 90–108.

Akinpelu D. (2015). "Treadmill Stress Testing Technique." *Medscape*, 1-3.

Akula R., and Mohamed H. (2019). "Automation algorithm to detect and quantify electrocardiogram waves and intervals." *Procedia Computer Science*, 151, 941–946.

Al-ani M. (2014). “A rule-based expert system for automated ecg diagnosis.” *International Journal of Advances in Engineering & Technology,* 6(4), 1480-1493.

Al-Khatatneh A., Pitchay S. A., and Al-Qudah M. (2016). “A Review of Skew Detection Techniques for Document.” *Proceedings - UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim 2015*, 316–321.

Alarsan F. I., and Younes M. (2019). “Analysis and classification of heart diseases using heartbeat features and machine learning algorithms.” *Journal of Big Data,* 6(1), 1-15.

Alizadehsani R., Roshanzamir M., Abdar M., Beykikhoshk A., Khosravi A., Panahiazar M., Koohestani A., Khozeimeh F., Nahavandi S., and Sarrafzadegan N. (2019). “A database for using machine learning and data mining techniques for coronary artery disease diagnosis.” *Scientific Data,* 6(1), 1–13.

Alzubaidi L., Zhang J., Humaidi A. J., Al-Dujaili A., Duan Y., Al-Shamma O., Santamaría J., Fadhel M. A., Al-Amidie M., and Farhan L. (2021). “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions”. *Journal of Big Data,* 8(1), 1-74.

Amato F., López A., Peña-Méndez E. M., Vaňhara P., Hampl A., and Havel J. (2013). “Artificial neural networks in medical diagnosis.” *Journal of Applied Biomedicine.,* 11(2), 47–58.

Andreotti F., Carr O., Pimentel M. A. F., Mahdi A., and Vos M. De. (2017). “Comparing Feature Based Classifiers and Convolutional Neural Networks to Detect Arrhythmia from Short Segments of ECG.” *Computing in Cardiology,* 44, 1–4.

Anggraeni W., Nurmasari R., Riksakomara E., Samopa F., Wibowo R. P., Condro T. L., and Pujiadi. (2017). “Modified Regression Approach for Predicting Number of Dengue Fever Incidents in Malang Indonesia.” *Procedia Computer Science,* 124, 142–150.

Aqib Haqmi Abas M., Ismail N., Ihsan Mohd Yassin A., and Nasir Taib M. (2018).

"VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation." *International Journal of Engineering & Technology,* 7(4.11), 90-94.

Arafat S., Dohrmann M., and Skubic M. (2005). "Classification of Coronary Artery Disease Stress ECGs using Uncertainty Modeling." *2005 ICSC Congress on Computational Intelligence Methods and Applications, IEEE*, 2005, 1–4.

Atkov O. Y., Gorokhova S. G., Sboev A. G., Generozov E. V., Muraseyeva E. V., Moroshkina S. Y., and Cherniy N. N. (2012). "Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters." *Journal of Cardiology*, 59(2), 190–194.

Azar A. T., and El-Metwally S. M. (2013). "Decision tree classifiers for automated medical diagnosis." *Neural Computing and Applications*, 23(7), 2387–2403.

Ay B., Yildirim O., Talo M., Baloglu U.B., Aydin G., Puthankattil S.D., Acharya U.R., (2019). "Automated Depression Detection Using Deep Representation and Sequence Learning with EEG Signals." *Journal of Medical Systems,* 43(7), 1-12.

Babaoglu I., Baykan O. K., Aygul N., Ozdemir K., and Bayrak M. (2009). "Assessment of exercise stress testing with artificial neural network in determining coronary artery disease and predicting lesion localization." *Expert Systems with Applications*, 36, 2562–2566.

Babaoglu İ., Findik O., and Ülker E. (2010). "A comparison of feature selection models utilizing binary particle swarm optimization and genetic algorithm in determining coronary artery disease using support vector machine." *Expert Systems with Applications*, 37(4), 3177–3183.

Badawy M. M., and Muaidi Q. I. (2019). "Cardio respiratory response: Validation of new modifications of Bruce protocol for exercise testing and training in elite Saudi triathlon and soccer players." *Saudi Journal of Biological Sciences*, 26(1), 105–111.

Badilini F., Erdem T., Zareba W., and Moss A. J. (2003). "ECGScan: a method for conversion of paper electrocardiographic printouts to digital electrocardiographic files.” *Journal of Electrocordialogy,* 38, 310–318.

Baek Y., Lee B., Han D., Yun S., and Lee H. (2019). “Character region awareness for text detection.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2019, 9357–9366.

Baloglu U. B., Talo M., Yildirim O., Tan R. S., and Acharya U. R. (2019). “Classification of myocardial infarction with multi-lead ECG signals and deep CNN.” *Pattern Recognition Letters*, 122, 23–30.

Baxt W. G. (1991). “Use of an artificial neural network for the diagnosis of myocardial infarction.” *Annals of Internal Medicine,* 115(11), 843–848.

Baydoun M., Safatly L., Abou Hassan O. K., Ghaziri H., Hajj A. El, and Isma’eel H. (2019). “High Precision Digitization of Paper-Based ECG Records: A Step Toward Machine Learning.” *IEEE Journal of Translational Engineering in Health and Medicine*, 7, 1–8.

Bayoh M. N., and Lindsay S. W. (2003). “Effect of temperature on the development of the aquatic stages of Anopheles gambiae sensu stricto (Diptera: Culicidae).” *Bulletin of Entomological Research*, 93(5), 375–381.

Bayoh M. N., and Lindsay S. W. (2004). “Temperature-related duration of aquatic stages of the Afrotropical malaria vector mosquito Anopheles gambiae in the laboratory.” *Medical and Veterinary Entomology*, 18(2), 174–179.

Belavagi M. C., and Muniyal B. (2016). “Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection.” *Procedia Computer Science,* 89, 117–123.

Bertalmío M., Bertozzi A. L., and Sapiro G. (2001). “Navier-Stokes, fluid dynamics, and image and video inpainting.” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1, 355-362.

Boden M. A. (1998). “Creativity and artificial intelligence.” *Artificial Intelligence,* 103(1–2), 347–356.

Cairns A. W., Bond R. R., Finlay D. D., Breen C., Guldenring D., Gaffney R., Gallagher A. G., Peace A. J., and Henn P. (2016). “A computer-human interaction model to improve the diagnostic accuracy and clinical decision-making during 12-lead electrocardiogram interpretation.” Journal of Biomedical Informatics, 64, 93–107.

Carolina Sparavigna A. (2015). “An Image Processing Approach Based on Gnu Image Manipulation Program Gimp to the Panoramic Radiography.” *International Journal of Sciences*, 1(05), 57–67.

Celin S., and Vasanth K. (2018). “ECG Signal Classification Using Various Machine Learning Techniques.” *Journal of Medical Systems*, 42(12), 1-11.

Chen T. M., Huang C. H., Shih E. S. C., Hu Y. F., and Hwang M. J. (2020). “Detection and Classification of Cardiac Arrhythmias by a Challenge-Best Deep Learning Neural Network Model.” *iScience*, 23(3), 1-27.

Cireşan D. C., Meier U., Gambardella L. M., and Schmidhuber J. (2011). “Convolutional neural network committees for handwritten character classification.” *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, 10, 1135–1139.

Clifford G., Liu C., Moody B., Lehman L., Silva I., Li Q., Johnson A., and Mark R. (2017). “AF Classification from a Short Single Lead ECG Recording: the Physionet Computing in Cardiology Challenge 2017.” *Computing in Cardiology*, 44, 1–4.

Çolak M. C., Çolak C., Kocatürk H., Sağiroglu Ş., and Barutçu I. (2008). “Predicting coronary artery disease using different artificial neural network models.” *Anadolu Kardiyoloji Dergisi*, 8(4), 249–254.

Dakappa P. H., Prasad K., Rao S. B., Bolumbu G., Bhat G. K., and Mahabala C. (2017). “A Predictive Model to Classify Undifferentiated Fever Cases Based on Twenty-Four-Hour Continuous Tympanic Temperature Recording.” *Journal of Healthcare Engineering*, 2017, 1-6.

Dakappa P. H., Prasad K., Rao S. B., Bolumbu G., Bhat G. K., and Mahabala C. (2018).

“Classification of Infectious and Noninfectious Diseases Using Artificial Neural Networks from 24-Hour Continuous Tympanic Temperature Data of Patients with Undifferentiated Fever.” *Critical Reviews in Biomedical Engineering*, 46(2), 173–183.

Davalagi S., Palicheralu B., Murthy S., and Kashyup M. (2020). “Acceptance and feasibility for handheld tele-ECG.” *Indian Journal of Community Health.*, 32(3), 586–589.

Davari Dolatabadi A., Khadem S. E. Z., and Asl B. M. (2017). “Automated diagnosis of coronary artery disease (CAD) patients using optimized SVM.” *Computer Methods and Programs in Biomedicine*, 138, 117–126.

Debasish Biswas, et. al. (2011). “Novel Gray Scale Conversion Techniques Based on Pixel Dept.” *Journal of Global Research in Computer Science*, 2(7), 79–82.

Rosack, Jim (2006). “Recommendations for Treatment.” *Psychiatric News*, 41(1), 1–37.

Donida Labati R., Muñoz E., Piuri V., Sassi R., and Scotti F. (2018). “Deep-ECG: Convolutional Neural Networks for ECG biometric recognition.” *Pattern Recognition Letters*, 0, 1–8.

Doshi R., Falzon D., Thomas B. V., Temesgen Z., Sadasivan L., Migliori G. B., and Raviglione M. (2017). “Tuberculosis control, and the where and why of artificial intelligence.” *ERS Monograph*, 3(2), 1–5.

Dr.N.Kannathasan. (2018). “Dengue disease prediction in India using Machine learning approaches.” *International Journal of Computer Engineering and Applications*, XII(I), 121–126.

Dyego E., Batista D. A., Bublitz F. M., and Lira R. V. (n.d.). “Dengue Prediction Through Machine Learning and Deep Learning : A Scoping Review Protocol .” 1–9.

Ebrahimi Z., Loni M., Daneshtalab M., and Gharehbaghi A. (2020). “A review on deep learning methods for ECG arrhythmia classification.” *Expert Systems with Applications X*, 7, 100033.

Ebrahimzadeh E., and Pooyan M. (2011). “Early detection of sudden cardiac death by using classical linear techniques and time-frequency methods on electrocardiogram signals.” *Journal of Biomedical Science and Engineering*, 04(11), 699–706.

Falkinham J. O. (2003). “Mycobacterial aerosols and respiratory disease.” *Emerging Infectious Diseases*, 9(7), 763–767.

Fan X., Hu Z., Wang R., Yin L., Li Y., and Cai Y. (2020). “A novel hybrid network of fusing rhythmic and morphological features for atrial fibrillation detection on mobile ECG signals.” *Neural Computing and Applications*, 32(12), 8101–8113.

Fariha M. A. Z., Ikeura R., Hayakawa S., and Tsutsumi S. (2020). “Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals.” *J. Phys. Conf. Ser.*, 1532(1).

Fathima K., and Vimina E. R. (2022). “Heart Disease Prediction Using Deep Neural Networks: A Novel Approach.” *Lect. Notes Networks Syst.*, 213, 725–736.

Frieden T. R., Brudney K. F., and Harries, A. D. (2014). “Global tuberculosis: perspectives, prospects, and priorities.” *JAMA*, 312(14), 1393–4.

Fujino A., Ueda N., and Saito K. (2007). “A hybrid generative / discriminative approach to text classification with additional information.” 43, 379–392.

Garg D. K. (2012). “ECG Paper Records Digitization through Image Processing Techniques.” *Int. J. Comput. Appl.*, 48(13), 35–38.

Ghiasi S., Abdollahpur M., Madani N., Kiani K., and Ghaffari A. (2017). “Atrial fibrillation detection using feature based algorithm and deep convolutional neural network.” *Comput. Cardiol. (2010).*, 44, 1–4.

Ghosh B., Dutta I. K., Carlson A., Totaro M., and Bayoumi M. (2020). “An Empirical Analysis of Generative Adversarial Network Training Times with Varying Batch Sizes.” *2020 11th IEEE Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2020,* (October), 0643–0648.

Giri D., Rajendra Acharya U., Martis R. J., Vinitha Sree S., Lim T. C., Ahamed T., and Suri J. S. (2013). “Automated diagnosis of Coronary Artery Disease affected patients using LDA, PCA, ICA and Discrete Wavelet Transform.” *Knowledge-Based Syst.*, 37, 274–282.

Gupta V., and Mittal M. (2021). “R-peak detection for improved analysis in health informatics.” *Int. J. Med. Eng. Inform.*, 13(3), 213–223.

Gupta V., Mittal M., and Mittal V. (2021). “FrWT-PPCA-Based R-peak Detection for Improved Management of Healthcare System.” *IETE J. Res.*, 1–15.

Guzman M. G., Halstead S. B., Artsob H., Buchy P., Farrar J., Nathan M. B., Pelegrino J. L., Simmons C., and Yoksan S. (2010). “Europe PMC Funders Group Dengue : a continuing global threat Europe PMC Funders Author Manuscripts.” *Nat Rev Microbiol*, 8(12 0), 7–16.

Hameed N., Shabut A. M., Ghosh M. K., and Hossain M. A. (2020). “Multi-class multi-level classification algorithm for skin lesions classification using machine learning techniques.” *Expert Syst. Appl.*, 141, 112961.

Hamilton P. S., and Tompkins W. J. (1986). “Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database.” *IEEE Trans. Biomed. Eng.*, BME-33(12), 1157–1165.

Han C., and Shi L. (2020). “ML–ResNet: A novel network to detect and locate myocardial infarction using 12 leads ECG.” *Comput. Methods Programs Biomed.*, 185, 105138.

Hao P., Gao X., Li Z., Zhang J., Wu F., and Bai C. (2020). “Multi-branch fusion network for Myocardial infarction screening from 12-lead ECG images.” *Comput. Methods Programs Biomed.*, 184, 105286.

Haq A. U., Li J. P., Memon M. H., Nazir S., and Sun R. (2018). “A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms.” *Mob. Inf. Syst.*, 2018, 1–21.

Hartmann J., Huppertz J., Schamp C., and Heitmann M. (2019). “Comparing automated text classification methods.” *Int. J. Res. Mark.,* 36(1), 20–38.

Haverkamp H. T., Fosse S. O., and Schuster P. (2019). “Accuracy and usability of single-lead ECG from smartphones - A clinical study.” *Indian Pacing Electrophysiol. J.,* 19(4), 145–149.

He K., Zhang X., Ren S., and Sun J. (2015). “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.” *Proc. IEEE Int. Conf. Comput. Vis.,* 2015 Inter, 1026–1034.

Hmeidi I., Hawashin B., and El-qawasmeh E. (2008). “Performance of KNN and SVM classifiers on full word Arabic articles.” 22, 106–111.

Hong S., Zhou Y., Shang J., Xiao C., and Sun J. (2020). “Opportunities and challenges of deep learning methods for electrocardiogram data: A systematic review.” *Comput. Biol. Med.,* 122(April), 103801.

Hooda R., Sofat S., Kaur S., Mittal A., and Meriaudeau F. (2017). “Deep-learning: A potential method for tuberculosis detection using chest radiography.” *Proc. 2017 IEEE Int. Conf. Signal Image Process. Appl. ICSIPA 2017,* 497–502.

Ittyachen A. M., and Ramachandran R. (2015). “Study of acute febrile illness: A 10-year descriptive study and a proposed algorithm from a tertiary care referral hospital in rural Kerala in southern India.” *Trop. Doct.*, 45(2), 114–117.

Jacobé de Naurois C., Bourdin C., Stratulat A., Diaz E., and Vercher J. L. (2019). “Detection and prediction of driver drowsiness using artificial neural network models.” *Accid. Anal. Prev.,* 126(October 2017), 95–104.

Jahmunah V., Oh S. L., Wei J. K. E., Ciaccio E. J., Chua K., San T. R., and Acharya U. R. (2019). “Computer-aided diagnosis of congestive heart failure using ECG signals – A review.” *Phys. Medica,* 62(May), 95–104.

Jambukia S. H., Dabhi V. K., and Prajapati H. B. (2015). “Classification of ECG signals using machine learning techniques: A survey.” *Conf. Proceeding - 2015 Int. Conf. Adv.*

Comput. Eng. Appl. ICACEA 2015, (December), 714–721.

Jayaraman S., Swamy P., Damodaran V., and Venkatesh N. (2012). “A Novel Technique for ECG Morphology Interpretation and Arrhythmia Detection Based on Time Series Signal Extracted from Scanned ECG Record.” *Adv. Electrocardiograms - Methods Anal.*, InTech.

Jian Jia-zheng., Ger Tzong-rong Lai., Han-hua Ku, Chi-ming. (2021). “Detection of Myocardial Infarction Using ECG and Multi-Scale Feature Concatenate.” *Sensors*, 21(5), 1906, 1–17.

Jiang W., and Kong S. G. (2007). “Block-based neural networks for personalized ECG signal classification.” *IEEE Trans. Neural Networks*, 18(6), 1750–1761.

Johnson M. M., and Odell J. A. (n.d.). “Nontuberculous mycobacterial pulmonary infections.” (4).

Joshi R., and Kalantri S. (2014). “Acute Undifferentiated Fever: Management Algorithm. Update on Tropical Fever.” *API Med. Updat.*, 1–14.

Jovic A., Brkic K., and Krstacic G. (2019). “Detection of congestive heart failure from short-term heart rate variability segments using hybrid feature selection approach.” *Biomed. Signal Process. Control*, 53, 101583.

Jufriadif Naama, Catur Suharintob S. (2017). “Digitalisasi Grafik Elektrokardiogram dengan Teknik Pixel Indexing.” Universitas Putra Indonesia: Sistem Informasi dan Teknologi Informasi.

kaisJameel S., and Manza R. (2012). “Color image segmentation using wavelet.” *Int. J. Appl. Inf. Syst.*, 1(6), 1–4.

Kanani P., and Padole M. (2018). *Recognizing real time ecg anomalies using arduino, ad8232 and java*. *Commun. Comput. Inf. Sci.*, Springer Singapore.

Kaveh A., and Chung W. (2013). “Temporal and spectral features of single lead ECG for human identification.” *2013 IEEE Work. Biometric Meas. Syst. Secur. Med. Appl.*

BioMS 2013 - Proc., 17–21.

Khatibi T., and Rabinezhadsadatmahaleh N. (2019). “Proposing feature engineering method based on deep learning and K-NNs for ECG beat classification and arrhythmia detection.” *Australas. Phys. Eng. Sci. Med.*, 43(1), 49–68.

Khleaf H. K., Gazali K. H. Bin, Abdalla A. N., and Na, M. (2015). “ECG-Trace-Digitization-Using-Image-Processing-Techniques.doc.” *Int. J. Sci. Eng. Res.*, 6(4), 1113–1116.

Kruger G. H., Latchamsetty R., Langhals N. B., Yokokawa M., Chugh A., Morady F., Oral H., and Berenfeld O. (2019). “Bimodal classification algorithm for atrial fibrillation detection from m-health ECG recordings.” *Comput. Biol. Med.*, 104(July), 310–318.

Kumar M., Pachori R. B., and Acharya U. R. (2017). “Characterization of coronary artery disease using flexible analytic wavelet transform applied on ECG signals.” *Biomed. Signal Process. Control*, 31, 301–308.

Kurt I., Ture M., and Kurum A. T. (2008). “Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease.” *Expert Syst. Appl.*, 34(1), 366–374.

Kusuma S., and J D. U. (2018). “Machine Learning and Deep Learning Methods in Heart Disease (HD) Research.” 119(18), 1483–1496.

Lakhani P., and Sundaram B. (2017). “Deep Learning at Chest Radiography : Automated Classification of Pulmonary Tuberculosis by Using Convolutional.” 000(0).

Laureano-rosario A. E., Duncan A. P., Mendez-lazaro P. A., Garcia-rejon J. E., Id S. G., Farfan-ale J., Savic D. A., and Muller-karger F. E. (n.d.). “Application of Artificial Neural Networks for Dengue Fever Outbreak Predictions in the Northwest Coast of Yucatan , Mexico and San Juan , Puerto Rico.” 1–16.

Lee H. G., Noh K. Y., and Ryu K. H. (2007). “Mining biosignal data: Coronary artery disease diagnosis using linear and nonlinear features of HRV.” *Lect. Notes Comput.*

Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 4819 LNAI, 218–228.

Lee I., Hsieh C., Lee C., and Liu J. (2017). “Diabetic patients suffering dengue are at risk for development of dengue shock syndrome/severe dengue: emphasizing the impacts of co-existing comorbidity(ies) and glycemic control on dengue severity.” *J. Microbiol. Immunol. Infect.,* (2018).

Lee K. Y., Chung N., and Hwang S. (2016). “Application of an artificial neural network (ANN) model for predicting mosquito abundances in urban areas.” *Ecol. Inform.,* 36, 172–180.

Lehtinen R., Holst H., Turjanmaa V., Edenbrandt L., Pahlm O., and Malmivuo J. (1998). “Artificial neural network for the exercise electrocardiographic detection of coronary artery disease.” *Proc. Int. Conf. Bioelectromagn.,* (February), 57–58.

Li H., and Boulanger P. (2020). “A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG).” *Sensors,* 20(5), 1461.

Li H., Yuan D., Ma X., Cui D., and Cao L. (2017). “Genetic algorithm for the optimization of features and neural networks in ECG signals classification.” *Sci. Rep.,* 7(January), 1–12.

Li S., Wang L., Li J., and Yao Y. (2021). “Image Classification Algorithm Based on Improved AlexNet.” *J. Phys. Conf. Ser.,* 1813(1).

Li X., Bian D., Yu J., Mao H., Li M., and Zhao D. (2019). “Using machine learning models to classify stroke risk level based on national screening data ∗.” *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS,* 2, 1386–1390.

Liang Y., Yin S., Tang Q., Zheng Z., Elgendi M., and Chen Z. (2020). “Deep Learning Algorithm Classifies Heartbeat Events Based on Electrocardiogram Signals.” *Front. Physiol.,* 11(October).

Loganathan P., Karthikeyan R., and Scholar R. (2021). *Residual Neural Network (ResNet) Based Plant Leaf Disease Detection and Classification*. *Turkish Online J.*

Qual. Inq.

Luger G. F. (2005). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. *Zyw. Czlowieka I Metab.*

Luukka P., and Lampinen J. (2010). “A Classification method based on principal component analysis and differential evolution algorithm applied for prediction diagnosis from clinical EMR heart data sets.” 263–283.

Lyman M. M., Grigg C., Kinsey C. B., Keckler M. S., Moulton-Meissner H., Cooper E., Soe M. M., Noble-Wang J., Longenberger A., Walker S. R., Miller J. R., Perz J. F., and Perkins K. M. (2017). “Invasive Nontuberculous Mycobacterial Infections among Cardiothoracic Surgical Patients Exposed to Heater–Cooler Devices1.” *Emerg. Infect. Dis.*, 23(5), 796–805.

Machuca I., Vidal E., la Torre-Cisneros J. de, and Rivero-Román A. (2018). “Tuberculosis in immunosuppressed patients.” *Enfermedades Infecc. y Microbiol. Clin. (English ed.),* 36(6), 366–374.

Maclin P. S., and Dempsey J. (1994). “How to improve a neural network for early detection of hepatic cancer.” *Cancer Lett.*, 77(2–3), 95–101.

Mahamdi Y., Boubakeur A., Mekhaldi A., and Benmahamed Y. (2022). “Power Transformer Fault Prediction using Naive Bayes and Decision tree based on Dissolved Gas Analysis.” *ENP Eng. Sci. J.*, 2(1), 1–5.

Mahmud S., Guo Z., Quadir F., Liu J., and Cheng J. (2022). “Multi-head attention-based U-Nets for predicting protein domain boundaries using 1D sequence features and 2D distance maps.” *BMC Bioinformatics*, 23(1), 1–15.

Maniruzzaman M., Rahman M. J., Ahammed B., and Abedin M. M. (2020). “Classification and prediction of diabetes disease using machine learning paradigm.” *Heal. Inf. Sci. Syst.*, 8(1), 1–14.

Marker B., Hoffman J. I. E., Practitioners B., and Edition S. (2019). “Create account Related terms : View all Topics.”

Marras T. K., Chedore P., Ying A. M., and Jamieson F. (2007). “Isolation prevalence of pulmonary non-tuberculous mycobacteria in Ontario, 1997-2003.” *Thorax*, 62(8), 661–666.

Marston H. R., Hadley R., Banks D., and Duro M. D. C. M. (2019). “Mobile Self-Monitoring ECG Devices to Diagnose Arrhythmia that Coincide with Palpitations: A Scoping Review.” *Healthcare*, 7(3), 96.

Martis R. J., Acharya U. R., and Adeli H. (2014). “Current methods in electrocardiogram characterization.” *Comput. Biol. Med.*, 48(1), 133–149.

Mathews S. M., Kambhamettu C., and Barner K. E. (2018). “A novel application of deep learning for single-lead ECG classification.” *Comput. Biol. Med.*, 99(May 2017), 53–62.

McManus K. H. C. and D D. (2019). “Detection of atrial fibrillation using a wrist-worn device.” *Physiol. Meas.*, 40(2), 657–658.

Mehrotra R., Ansari M. A., Agrawal R., and Anand R. S. (2020). “A Transfer Learning approach for AI-based classification of brain tumors.” *Mach. Learn. with Appl.*, 2(October), 100003.

Miller T. D., Askew J. W., and Anavekar N. S. (n.d.). “Noninvasive Stress Testing for Coronary Artery Disease.” *Heart Fail. Clin.*, 12(1), 65–82.

Mirsaeidi M., Machado R. F., Garcia J. G. N., and Schraufnagel D. E. (2014). “Nontuberculous Mycobacterial Disease Mortality in the United States, 1999–2010: A Population-Based Comparative Study.” *PLoS One*, (J. L. Herrmann, ed.), 9(3), e91879.

Mirza A. H., Nurmaini S., and Partan R. U. (2022). “Automatic Classification of 15 Leads ECG Signal of Myorcadial Infarction Using Convolutional Neural Network.” 1–12.

Mishra S., Khatwani G., Patil R., Sapariya D., Shah V., Parmar D., Dinesh S., Daphal P., and Mehendale N. (2021). “ECG Paper Record Digitization and Diagnosis Using Deep Learning.” *J. Med. Biol. Eng.*, 41(4), 422–432.

Moody G. B. (2004). “Spontaneous termination of atrial fibrillation: A challenge from PhysioNet and Computers in Cardiology 2004.” *Comput. Cardiol.*, 31, 101–104.

Moody G. B., and Mark R. G. (2001). “The impact of the MIT-BIH arrhythmia database.” *IEEE Eng. Med. Biol. Mag.*, 20(3), 45–50.

Mourad O., Palda V., and Detsky A. S. (2003). “A Comprehensive Evidence-Based Approach to Fever of Unknown Origin.” *Arch. Intern. Med.*, 163(5), 545.

Mukhopadhyay P., and Chaudhuri B. B. (2014). “Author ’ s Accepted Manuscript A Survey of Hough Transform.” *Pattern Recognit.*

Murat F., Yildirim O., Talo M., Baloglu U. B., Demir Y., and Acharya U. R. (2020). “Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review.” *Comput. Biol. Med.*, 120(March), 103726.

Nazareth T., Seixas G., and Sousa C. A. (2016). “Climate Change and Mosquito-Borne Diseases.” *Clim. Chang. Manag.*, 109(March), 187–199.

Nejedly P., Ivora A., Viscor I., Koscova Z., Smisek R., Jurak P., and Plesinger F. (2022). “Classification of ECG using ensemble of residual CNNs with or without attention mechanism.” *Physiol. Meas.*, 43(4).

Ogoina, D. (2011). “Fever , fever patterns and diseases called ‘ fever ’ — A review.” *J. Infect. Public Health*, 4(3), 108–124.

Oh S. L., Adam M., Tan J. H., Hagiwara Y., Sudarshan V. K., Koh J. W., Chua K. C., Chua, K. P., Tan R. S., and Ng Y. K. (2017). “Automated identification of coronary artery disease from short-term 12 lead electrocardiogram signals by using wavelet packet decomposition and common spatial pattern techniques.” *J. Mech. Med. Biol.*, 17(07), 1740007.

Oh S. L., Ng Eddie Y. K., Tan R. S., and Acharya U. R. (2018). “Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats.” *Comput. Biol. Med.*, 102(June), 278–287.

Oquab M., Bottou L., Laptev I., and Sivic J. (2015). "Is object localization for free? - Weakly-supervised learning with convolutional neural networks." *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 07-12-June(iii), 685–694.

Pan J., and Tompkins W. J. (1985). "A Real-Time QRS Detection Algorithm." *IEEE Trans. Biomed. Eng.*, BME-32(3), 230–236.

Pandey A. K., and Gupta M. (2018). "Software fault classification using extreme learning machine: a cognitive approach." *Evol. Intell.*, 0(0), 0.

Patil R., and Karandikar R. (2018). "Image digitization of discontinuous and degraded electrocardiogram paper records using an entropy-based bit plane slicing algorithm." *J. Electrocardiol.*, 51(4), 707–713.

Pe A. B., Rizo D., Izquierdo A., Sa L., Pacheco B., Hirayama K., and Guzma M. G. (2011). "International Journal of Infectious Diseases Long-term persistence of clinical symptoms in dengue-infected persons and its association with immunological disorders." 15, 38–43.

Pławiak P. (2018). "Novel genetic ensembles of classifiers applied to myocardium dysfunction recognition based on ECG signals." *Swarm Evol. Comput.*, 39(October 2017), 192–208.

Poddar M. G., Kumar V., and Sharma Y. P. (2015). "Automated diagnosis of coronary artery diseased patients by heart rate variability analysis using linear and non-linear methods." *J. Med. Eng. Technol.*, 39(6), 331–341.

Podgorelec V. (2012). "Analyzing EEG signals with machine learning for diagnosing alzheimer's disease." *Elektron. ir Elektrotechnika*, 18(8), 61–64.

Pratt W. K. (2003). "Morphological Image Processing." *Digit. Image Process.*, 401–441.

Price D., Cardiologist C., Mary S., and Wight I. (2010). "How to read an electrocardiogram (ECG). Part 1: Basic principles of the ECG. The normal ECG." *South Sudan Med. J.*, 3(2), 26-31–31.

Qibin Zhao and Liqing Zhang. (2005). “ECG Feature Extraction and Classification Using Wavelet Transform and Support Vector Machines.” *2005 Int. Conf. Neural Networks Brain*, IEEE, 1089–1092.

Rahhal M. M. Al, Bazi Y., AlHichri H., Alajlan N., Melgani F., and Yager R. R. (2016). “Deep learning approach for active classification of electrocardiogram signals.” *Inf. Sci. (Ny).*, 345, 340–354.

Rai A., and Mitra M. (2021). “Lamb wave based damage detection in metallic plates using multi-headed 1-dimensional convolutional neural network.” *Smart Mater. Struct.*, 30(3), 035010.

Raid A., Khedr W., El-dosuky M., and Aoud M. (2014). “Image Restoration Based on Morphological Operations.” *Int. J. Comput. Sci. Eng. Inf. Technol.*, 4(3), 9–21.

Rajeswari K., Vaithiyanathan V., and Neelakantan T. R. (2012). “Feature Selection in Ischemic Heart Disease identification using feed forward neural networks.” *Procedia Eng.*, 41(Iris), 1818–1823.

Rani A., Kumar S., Micheloni C., and Foresti G. L. (2013). “Incorporating linear discriminant analysis in neural tree for multidimensional splitting.” *Appl. Soft Comput. J.*, 13(10), 4219–4228.

Ravichandran L., Harless C., and Member S. (2013). “Novel Tool for Complete Digitization of Paper Electrocardiography Data.” *IEEE J. Transl. Eng. Heal. Med.*, 1(March), 1–7.

Ribeiro A. H., Ribeiro M. H., Paixão G. M. M., Oliveira D. M., Gomes P. R., Canazart J. A., Ferreira M. P. S., Andersson C. R., Macfarlane P. W., Wagner M., Schön T. B., and Ribeiro A. L. P. (2020). “Automatic diagnosis of the 12-lead ECG using a deep neural network.” *Nat. Commun.*, 11(1), 1–9.

Roth A. R., and Basello G. M. (2003). “Approach to the adult patient with fever of unknown origin.” *Am. Fam. Physician*, 68(11), 2223–8.

Rubin J., Parvaneh S., Rahman A., Conroy B., and Babaeizadeh S. (2018). “Densely connected convolutional networks for detection of atrial fibrillation from short single-lead ECG recordings.” *J. Electrocardiol.*, 51(6), S18–S21.

Russo C. (2002). “[Role of hemostatic markers in the pathogenesis of coronary diseases].” *Recenti Prog. Med.*, 93(2), 75–7.

Saleh M. R. (2011). “Expert Systems with Applications Experiments with SVM to classify opinions in different domains.” *Expert Syst. Appl.*, 38(12), 14799–14804.

Salim N. A. M., Wah Y. B., Reeves C., Smith M., Yaacob W. F., Mudin R. N., Dapari R., Sapri N. F., and Haque U. (2021). “Prediction of dengue outbreak in Selangor Malaysia using machine learning techniques.” *Sci. Rep.*, 11(1), 1–9.

Samuel A. L. (1969). “Some studies in machine learning using the game of checkers. II-Recent progress.” *Annu. Rev. Autom. Program.*, 6(PART 1), 1–36.

Sandler M., Howard A., Zhu M., Zhmoginov A., and Chen L. C. (2018). “MobileNetV2: Inverted Residuals and Linear Bottlenecks.” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 4510–4520.

Sankar A. B., Selvi J. A. V., Kumar D., and Lakshmi K. S. (2013). “Effective enhancement of classification of respiratory states using feed forward back propagation neural networks.” *Sadhana - Acad. Proc. Eng. Sci.*, 38(3), 377–395.

Sannino G., and Pietro G. De. (2018). “A deep learning approach for ECG-based heartbeat classification for arrhythmia detection.” *Futur. Gener. Comput. Syst.*, 86, 446–455.

Sarma D., Hossain S., Mittra T., Bhuiya M. A. M., Saha I., and Chakma R. (2020). “Dengue Prediction using Machine Learning Algorithms.” *IEEE Reg. 10 Humanit. Technol. Conf. R10-HTC*, 2020-Decem.

Sassi R., Bond R. R., Cairns A., Finlay D. D., Guldenring D., Libretti G., Isola L., Vaglio M., Poeta R., Campana M., Cuccia C., and Badilini F. (2017). “PDF–ECG in clinical practice: A model for long–term preservation of digital 12–lead ECG data.” *J. Electrocardiol.*, 50(6), 776–780.

Schmidhuber J. (2015). “Deep Learning in neural networks: An overview.” *Neural Networks*, 61, 85–117.

Schober P., Boer C., and Schwarte L. A. (2018). “Correlation Coefficients: Appropriate Use and Interpretation.” 126(5), 1763–1768.

Scott T. W. (2009). “Dengue.” *Encycl. Insects*, Elsevier, 257–259.

Sedghamiz H. (2014). “Matlab Implementation of Pan Tompkins ECG QRS.” 1–3.

Seera M., and Lim C. P. (2014). “A hybrid intelligent system for medical data classification.” *Expert Syst. Appl.*, 41(5), 2239–2249.

Selvaretnam A. P., Sahu P. S., Sahu M., and Ambu S. (2016). “Asian Paci fi c Journal of Tropical Biomedicine A review of concurrent infections of malaria and dengue in Asia.” *Asian Pac. J. Trop. Biomed.*, 6(7), 633–638.

Serhani M. A., Navaz A. N., Ashwal H. Al, and Qirim N. Al. (2020). “Ecg-based arrhythmia classification & clinical suggestions: An incremental approach of hyperparameter tuning.” *ACM Int. Conf. Proceeding Ser.*, (January 2021), 13–19.

Sharma J., Kumar V., Ayub S., and Saini J. (2012). “Uniform Sampling of ECG Waveform of MIT-BIH Normal Sinus Rhythm Database at Desired Intervals.” *Int. J. Comput. Appl.*, 50(15), 6–9.

Sharma S., and Mehra R. (2020). “Conventional Machine Learning and Deep Learning Approach for Multi-Classification of Breast Cancer Histopathology Images—a Comparative Insight.” *J. Digit. Imaging*, 33(3), 632–654.

Shaukat Dar K., and Ulya Azmeen S. M. (2015). “Dengue Fever Prediction: A Data Mining Problem.” *J. Data Mining Genomics Proteomics*, 06(03).

Shaw A. J., Ingham S. A., Atkinson G., and Folland J. P. (2015). “The correlation between running economy and maximal oxygen uptake: Cross-sectional and longitudinal relationships in highly trained distance runners.” *PLoS One*, 10(4), 37–38.

Shrivastava P. R. K., Panbude S., and Narayanan G. (2014). “Digitization of ECG Paper Records using MATLAB.” *Int. J. Innov. Technol. Explor. Eng.*, (6), 2278–3075.

Siegel N., Horvitz Z., Levin R., Divvala S., and Farhadi A. (2016). “Figureseer: Parsing result-figures in research papers.” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 9911 LNCS, 664–680.

Singla N., Arora S., Goel P., Chander J., and Huria A. (2015). “Dengue in pregnancy : an under-reported illness , with special reference to other existing co-infections.” *Asian Pac. J. Trop. Med.*, 8(3), 206–208.

Srivastava R. K., Greff K., and Schmidhuber J. (2015). “Highway Networks.”

Steinmetzer T., Bönninger I., Reckhardt M., Reinhardt F., Erk D., and Travieso C. M. (2019). “Comparison of algorithms and classifiers for stride detection using wearables.” *Neural Comput. Appl.*

Stylianides N., and Kontou E. (2020). “BAYES THEOREM AND ITS RECENT APPLICATIONS Nicholas Stylianides What is Bayes Theorem ? Extended Form of Bayes ’ Formula.” *MA3517 Math. Res. J.*, 1(March), 1–7.

Subbiah S., and Patro R. (2015). “Feature Extraction and Classification for ECG Signal Processing based on Artificial Neural Network and Machine Learning Approach.” *Int. Conf. Inter Discip. Res. Eng. Technol.*, (February), 50–57.

Suresh V., Janik P., Rezmer J., and Leonowicz Z. (2020). “Forecasting solar PV output using convolutional neural networks with a sliding window algorithm.” *Energies*, 13(3).

Susilawati T. N., and McBride W. J. H. (2014). “Acute undifferentiated fever in Asia: a review of the literature.” *Southeast Asian J. Trop. Med. Public Health*, 45(3), 719–26.

Swapna G., Soman K. P., and Vinayakumar R. (2018). “Automated detection of cardiac arrhythmia using deep learning techniques.” *Procedia Comput. Sci.*, 132, 1192–1201.

Szegedy C., Wei Liu, Yangqing Jia, Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., and Rabinovich A. (2015). “Going deeper with convolutions.” *2015 IEEE Conf. Comput. Vis. Pattern Recognit.*, IEEE, 1–9.

Tabassum T., and Ahmad M. (2020). “Numerical Data Extraction from ECG Paper Recording Using Image Processing Technique.” *2020 11th Int. Conf. Electr. Comput. Eng.*, IEEE, 355–358.

Tan J. H., Hagiwara Y., Pang W., Lim I., Oh S. L., Adam M., Tan R. S., Chen M., and Acharya U. R. (2018). “Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals.” *Comput. Biol. Med.*, 94(December 2017), 19–26.

Tan M., and Le Q. V. (2019). “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.” *36th Int. Conf. Mach. Learn. ICML 2019*, 2019-June, 10691–10700.

The Theano Development Team, Al-Rfou R., Alain G., Almahairi A., Angermueller C., Bahdanau D., Ballas N., Bastien F., Bayer J., Belikov A., Belopolsky A., Bengio Y., Bergeron A., Bergstra and Zhang Y. (2016). “Theano: A Python framework for fast computation of mathematical expressions.” 1–19.

Tripathy A., Anand A., and Rath S. K. (2017). “Document-level sentiment classification using hybrid machine learning approach.” *Knowl. Inf. Syst.*, 53(3), 805–831.

Trstenjak B., Mikac S., and Donko D. (2014). “KNN with TF-IDF Based Framework for Text Categorization.” *Procedia Eng.*, 69, 1356–1364.

U.R. Acharya, sree, vinitha, swapna, et. al. (2014). “Linear and nonlinear analysis of normal and CAD-affected heart rate signals.” *Comput. Methods Programs Biomed.*, 113(1), 55–68.

Ullah A., Anwar S. M., Bilal M., and Mehmood R. M. (2020). “Classification of arrhythmia by using deep learning with 2-D ECG spectral image representation.”

Remote Sens., 12(10).

Vasudeva S. T., Rao S. S., Panambur N. K., Mahabala C., Dakappa P. H., and Prasad K. (2020). “Diagnostic classification of undifferentiated fevers using artificial neural network.” *AIP Conf. Proc.*, 070001.

Vijayashree J., and Sultana H. P. (2018). “A Machine Learning Framework for Feature Selection in Heart Disease Classification Using Improved Particle Swarm Optimization with Support Vector Machine Classifier.” *Program. Comput. Softw.*, 44(6), 388–397.

Vilcant V., and Zeltser R. (2019). “Contraindications.” 4–9.

Viliane Vilcant, and Roman Zeltser. (2022). “Treadmill Stress Testing.” *StatPearls*.

Virgin G. A., and Baskar V. V. (2018). “Conversion of ECG Graph into Digital Format.” *Int. J. Pure Appl. Math.*, 118(17), 469–484.

Wang J., Wang P., and Wang S. (2020). “Automated detection of atrial fibrillation in ECG signals based on wavelet packet transform and correlation function of random process.” *Biomed. Signal Process. Control*, 55, 101662.

Wang, Yan, Z., Yang Y., Chen J., Yang T., Kuo I., Abu P. R., Huang P, Chen C., and Chen S. (2021). “A Classification and Prediction Hybrid Model Construction with the IQPSO-SVM Algorithm for Atrial Fibrillation Arrhythmia.” *Sensors*, 21(15), 5222.

Wu L., Xie X., and Wang Y. (2021). “ECG enhancement and r-peak detection based on window variability.” *Healthc.*, 9(2).

Xing Z., Ding W., Zhang S., Zhong L., Wang L., Wang J., Wang K., Xie Y., Zhao X., Li N., and Ye Z. (2020). “Machine Learning-Based Differentiation of Nontuberculous Mycobacteria Lung Disease and Pulmonary Tuberculosis Using CT Images.” 2020.

Xu S. S., Mak M.W., and Cheung C. (2019). “Towards End-to-End ECG Classification With Raw Signal Extraction and Deep Neural Networks.” *IEEE J. Biomed. Heal. Informatics*, 23(4), 1574–1584.

Yildirim O., Baloglu U. B., Tan R., Ciaccio E. J., and Acharya U. R. (2019). “A new approach for arrhythmia classification using deep coded features and LSTM networks.” *Comput. Methods Programs Biomed.*, 176(May), 121–133.

Yildirim O., Tan R. S., and Acharya U. R. (2018). “An efficient compression of ECG signals using deep convolutional autoencoders.” *Cogn. Syst. Res.*, 52, 198–211.

Yochum M., Renaud C., and Jacquir S. (2016). “Automatic detection of P, QRS and T patterns in 12 leads ECG signal based on CWT.” *Biomed. Signal Process. Control*, 25, 46–52.

Yoo H., Han S., and Chung K. (2020). “A Frequency Pattern Mining Model Based on Deep Neural Network for Real-Time Classification of Heart Conditions.” *Healthcare*, 8(3), 234.

Young K., Chung N., and Hwang S. (2016). “Ecological Informatics Application of an arti fi cial neural network (ANN) model for predicting mosquito abundances in urban areas.” *Ecol. Inform.*, 36, 172–180.

Zhang L., Jiang L., and Li C. (2019a). “A discriminative model selection approach and its application to text classification.” *Neural Comput. Appl.*, 31(4), 1173–1187.

Zhang Q., Zhang M., Chen T., Sun Z., Ma Y., and Yu B. (2019b). “Recent advances in convolutional neural network acceleration.” *Neurocomputing*, 323, 37–51.

Zhu R., and Wang Y. (2012). “Application of improved median filter on image processing.” *J. Comput.*, 7(4), 838–841.

LIST OF PUBLICATIONS

1. Vasudeva, S.T.; Rao, S.S.; Panambur, N.K.; Shettigar, A.K.; Mahabala, C.; Kamath, P.; Gowdru Chandrashekarappa, M.P.; Linul, E. Development of a Convolutional Neural Network Model to Predict Coronary Artery Disease Based on Single-Lead and Twelve-Lead ECG Signals. Appl. Sci. 2022, 12, 7711. https://doi.org/10.3390/app12157711.
2. Shrivathsa, T. V., Kalyan, C., Rao, S. S., Navin Karanth, P., Chakrapani, M., & Kamath, P. (2022). Development and performance evaluation of a coronary artery disease prediction system with transfer learned model based on single lead and multi-lead ECG & TMT-ECG signals. International Journal of Health Sciences, 6(S4), 8865–8890. https://doi.org/10.53730/ijhs.v6nS4.10681.
3. Shrivathsa Thokur Vasudeva, Shrikantha Sasihithlu Rao, Navin Karanth Panambur, Chakrapani Mahabala, Pradeepa Hoskere Dakappa, and Keerthana Prasad , "Diagnostic classification of undifferentiated fevers using artificial neural network", AIP Conference Proceedings 2236, 070001 (2020) https://doi.org/10.1063/5.0007749.
4. Shrivathsa, T. V., Vikash., Rao, S. S., Navin Karanth, P., Chakrapani, M., & Kamath, P. “Automatic Detection and Quantification of Data and Feature Parameters of Electrocardiogram and Tread Mill Test Electrocardiogram (TMT-ECG)”. (Under Review)
5. Shrivathsa Thokur Vasudeva, Shrikantha Sasihithlu Rao, Navin Karanth Panambur, Chakrapani Mahabala, Pradeepa Hoskere Dakappa, and Keerthana Prasad, “A predictive intelligent system development for disease classification in diagnostic applications”. (Under Review)

APPENDIX – I

ALGORITHMIC CODE FOR MULTI-DIMENSIONAL ANALYSIS

(1) Algorithm code for fever analysis

```
import pandas
import numpy
import scipy
import sys
import sklearn
import matplotlib.pyplot as plt
# Load libraries
from pandas.plotting import scatter_matrix
from sklearn import model_selection
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import log_loss
data = pandas.read_excel (r"C:\Users\SHRI\Videos\PhD\temparature\Temp clasifi python code\Untitled Folder\data12.xlsx", sheet_name=0)
dta1= [data,data]
dta12 = pandas.concat((dta1), axis = 1)
dta12.to_csv ('newdata312.csv',  index=False)
df12= pandas.read_csv ("newdata312.csv")
```

```
trans12= df12.T
trans12.to_csv ('newtrans12.csv',  index=False)
df=pandas.read_csv("newtrans12.csv")
dff=df.values
X = dff[:,:-1]
array12 = trans12.values
X2 = array12[:,:-1]
Y2 = array12[:,-1]
# Split-out validation dataset
validation_size = 0.20
seed = 10
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X2,
Y2, test_size=validation_size, random_state=seed)
# Test options and evaluation metric
seed = 10
scoring = 'accuracy'
# Spot Check Algorithms
models = []
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
# evaluate each model in turn
results = []
names = []
for name, model in models:
  kfold = model_selection.KFold(n_splits=10, random_state=seed)
  cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, sc
oring=scoring)
  results.append(cv_results)
  names.append(name)
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  print(msg)
# Compare Algorithms
fig = plt.figure()
fig.suptitle('Algorithm Comparison')
ax = fig.add_subplot(111)
plt.boxplot(results)
ax.set_xticklabels(names)
plt.show()
# Make predictions on validation dataset
cart = DecisionTreeClassifier()
decesiontree = cart.fit(X_train, Y_train)
predictions = cart.predict(X_validation)
predictions
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
y_pred = cart.fit(X_train, Y_train).predict(X_validation)
pred = decesiontree.predict(X_validation)
class_names =  Y_validation
cm=confusion_matrix(class_names,pred)
from sklearn import preprocessing, model_selection, tree
model = tree.DecisionTreeClassifier()
model.fit(X_train, Y_train)
accuracy = model.score(X_validation, Y_validation)
print("Accuracy = %g" % (accuracy))
print("Test Error = %g" % (1.0 - accuracy))
print("Test score: {0:.2f} %".format(100 * accuracy))
```

(2) ALGORITHM CODE FOR ECG ANALYSIS

1. Data collection

(a) Folder renaming

```
import os
import pandas as pd
#i=136
#C:\Users\SHRI\Videos\Final DATA\DATA F\CAD

os.chdir(r'C:\Users\SHRI\Videos\Final DATA\DATA F\NoCAD\103N\EXCER')
for file in os.listdir():
    src=file
    if src=='aVR.xlsx':
        dst="4r.xlsx"
        os.rename(src,dst)
    if src=='aVL.xlsx':
        dst="5l.xlsx"
        os.rename(src,dst)
    if src=='aVF.xlsx':
        dst="6f.xlsx"
        os.rename(src,dst)
os.chdir(r'C:\Users\SHRI\Videos\Final DATA\DATA F\NoCAD\103N\NO_EXCER')
for file in os.listdir():
    src=file
    if src=='aVR.xlsx':
        dst="4r.xlsx"
        os.rename(src,dst)
    if src=='aVL.xlsx':
        dst="5l.xlsx"
        os.rename(src,dst)
    if src=='aVF.xlsx':
        dst="6f.xlsx"
        os.rename(src,dst)
```

(b) Data collection

```
#Combining multiple excel files in to single excel file containg in single folder
import matplotlib as plt
import numpy as np
import openpyxl
from openpyxl import load_workbook, Workbook
import os
for i in range(1, 158):
    print('ECG {}:'.format(i))
    data1 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\1st.xlsx'.format(i))
    data2 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\2nd.xlsx'.format(i))
    data3 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\3rd.xlsx'.format(i))
    data4 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\4r.xlsx'.format(i))
    data5 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\5l.xlsx'.format(i))
    data6 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\6f.xlsx'.format(i))
    data7 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V1.xlsx'.format(i))
    data8 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V2.xlsx'.format(i))
    data9 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V3.xlsx'.format(i))
    data10 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V4.xlsx'.format(i))
    data11 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V5.xlsx'.format(i))
    data12 = pd.read_excel (r'C:\Users\SHRI\Videos\Final DATA\DATA F\CAD\{}C\EXCER\V6.xlsx'.format(i))
```

```
ecgxt = data1.iloc [:,0]
ecgya = data1.iloc [:,1]
PR_int= ecgya.iloc[98:197]
QRS_comp= ecgya.iloc[197:252]
ST_seg= ecgya.iloc[252:319]
T_seg= ecgya.iloc[319:418]
QT_seg= ecgya.iloc[197:418]
P= max(PR_int)   # P-value
R= max(QRS_comp)  # R-value
S= min(QRS_comp)  # S-value
T= max(T_seg) # T-value
s = pd.Series([P, R, S, T])
frames= [ecgya, QRS_comp,ST_seg, s]
result1 = pd.concat(frames)
#result.to_excel("output.xlsx", sheet_name='ecg_modify_data', index=False)
.
.
.       #Similarlly continued for twelve leads
.
ecgxt = data12.iloc [:,0]
ecgya = data12.iloc [:,1]
PR_int= ecgya.iloc[98:197]
QRS_comp= ecgya.iloc[197:252]
ST_seg= ecgya.iloc[252:319]
T_seg= ecgya.iloc[319:418]
QT_seg= ecgya.iloc[197:418]
P= max(PR_int)   # P-value
R= max(QRS_comp)  # R-value
S= min(QRS_comp)  # S-value
T= max(T_seg) # T-value
s = pd.Series([P, R, S, T])
```

```
    frames= [ecgya, QRS_comp,ST_seg, s]
    result12 = pd.concat(frames)
    combineN= [result1, result2, result3, result4, result5, result6, result7, result8, result9, result10, result11, result12]
    result=  pd.concat(combineN)

    path = r'C:\Users\SHRI\Videos\Final DATA\DATA F\data_prepare\CAD_ecg_exwithfe\CAD_ecg_exwithfe.xlsx'
    book = load_workbook(path)
    writer = pd.ExcelWriter(path, engine = 'openpyxl')
    writer.book = book

    result.to_excel(writer, sheet_name = '{}'.format(i), index=False)
    #result.to_excel(writer, sheet_name = 'c2')
    writer.save()
    writer.close()
```

Or Second code

```
# Give an excel filename and worksheet name
output=r'C:\Users\SHRI\Videos\PhD\Ecg\ECG python\Untitled Folder\data_prepare\output.xlsx'
worksheet = 'Sheet'
wb = Workbook()

# If file not present at location, then create one
if os.path.isfile(output):
    print('File Present')
else:
    print('Creatted New file')
    ws = wb.create_sheet(worksheet)
    wb.save(output)
```

```
# Loop for all 350 files
for i in range(1, 13):
    print('File {}:'.format(i))
    data24 = pd.read_excel(r'C:\Users\SHRI\Videos\PhD\Ecg\ECG python\KMC_ecg_data\CAD_DATA\1C_C\EXCER\{}.xlsx'.format(i))

    ecgxt = data24.iloc [:,0]
    ecgya = data24.iloc [:,1]
    PR_int= ecgya.iloc[98:197]
    QRS_comp= ecgya.iloc[197:252]
    ST_seg= ecgya.iloc[252:319]
    T_seg= ecgya.iloc[319:418]
    QT_seg= ecgya.iloc[197:418]
    P= max(PR_int)   # P-value
    R= max(QRS_comp)  # R-value
    S= min(QRS_comp)  # S-value
    T= max(T_seg) # T-value
    s = pd.Series([P, R, S, T])
    frames= [ecgya, QRS_comp,ST_seg, s]
    result = pd.DataFrame(pd.concat(frames))
    ws = wb.active
    result_list = result.to_numpy()
    print('Total rows = ', len(result_list))
    for row in result_list.tolist():
        ws.append(row)
    wb.save(output)
```

(c) Combine all the sheets of excel into single sheet

```
import pandas as pd
import numpy as np
import os, collections, csv
from os.path  import basename
```

```
#df = []
dat = r'C:\Users\SHRI\Videos\Final DATA\DATA F\data_prepare\NOCAD_ecg_ex\
NOCAD_ecg_ex.xlsx'
#C:\Users\SHRI\Videos\Final DATA\DATA F\data_prepare\CAD_ecg_exwithfe\CA
D_ecg_exwithfe
list_sheets = []
#numberOfSheets = 125 #Modify this (n+1).
for i in range(1,119):
#for i in range(1,158):
    data = pd.read_excel(dat, sheet_name = str(i), header=None)
    list_sheets.append(data)

#remember python is very strict on how you arrange stuff so be aware of this
final = r'C:\Users\SHRI\Videos\Final DATA\DATA F\data_prepare\NOCAD_ecg_ex
\To_NOCAD_ecg_ex.xlsx'
out_data= pd.concat(list_sheets,axis = 1)
out_data.to_excel(final)
```

2. Image preprocessing

(a) Background removal

```
# white background
from scipy import ndimage, misc
import numpy as np
import os
import cv2
from PIL import Image
import PIL.ImageOps
import matplotlib.pyplot as plt
import skimage
from skimage import io
from google.colab.patches import cv2_imshow
```

```
path = "/content/drive/MyDrive/1C.jpg"

# iterate through the names of contents of the folder

# create the full input path and read the file
input_path = path
img1= cv2.imread(input_path)

print(BLACK_MIN)
print(BLACK_MAX)

BLACK_MIN = np.array([0,0,0],np.uint8)
BLACK_MAX = np.array([255,255,160],np.uint8)
hsv_img = cv2.cvtColor(img1,cv2.COLOR_BGR2HSV)
frame_threshed = cv2.inRange(hsv_img, BLACK_MIN, BLACK_MAX)
cv2.imwrite('threshed.jpg', frame_threshed)
image = Image.open('threshed.jpg')
inverted_image = PIL.ImageOps.invert(image)
inverted_image.save('invert.jpg')
img2=cv2.imread("invert.jpg")
kernel = np.ones((2,2), np.uint8)
dilate = cv2.morphologyEx(img2, cv2.MORPH_DILATE, kernel)
```

(b) Selection of Images

```
import os
import random
import shutil

files_list = []
for root, dirs, files in os.walk(r'C:\Users\SHRI\Videos\Final DATA\Shri ECG images\single_lead_bw_crop_final\Validation\CAD'):
```

```
##for root, dirs, files in os.walk(r'C:\Users\SHRI\Videos\Final DATA\Shri ECG images\single_lead_bw_crop_final\Validation\NOCAD'):

    for file in files:
        #all
        if file.endswith(".jpg"):
            files_list.append(os.path.join(root, file))


#print images
#lets me count and print the amount of jpeg,jpg,pmg
file_count = len(files_list)
print (file_count)

#print files_list
# Get the remaining files

selected_files = random.sample(files_list, 3015)  #assign to a list
##selected_files = random.sample(files_list, 2265)  #assign to a list

dest_path = r'C:\Users\SHRI\Videos\Final DATA\Shri ECG images\single_lead_bw_crop_final\Train\CAD'

for src_path in selected_files:
    shutil.move(src_path, os.path.join(dest_path, os.path.basename(src_path)))
```

3. Hyperparameter tuning

```
INPUT_SHAPE=(SIZE,SIZE,3)
NUM_CLASSES=1

from kerastuner import HyperModel
```

```
class CNNHyperModel(HyperModel):
    def __init__(self, input_shape, num_classes):
        self.input_shape = input_shape
        self.num_classes = num_classes

    def build(self, hp):
        model = keras.Sequential()
        #conv_1
        model.add(
            Conv2D(
                filters=hp.Choice(
                    'num_filters_conv_1',
                    values=[16,32, 64,128],
                    default=64,
                ),
                activation='relu',
                kernel_size=hp.Int('conv1_kernel_size', min_value=3, max_value=9),padding='same'
            )
        )
        model.add(MaxPooling2D(pool_size=hp.Int('pool1_pool_size', min_value=2, max_value=7),padding='same'))

        model.add(
            Conv2D(
                filters=hp.Choice(
                    'num_filters_conv_2',
                    values=[16,32, 64,128],
                    default=32,
                ),
                activation='relu',
                kernel_size=hp.Int('conv2_kernel_size', min_value=3, max_value=9),padding='same'
            )
        )
# Similarly add number of layers need to tune.
        model.add(
            Dropout(
                rate=hp.Float(
                    'dropout_7',
                    min_value=0.2,
                    max_value=0.8,
                    default=0.4,
                    step=0.5
                )
            )
        )
        model.add(Dense(self.num_classes, activation='sigmoid'))

        model.compile(
            optimizer=keras.optimizers.Adam(
                hp.Float(
                    'learning_rate',
                    min_value=1e-9,
                    max_value=1e-1,
                    sampling='LOG',
                    default=1e-3
                )
            ),
            loss='binary_crossentropy',
            metrics=['accuracy']
        )
        return model
```

```
hypermodel = CNNHyperModel(input_shape=INPUT_SHAPE, num_classes=NUM_CLASSES)

from kerastuner.tuners import RandomSearch
SEED=42
MAX_TRIALS=10
EXECUTION_PER_TRIAL=5
NUM_CLASSES = 1
INPUT_SHAPE = (SIZE, SIZE, 3)

hypermodel = CNNHyperModel(input_shape=INPUT_SHAPE, num_classes=NUM_CLASSES)

tuner = RandomSearch(
    hypermodel,
    objective='val_accuracy',
    seed=SEED,
    max_trials=MAX_TRIALS,
    executions_per_trial=EXECUTION_PER_TRIAL,
    directory='SHRI',
    project_name='CAD',
    overwrite=True
)

tuner.search_space_summary()

N_EPOCH_SEARCH = 40

tuner.search(X_train, y_train, epochs=40, validation_data=(X_val,y_val))
```

4. ONE DIMENSIONAL NEURAL NETWORKS

(a) Data Calling

```
import pandas
from numpy import loadtxt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Activation

data2 = pandas.read_excel (r'C:\Users\SHRI\Videos\Ecg\ECG python\Untitled Folder\data_prepare\final data1\step-4 data for algorithm\Random data selection\ecg_ex_nonex_sup_train.xlsx', sheet_name=0)


df1 = data2.sample(frac=0.20,axis='columns')
df2 = df1.sample(frac=1,axis='columns')
df1.to_excel("testp2.xlsx")
trans1= df1.T
array1 = trans1.values
X1 = array1[:,:-1]
y1 = array1[:,-1]

df3 = data2.drop(df2.columns,axis=1)
df3.to_excel("trainp2.xlsx")
trans2= df3.T
array2 = trans2.values
X2 = array1[:,:-1]
y2 = array1[:,-1]
```

(b) Training Algorithm with three convolutional layers

```
import tensorflow as tf
from functools import partial
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import Flatten
from tensorflow.python.keras.layers import Conv1D
n_features = 1
X22 = X2.reshape((X2.shape[0], X2.shape[1], n_features))

X11 = X1.reshape((X1.shape[0], X1.shape[1], n_features))

# from keras.utils import normalize
# X22 = normalize(X22, axis=1)
# X11 = normalize(X11, axis=1)


# define model
#n_steps = 9792
n_steps = 815
model = Sequential()
model.add(Conv1D(filters=64, kernel_size=2, activation='relu', input_shape=(n_steps, n_features)))
model.add(Conv1D(filters=32, kernel_size=2, activation='relu'))
model.add(Conv1D(filters=16, kernel_size=2, activation= tf.keras.layers.LeakyReLU(alpha=0.1)))
model.add(Flatten())
model.add(Dense(16, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.summary()
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# fit model
X11 = X1.reshape((X1.shape[0], X1.shape[1], n_features))
history1 = model.fit(X11,y1,epochs = epoch, batch_size=10, verbose=2, validation_data = (X22, y2))
```

```
score, acc = model.evaluate(X22, y2,
                batch_size=10,
                verbose=1)
print('Test score:', score)
print('Test accuracy:', acc)

# plot training and validation accuracy along with training and validation loss
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report,confusion_matrix

acc = history1.history['accuracy']
val_acc = history1.history['val_accuracy']
loss = history1.history['loss']
val_loss = history1.history['val_loss']

epochs_range = range(epoch)

plt.figure(figsize=(15, 15))
plt.subplot(2, 2, 1)
plt.plot(epochs_range, acc, label='Training Accuracy')
plt.plot(epochs_range, val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('Training and Validation Accuracy')

plt.subplot(2, 2, 2)
plt.plot(epochs_range, loss, label='Training Loss')
plt.plot(epochs_range, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

```
# print out the classification report to see the precision and accuracy.
predictions = model.predict_classes(X22)
predictions = predictions.reshape(1,-1)[0]
print(classification_report(y2, predictions, target_names = ['CAD (Class 0)','NOCAD (Class 1)']))
```

5 TWO-DIMENSIONAL CONVOLUTIONAL NETWORK

```
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.image import ImageDataGenerator
#from keras.utils import to_categorical
```

(a) Model definition

```
X_train, X_test, y_train, y_test = train_test_split(dataset, label, test_size = 0.15, random_state = 1,)
X_train = X_train / 255.
X_test = X_test / 255.

INPUT_SHAPE = (SIZE, SIZE, 3)   #change to (SIZE, SIZE, 3)
# create model
model = Sequential()

model.add(Conv2D(64, (3, 3), input_shape=INPUT_SHAPE))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(32, (3, 3), kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2)
model.add(Conv2D(32, (3, 3), kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(32, (3, 3), kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(Conv2D(128, (3, 3), kernel_initializer = 'he_uniform'))
model.add(Activation('relu'))
# Number of layers
model.add(Dense(1))
model.add(Activation('sigmoid'))
# load weights
model.load_weights("model.weights.best.hdf5")
# Compile model (required to make predictions)
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
print("Created model and loaded weights from file")
```

(b) Optimizer Definition

```
import tensorflow as tf
opt=tf.keras.optimizers.Adam(
   learning_rate=0.00001,
   beta_1=0.1,
   beta_2=0.1,
   epsilon=1e-07,
   amsgrad=False,
   name="Adam",

)


model.compile(loss= 'binary_crossentropy',
          optimizer='adam',  #'adam',
          metrics=['accuracy'])

print(model.summary())
```

(c) Print Model Plot

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
keras.utils.plot_model(model, show_shapes=True)
```

(d) Model Fit

```
import tensorflow as tf
from tensorflow.keras.callbacks import ModelCheckpoint
es=tf.keras.callbacks.EarlyStopping( monitor="val_loss", patience=3,
                                     verbose=1,  restore_best_weights=True)
rlronp=tf.keras.callbacks.ReduceLROnPlateau( monitor="val_loss", factor=0.5, patie
nce=2,
                                             verbose=1)
checkpointer = ModelCheckpoint(filepath='model.weights.best.hdf5', verbose = 2, sav
e_best_only=True)
callback_list=[es, rlronp, checkpointer]

history = model.fit(X_train,
                    y_train,
                    batch_size = 32,
                    verbose = 2,
                    epochs = 25,
                    validation_split=0.10,
                    shuffle = False,callbacks=[callback_list])
```

(e) Random Testing

```
#Test the model on single images
n = random.randint(0, len(X_test)-1)
img = X_test[n]
plt.imshow(img)
input_img = np.expand_dims(img, axis=0) #Expand dims so the input is (num images, x, y, c)
print("The prediction for this image is: ", model.predict(input_img))
print("The actual label for this image is: ", y_test[n])
```

*Testing dataset

```
_, acc = model.evaluate(X_test, y_test)
print("Accuracy = ", (acc * 100.0), "%")
```

*Cross Validation

```
from sklearn.model_selection import StratifiedKFold
import numpy
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=seed)
cvscores = []
# evaluate the model
for train, test in kfold.split(x_train,y_train):
  scores = model.evaluate([x_test], [y_ test], verbose=0)
  print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
  cvscores.append(scores[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (numpy.mean(cvscores), numpy.std(cvscores)))
```

```
#ROC
from sklearn.metrics import roc_curve
y_preds = model.predict(X_test).ravel()
```

```
fpr, tpr, thresholds = roc_curve(y_test, y_preds)
plt.figure(1)
plt.plot([0, 1], [0, 1], 'y--')
plt.plot(fpr, tpr, marker='.')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.show()

#Threshold
import pandas as pd
i = np.arange(len(tpr))
roc = pd.DataFrame({'tf' : pd.Series(tpr-(1-
fpr), index=i), 'thresholds' : pd.Series(thresholds, index=i)})
ideal_roc_thresh = roc.iloc[(roc.tf-
0).abs().argsort()[:1]]  #Locate the point where the value is close to 0
print("Ideal threshold is: ", ideal_roc_thresh['thresholds'])

#Confusion matrix
mythreshold=float(ideal_roc_thresh['thresholds'])
from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
y_pred = (model.predict(X_test)>= mythreshold).astype(int)
cm=confusion_matrix(y_test, y_pred)
clr = classification_report(y_test, y_pred, labels=[0, 1], target_names=["CAD", "NOC
AD"])
plt.figure(figsize=(6, 6))
sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='Blues', cbar=False)
plt.xticks(ticks=[0.5, 1.5], labels=["CAD", "NOCAD"])
plt.yticks(ticks=[0.5, 1.5], labels=["CAD", "NOCAD"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()

print("Classification Report:\n----------------------\n", clr)
```

6 TRANSFER LEARNING APPROACH

(a) EFFICIENT NET B0

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import layers
from tensorflow.keras import Model
import matplotlib.pyplot as plt

!pip install git+https://github.com/qubvel/segmentation_models
import efficientnet.tfkeras as efn
base_model = efn.EfficientNetB0(input_shape = (150, 150, 3), include_top = False, w
eights = 'imagenet')
for layer in base_model.layers:
   layer.trainable = False
# Flatten the output layer to 1 dimension
x = layers.Flatten()(base_model.output)
# Add a fully connected layer with 512 hidden units and ReLU activation
x = layers.Dense(512, activation='relu')(x)
# Add a dropout rate of 0.5
x = layers.Dropout(0.5)(x)

# Add a final sigmoid layer for classification
x = layers.Dense(1, activation='sigmoid')(x)

model = tf.keras.models.Model(base_model.input, x)
model.compile(loss='binary_crossentropy',#BinaryFocalLoss(gamma=5)
            optimizer='adam',  #'adam',           #also try rmsprop or custom optimizer "opt"
            metrics=['accuracy'])
print(model.summary())
```

(b) Inception V3

```
from tensorflow.keras.applications.inception_v3 import InceptionV3
base_model = InceptionV3(input_shape = (150, 150, 3), include_top = False, weights = 'imagenet')
```

(c) Mobilenet V2

```
import tensorflow as tf
base_model = tf.keras.applications.MobileNetV2(input_shape = (150, 150, 3), include_top = False, weights = "imagenet")
```

(d) ResNet50

```
from tensorflow.keras.applications import ResNet50
base_model = ResNet50(input_shape=(150, 150,3), include_top=False, weights="imagenet")
```

(e) VGG16

```
from tensorflow.keras.applications.vgg16 import VGG16
base_model = VGG16(input_shape = (150, 150, 3), # Shape of our images
include_top = False, # Leave out the last fully connected layer
weights = 'imagenet')
```

7. MULTI HEADED MODEL

(a) Data handling

```
from google.colab import drive
drive.mount('/content/drive')
import os
import cv2
from PIL import Image
import numpy as np
#Single Lead Images
image_directory = '/content/drive/MyDrive/PhD Data File/Images/'
#Twelve Lead Images
image_directory = '/content/drive/MyDrive/PhD Data File/12lead/'
SIZE = 256

dataset = []  #Many ways to handle data, you can use pandas. Here, we are using a list format.
label = []  #Place holders to define add labels. We will add 0 to all parasitized images and 1 to uninfected.

CAD_images = os.listdir(image_directory + 'CAD/')
for i, image_name in enumerate(CAD_images):    #Remember enumerate method adds a counter and returns the enumerate object

    if (image_name.split('.')[1] == 'jpg' or 'JPG'):
        image = cv2.imread(image_directory + 'CAD/' + image_name)
        #image=cv2.medianBlur(image,1)
        #image=~image00
        image = Image.fromarray(image, 'RGB')
        image = image.resize((SIZE, SIZE))
        dataset.append(np.array(image))
        label.append(1)
```

```
#Iterate through all images in Uninfected folder, resize to 64 x 64
#Then save into the same numpy array 'dataset' but with label 1

NOCAD_images = os.listdir(image_directory + 'NOCAD/')
for i, image_name in enumerate(NOCAD_images):
    if (image_name.split('.')[1] == 'jpg' or 'JPG'):
        image = cv2.imread(image_directory + 'NOCAD/' + image_name)
        #image=cv2.medianBlur(image,1)
        #image=~image
        image = Image.fromarray(image, 'RGB')
        image = image.resize((SIZE, SIZE))
        dataset.append(np.array(image))
        label.append(0)

dataset = np.array(dataset)
label = np.array(label)

#Sanity check, view few mages
import random
import numpy as np
import matplotlib.pyplot as plt

image_number = random.randint(0, len(dataset)-1)
plt.imshow(np.reshape(dataset[image_number], (SIZE, SIZE, 3)))
print("Label for this image is: ", label[image_number])

*Training images split
from sklearn.model_selection import train_test_split
#from keras.utils import to_categorical
x_train, x_val, y_train, y_val = train_test_split(dataset, label, test_size = 0.016, random_state=41, stratify = label)
```

*Data handling

```
# multivariate multi-headed 1d cnn example
from numpy import array
from numpy import hstack
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Input
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Conv1D
from tensorflow.keras.layers import MaxPooling1D
from tensorflow.keras.layers import concatenate
import tensorflow as tf
import matplotlib as plot
import pandas
from numpy import loadtxt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Activation


# load the dataset
# Single lead data
data1 = pandas.read_excel (r'/content/drive/MyDrive/PhD Data File/CAD_NC_wi_feat_1lde.xlsx', sheet_name=0)

# Twelve lead data

data1 = pandas.read_excel (r'/content/drive/MyDrive/PhD Data File/CAD_NC_wi_feat.xlsx', sheet_name=0)

datad= data1.T
datasetd= datad.values
```

```
data1= datasetd[:,:-1]
label1= datasetd[:,-1]

*Training data split
x_traind, x_vald, y_traind, y_vald = train_test_split(data1, label1, test_size = 0.016, random_state=41, stratify = label1)
```

(b) Data correlations

```
# calculate the Pearson's correlation between two variables
from scipy.stats import pearsonr

# calculate validation Pearson's correlation
corr, _ = pearsonr(y_val, y_vald)
print('Pearsons correlation: %.3f' % corr)

# calculate train Pearson's correlation
corr, _ = pearsonr(y_train, y_traind)
print('Pearsons correlation: %.3f' % corr)
```

(c) Scaling data

```
from sklearn.preprocessing import StandardScaler
X11= rescaledX1 =  StandardScaler().fit_transform(X1)
X22= rescaledX2 = StandardScaler().fit_transform(X2)
```

(d) Algorithm code

```
from tensorflow.keras.layers import Dense, Conv2D , MaxPool2D , Flatten , Dropout, BatchNormalization
from tensorflow.python.keras.layers import MaxPooling2D, GlobalMaxPooling2D
from tensorflow.keras.applications.mobilenet_v2 import preprocess_input
```

```
# first input model
visible1 = Input(shape=(815,1))
cnn1 = Conv1D(filters=8, kernel_size=5, activation= 'relu')(visible1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Conv1D(filters=16, kernel_size=5, activation= 'relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Dropout(0.2)(cnn1)
cnn1 = Conv1D(filters=32, kernel_size=5, activation='relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Conv1D(filters=32, kernel_size=5, activation='relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Dropout(0.2)(cnn1)
cnn1 = Conv1D(filters=64, kernel_size=5, activation='relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Conv1D(filters=64, kernel_size=5, activation='relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Dropout(0.2)(cnn1)
cnn1 = Conv1D(filters=128, kernel_size=5, activation='relu')(cnn1)
cnn1 = MaxPooling1D(pool_size=2)(cnn1)
cnn1 = Dropout(0.2)(cnn1)
cnn1 = Flatten()(cnn1)
cnn1 = Dense(256, activation='relu')(cnn1)
cnn1 = Dropout(0.4)(cnn1)
cnn1 = Dense(128, activation='relu')(cnn1)

# second input model
visible2 = Input(shape=(100, 100, 3))
cnn2 = Conv2D(filters=8,kernel_size= (5,5),activation='relu',padding='same')(visible2)
cnn2 = Conv2D(filters=16,kernel_size= (5,5),activation='relu',padding='same')(cnn2)
cnn2 = MaxPooling2D(pool_size=(2,2))(cnn2)
cnn2 = Dropout(0.2)(cnn2)
```

```
cnn2 = Conv2D(filters=32,kernel_size= (5,5),activation='relu',padding='same')(cnn2)
cnn2 = Conv2D(filters=32,kernel_size= (5,5),activation='relu',padding='same')(cnn2)
cnn2 = MaxPooling2D(pool_size=(2,2))(cnn2)
cnn2 = Dropout(0.2)(cnn2)
cnn2 = Conv2D(filters=64,kernel_size= (5,5),activation='relu',paddi
cnn2 = Conv2D(filters=64,kernel_size= (5,5),activation='relu',padding='same')(cnn2)
cnn2 = MaxPooling2D(pool_size=(2,2))(cnn2)
cnn2 = Dropout(0.2)(cnn2)
cnn2 = Conv2D(filters=128,kernel_size= (5,5),activation='relu',padding='same')(cnn2
)
cnn2 = Conv2D(filters=128,kernel_size= (5,5),activation='relu',padding='same')(cnn2
)
cnn2 = MaxPooling2D(pool_size=(2,2))(cnn2)
cnn2 = Dropout(0.2)(cnn2)
cnn2 = Flatten()(cnn2)
cnn2 = Dense(256, activation='relu')(cnn2)
cnn2 = Dropout(0.4)(cnn2)
cnn2 = Dense(128, activation='relu')(cnn2)

# third input model
visible3 = Input(shape=(100, 100, 3))
cnn3 = Flatten()(visible3)
cnn3 = Dense(8, activation='relu')(cnn3)
cnn3 = Dense(32, activation='relu')(cnn3)
cnn3 = Dense(256, activation='relu')(cnn3)
cnn3 = Dropout(0.4)(cnn3)
cnn3 = Dense(128, activation='relu')(cnn3)

# merge input models
merge = concatenate([cnn1, cnn2, cnn3])
output = Dense(1, activation='sigmoid')(merge)
model = Model(inputs=[visible1, visible2, visible3], outputs=output)
```

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

(e) Model Architecture

```
from tensorflow.keras.utils import plot_model
# plot graph
plot_model(model, to_file='multiple_inputs.png')
```

(f) Model Fitting and End conditions

```
from tensorflow.keras.callbacks import ModelCheckpoint
checkpointer = ModelCheckpoint(filepath='model.weights.best.hdf121', verbose = 2,
save_best_only=True)
Es= tf.keras.callbacks.EarlyStopping(monitor='val_loss',patience=10,restore_best_weights=True)
# fit model
Deepmod = model.fit([x_traind, x_train, x_train], y_train, epochs=100, verbose=2, batch_size=18, callbacks=[checkpointer, Es], validation_data = ([x_vald, x_val, x_val],
y_val))

#plot the training and validation accuracy and loss at each epoch
loss = Deepmod.history['loss']
val_loss = Deepmod.history['val_loss']
epochs = range(1, len(loss) + 1)
plt.plot(epochs, loss, 'y', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
plt.title('Training and validation loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

```
acc = Deepmod.history['accuracy']
val_acc = Deepmod.history['val_accuracy']
plt.plot(epochs, acc, 'y', label='Training acc')
plt.plot(epochs, val_acc, 'r', label='Validation acc')
plt.title('Training and validation accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()
plt.plot(epochs, loss, label='Training Loss')
plt.plot(epochs, val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('Training and Validation Loss')
plt.show()
```

(g) Cross Validation

```
from sklearn.model_selection import StratifiedKFold
import numpy
# fix random seed for reproducibility
seed = 7
numpy.random.seed(seed)
kfold = StratifiedKFold(n_splits=10, shuffle=True, random_state=seed)
cvscores = []
# evaluate the model
for train, test in kfold.split(x_train,y_train):
  scores = model.evaluate([x_vald, x_val, x_val], [y_val], verbose=0)
  print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
  cvscores.append(scores[1] * 100)
print("%.2f%% (+/- %.2f%%)" % (numpy.mean(cvscores), numpy.std(cvscores)))
```

(h) ROC Plot

```
#ROC can help identify the right threshold.
```

```
from sklearn.metrics import roc_curve
y_preds = model.predict([x_vald, x_val, x_val]).ravel()

fpr, tpr, thresholds = roc_curve(y_val, y_preds)
plt.figure(1)
plt.plot([0, 1], [0, 1], 'y--')
plt.plot(fpr, tpr, marker='.')
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.show()
```

(i) Confusion matrix

```
#We compare labels and plot them based on correct or wrong predictions.
#Since sigmoid outputs probabilities we need to apply threshold to convert to label.
x_test = x_val
x_testd = x_vald
y_test = y_val
mythreshold=float(ideal_roc_thresh['thresholds'])

from sklearn.metrics import confusion_matrix, classification_report
import seaborn as sns
y_pred = (model.predict([x_testd, x_test, x_test])>= mythreshold).astype(int)
cm=confusion_matrix(y_test, y_pred)
clr = classification_report(y_test, y_pred, labels=[1, 0], target_names=["CAD", "NOCAD"])
plt.figure(figsize=(6, 6))
sns.heatmap(cm, annot=True, fmt='g', vmin=0, cmap='Blues', cbar=False)
plt.xticks(ticks=[0.5, 1.5], labels=["NOCAD", "CAD"])
plt.yticks(ticks=[0.5, 1.5], labels=["NOCAD", "CAD"])
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.title("Confusion Matrix")
plt.show()
print("Classification Report:\n----------------------\n", clr)
```

APPENDIX – II

WORKING PRINCIPLE OF DEEP LEARNING MODELS

(A) Hyperparameter of Convolutional neural networks

A.1 Activation functions

Activation function generates a scalar value from a weighted sum of inputs in artificial neural networks(Pandey and Gupta 2018). These are basically divided into two types:

a. Linear Activation Function

b. Non-linear Activation Functions

a. Linear activation function: The function, where the input values are linearly related (Figure A1) to output is known as linear activation function. That is mathematically represented as:

$$f(x) = x$$

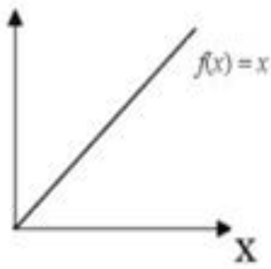


Figure A1**:** Linear activation function

b. Non-linear Activation Functions: These types of activation functions show nonlinearity behavior between input and output values. These functions help in stacking of multiple layers of neurons and generating non-linear output from the input passed through multiple layers. Some of the major nonlinear activation functions are

(i) Sigmoid Activation Functions:

The output of sigmoid activation function is always in the range of 0 -1. The function assumes the shape shown in Figure A2 and is mathematically represented as:

$$f(x) = \frac{1}{1+e^{x}}$$

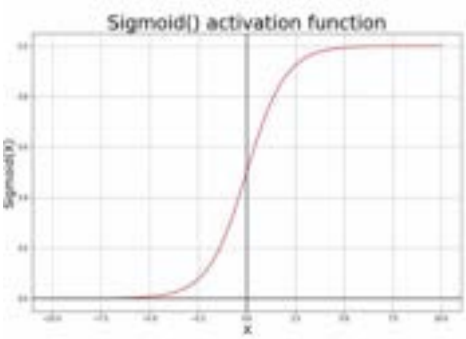


Figure A2: Sigmoid activation function (Rai and Mitra 2021)

(ii) Rectified Linear Unit (Relu) Activation Functions:

The rectified linear activation function (He et al. 2015) is similar to the linear function but the difference is, it generates output directly if the input is positive. Otherwise, it remains zero (Figure A3). It is the default activation function for many types of neural networks due to the ease of training and often achieves better performance. This is mathematically represented as:

$$f(x) = \max(0, x)$$

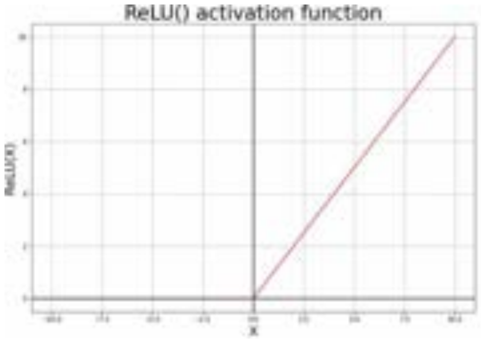


Figure A3**:** Relu activation function (Rai and Mitra 2021)

(iii) Softmax Function:

The softmax function is similar to the sigmoid function with a range of -1 to +1 (Figure A4) but it is flexible to handle multi-class classification algorithms. It is mathematically represented as:

$$f(x) = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

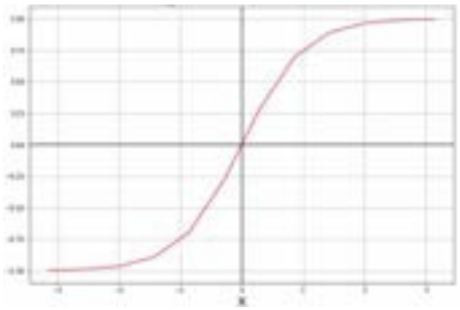


Figure A4**:** Softmax activation function (Rai and Mitra 2021)

<u>(iv) Leaky Relu Activation Functions</u>

Leaky ReLU indicates the upgraded behavior compared to the ReLU activation function and shows a small slope in the negative area (Figure A5) which is indicated as zero in Relu activation. This is mathematically expressed as:

$$\mathrm{f(x) = max\,(0.1x, x)}$$

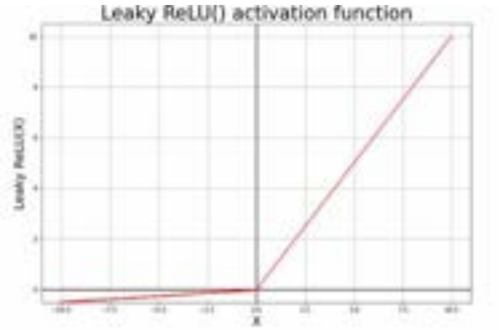


Figure A5**:** Leaky Relu activation function (Rai and Mitra 2021)

A.2 Filters

A filter is like a single pattern, which convolves across the input and finds similarities between the stored template which is related to different regions in the input image. The common range of filters defined is 16 to 512.

A.3 Kernel Size

Kernel size refers to the width and height of the filter matrix. Commonly used kernel sizes will be varying from 1x1 to 5x5.

A.4 Stride and padding

Stride (Steinmetzer et al. 2019) defines the size of the step consideration of the kernel when sliding through the image. The Padding describes the addition of zero pixels around the edges of an image. The requirement of padding is to maintain the original size of an image during the application of a convolutional filter.

A.5 Weight and Bias

Weights and biases (Li et al. 2017) are initially unknown, learnable parameters in any prediction model. Weights are parameters which determine how strongly each of the neurons affects other neurons (strength of connection). Bias are constant values, which are additional input parameter into the next layer.

A.6 Normalization

Normalization is one of the preprocessing (rescaling technique) operations of data.

A.7 Regularization

Regularization is a technique related to modification of the prediction model to overcome overfitting problems.

A.8 Overfitting and Underfitting

Overfitting errors occur when the model tries to learn all the data points or more than defined data points from given dataset. But underfitting errors occur when model is unable to rectify the underlying trend of the dataset.

A.9 Maxpooling

A pooling operation which calculates the maximum/largest, value in the region of each feature map selected by the filter is known as max-pooling. If it is considered as an averaged value of region, then it is known as average pooling.

A.10 Dropout

Dropout is a regularization technique, which affects the training process and is ideal for evaluation. The dropout minimizes the unnecessary feature which effects the network, making the network simpler and improves its generalization capabilities.

A.11 Learning rate

Learning rate is a tuning parameter to optimize the algorithm which determines the step size at each iteration to achieve a minimum of a loss function.

A.12 Optimizer

Optimizers are algorithms or mathematical methods used to minimize loss (cost) function (error function) and maximize the efficiency of the model. These mathematical functions depend on the model's learnable parameters. The major optimizer used for the development of the model is Gradient descent (Srivastava et al. 2015; Yoo et al. 2020). Gradient Descent is the commonly known iterative optimization algorithm. This optimization algorithm works based on a convex function (Figure A6) and squeezes the parameters by iteration process to minimize a loss function to its local minima. One of the famous variants of the gradient decent optimizer is the Adam optimizer algorithm. The method uses the computing adaptive learning rates for every parameter. It is based on the combination of RMS-Prop (Patil and Karandikar 2018) and Adadelta optimizer function (which are other variants of gradient descent optimizer). It saves the i.e. decaying average of the past gradients (momentum), as well as a decaying average of the past squared gradients.

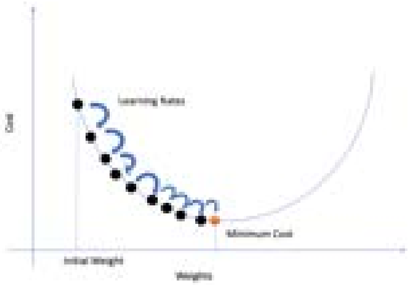


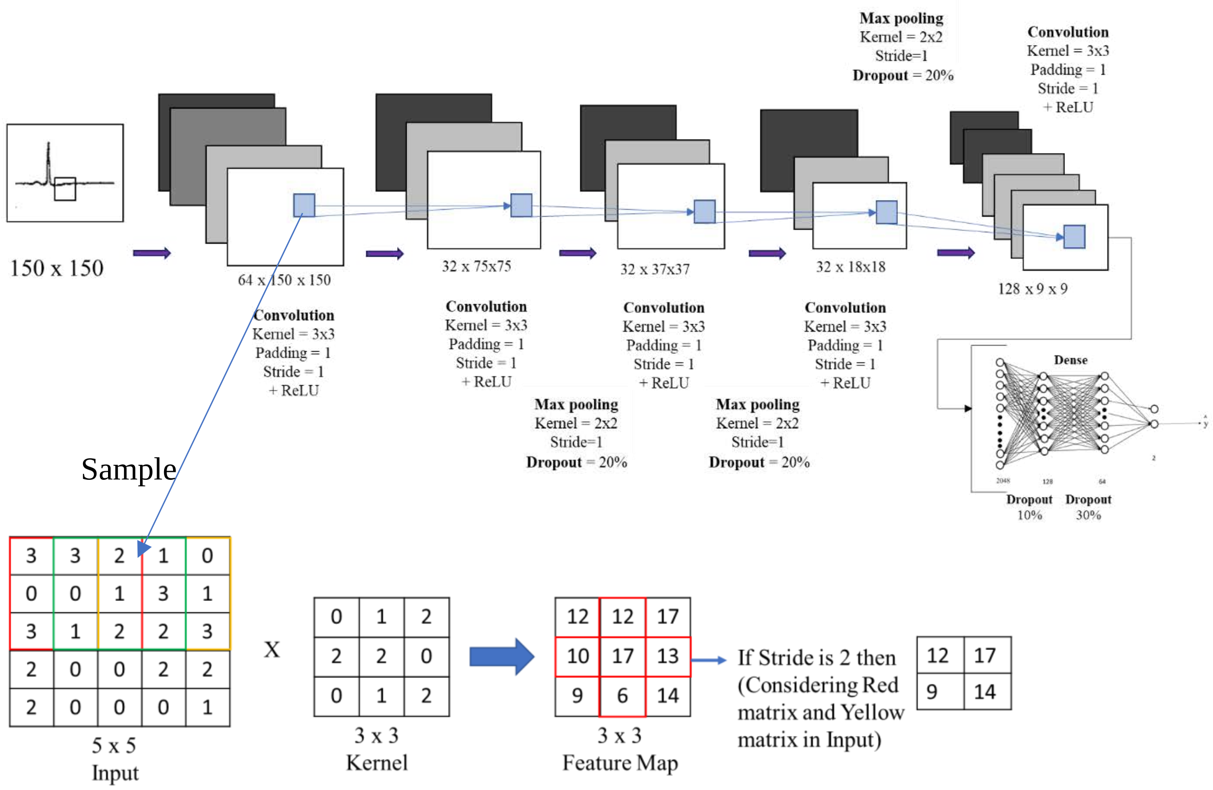
Figure A6**:** General principle of Gradient descent. (Ghosh et al. 2020)

A.13 Loss Function

The loss function provides the comparative relation between the target and predicted output values. The well-known loss functions are cross-entropy, mean square error, mean absolute error, etc. The cross-entropy loss function measures the difference between two probability distribution functions. Major classes of cross entropy functions are binary cross entropy function used for binary classification, and categorical cross-entropy for multiclass classification.

A.14 Deep convolutional neural network

The conceptual methodology followed in one dimensional and two-dimensional ECG analysis.



3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

0	1	2
2	2	0
0	1	2

12	12	17
10	17	13
9	6	14

If Stride is 2 then (Considering Red matrix and Yellow matrix in Input)

12	17
9	14

If Kernel size => 3x3

Feature Map (Output) Size = (Input – Kernel Size) +1

= (5-3) +1

= 3

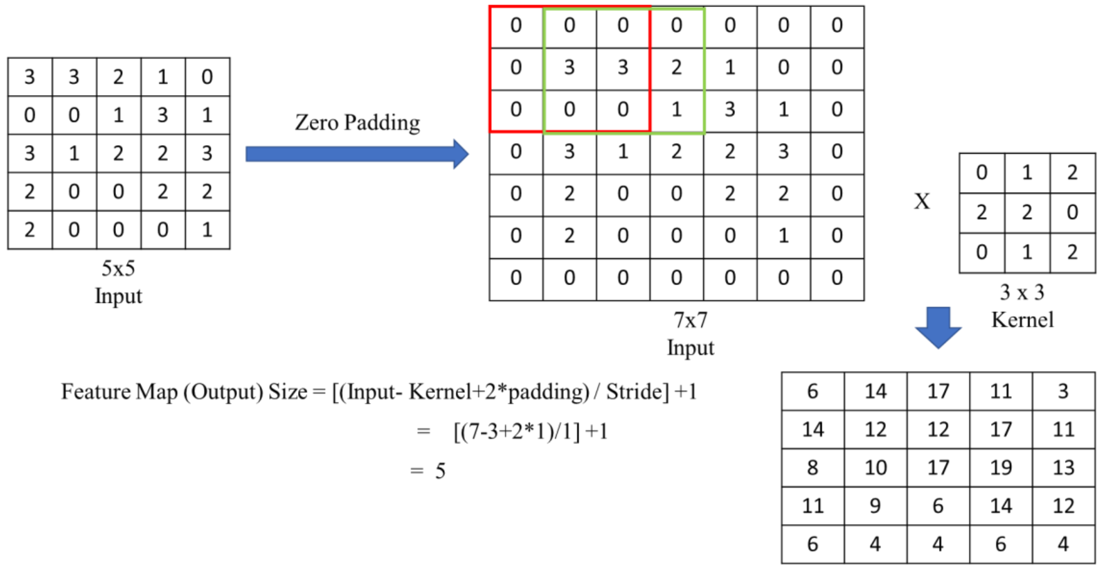
If Stride => Movement of Kernel (Horizontal and Vertical)= 2 x 2

Feature Map (Output) Size = [(Input- Kernel) / Stride] +1

= [(5-3)/2] +1

= 2

Padding : Adding zero as boundary elements to protect actual boundary features



5x5 Input

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

7x7 Input

0	0	0	0	0	0	0
0	3	3	2	1	0	0
0	0	0	1	3	1	0
0	3	1	2	2	3	0
0	2	0	0	2	2	0
0	2	0	0	0	1	0
0	0	0	0	0	0	0

3 x 3 Kernel

0	1	2
2	2	0
0	1	2

6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

Feature Map (Output) Size = [(Input- Kernel+2*padding) / Stride] +1

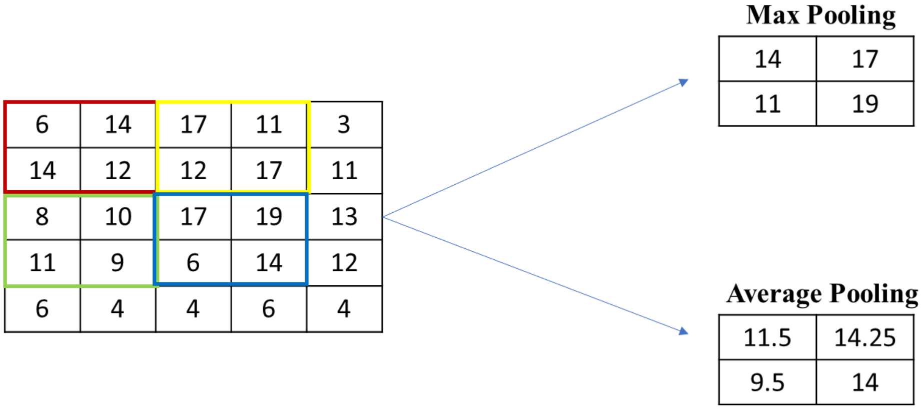
= [(7-3+2*1)/1] +1

= 5

Pooling: Reducing output feature map by retaining the features.

(i) **Max Pooling:** Pooling maximum Number

(ii) **Average Pooling :** Pooling average Number



6	14	17	11	3
14	12	12	17	11
8	10	17	19	13
11	9	6	14	12
6	4	4	6	4

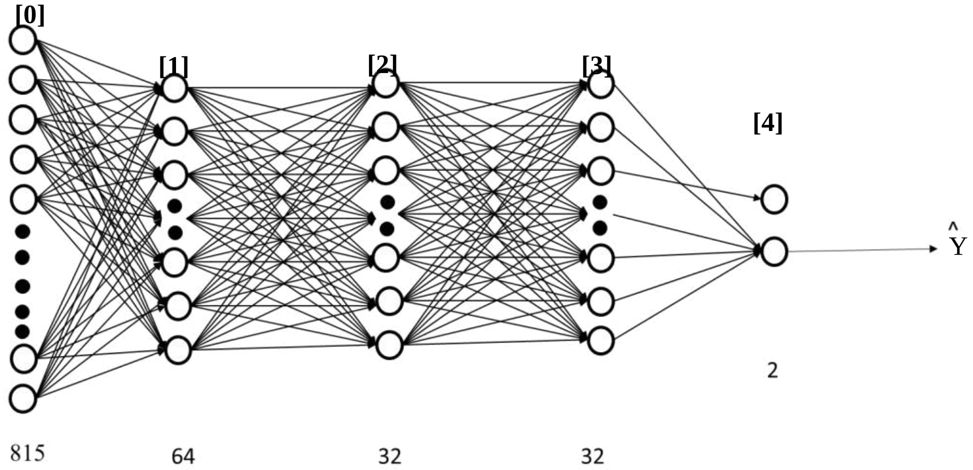
Max Pooling

14	17
11	19

Average Pooling

11.5	14.25
9.5	14

A.15 Neural network working principle



Forward Propagation:

- ⇨ Input Layer

$$A^{(0)} = X\ (815 \text{ x } m)$$

- ⇨ Obtain first layer Linear combination of weight and bias

$$\underset{(64 \text{ x m})}{Z^{(1)}} = \underset{(64 \text{ x } 815)}{w^{(1)}} \underset{(815 \text{ x m})}{A^{(0)}} + \underset{(64 \text{ x m})}{b^{(1)}}$$

- ⇨ Appling the activation function (Nonlinear function)

$$A^{(1)} = g\left(Z^{(1)}\right) = ReLU(Z^{(1)})$$

$$\text{ReLU } (x) = \begin{cases} x \quad if\ x > 0 \\ \quad or \\ 0 \ \ if\ x \le 0 \end{cases}$$

- ⇨ Obtain Second layer Linear combination of weight and bias

$$\underset{(32 \text{ x m})}{Z^{(2)}} = \underset{(32 \text{ x } 64)}{w^{(2)}} \underset{(64 \text{ x m})}{A^{(1)}} + \underset{(32 \text{ x m})}{b^{(2)}}$$

$$A^{(2)} = g\left(Z^{(2)}\right) = ReLU(Z^{(2)})$$

- ⇨ Obtain Third layer Linear combination of weight and bias

$$\underset{(32 \text{ x m})}{Z^{(3)}} = \underset{(32 \text{ x } 32)}{w^{(3)}} \underset{(32 \text{ x m})}{A^{(2)}} + \underset{(32 \text{ x m})}{b^{(3)}}$$

$$A^{(3)} = ReLU\,(Z^{(3)})$$

- ⇨ Obtain Forth layer Linear combination of weight and bias

$$\underset{(2\text{ x }m)}{Z^{(4)}} = \underset{(2\text{ x }32)}{w^{(4)}} \quad \underset{(32\text{ x }m)}{A^{(3)}} + \underset{(2\text{ x }m)}{b^{(4)}}$$

$$A^{(4)} = sigmoid\,(Z^{(4)})$$

Output Layer — Sigmoid Activation Function — Probabilities

$$\begin{Bmatrix} 1.2 \\ 5.6 \\ 2.1 \\ 0.8 \\ 2.3 \end{Bmatrix} \longrightarrow \boxed{\frac{1}{1-e^{-z}}} \longrightarrow \begin{Bmatrix} 0.03 \\ 0.91 \\ 0.06 \\ 0.01 \\ 0.07 \end{Bmatrix}$$

Backword Propagation:

From Predictions getting errors to update Algorithm

Error related to final layer

$$\underset{(2\text{ x }m)}{dZ^{(4)}} = \underset{(2\text{ x }m)}{A^{(4)}} - \underset{(2\text{ x }m)}{Y}$$

Error related to weights of forth layer

$$\underset{(2\text{ x }32)}{dw^{(4)}} = \frac{1}{m} \quad \underset{(2\text{ x }m)}{dZ^{(4)}} \quad \underset{(m\text{ x }32)}{A^{(3)T}}$$

Error related to bias of forth layer

$$\underset{(2\text{ x }1)}{db^{(4)}} = \frac{1}{m} \quad \underset{(2\text{ x }1)}{dZ^{(4)}}$$

Error related to third layer

$$\underset{(32\text{ x }m)}{dZ^{(3)}} = \underset{(32\text{ x }2)}{W^{(4)T}} \quad \underset{(2\text{ x }m)}{dZ^{(4)}} * \underset{(32\text{ x }m)}{g'\left(Z^{(3)}\right)}$$

Error related to weights of third layer

$$\underset{(32\text{ x }32)}{dw^{(3)}} = \frac{1}{m} \quad \underset{(32\text{ x }m)}{dZ^{(3)}} \quad \underset{(m\text{ x }32)}{A^{(2)T}}$$

Error related to bias of third layer

$$\underset{(32\text{ x }1)}{db^{(3)}} = \frac{1}{m} \quad \underset{(32\text{ x }1)}{dZ^{(3)}}$$

Error related second layer

$$\underset{(32\text{ x }m)}{dZ^{(2)}} = \underset{(32\text{ x }32)}{W^{(3)T}} \quad \underset{(32\text{ x }m)}{dZ^{(3)}} * \underset{(32\text{ x }m)}{g'\left(Z^{(2)}\right)}$$

Error related to weights of second layer

$$\underset{(32\text{ x }64)}{dw^{(2)}} = \frac{1}{m} \quad \underset{(32\text{ x }m)}{dZ^{(2)}} \quad \underset{(m\text{ x }64)}{A^{(1)T}}$$

Error related to bias of second layer

$$db^{(2)} = \frac{1}{m} \; dZ^{(2)}$$

(32 x 1) (32 x 1)

Error related first layer

$$dZ^{(1)} = W^{(2)T} \; dZ^{(2)} * g'\left(Z^{(1)}\right)$$

(64 x m) (64 x 32) (32 x m) (64 x m)

Error related to weights of first layer

$$dw^{(1)} = \frac{1}{m} \; dZ^{(1)} \; X^{T}$$

(64 x 815) (64 x m) (m x 815)

Error related to bias first layer

$$db^{(1)} = \frac{1}{m} \; dZ^{(1)}$$

(64 x 1) (64 x 1)

Updating learning parameters:

$$W^{(1)} = W^{(1)} - \propto dW^{(1)}$$
$$b^{(1)} = b^{(1)} - \propto db^{(1)}$$
$$W^{(2)} = W^{(2)} - \propto dW^{(2)}$$
$$b^{(2)} = b^{(2)} - \propto db^{(2)}$$
$$W^{(3)} = W^{(3)} - \propto dW^{(3)}$$
$$b^{(3)} = b^{(3)} - \propto db^{(3)}$$
$$W^{(4)} = W^{(4)} - \propto dW^{(4)}$$
$$b^{(4)} = b^{(4)} - \propto db^{(4)}$$

Where,

$\propto$ = Learning Rate

$A^{(0)}$ = Input Layer

$Z^{(1)}, Z^{(2)}, Z^{(3)}, Z^{(4)}$ = Results of First Layer, second layer, third layer, forth layer respectively.

$dZ^{(1)}, dZ^{(2)}, dZ^{(3)}, dZ^{(4)}$ = Error of first layer, second layer, third layer, forth layer respectively.

Y = Prediction = $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ = Binary prediction

(B) Different transfer learning approaches

B.1 VGG16

Model VGG16 is based on a convolutional neural network and is one of the best vision model architectures. Compare to normal networks VGG resolves problems, in dealing with large data with a huge number of hyper-parameters. (Aqib Haqmi Abas et al. 2018; Sharma and Mehra 2020) which concentrated on a tiny convolution layer with filter size (3x3) and stride one. Even, the padding and max-pool layers have filters(2x2) and stride two. A similar arrangement is repeated throughout the complete architecture. At the end of the architecture, it is connected to the output followed by two fully connected layers. For the classification, the output layer is coupled to a sigmoid activation function. The number 16 in VGG16 refers to the number of weighted layers present in the network and the network stands as large with over 138 million parameters.

The transfer learning method is used to get a pre-trained configuration of VGG16 architecture (Figure B1) with the ImageNet dataset to predict the collected ECG signal image dataset. Initially, the architecture model is trained with a large labelled image dataset (1.4 million) with thousand different classes. This helps to get a good pre-trained model and feature learned model to classify defined ECG signal images.

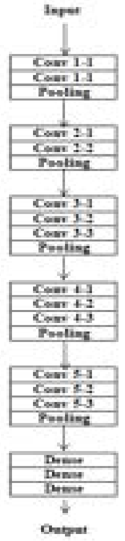


Figure B1 Architecture of VGG16 (Sharma and Mehra 2020)

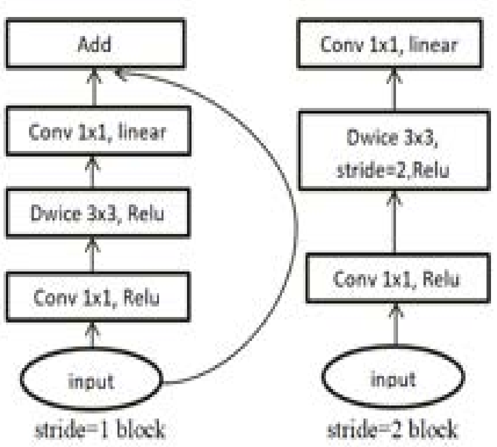


Figure B2 MobileNetV2 Architecture (Sandler et al. 2018)

B.2 MobileNetV2

MobileNetV2 is a Google-built model architecture to give real-time categorization capabilities in devices such as smartphones with limited computational capabilities. The network architecture uses an inverted residual structure as the input and output of the residual block are like thin bottle neck layers. It also employs lightweight convolution filters to increase the features in the expansion layer, as well as to remove the non-linearities in the narrow layer. This implementation of leverage is done by transfer learning method from ImageNet data to a defined dataset (Sandler et al. 2018). This Mobile Net improves the efficiency based on the state-of-the-art performance of mobile models across the spectrum of different model sizes. The overall architecture of Mobile Net is represented as shown in Figure B2.

B.3 Inception

Inception is a more advanced version of Google-Net (Szegedy et al. 2015), which was first demonstrated in the ImageNet recognition competition. Using the transfer learning (Mehrotra et al. 2020) approach, the network also obtained good classification performance in numerous bio-medical applications. The inception model is used to combine multiple different-sized convolutional filters into a single filter, reducing the number of parameters that must be learned (i.e. during a deeper network with controlling parameters under 25 million compared better against 60 million for Alex-Net (Li et al. 2021)). This resulted in reduction of computational complexity throughout the network. Figure B3 depicts the overall architecture of inception.

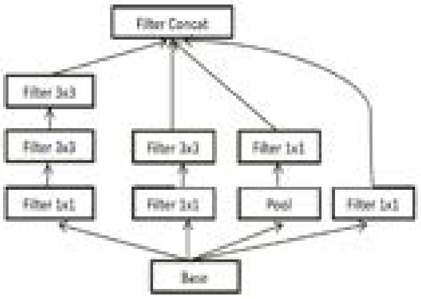


Figure B3 Architecture of Inception
(Szegedy et al. 2015)

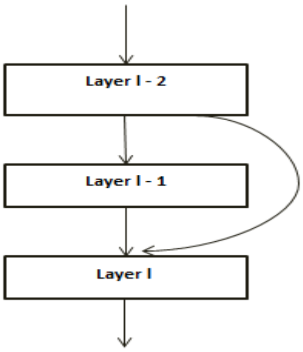


Figure B4 Canonical form of a residual neural network. (Sharma and Mehra 2020)

B.4 ResNet

Residual neural network is another well-known network for transfer learning (ResNet). This network utilizes skip connections, shortcuts, and bypass links to optimize layer weights. The non-linearity activation function (ReLu) and batch normalization are incorporated in the middle layers of the architecture in general ResNet models (Loganathan et al. 2021). The extra weight matrix may be used to train or learn skip weight method adaptation. Hence this model is also called Highway Nets (Srivastava et al. 2015). Figure B4 illustrates the Architecture of ResNet.

B.5 EfficientNet

The EfficientNet is a family of CNN-based architecture developed by the google team. The EfficientNet is a part of the baseline model neural network developed based on the ImageNet dataset. This architecture not only improves the accuracy but also provides better efficiency by reducing number of parameters as compared to state of art methods. (Tan and Le 2019) states about several models developed from a baseline models by performing compound scaling methods (EfficientNet B1 to B7). Figure B5 depicts the baseline efficient model.

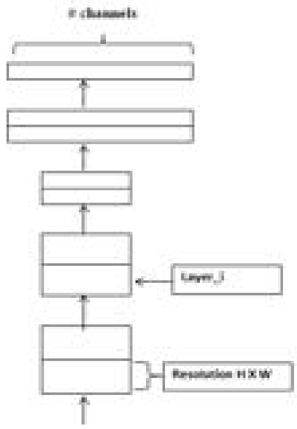


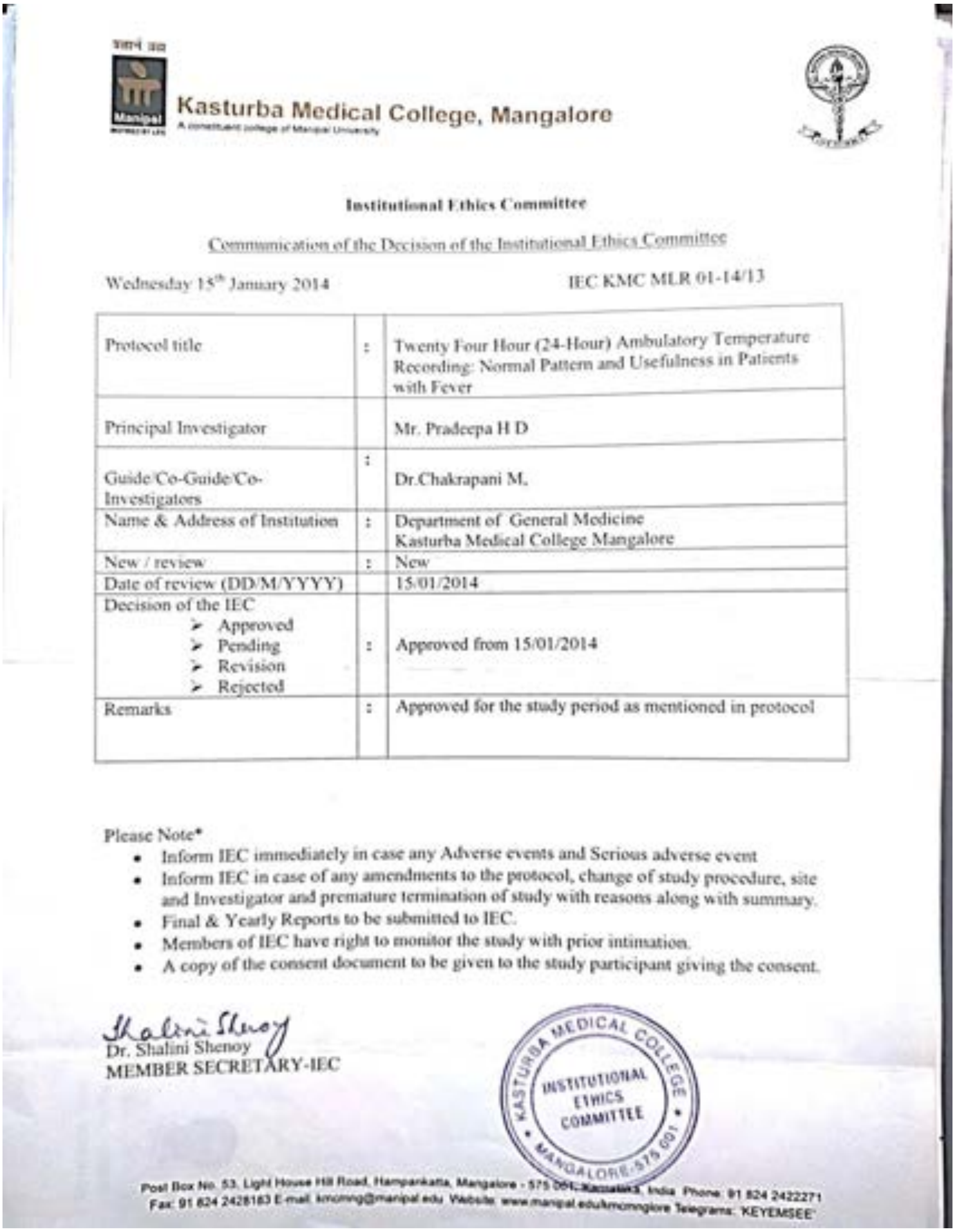
Figure B5 EfficientNet baseline architecture. (Tan and Le 2019)

Deep learning and machine learning models have already shown remarkable success in some medical disease data analysis (Acharya et al. 2017c), enabling more accurate diagnosis and treatment of a wide range of diseases, from infectious diseases to neurodegenerative disorders. As the field continues to evolve, it is likely that it will see even more exciting applications of deep learning models in the field of healthcare.

Application of these machine learning, deep learning and transfer learning models are considered to evaluate and classify the fever symptomatic diseases condition and heart related diseases condition based on the respective data types (temperature and ECG respectively). The understanding of the behavior of diseases and datasets will help to analyze the problem. The following sections briefly explain the cases related to the diseases and terminologies which are considered in the application machine learning/ deep learning algorithm.

APPENDIX-III

ETHICAL COMMITTEE APPROVAL LETTER



Manipal

Kasturba Medical College, Mangalore
A constituent college of Manipal University

Institutional Ethics Committee

Communication of the Decision of the Institutional Ethics Committee

Wednesday 15th January 2014 IEC KMC MLR 01-14/13

Protocol title	:	Twenty Four Hour (24-Hour) Ambulatory Temperature Recording: Normal Pattern and Usefulness in Patients with Fever
Principal Investigator		Mr. Pradeepa H D
Guide/Co-Guide/Co-Investigators	:	Dr.Chakrapani M.
Name & Address of Institution	:	Department of General Medicine Kasturba Medical College Mangalore
New / review	:	New
Date of review (DD/M/YYYY)		15/01/2014
Decision of the IEC ➢ Approved ➢ Pending ➢ Revision ➢ Rejected	:	Approved from 15/01/2014
Remarks	:	Approved for the study period as mentioned in protocol

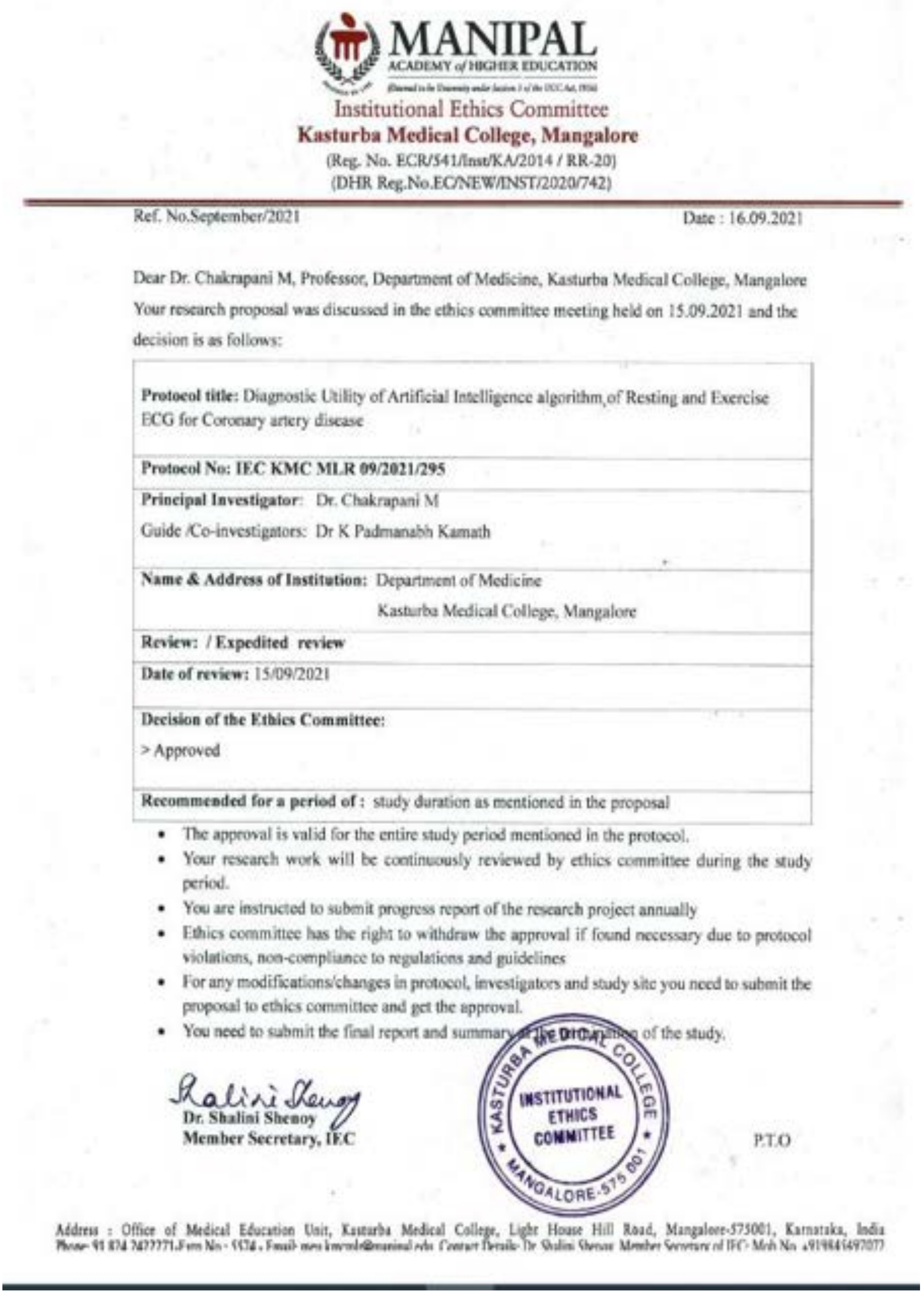
Please Note*

- Inform IEC immediately in case any Adverse events and Serious adverse event
- Inform IEC in case of any amendments to the protocol, change of study procedure, site and Investigator and premature termination of study with reasons along with summary.
- Final & Yearly Reports to be submitted to IEC.
- Members of IEC have right to monitor the study with prior intimation.
- A copy of the consent document to be given to the study participant giving the consent.

Shalini Shenoy
Dr. Shalini Shenoy
MEMBER SECRETARY-IEC

KASTURBA MEDICAL COLLEGE • MANGALORE-575 001 •
INSTITUTIONAL ETHICS COMMITTEE

Post Box No. 53, Light House Hill Road, Hampankatta, Mangalore - 575 001, Karnataka, India. Phone: 91 824 2422271
Fax: 91 824 2428183 E-mail: kmcmng@manipal.edu Website: www.manipal.edu/kmcmnglore Telegrams: 'KEYEMSEE'



MANIPAL
ACADEMY of HIGHER EDUCATION
(Deemed to be University under Section 3 of the UGC Act, 1956)
Institutional Ethics Committee
Kasturba Medical College, Mangalore
(Reg. No. ECR/541/Inst/KA/2014 / RR-20)
(DHR Reg.No.EC/NEW/INST/2020/742)

Ref. No.September/2021

Date : 16.09.2021

Dear Dr. Chakrapani M, Professor, Department of Medicine, Kasturba Medical College, Mangalore

Your research proposal was discussed in the ethics committee meeting held on 15.09.2021 and the decision is as follows:

Protocol title: Diagnostic Utility of Artificial Intelligence algorithm of Resting and Exercise ECG for Coronary artery disease

Protocol No: IEC KMC MLR 09/2021/295

Principal Investigator: Dr. Chakrapani M

Guide /Co-investigators: Dr K Padmanabh Kamath

Name & Address of Institution: Department of Medicine
Kasturba Medical College, Mangalore

Review: / Expedited review

Date of review: 15/09/2021

Decision of the Ethics Committee:

> Approved

Recommended for a period of : study duration as mentioned in the proposal

- The approval is valid for the entire study period mentioned in the protocol.
- Your research work will be continuously reviewed by ethics committee during the study period.
- You are instructed to submit progress report of the research project annually
- Ethics committee has the right to withdraw the approval if found necessary due to protocol violations, non-compliance to regulations and guidelines
- For any modifications/changes in protocol, investigators and study site you need to submit the proposal to ethics committee and get the approval.
- You need to submit the final report and summary of the outcome of the study.

Shalini Shenoy
Dr. Shalini Shenoy
Member Secretary, IEC

INSTITUTIONAL ETHICS COMMITTEE

P.T.O

Address : Office of Medical Education Unit, Kasturba Medical College, Light House Hill Road, Mangalore-575001, Karnataka, India
Phone- 91 824 [illegible] Fax No - [illegible], Email- [illegible] Contact Details- Dr Shalini Shenoy Member Secretary of IEC- Mob No. [illegible]

BIODATA

SHRIVATHSA TV
#2/101, "SHRI PADMANABHA",
Thokur Post and Village,
Haleyangadi Via, Mangaluru, D.K., Karnataka, 574146
Ph.No.: 9964009281
Email. Id: shrishatv@gmail.com

Education Qualification

#	Details of Qualifications with specialization,	Institution/University	Month and Year of acquiring Qualification	Class/ Grade
1	XII	Govinda Dasa PU College, Surathkal	May / 2007	67.5%
2	BE (Mechanical Engineering)	Vemana Institute of Technology Bangaluru (VTU)	September / 2011	72.6%
3	M.Tech (Machine Design)	Bangalore Institute of Technology Bangaluru (VTU)	July / 2014	74.91%
4	(Ph.D) (AI in Healthcare)	NITK, Surathkal	To Date	CGPA: 8.5

Work Experience

#	Organization with full address	Period (MM/YYYY)			Designation & nature of work
		From	To	Duration	
1	Srinivas Institute of Technology, Valachil, Mangaluru	March 2015	July 2017	Two years- Three months	Assistant Professor

Publication Details

Journal Paper Publications Paper Published - 5

Conference Papers - 3

Journal Paper - 2

Journal Papers Under Review - 2