Heart Diseases Diagnosis Using Artificial Neural Networks

Alsalamah, Mashail

Award date:
2017

Awarding institution:
Coventry University

HEART DISEASES DIAGNOSIS USING ARTIFICIAL NEURAL NETWORKS

Submitted by:

Mashail Alsalamah

A thesis submitted in partial fulfilment of the university's requirements for the Degree of Doctor of Philosophy

January 2017

Certificate of Ethical Approval

Applicant:

Mashail Alsalamah

Project Title:

HEART DISEASE DIAGNOSIS USING DEEP LEARNING ARTIFICIAL NEURAL NETWORK

This is to certify that the above named applicant has completed the Coventry University Ethical Approval process and their project has been confirmed and approved as Low Risk

Date of approval:

27 January 2017

Project Reference Number:

P50627

ABSTRACT

Information technology has virtually altered every aspect of human life in the present era. The application of informatics in the health sector is rapidly gaining prominence and the benefits of this innovative paradigm are being realized across the globe. This evolution produced large number of patients' data that can be employed by computer technologies and machine learning techniques, and turned into useful information and knowledge. This data can be used to develop expert systems to help in diagnosing some life-threating diseases such as heart diseases, with less cost, processing time and improved diagnosis accuracy. Even though, modern medicine is generating huge amount of data every day, little has been done to use this available data to solve challenges faced in the successful diagnosis of heart diseases. Highlighting the need for more research into the usage of robust data mining techniques to help health care professionals in the diagnosis of heart diseases and other debilitating disease conditions.

Based on the foregoing, this thesis aims to develop a health informatics system for the classification of heart diseases using data mining techniques focusing on Radial Basis functions and emerging Neural Networks approach. The presented research involves three development stages; firstly, the development of a preliminary classification system for Coronary Artery Disease (CAD) using Radial Basis Function (RBF) neural networks. The research then deploys the deep learning approach to detect three different types of heart diseases i.e. Sleep Apnea, Arrhythmias and CAD by designing two novel classification systems; the first adopt a novel deep neural network method (with Rectified Linear unit activation) design as the second approach in this thesis and the other implements a novel multilayer kernel machine to mimic the behaviour of deep learning as the third approach. Additionally, this thesis uses a dataset obtained from patients, and employs normalization and feature extraction means to explore it in a unique way that facilitates its usage for training and validating different classification methods. This unique dataset is useful to researchers and practitioners working in heart disease treatment and diagnosis.

The findings from the study reveal that the proposed models have high classification performance that is comparable, or perhaps exceed in some cases, the existing automated and manual methods of heart disease diagnosis. Besides, the proposed deep-learning models provide better performance when applied on large data sets (e.g., in the case of Sleep Apnea), with reasonable performance with smaller data sets.

The proposed system for clinical diagnoses of heart diseases, contributes to the accurate detection of such disease, and could serve as an important tool in the area of clinic support system. The outcome of this study in form of implementation tool can be used by cardiologists to help them make more consistent diagnosis of heart diseases.

ACKNOWLEDGEMENTS

This thesis became a reality with the kind support and help of many individuals; I would like to extend my sincere thanks to all of them.

My deepest thanks to ALLAH for giving me the strength and ability to understand my studies, learn from them, and complete this thesis.

The completion of this project would not have been possible without the great assistance of my Director of Study, Dr. Saad Amin, and his encouragement, giving me creative and comprehensive advice until the work came into existence. I am also thankful to my supervision team, Dr. John and Dr Vasile, for their great help, and special thanks for Cardiology Department at King Abdullah Medical City especially Dr. Osama Elkhateeb , for his collaboration in collecting and compiling data.

I am highly indebted to my parents and the countless time they have helped me throughout my life and I hope this achievement will complete the dreams they had for me all those many years ago when they chose to give me the best education possible.

I would like to acknowledge and extend my heartfelt gratitude to my soul, my son, Khalid. I really would not have successfully met a number of challenges without your humble and kind heart! May ALLAH bless you and give you a happy live.

I would like to express my gratitude towards my whole family, including my brothers and sisters, particularly Halah and Lolo, for all the support and love you gave me which helped in the completion of this work.

I would express a deep sense of thanks to Abdulrahman, I won't face all the struggles without you.

I am grateful to my best friends, Asawer and Sally, who always stood with me and provided me strength, perseverance, support, and love.

Finally, my thanks and appreciation goes to all those people who have willingly helped me out with their abilities, suggestions, and support; you know who you are—thank you!

LIST OF PUBLICATIONS

This research has been documented, in part, within the following publications:

Alsalamah, M., Amin, S. & Halloran, J., 2014. Diagnosis of heart disease by using a radial basis function network classification technique on patients' medical records. 2014 IEEE MTT-S International Microwave Workshop Series on RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-Bio2014), IEEE.

Al-Salamah, M. & Amin, S., 2015. Heart Disease Diagnosis Using Reconstructive Radial Basis Function Networks with Overlapping Prevention Method. 5th EAI International Conference on Wireless Mobile Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies", ACM.

Alsalamah, M. & Amin, S., 2016. Medical Image Inpainting with RBF Interpolation Technique. International Journal of Advanced Computer Science and Applications(ijacsa), 7(8), pp. 91-99.

Al-Salamah, M. & Amin, S., 2016. Multilayer Radial Basis Function Kernel Machine. 6th EAI International Conference on Wireless Mobile Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies", Springer.

.

Al-Salamah, M. & Amin, S., 2016. Improving the Probability of Clinical Diagnosis of Coronary-Artery Disease using Extended Kalman Filters with Radial Basis Function Network. 6th EAI International Conference on Wireless Mobile Communication and Healthcare - "Transforming healthcare through innovations in mobile and wireless technologies", Springer.

TABLE OF CONTENTS

ABSTRACT I
ACKNOWLEDGEMENTS II
LIST OF PUBLICATIONS III
LIST OF FIGURES VIII
LIST OF TABLES XIII
LIST OF ABBREVIATIONS XV
LIST OF APPENDICES XVI
Chapter 1 : Introduction 1
1.1. Overview 1
1.2. Research Motivation 5
1.3. Research Aim and Objectives 6
1.3.1. Aim 6
1.3.2. Objectives 7
1.4. Research Contribution 9
1.5. Thesis Organization 12
Chapter 2 : Background about Heart Diseases and Artificial Neural Networks 14
2.1. Overview 14
2.2. Heart Diseases 15
2.2.1. Coronary Artery Disease 15
2.2.2. Arrhythmia 19
2.2.3. Sleep Apnea 23
2.3. Classification Techniques 26
2.3.1. Linear Classifier 27
2.3.2. Nonlinear Classifier 29
2.3.3. k-nearest neighbours (KNN) 32
2.3.4. Artificial Neural Networks 34
2.4. Summary 61
Chapter 3 : Heart Diseases Classifications using Data Mining and Deep Neural Networks 62
3.1. Overview 62
3.2. Heart Diseases Classification with Neural Networks and Deep Learning Methods 62

3.2.1. General Heart Disease 62
3.2.2. Coronary artery disease 64
3.2.3. Arrhythmia 65
3.2.4. Sleep Apnea 66
3.3. Other Classification Methods for Heart Diseases 67
3.3.1. General Heart Disease 67
3.3.2. Coronary artery disease 69
3.3.3. Arrhythmia 70
3.3.4. Sleep Apnea 73
3.4. Summary 74
Chapter 4 : Research Methodology and Data Collection 75
4.1. Overview 75
4.2. Research Design 75
4.3. Location of Research Study 76
4.4. Data Collection 78
4.4.1. Primary Data 78
4.4.2. Secondary Data 79
4.5. Data Processing 80
4.5.1. CAD Dataset Processing 80
4.5.2. Arrhythmia Dataset Processing 92
4.5.3. Sleep Apnea Dataset Processing 92
4.6. Classification Model 97
4.6.1. Radial Basis Function Neural Networks 98
4.6.2. Traditional Classifiers 98
4.6.3. Deep Neural Networks 98
4.6.4. Multilayer Radial Basis Function Kernal Machine 99
4.7. Performance Measures (Cornell University, 2003) 100
4.8. Summary 102
Chapter 5 : Prediction of Coronary Artery Disease Using a Combination of Methods for Training Radial Basis Function Networks 104
5.1. Overview 104

5.2. Background 104
5.3. Methodology 106
5.3.1. Data Set Description 107
5.3.2. Training the RBF Neural Network 108
5.4. Discussion 130
5.5. Summary 132
Chapter 6 : Heart Disease Detection Using Data Mining Algorithms and Deep Neural Networks 134
6.1. Overview 134
6.2. Datasets Description 134
6.3. Heart Diseases Detection Using Traditional Data Mining Algorithms 135
6.3.1. Methodology 135
6.3.2. Experimental Results 138
6.4. Heart Disease Detection Using Deep Neural Networks 152
6.4.1 Methodology 152
6.4.2 Model Classification 153
6.4.3 Experimental Results 161
6.5. Discussion 167
6.6. Summary 172
Chapter 7 : Heart Disease Diagnosis Using Deep Radial Basis Function Kernel Machine 174
7.1. Overview 174
7.2. Background 174
7.2.1. Deep Kernel Methods 174
7.2.2. Unsupervised Kernel Techniques 178
7.3. The proposed Deep RBF kernel machine 178
7.4. Methodology 180
7.4.1. Multilayer RBF kernel machine based on Kernel PCA 182
7.4.2. Multilayer RBF kernel machine based on supervised kernel regression 184
7.4.3. Multilayer RBF machine based on unsupervised kernel regression 185
7.5. Practical Implementation 190
Description of the main functions .....**Error! Bookmark not defined.**

7.6. Experimental Results..... 191
7.6.1. Datasets Description 191
7.6.2. Results Discussion and Analysis 192
7.7. Discussion 205
7.8. Summary 207
Chapter 8 : Conclusion and Future Work 209
8.1 Overview 209
8.2 Review of the Work and Contributions..... 210
8.3 Future Work 213
REFERENCES 215
APPENDICES .....**Error! Bookmark not defined.**

LIST OF FIGURES

Figure 1.1: Informatics pyramid 2

Figure 1.2: Relation between deep learning, machine learning and AI 5

Figure 1.3: Approaches of the research 7

Figure 1.4: overall approches 8

Figure 2.1: Figure (A) shows the location of the heart in the body. Figure (B) shows a normal coronary artery with normal blood flow. The inset image shows a cross-section of a normal coronary artery. Figure (C) shows a coronary artery narrowed by plaque. The inset image shows a cross-section of the plaque-narrowed artery. 16

Figure 2.2: Types of Arrhythmias 20

Figure 2.3: Types of Sleep Apnea 23

Figure 2.4: 1-Dimensional Linear Classifier 27

Figure 2.5: 2-Dimensional Three Classes Linear Classifier 28

Figure 2.6: Linear Decision Boundary on 2-Dimensional Decision Plane 29

Figure 2.7: Nonlinear Decision Boundary on 2-Dimensional Decision Plane 30

Figure 2.8: Local and Global Minimum 31

Figure 2.9: K-nearest Neighbours classification 32

Figure 2.10: Classification of new point with different *k* values 33

Figure 2.11: Schematic of biological neuron 34

Figure 2.12: Perceptron Structure 35

Figure 2.13: Two perceptron network 36

Figure 2.14: 2-Perceptrons Network Classification Surface 36

Figure 2.15: Linear Separability 37

Figure 2.16: Multilayer Perceptron Network 37

Figure 2.17: Fully-Connected, Feed-Forward Neural Network..... 39

Figure 2.18: Perceptron's Mathematical Structure 39

Figure 2.19: Distance from Centre Point 40

Figure 2.20: Distance in 1-Dimensional Space..... 41

Figure 2.21: Linear Radial Basis Function 41

Figure 2.22: Parameters of Gaussian Function 42

Figure 2.23: A node of RBF Network 42

Figure 2.24: RBF Network Classifier Structure 43

Figure 2.25: Two inputs create 2D space..... 43

Figure 2.26: Third layer of Network..... 45

Figure 2.27: A schematic diagram of the feed-forward layered network model represented by the radial basis function expansion..... 47

Figure 3.1: The methodology of the proposed algorithm 67

Figure 4.1: Proposed Methodology of the research..... 77

Figure 4.2: Research study location..... 70

Figure 4.3: Calculation of RR Intervals..... 93

Figure 4.4: Extracted RR Intervals from (a01) file..... 94

Figure 4.5: Extracted Features from RR Intervals of (a01) file..... 95

Figure 4.6: Feature Importance..... 96

Figure 4.7: Multilayer Radial Basis Function Kernel Machine Algorithms..... 100

Figure 5.1: Methodology Overview..... 107

Figure 5.2: Flow of the RBF network build-up to obtain the CAD diagnosis result..... 117

Figure 5.3: Three Combinations of Data sets and 4 different training and Activation functions without Matrix Completion..... 117
Figure 5.4: Three Combinations of Data sets and 4 different training and Activation functions with Matrix Completion..... 119
Figure 5.5: CAD Prediction Error..... 122
Figure 5.6: Testing results for set no 1 123
Figure 5.7: CAD estimate values for validation set no 1..... 124
Figure 5.8: Testing results for set no 2 125
Figure 5.9: CAD estimate values for validation set no 2..... 125
Figure 5.10: Testing results for set no 3 126
Figure 5.11: CAD estimate values for validation set no 3..... 126
Figure 5.12: Simple Data Set with three Combinations of Data sets and 4 different training and Activation functions.....**Error! Bookmark not defined.**
Figure 5.13: Cancer Data Set with three Combinations of Data sets and 4 different training and Activation functions.....**Error! Bookmark not defined.**
Figure 5.14: General steps of the gravitational search algorithm 128
Figure 5.15: Steps of PSOGSA..... 128
Figure 5.16: Classification rate for Cancer data set..... 129
Figure 5.17: Classification rate for CAD data set..... 129
Figure 6.1: Block Diagram of the proposed methodology 135
Figure 6.2: Values of ANOVA test for features set of Apnea dataset..... 136
Figure 6.3: Values of ANOVA test for features set of CAD dataset..... 137
Figure 6.4: Values of ANOVA results for features of Arrhythmia Dataset 138

Figure 6.5: Workflow of the proposed classification scheme..... 139
Figure 6.6: Classification performance obtained : (a) before features selection (b) after features selection..... 140
Figure 6.7: Accuracy comparison of classifiers performance before and after features selection 140
Figure 6.8: ROC Analysis for apnea class..... 143
Figure 6.9: ROC Analysis for non-apnea class..... 144
Figure 6.10: Classification performance obtained : (a) After features selection (b) Before features selection..... 145
Figure 6.11: ROC Analysis for CAD class..... 147
Figure 6.12: ROC Analysis for non-CAD class..... 148
Figure 6.13: Classification Performance for Arrhythmia Dataset 149
Figure 6.14: ROC Analysis for Arrhythmia class..... 151
Figure 6.15: ROC Analysis for non- Arrhythmia class 151
Figure 6.16: Block diagram of the proposed methodology 153
Figure 6.17: DNN Model Architecture..... 160
Figure 6.18: Structure of B-DNN Model.....**Error! Bookmark not defined.**
Figure 6.19: DNN-DT Classifier Architecture 161
Figure 6.20: B-DNN Model performance obtained : (a) before features selection (b) after features selection 162
Figure 6.21: Performance of the minute-class-based classification at DNN-DT scheme..... 164
Figure 6.22: Average Performance of the class-based classification stage at DNN-DT scheme 165
Figure 6.23: Confusion matrix for the class-based classification at DNN-DT scheme.. 165

Figure 6.24: Performance of B- DNN-DT at CAD dataset 166

Figure 6.25: Confusion matrix of B- DNN scheme at CAD dataset 166

Figure 6.26: Performance of B- DNN-DT at Arrhythmia dataset 167

Figure 6.27: Confusion matrix of B- DNN scheme at Arrhythmia dataset 167

Figure 7.1: Methodology Overview..... 181

Figure 7.2: Multilayer kernels machine for the three different transformations..... 182

Figure 7.3: The procedure of unsupervised regression (before improvement)..... 189

Figure 7.4: Accuracy vs. number of layers when applying the four algorithms for CAD dataset..... 195

Figure 7.5: Accuracy vs. number of layers when applying the four algorithms for Arrhythmia dataset..... 199

Figure 7.6: Accuracy vs. number of layers when applying the four algorithms for Apnea dataset..... 203

LIST OF TABLES

Table 4.1: Distribution of independent variables (Continuous)..... 81

Table 4.2: IV Values..... 82

Table 4.3: VIF of the variables..... 83

Table 4.4: Concordance values..... 84

Table 4.5: Model Summary..... 84

Table 4.6: Goodness of fit – Hosmer-Lemeshow..... 84

Table 4.7: Top 10 other diagnosis outcomes when CAD is present..... 86

Table 5.1: Summary of results of the 4 training and activation functions without matrix completion..... 118

Table 5.2: Summary of the best result obtained from Figure 5.4..... 120

Table 5.3: Summary of the best results obtained from Figure 5.12. **Error! Bookmark not defined.**

Table 5.4: Summary of the best results obtained from Figure 5.13. **Error! Bookmark not defined.**

Table 5.5: Classification Accuracy for PSOGSA with varying Hidden nodes..... 130

Table 6.1: Model parameters for the applied classifiers..... 139

Table 6.2: A summary of the classifiers performance for Apnea Dataset..... 140

Table 6.3: Confusion Matrix for the classifiers for Apnea Dataset..... 141

Table 6.4: A summary of the classifiers performance for CAD Dataset..... 145

Table 6.5: Confusion Matrix for the classifiers for CAD Dataset..... 146

Table 6.6: Confusion Matrix for the classifiers for Arrhythmia Dataset..... 149

Table 6.7: A summary of classifier performance for minute-based Classification..... 162

Table 6.8: Confusion Matrix before and after features selection for Apnea dataset 162
Table 6.9: Performance Summary for minute-class-based model of DNN-DT scheme for minute's quantification and class detection for each file of the Apnea dataset 163
Table 7.1: Results Summary of kPCA algorithm at CAD dataset 192
Table 7.2: Results Summary of Supervised Regression algorithm at CAD dataset 193
Table 7.3: Results Summary of Unsupervised Latent Regression algorithm at CAD dataset .. 194
Table 7.4: Results Summary of Unsupervised Latent Regression with projection algorithm at CAD dataset..... 195
Table 7.5: Summary of Confusion Matrix for kPCA algorithm for CAD dataset..... 196
Table 7.6: Results Summary of kPCA algorithm at Arrhythmia dataset..... 197
Table 7.7: Results Summary of Supervised Regression algorithm at Arrhythmia dataset 198
Table 7.8: Results Summary of Unsupervised Latent Regression algorithm at Arrhythmia dataset 198
Table 7.9: Results Summary of Unsupervised Latent Regression with projection algorithm at Arrhythmia dataset 199
Table 7.10: Summary of Confusion Matrix for Unsupervised Latent Regression with projection algorithm for Arrhythmia dataset 200
Table 7.11: Results Summary of kPCA algorithm at Apnea dataset 201
Table 7.12: Results Summary of Supervised Regression algorithm at Apnea dataset ... 201
Table 7.13: Results Summary of Unsupervised Latent Regression algorithm at Apnea dataset 202
Table 7.14: Results Summary of Unsupervised Latent Regression with Projection algorithm at Apnea dataset..... 202
Table 7.15: Summary of Confusion Matrix for Supervised Regression algorithm for Apnea dataset 204

LIST OF ABBREVIATIONS

CAD	Coronary Artery Disease
OSA	Obstructive Sleep Apnea
CSA	Central Sleep Apnea
MIX	Mix Sleep Apnea
PSG	Polysomnography
ECG	Electrocardiogram
KAMC	King Abdullah Medical City
ML	Machine Learning
ANN	Artificial Neural Network
DNN	Deep Neural Network
RBFN	Radial Basis Function Networks
MLKM	Multilayer Kernel Machine
MLP	Multilayer Perceptron
SVM	Support Vector Machines
KNN	K- Nearest Neighbour
PCA	Principal Component Analysis

LIST OF APPENDICES

Appendix I: Data Collection Form

Appendix II: The Statistic analysis study of Coronary-Artery Disease Data Based on King Abdullah Medical City in Saudi Arabia (KAMC-CAD)

Appendix III: ANOVA test results for arrhythmia data set

Appendix IV: Performance Summary for minute-class-based model of DNN-DT scheme for each file of Apnea data set

Appendix V: Results of Deep RBF kernel Machine Model

1 : Introduction

1.1. Overview

Information and communication technologies have revolutionized the way people lead their lives and conduct business in the 21st century. It would not be wrong to say that information technology has altered virtually every aspect of human lifestyle in the present era. Rouse (2016) reported that "health informatics is the study of resources and methods for the management of health information. This area of study supports health information technology, medical practice, medical research and medical informatics". The application of informatics in the health sector has been rapidly gaining prominence and the benefits of this innovative paradigm are being realized across the globe.

The importance of health informatics, as it is popularly known as, has risen significantly in the recent years due to the need for a secured and efficient management of medical data. The discipline of health informatics is involved in the efficient collection of medical data, secured archival and rapid retrieval; this improves patients' diagnoses and treatment. Health informatics also facilitates proper management, analysis and use of health-related data for more efficient healthcare delivery and service to the clients and patients. The inherent philosophy of health informatics is to transfer greater control of healthcare to the care providers and patients. It also emphasizes the importance and sensitivity of the roles played by healthcare professionals who handle and manage the data (University of Toronto, 2013).

Other terms such as clinical informatics and health information management are also very popular and they all focus on the aspect of incorporating the power of information technology into modern health practices and medical data management. Even though there are different names for the concept, the inherent philosophy of all of them is essentially the same. Whether it is called health informatics, biomedical informatics, or clinical informatics, it represents a process through which data is analysed and utilised to

generate knowledge that can be applied successfully to address clinical problems and facilitate rapid health care delivery in a highly time-sensitive manner (Dalrymple, 2011). Figure 1.1 below is an informatics pyramid highlighting the intricate relationship between data, information and knowledge that all the above-mentioned fields or paradigms subscribe to.

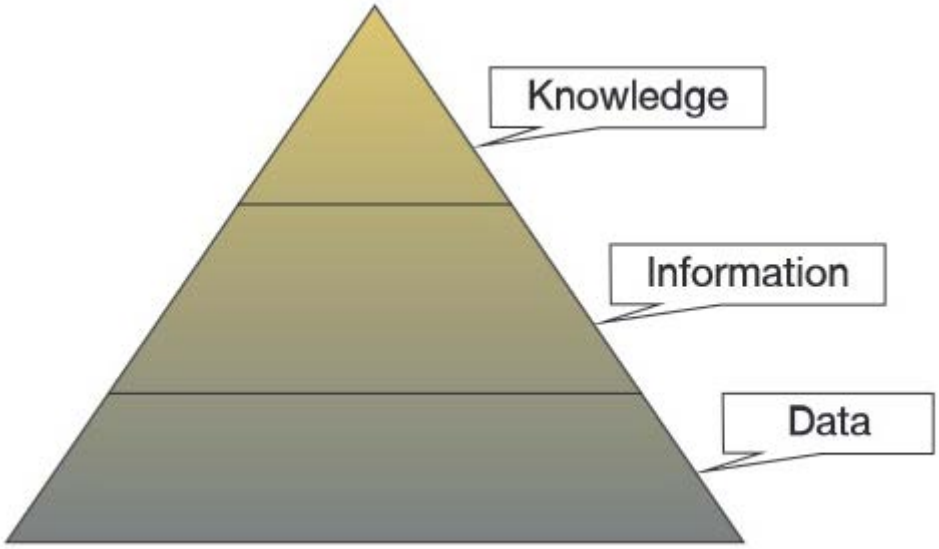


Figure 1.1: Informatics pyramid (Dalrymple, 2011)

The Electronic Health Record (EHR) is a systematic collection of electronic health data about individual patients or populations. It is capable of being shared across healthcare providers in a certain state or throughout a country (Gunter & Terry, 2005). Health records may include a range of data including general medical records, patient examinations, patient treatments, medical history, allergies, immunization status, laboratory results, radiology images, and some useful information for examination. This rich information dataset may help researchers in examining and diagnosing diseases using computer techniques. Using EHRs can help in reducing the cost of legacy systems, improving the quality of care, and increasing the mobility or sharing of records.

The existence of EHRs have encouraged researchers to the idea of electronic healthcare system where the components of the legacy healthcare systems come together and

electronically share and transfer patient information across the public infrastructure across a country.

The Arab World, and specifically Saudi Arabia, is moving fast toward electronic health care information systems. This movement will lead to the production of huge health-related information and data that can be a great asset in increasing overall quality of healthcare and wellbeing of the people, if used judiciously. The aim of this current work is to investigate the aspects of utilising health data using novel machine learning and data mining techniques to better diagnose and treat specific heart diseases.

Heart disease or cardiovascular disease is the class of diseases that involve the heart or blood vessels (arteries and veins). Today, most countries face high and increasing rates of heart disease and it has become a leading cause of debilitation and death worldwide in men and women over age sixty-five and today in many countries heart disease is viewed as a "second epidemic," replacing infectious diseases as the leading cause of death (Gale Nutrition Encyclopedia, 2011).
Early diagnosis of heart diseases can help reduce the rate of mortality. One of the ways to diagnose heart diseases is by using echocardiography. Echocardiography, or echo, is a painless test that uses sound waves to create pictures of the heart. The test gives information about the size and shape of the heart and how well the heart chambers and valves are working. Echo also can be done to detect heart problems in infants and children (NCBI, 2014).

The analysis of echo data by experts is time consuming and this is in concomitant with the shortage of experts possessing knowledge on the analysis of echo data. Thus, automated methods can solve limitations of traditional diagnostic methods and provide medical knowledge for diagnoses purposes (DAMTEW, 2011). To solve this and many other problems in the health sector related to heart diseases diagnosis, one must come up with a way to extract hidden information from enormous datasets that are collected in the past. Data mining and machine learning can be a solution by generating rules from those enormous datasets which can be used in echo readings.

Data mining has recently become one of the most progressive and promising fields for the extraction and manipulation of data to produce useful information (Payam Mohebbi, 2016). Often the process of Machine Learning is like that of data mining. Both employ the same methods and most likely overlap. With a good study and protocol, they can be distinguished as machine learning that focuses on prediction, based on known properties learned from the training data within the dataset, while data mining focuses on the discovery of previously unknown properties in the data (Mitchell, 1999). The concept of Machine Learning is based on identifying unique patterns in data and extracting feasible knowledge from them (Clifton, et al., 2012). It uses extracted data to detect patterns and adjust program actions accordingly (Rouse, 2016).

Machine learning algorithms are often categorized as being supervised or unsupervised. In supervised learning, the training set contains data and the correct output of the task with that data. Supervised learning includes classification algorithms such as logic regression, classification trees, support vector machines, random forests, artificial neural networks (ANNs). In unsupervised learning, the training set contains data but no labels or headings to categorize the data. Therefore, under this classification algorithm, e.g., K-means (Hartigan, 1979), the solution must not only be provided, but also the categorization based on the similarity of the data as well. Unsupervised learning includes clustering algorithms, k-means clustering algorithm, Distances and Normalization and Self-Organising Maps (Louridas & Ebert, 2016).

Deep Learning is a set of Machine Learning algorithms which has one or more hidden layers in Neural Networks. The hidden layers do not provide direct functions that map data for classification, but rather afford useful information to classify a data/set of data to a cluster, and extracts features and aspects from the input space. Deep Learning is a promising avenue of research in the field of machine learning and data mining. Deep Learning solutions have yielded outstanding results in different machine learning applications, including speech recognition, computer vision, and natural language processing (Najafabadi, et al., 2015). Figure 1.2 shows the relationship of deep learning with machine learning and Artificial intelligence. Artificial intelligence uses machine

learning algorithms for decision making and one of the technique of machine learning is the representation learning. Deep learning is part of the representation learning, which uses more than one hidden layers to complete its neural networking learning and classification procedures.

Employing data mining and machine learning in the health sector has been rapidly gaining high importance around the world. The importance of health informatics has risen significantly in the recent years due to the need for a secured and efficient management of medical data. Health informatics also facilitates proper management, analysis and use of health-related data for the purpose of more efficient healthcare delivery. It also plays a vital role in helping physicians to identify effective treatments, and patients to receive better and more affordable health services (Zarb, 2016). With regards to health informatics, a viable application of data classification is to establish if an admitted patient in a hospital is improving or deteriorating.

1.2. Research Motivation

Heart Disease has become a common disease around the world. Most countries face high and increasing rates of heart disease or Cardiovascular Disease. Even though modern

medicine is generating huge amount of data every day, little has been done to use this available data to solve the challenges that face a successful interpretation of echocardiography examination results.

Discovering the disease in its early stages may reduce the severity of heart disease. Computing technologies and machine learning tools can be used to assist physicians in diagnosing and predicting the disease so they can provide the necessary treatment and prevent the impact, including the possibility of death.

Predicting the outcome of a disease is one of the most interesting and challenging tasks in which to develop data mining applications. Automatic classification systems with a high accuracy of heart diseases screening and classification will help in decreasing the workload for healthcare personnel in the process of the early detection of heart diseases. Therefore, this thesis intends to utilize the latest technologies, focusing on deep-learning Neural Networks approach, in data mining science to produce models and methods that can assist physicians in the process of detection of heart diseases. Moreover, the work of this research will perform an empirical evaluation of deep learning approaches, and discuss the conclusions from these findings. Also, this work introduces a new dataset of Coronary Artery Disease, which would be beneficial to heart diseases researchers and practitioners, especially in the screening field.

1.3. Research Aim and Objectives

1.3.1. Aim

The aim of this thesis is to propose a novel technique for the classification of heart disease (i.e., Coronary Artery Disease – CAD) data by using Neural Networks techniques, and to evaluate the performance and accuracy measures by the comparison of different classification models' results.

This research is designed to make use of computerized available patients' data and pertinent health information to develop a decision support system for clinical purposes.

This work is intending to develop some framework that can help physicians in diagnosing heart diseases, by employing machine learning techniques. This can help in the early detection of the disease, and ultimately facilitate treatment plans to save patient life. Furthermore, this work aims to evaluate deep learning methods in the medical domain and to examine whether deep learning methods outperform state of the art approaches. Figure 1.3 shows the approaches that will be investigated in this research.

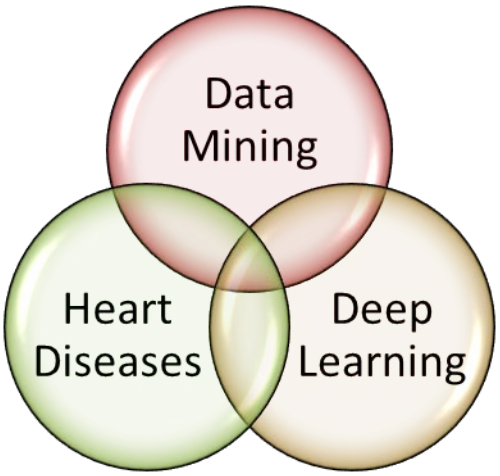


Figure 1.3: Approaches of the research

Basically, the proposed algorithm for classification of heart diseases (CAD) consists of three types of development. Firstly, this thesis will present the development of a basic system for the classification of Coronary Artery Disease (CAD) using Radial Basis Function (RBF) neural networks. The research will then focus on the detection of CAD and other heart diseases types (e.g. Arrhythmia and Sleep Apnea) using deep learning techniques based on two different approaches, which are the Deep Neural Networks and Deep RBF Neural Networks. Figure 1.4 shows the scope of the research.

1.3.2. Objectives

To achieve the above stated aim, the following are the objectives of the research:

- To identify key patterns or features from the datasets, using effective feature selection techniques to highlight key features and obtain higher classification accuracy

- To investigate important concerns regarding missing values in a dataset, and how it can impact the accuracy of classification on the performance of machine learning algorithms.
- Design a data classification model based on Neural Networks for heart disease's data. Verify the performance and accuracy of this classification model by using different standard datasets, hence to obtain a comparison of the proposed system to the existing classification models.

Therefore, this thesis proposes new methods for constructing missing feature values, investigates feature selection techniques and develops a novel data mining algorithms for providing automatic computer-aided and decision support systems for accurate/correct heart diseases diagnoses.

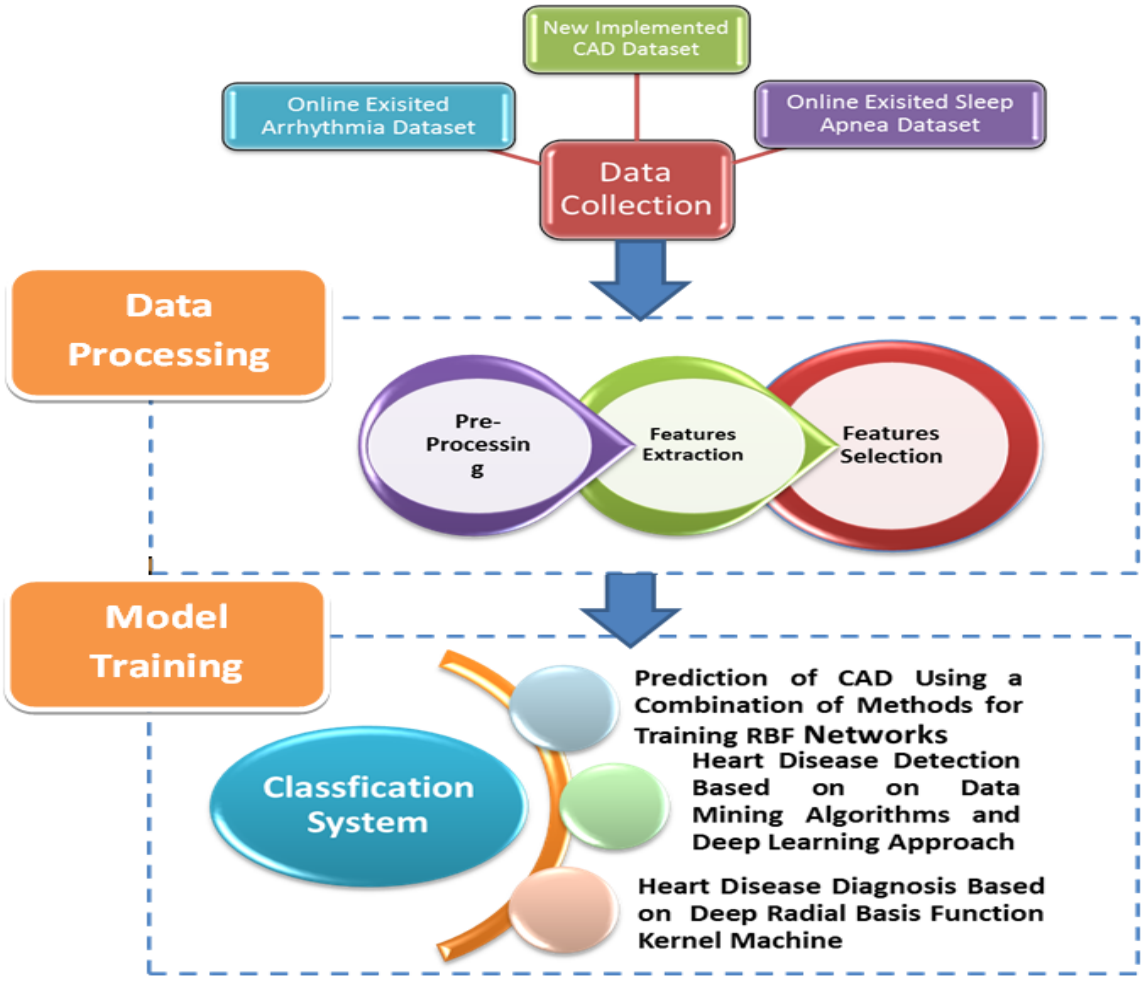


Figure 1.4: overall approaches

1.4. Research Contributions

1.4.1 Key contributions

- Develop a clinical support system for classification of heart disease

This research aims to contribute to the health informatics evolution, through analysing large set of data obtained from electronic health systems by employing data mining and machine learning algorithms. Utilizing such data assists in proposing novel automated approaches for diagnosing of heart diseases based on previous history obtained from patients.

- Combination of classification techniques with improved results

One of the major contributions of this work is to combine classification techniques for improvement of heart disease classification. This is to close any gaps between the use of emerging classification techniques and the existing methods for detection of heart diseases. Since the previous studies have provided varying results for classification of existence or absence of a disease, with use of different Machine learning methods. This work contributes by finding and combining different classical and emerging deep learning techniques to provide the optimal results. As regards diagnosis approach, this research work proposes three approaches for binary classification of heart diseases based on data mining algorithms.

– The first approach presents a novel method for Coronary Artery Disease (CAD) diagnosis using a combination of methods for training Radial Basis Function (RBF) networks. Two methods are used for training process. The first one involves usage of Extended Kalman Filter for the learning procedure, which employs different training algorithms, such as the Quasi Newton and Scaled Conjugate Gradient (SCG). The second prediction method is PSOGSA - the Gravitational Search Algorithm (GSA) - which is a novel heuristic optimization method based on the law of gravity and mass interactions.

– The second approach utilises deep learning techniques to design a Deep Neural Network with Rectified Linear unit (RELU) activation, to diagnose different types of heart diseases. In addition, some of the most well-known data mining algorithms that were used in the existing literature to diagnose heart diseases were explored and evaluated. The novel aspect of the research is that a prediction model is constructed using Logistic Regression, K-Nearest Neighbours (KNN), Support Vector Machine (SVM) and Naïve Bayes Classifiers, and with use of some repository data sets and real patient's data obtained from King Saud Medical complex, classification of Apnea-ECG, CAD and Arrhythmias is performed. This work is unique in the sense that no other research has combined these approaches for heart disease detection, and have obtained results of up to 95% precision.

- The third approach combines the first approach and the second concept to develop an automatic heart disease detection system using Multilayer RBF kernel machines. Deep neural networks with rectified linear unit activation are used in this work and based on Multilayer RBF Kernels. Three different kernels are used; i.e. Kernel PCA, unsupervised regression and supervised regression. The novelty of the work is to transparently combine these different methods to achieve optimal classification performance.

- Different sets of experiments were performed to evaluate the proposed classification schemes on the different datasets currently in use. Based on those experiments, the best classification technique depends on the application and on the classifiers characteristics. Moreover, there is no best classifier that effectively fits all datasets.

- Handling features selection in clinical practice

One final key contribution is to use the feature selection methods to improve the performance of the proposed system. The data sets especially for patients with heart

disease can have multiple attributes and not all may necessarily contribute towards the existence or absence of the disease. Also, not all contributing features have equal importance for training the neural network to obtain correct results. The features selection is used to obtain the most relevant attributes that participate positively in the classification process. Regarding this aspect, different features selections methods were applied. Based on the foregoing, the findings of this research confirmed that featured selection methods can improve the performance of the classification models. The finding of the research contributes to the fact that not a single feature selection technique performs best for all data sets, but different schemes can be tried to get best performance. Besides, the results provide enough experimental proof that the feature selection improve the performance of the classifier. Therefore, before applying a specific feature selection technique, the nature of the data sets and the characteristics of the classification model is kept in consideration to obtain better outcomes.

1.4.2 Other contributions

- An approach for tackling data problems identified in previous research and literature. Concerning missing features problem, this study used an existing approach for constructing missing features values based on the Matrix Completion technique (Candes, 2012). The proven and proposed method showed good improvements in classification accuracy as compared to the original dataset which contained some missing features values. The observed improvements for CAD classification using quasi-newton training and R4RlogR activation function, yielded a 2% accuracy improvement with dataset when matrix completion was applied. On Average 0.5 to 1.0% improvement was observed for a combination of different training and activation methods with matrix completion.
- Also, this research utilises a dataset based on CAD patients with unique parameters to help detect the disease effectively and have minimum error in the obtained results. The data set in this regard is unique with up to 50 different patient attributes, and as such could be used by any future researcher for further exploration of pertinent heart disease conditions.

1.5. Thesis Organization

The thesis is organised into eight chapters, each focusing on different features of the research work. The following is a summary of the contents of each chapter.

Chapter 1 provides an overview of heart diseases detection and a more detailed investigation of the problems. The research aims and objectives of this study are also presented. In addition, the motivations which have led to this research and the contributions of the thesis are presented in this introductory chapter.

Chapter 2 describes the available information on Heart Diseases from clinical view; particularly the three common types; coronary artery disease (CAD), Arrhythmia and sleep apnea. As well, it investigates classification techniques focusing on Artificial Neural Networks and deep learning approach.

Chapter 3 considers the state-of-the-art algorithms proposed in the field of heart diseases, neural networks and deep learning. In addition to other classification methods proposed for heart diseases diagnosis.

Chapter 4 discusses the research methodology used, including the process of data collection for this study. The research design, as a guide for planning the research development, is also presented. The experimental datasets, which consist of the existing datasets and a novel developed dataset, are presented in Chapter 3 in greater detail. Finally, the chapter provides detailed information on the data processing procedures adopted in the research to explore the collected data.

Chapter 5 explains the development of automatic system to predict Coronary Artery Disease based on RBF networks. The proposed system presents a combination of different techniques, such as the Extended Kalman Filtering and Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA) for radial basis functions

training. The evaluations of the developed systems are also presented, where it presents the efficiency and the validity of the proposed approach.

Chapter 6 describes the scheme proposed that is based on a deep learning approach to address the automatic detection and classification of heart diseases. It presents the different components of the approach that were derived by drawing upon the results of this research study. The chapter also provides information pertaining to the approach validation. It also presents the application of some of the traditional data mining algorithms to the used datasets to check its soundness.

Chapter 7 presents the development of the deep RBF Kernel Machine framework. It also explores the different approaches of features selection and dimensionality reduction to train RBF based on Multilayer Kernel Learning. It also provides the available information pertaining to the effectiveness of the framework validation. In addition, some discussions on the findings of this study are presented.

Chapter 8 summarises the accomplishments of the research work. It concludes the contents of the thesis and provides information regarding the research contributions, which have benefited several areas. It also highlights some recommendations for future research work.

2 : Background: Heart Diseases and Artificial Neural Networks

2.1. Overview

One of the most prevalent diseases that people suffer from is heart disease. Among various life- threatening diseases, heart diseases have received a great deal of attention in medical research. The World Health Organization reported that heart disease is the first leading cause of death in high and low income countries (WHO, 2007). The Economic and Social Commission of Asia and the Pacific reported that in one fifth of Asian countries, most of the diseases from which lives are lost, such as cancer, diabetes and cardiovascular diseases are non-communicable (ESCAP, 2010). The Australian Bureau of Statistics reported that 33.7% of all deaths in Australia are due to heart and circulatory system diseases which are the leading cause of death (Australian Bureau of Statistics, 2010). The American Heart Association also reported that heart disease is the number one cause of death in the United States, killing over 375,000 people a year (American Heart Association, 2015).

Statistical analysis has identified the risk factors associated with heart disease to be age, blood pressure, smoking habit (Heller, et al., 1984), total cholesterol (Wilson, et al., 1998), diabetes (Simons, et al., 2003), hypertension, family history of heart disease (Din & Rabbi, 2006), obesity, and lack of physical activity (Shahwan-Akl, 2010). Knowledge of the risk factors associated with heart disease helps health care professionals to identify patients with a high risk of having heart disease.

Coronary heart disease (CAD) is a common type of heart disease. This condition results from a build-up of plaque on the inside of the arteries, which reduces blood flow to the heart and increases the risk of a heart attack and other heart complications. Other forms of heart disease include: irregular heartbeat (arrhythmias), congenital heart defects, weak heart muscles (cardiomyopathy), heart valve problems, heart infections and cardiovascular disease (Higuera, 2014).

This chapter provides the background for each of the main components involved in the research. It considers the available information on heart diseases; particularly the three major types; CAD, arrhythmia and sleep apnea. This chapter comprises five main components which are CAD, arrhythmia, sleep apnea, neural networks and deep learning methods, classification of heart disease using deep learning and neural networks, and classification of heart disease using the traditional data mining methods.

2.2. Heart Diseases

2.2.1. Coronary Artery Disease

The Prevalence of CAD is increasing across the globe, and costing government and other healthcare stakeholders a lot of money to manage (Kuulasmaa, et al., 2000). In addition to the financial pressure it imposes, CAD frequently results into mortality and has been labelled as one of the world's most prevalent causes of death (WHO, 2016); (Genders, et al., 2012).

CAD is caused by the build-up of plaque in the walls of the arteries that supply blood to the heart (coronary arteries) and other areas of the thoracic region of the body (BBC, 2013). This Plaque consists of cholesterol and other substances that are deposited in the arterial wall (National Heart, 2016). CAD is reported to account for 7 million deaths over the world per annum.

The Prevalence of CAD is increasing across the globe, and management of the disease is costing government and other healthcare stakeholders a lot of money (Kuulasmaa, et al., 2000). In addition to the financial pressure imposed, CAD frequently results into mortality and has been labelled as one of the world's most prevalent causes of death (WHO, 2016); (Genders, et al., 2012). Figure 2.1 shows the difference between a normal coronary artery and one with CAD.

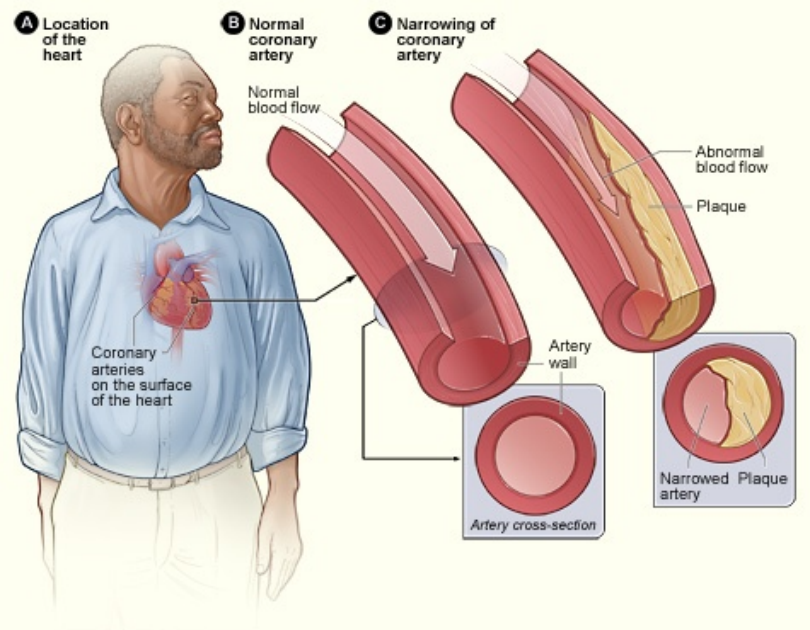


Figure 2.1: Figure (A) shows the location of the heart in the body. Figure (B) shows a normal coronary artery with normal blood flow. The inset image shows a cross-section of a normal coronary artery. Figure (C) shows a coronary artery narrowed by plaque. The inset image shows a cross-section of the plaque-narrowed artery.

Over time, CAD can weaken the heart muscle and lead to heart failure; an often-acute condition where the heart is unable to pump blood the way that it should. An initial sign of this is irregular heartbeat/heart rhythm which is called arrhythmia (National Heart, 2016) (BBC, 2013).

A host of factors are used in the diagnosis of CAD, and these include patient's blood pressure, cholesterol, sugar levels, high BMI (overweight/obesity), physical inactivity, unhealthy eating, and smoking (National Heart, 2016). Other factors such as age, gender, and family history of heart disease are also likely risk factors for CAD (Foundation, 2015) (National Heart, 2016).

Furthermore, if a patient is showing symptoms or at high risk of heart, a doctor (most often a highly trained cardiologist) uses several tests to diagnose CAD and prescribe an appropriate treatment regime (NHS, 2015). This process is both resource and labour intensive, making the diagnosis and treatment very expensive.

“Coronary artery disease begins in childhood, so that by the teenage years, there is evidence that plaques that will stay with us for life, are formed in most people,” said Fisher, who is former editor of the American Heart Association journal, ATVB. “Preventive measures instituted early are thought to have greater lifetime benefits. Healthy lifestyle will delay the progression of CAD, and there is hope that CAD can be regressed before it causes CHD”.

Living a healthy lifestyle that incorporates good nutrition, weight management and getting plenty of physical activity can play a big role in avoiding CAD.

Risk Factors

As per the study conducted by the American heart society (American Heart Association, 2016), the following are likely risk factors for CAD condition:

- **Age**

  The majority of people who die of coronary heart disease are 65 or older. At ages, greater than 65, women who have heart attacks are more likely than men to die from them within a few weeks.

- **Gender**

  Men have a greater risk of heart attack than women, and they have attacks earlier in life. In women, even after the menopause, the increase in the death rate from heart disease is not as great as men.

- **Heredity (Including Race)**

  Where children have parents with heart disease, they are more likely to develop it themselves. Most people with a strong family history of heart disease have one or more other risk factors likely to cause the disease.

- **Tobacco smoke**

  The risk of a smoker developing CHD is much higher than that for non-smokers. Cigarette smoking is a powerful independent risk factor for sudden cardiac death in patients with CHD. Cigarette smoking also acts with other risk factors to greatly increase the risk of CHD. Exposure to other people's smoke increases the risk of heart disease even for non-smokers.

- **High blood cholesterol**

As blood cholesterol rises, so does the risk of CHD. When other risk factors (such as high blood pressure and tobacco smoke) are present, this risk increases even more. A person's cholesterol level is also affected by age, sex, heredity and diet.

- **High blood pressure**

  High blood pressure increases the heart's workload, causing the heart muscle to thicken and become stiffer. This stiffening of the heart muscle is not normal, and prevents the heart working properly. It also increases the risk of stroke, heart attack, kidney failure and congestive heart failure.

- **Physical inactivity**

  An inactive lifestyle is a risk factor for coronary heart disease. Regular, moderate-to-vigorous physical activity helps reduce the risk of heart and blood vessel disease. Even moderate-intensity activities are helpful if done regularly and over a long term. Physical activity can help control blood cholesterol, diabetes and obesity, and with some people, lower blood pressure.

- **Obesity and overweight**

  People who have excess body fat — especially at the waist — are more likely to develop heart disease and stroke even if no other risk factors are present. Overweight and obese adults with risk factors for cardiovascular disease such as high blood pressure, high cholesterol, or high blood sugar can make lifestyle changes to lose weight and produce clinically meaningful reductions in triglycerides, blood glucose, HbA1c, and the risk of developing Type 2 diabetes. Many people may have difficulty losing weight, but a sustained weight loss of 3 to 5 per cent body weight may lead to clinically meaningful reductions in some risk factors. Above 5 per cent weight loss can lower blood pressure, cholesterol, and blood glucose levels.

- **Diabetes mellitus**

  Diabetes seriously increases the risk of developing cardiovascular disease. Even when glucose levels are under control, diabetes increases the risk of heart disease and stroke, but the risks are even greater if blood sugar is not well controlled. At least 68 per cent of people greater than 65 years of age with diabetes die of some form of heart disease and 16 per cent die of stroke. Persons with diabetes who are

obese or overweight should make lifestyle changes (e.g., eat better, get regular physical activity, lose weight) to help manage blood sugar levels.

2.2.2. Arrhythmia

Arrhythmia is a problem with the rate or rhythm of the heartbeat. It means that the heart beats too quickly, too slowly, or with an irregular pattern. When the heart beats faster than normal, it is called tachycardia. When the heart beats too slowly, it is called bradycardia. The most common type of arrhythmia is atrial fibrillation, which causes both an irregular and fast heartbeat.

Many factors can affect the heart's rhythm, such as having a heart attack, smoking, congenital heart defects, and stress. Some substances or medicines may also cause arrhythmia.

Symptoms of arrhythmias include: fast or slow heartbeat, skipping beats, light-headedness or dizziness, chest pain, shortness of breath and sweating (Medicine, 2016).

There are various types of arrhythmias and each type is associated with a pattern, and as such, it is possible to identify and classify its type. The arrhythmias can be classified into two major categories. The first category consists of arrhythmias formed by a single irregular heartbeat, herein called morphological arrhythmia. The other category consists of arrhythmias formed by a set of irregular heartbeats, herein called rhythmic arrhythmias. Both types of arrhythmias produce alterations in the morphology or wave frequency of the heartbeat and these can be identified by an ECG examination. Figure 2.2 shows the most common types of arrhythmias.

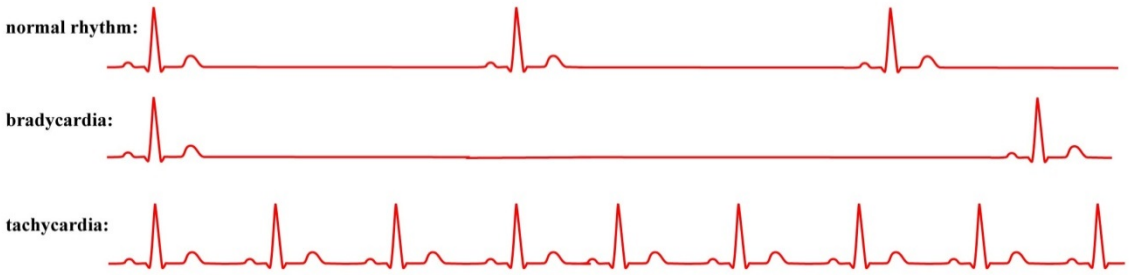


Figure 2.2: Types of Arrhythmias

The process of identifying and classifying arrhythmias can be very troublesome for a human being because sometimes it is necessary to analyse each heartbeat of the ECG records, acquired by the patient wearing a Holter monitor over several hours, or even days. In addition, there is the possibility of human error by the person performing the ECG records analysis, owing to fatigue. An alternative is to use computational techniques for automatic classification (Luz', et al., 2015).

Arrhythmias Causes (National Heart Foundation of Australia, 2016)

Arrhythmias are caused by a problem in the electrical system of the heart. Some causes of arrhythmias include:

- **Irritable heart cells**

  Sometimes heart cells begin to malfunction and start sending out abnormal electrical signals. Signals from these malfunctioning heart cells interfere with the proper signals from the natural pacemaker within the heart. This ‘confuses’ the heart causing an irregular heartbeat
- **Blocked signals**

  The electrical signals that tell the heart to beat may get ‘blocked’. This makes the heart beat very slowly.
- **Abnormal pathway**

  Sometimes the electrical signals start at the right place and time, but get interrupted and misdirected so they don’t follow the right path through the heart and cause an arrhythmia.
- **Medicines and stimulants**

In some cases, medicines and other substances, such as caffeine, nicotine and alcohol, can cause an arrhythmia.

Arrhythmias Types (National Heart Foundation of Australia, 2016)

- **Bradycardia (i.e. slow heartbeat)**

  This term is used to describe when the heart beats too slowly; generally, less than 60 beats per minute. It is serious when the heart beats so slowly that it can't pump enough blood to meet the needs of the body. Untreated, bradycardia can cause excessive tiredness, dizziness, light-headedness or fainting, because not enough blood reaches the brain. A slow heartbeat may be normal, and can be associated with improved physical fitness.

- **Sick sinus syndrome**

  This term describes when the natural pacemaker in the heart malfunctions and 'fires' too slowly, telling the heart to beat slowly. It can be caused by age or by fatty tissue in the arteries that take blood to the heart.

- **Heart block**

  When the signal passing from the collecting chambers (atria) to the pumping chambers (ventricles) of the heart is delayed or blocked this is called heart block. It is uncommon but can be serious. Symptoms can be mild or severe, depending on the location and seriousness of the blockage.

- **Tachycardia (a fast heartbeat)**

  Tachycardia is when the heart beats too fast, generally more than 100 beats per minute. Some forms of tachycardia are easily treated and not serious, while others can be life-threatening.

- **Supraventricular tachycardia**

  A rapid heartbeat that starts in the collecting chambers of the heart, the atria, or the electrical pathway from the atria is called "supraventricular tachycardia" (SVT). Common types of SVT are atrial flutter and atrial fibrillation.

- **Atrial flutter**

  An extra or early electrical signal which travels around the atria in a circle instead of along the normal signal pathway is called "atrial flutter". This

‘overstimulation’ causes the atria to contract quickly or ‘flutter’ at a much higher rate than normal. Most of this fluttering is blocked out by the electrical pathway from the atria to prevent the pumping chambers of the heart, the ventricles, from beating too fast. Atrial flutter is usually not life-threatening but can still cause chest pain, faintness or more serious heart problems.

- **Atrial fibrillation**

  The most common form of SVT is “atrial fibrillation”. This is when ‘waves’ of uncontrolled electrical signals, rather than the normal regulated signals, travel through the atria from the sinus node. These uncontrolled signals cause muscle fibres in the atria to contract out of time with each other, so that the atria ‘quiver’ or ‘fibrillate’. Some of this abnormal electrical activity reaches the ventricles, causing a rapid and irregular heartbeat. When the heart is in atrial fibrillation, it does not pump regularly or work as well as it should. Atrial fibrillation can cause a ‘fluttering’ heartbeat, an irregular pulse, chest pain or tightness, weakness and dizziness. Atrial fibrillation can also increase the risk of stroke, because blood trapped in the atria can clot. These clots may break loose from the heart, enter the bloodstream and travel to the brain, causing a stroke.

- **Paroxysmal supraventricular tachycardia**

  A ‘short circuit’ caused by an extra electrical connection or pathway in heart making the heart prone to episodes of sudden regular rapid heartbeats that may last for minutes or even hours is called “paroxysmal supraventricular tachycardia” (PSVT). Although these episodes may be frightening, they are rarely dangerous and can be very effectively treated.

- **Wolff-Parkinson-White syndrome**

  An extra or abnormal electrical pathway connecting the atria to the ventricles, causing attacks of SVT is called “Wolff-Parkinson-White syndrome”.

- **Ventricular tachycardia**

  When the ventricles beat too fast, called “ventricular tachycardia”, it is potentially very dangerous. Ventricular tachycardia that becomes so severe that the ventricles can’t pump effectively can lead to ventricular fibrillation. Ventricular fibrillation is when the electrical signal that should trigger the heartbeat splits away in

uncontrolled ‘waves’ around the ventricles. This life-threatening situation must be corrected immediately.

2.2.3. Sleep Apnea

Sleep apnea is a common disorder in which a person’s breathing may have one or more pauses during sleep. These pauses may last from a few seconds to minutes, and may occur hundreds of times during the night. If the obstruction to breathing is total and lasts for ten or more seconds, then the episode is called apnea. During the sleep apnea, the brain and the rest of the body may not get enough oxygen. Thus, the quality of sleep is poor, which makes the patient tired during the day. (Derrer, 2014). In addition, it is considered a risk factor for morbidity and mortality due to its long-term effect on the cardiovascular system (Caples, 2007).

Types of sleep apnea (Smith, et al., 2016). Figure 2.3 shows these types.

- **Obstructive sleep apnea (OSA)** is the most common type of sleep apnea. It occurs when the soft tissue in the back of the throat relaxes during sleep and blocks the airway, often causing loud snoring.
- **Central sleep apnea (CSA)** is a much less common type of sleep apnea that involves the central nervous system, occurring when the brain fails to signal the muscles that control breathing. People with central sleep apnea seldom snore.
- **Mix sleep apnea** (MIX) is a combination of obstructive sleep apnea and central sleep apnea.

Risk Factors (American Academy of Sleep Medicine, 2016)

The most common risk factors that may cause sleep apnea are listed below:

- Excess weight
- Large neck size
- Middle age
- Male gender
- Hypertension.
- Family history

The link between sleep apnea and heart disease has been studied for several years. Several studies showed that obstructive sleep apnea is associated with an increased risk of cardiovascular morbidity and mortality (Duna, et al., 2006). Apoor Gami & Neil Sanghvi (2013) said that "The presence and severity of sleep apnea are associated with a significantly increased risk of sudden cardiac death". There are several causes behind the association between sleep apnea and heart attacks. Sleep apnea can cause patient to stop breathing several times a night. When this happens, blood oxygen levels go down, which can cause heart rhythm to flutter. This is a heart complication that also occurs at the time of sudden cardiac death. (Eric Cohen, 2014)

Diagnose of Sleep Apnea

Traditionally, sleep-related breathing disorders are diagnosed by visual observation of polysomnography (PSG) signals. PSG is a sleep test that is performed at special laboratories. It consists of recording various signals including the breath airflow, respiratory movement, oxygen saturation, body position, Electroencephalogram (EEG), Electrooculogram (EOG), Electromyogram (EMG), and electrocardiogram (ECG) (Chazal, et al., 2004).

Even though PSG become the standard diagnostic tool for sleep disorder cases, there are some problems related to its implementation which make it expensive and time consuming. For example, its execution requires the patient to sleep in a sleep laboratory

for one or two nights, in the presence of technicians. Furthermore, patients must maintain a position throughout the night with special equipment attached to their bodies during measurement. Hence these limitations put a barrier to PSG acceptance and reduced its diagnostic power. Therefore, the need for a simpler alternative detection method has been arising. Automated methods that use artificial intelligence algorithms can solve PSG problems, since it is easier and faster to detect OSA cases. Furthermore, due to the increasing interest in wearable and portable sleep quality monitoring systems for home care which require the use of a minimum number of channels (i.e. fewer leads attached to the patient), OSA detection based on single-lead ECG is gaining keen interest in the sleep research community. In this light, there have been several algorithms proposed to tackle the problem of automatic OSA detection using ECG patterns obtained during PSG studies using machine learning techniques.

Apnea Challenge

ECG recording is one of the simpler and efficient technologies in sleep disorders detection. In 2000, Computers in Cardiology (CINC) and PhysioNet organized a competition to highlight the potential use of the ECG signals in diagnosing sleep apnea. They hosted a challenge where various research teams introduced several different methodologies for sleep apnea detection using the ECG.

PhysioNet provided free access to the database of ECG recordings and an automatic web-based scoring program. The recordings were arranged in three classes, as follows:

- Class A (apnea): recordings in this class contain at least one hour with an apnea index of 10 or more, and at least 100 minutes with apnea during the recording. The learning and test sets each contain 20 class A recordings;

- Class B (borderline): recordings in class B contain at least one hour with an apnea index of 5 or more, and between 5 and 99 minutes with apnea during the recording. The learning and test sets each contain 5 class B recordings; and

- Class C (control): recordings in class C contain fewer than 5 minutes with apnea during the recording. The learning and test sets each contain 10 class C recordings.

The competition consisted of two challenges. The first challenge was to identify the recordings in the test set with sleep apnea (class A) and the normal recordings (class C). Assignments for class B were not scored. The score was the total number of correct classifications of class A and class C, so that the maximum possible score was 30. The second challenge was to label each minute in all 35 test recordings as either containing apnea (A) or not (N). In this challenge, all 35 test recordings were scored (PhysioNet, 2012).

The literature above presents different heart diseases such as Apnea, Arrhythmias and CAD, and affords a critical review of work done on the identification of these diseases using computer aided and classification methods. To further extend on this review, in next section, basics on different classification techniques are presented, with relevant insight on how these classification methods have been used in detection of heart diseases.

2.3. Classification Techniques

Currently many different techniques are available for data classification. New techniques are developed either to improve the performance of a previous technique, or to find a new one which can solve the classification problem in a better way.

With a classification technique, the main purpose of the classifier is to label collected data with appropriate classes, when the class of a newly collected data is unknown, a new class can be allocated.

2.3.1. Linear Classifiers

As the name “classification” suggests, a classifier separates two or more classes from each other. In life data classification, on the feature plane, the separation line can be represented at different complexity levels mathematically.

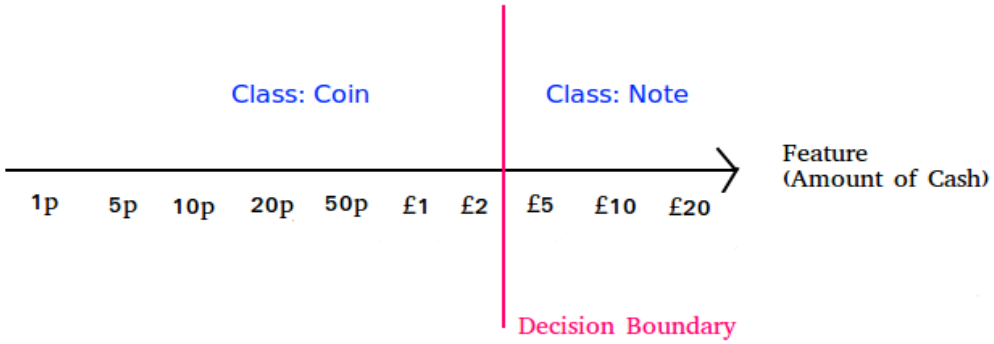


Figure 2.4: 1-Dimensional Linear Classifier

A linear classifier uses a polynomial decision boundary which is degree one at most. In Figure 2.4 above, a simple linear classifier is defined which classifies the type of cash per the amount with the classes “coin” or “note”. The separating decision boundary is found between £2 and £5 values, and it can be defined as,

$$x - 3 = 0$$

Because there is no a single coin between £2 and £5, the decision line can be defined at any point between them. In the above equation, it is chosen as $x = 3$ which is closer to £2. The distance from the closest data points to those that are at different sides of the decision boundary is important, and it is specifically used in Support Vector Machines while defining the decision boundary (Guyon, 2002).

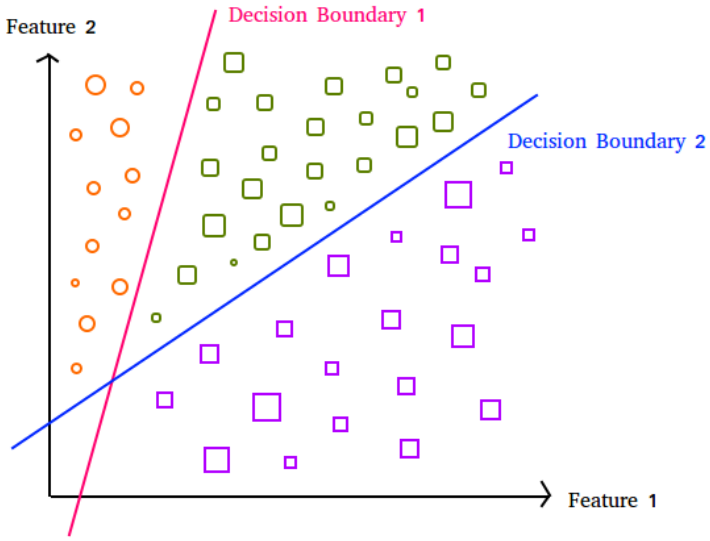


Figure 2.5: 2-Dimensional Three Classes Linear Classifier

If the polynomial rule of linear classifiers is applied, the linear classification technique can be applied to more dimensional classifiers as well (Carnegie Mellon University , 2009). In Figure 2.5, a two-dimensional classification surface with three different classes is presented. The number of dimensions of a classifier does not relate to the total number of defined classes or decision boundaries though, but the number of classes is generally supposed to be much less than the number in the training data provided.

Having too many classes of data which are less than the number of classes *ijn* the training data will result in less accurate classification because of the lack of class information. This effect can especially be seen in the k-nearest neighbours' classifier which uses the class of the closest training classes to do classification. With many classes, a point in the decision plane can give similar percentages to different classes at the same time, and this increases the error rate of the classifier. To solve this problem, powerful classifiers use feature extraction techniques to eliminate irrelevant input data features, and thus they can still generate a high percentage successful results with a small amount of training data. Since accurate and useful features are selected which have more weight in the decision of a binary classification problem, thus even with small amount of training data.

In health informatics, a linear classifier can be very useful when the record that is to be classified is well known. Because many symptoms of a sickness can be seen in the data, and can be determined by a doctor very easily, and this can be used to be able to increase the speed of the prediction process, and to decrease wrong predictions.

2.3.2. Nonlinear Classifier

With the increasing number of features of a classifier, separating different classes from each other with a straight line (i.e. by a linear decision boundary) can lead to wrong classifications.

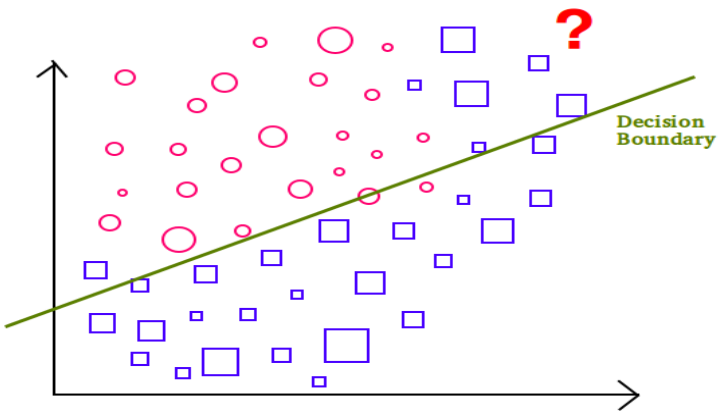


Figure 2.6: Linear Decision Boundary on 2-Dimensional Decision Plane

For linear classifiers, the polynomial of the decision boundary line is limited by a degree of one for its invariants. By removing this limit, very complex decision boundaries can be defined, and this can decrease the error rate of the designed classifier.

In Figure 2.6, two different classes are seen, and they are separated from each other by using a linear decision boundary. Because the border line of classes has a much more complex structure than a linear separator, different classes are seen at one side of the decision boundary which are seen to be wrongly classified. By defining the decision boundary in a much more complex way, the same class records can be collected at the same side of it, and therefore, a close-to-none level of error rate can be achieved.

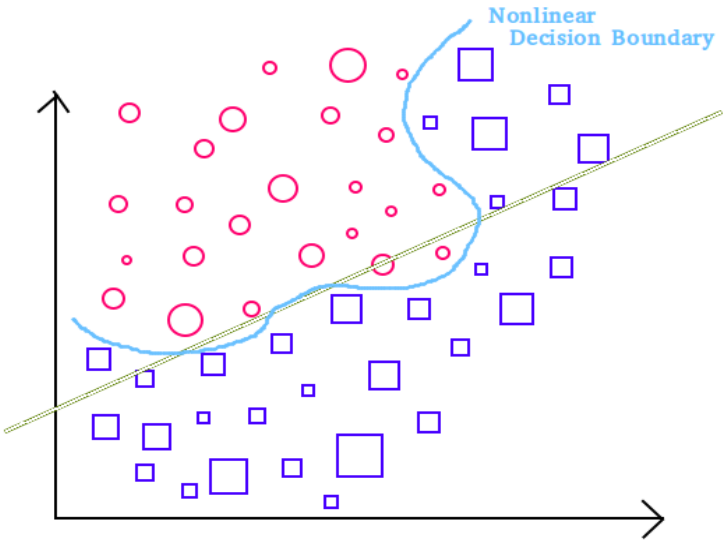


Figure 2.7: Nonlinear Decision Boundary on 2-Dimensional Decision Plane

One of the benefits of linear classifiers over nonlinear ones is the speed of training and classification of new data. A linear classifier has limitations on the degree of the polynomial, which also limits the flexibility of the decision boundary. The comparison of linear and nonlinear decision boundaries can be seen in Figure 2.7 as the nonlinear decision boundary precisely separates two different classes (Cambridge University Press, 2009).

In the automated environment, reducing the number of steps that are to be taken in accomplishing a task directly affects the performance. Therefore, linear classifiers can reach up to their optimal settings very quickly and with a minimum amount of training data. Also, although maybe not so noticeable with today's high performance computers, classification of new data requires a smaller number of mathematical operations. However, this can be very noticeable on open systems like web sites (e.g. Web search, Speech recognition) where the number of concurrent classification requests can reach up to thousands.

Compared to linear classifiers, nonlinear classifiers require a lot of training. A nonlinear decision boundary can be defined as:

Each term in the polynomial has a coefficient which determines the shape of the decision boundary. As the training process, continuous, the training algorithm changes those coefficients continuously until the classifier fails to achieve an acceptable error level.

In the training process, a computer cannot know the context of the data, and a polynomial doesn't show what the end decision boundary will be like, so the training algorithm tries to reduce the error level. One of the problems in training algorithms is that while there are better coefficients for the decision boundary, once the algorithm finds a suitable coefficient value, just to prevent an increasing error level, coefficients are set at this level and the algorithm accepts them as optimal values. This can be seen in Figure 2.8 below.

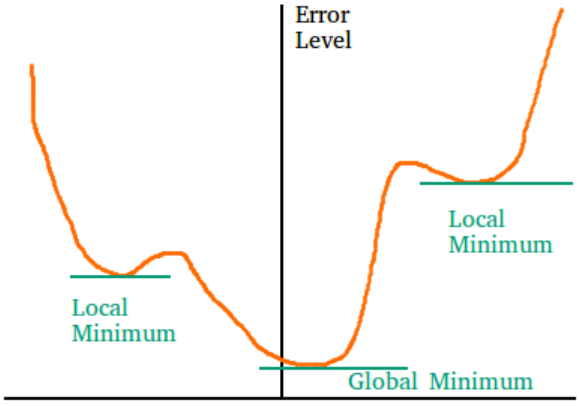


Figure 2.8: Local and Global Minimum

In the context of classification, the point of best coefficients is called the “global minimum”, and a suitable found point is called the “local minimum”. Because such a naïve[1] approach as to try the polynomial with different values continuously may require millions of iterations, and takes a long time, nonlinear classifiers use regression analysis techniques to give initial values to the coefficients of decision boundaries to decrease the total training time.

The Support Vector Machines (SVM) technique (Suykens, 1999) makes use of nonlinear classification together with an initial training approach. Compared to linear classification,

1 A naïve approach is to same equation with different values of invariants continuously until a suitable result is taken. Considering that numbers are real numbers, this process can take infinite time which is not suitable for classification purposes.

nonlinear classifiers can fit the needs of many applications and provide highly successful classification results. For this reason, in health informatics, classification of many features can be accomplished with this technique.

2.3.3. k-nearest neighbours (KNN)

Unlike linear and nonlinear classifiers, k-nearest the neighbours' algorithm follows a different technique. Most algorithms use training data to alter some internal structures as in linear and nonlinear classifiers. K-nearest neighbour's classifiers store training data in a database and for the classification of new features. For this reason, this classifier is generally seen as very simple to understand.

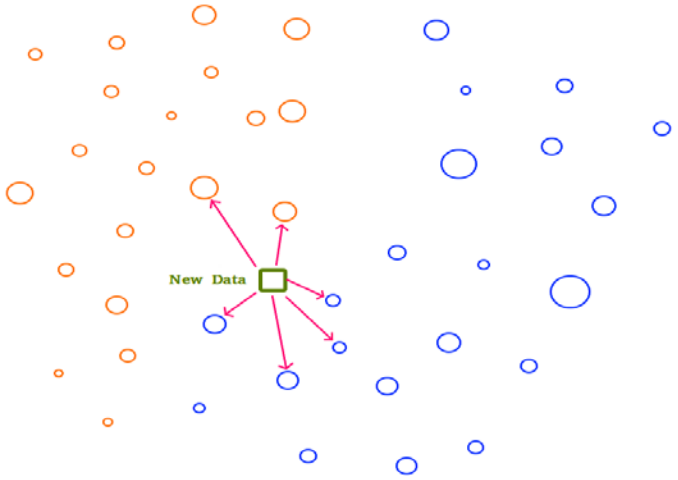


Figure 2.9: K-nearest Neighbours classification

When new data is provided to the classifier, because the training data is already available, the algorithm tries to find "k" number of training data records from database, which are those that are closest to the new data on the classification surface as shown in Figure 2.9. If there is a large amount of training data available, it is expected that the number of points in one class when compared to others will increase and that will be the result of classifier. In Figure 2.10, two circles are shown for k=3 and k=5.

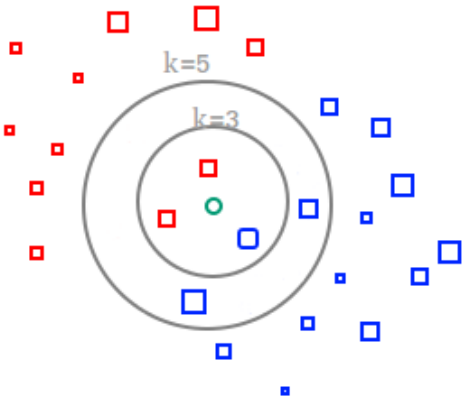


Figure 2.10: Classification of new point with different *k* values

In the smallest circle, which has 3 points in it, new data will be classified as “red”, because there are more red class points than blue ones. On the other hand, the bigger circle which consists of 5 different points has more blue class points than red ones. Therefore, if k=5 is chosen for classification, this time, any new data will be classified as blue.

This technique is applied in many applications such as medical diagnosis, pattern matching etc. However, KNN requires sizeable amount of training data to achieve higher accuracy; thus, this technique requires large storage space, and has been proven to provide better results as compared to linear classifiers.

2.3.4 SVM

Support Vector Machine – SVM (Suykens, 1999) is a non-statistical classifier supervised learning technique. To specify SVM two aspects are required, firstly the allowable classification functions and how it is decided that which one of these functions will be used for training the network. The SVM work by applying a linear function and thresholding the result. More precisely, this will specify a classifier by giving a vector w of weights, w1 through wn, where n is the number of coordinates in the future vectors. And a threshold b, given a data point x, with coordinates x_1 through x_m. They first compute the dot product, w.x. Which means that we add up w1 times x1, plus w2 times

x2, etc., up to wn time's xm. Comparison is then made on the resulting number to the threshold b. If it is larger than b, the point is put in the positive category, otherwise it is put in the negative category. This has a nice geometric interpretation. If say the feature vectors are two dimensional, instead of points with w.x exactly equal to b, it is just the line in the plane. The points with positive dot product lie on one side and the points with negative dot product lie on the other. The classifier is thus just cutting the plane into two pieces with a straight line. In three dimensions, the set of points with w.x equal to b is a two -dimensional plane, and the classifier uses it to divide the three-dimensional space into two regions.

2.3.5 Artificial Neural Networks

Artificial neural network (ANN) is an algorithm that was originally motivated by the goal of having machines that can mimic the brain. The base processing units of neural networks are known as neurons. Figure 2.11 below shows the schematic of a biological neuron.

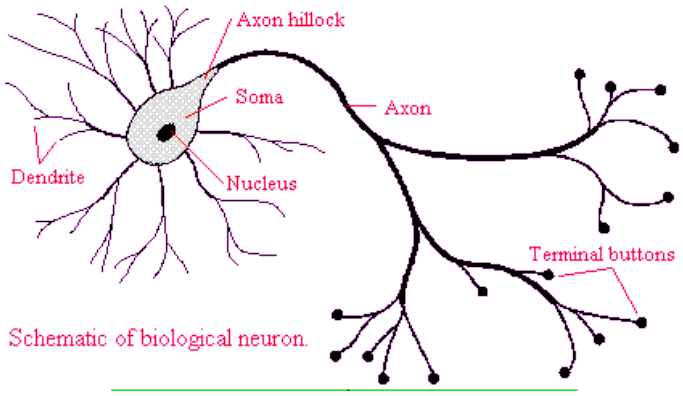


Figure 2.11: Schematic of biological neuron

A biological neuron consists of dendrite, soma, and axon parts which are indicated in the above figure. While different types of neurons have been discovered in the last century, generally, dendrites are accepted as data collectors, soma as the processing or decision making part, and the axon as the response distribution system (Gillies & Sterratt, 2012).

The artificial neural network consists of an interconnected group of neurons. They are physical cellular systems capable of obtaining/storing information, and using experiential knowledge. To be able to imitate the structure of the biological neuron in a computer system, a basic structure which is called perceptron was developed in 1943 (Corbett, et al., 1943).

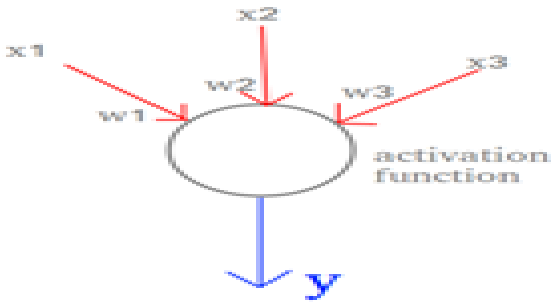


Figure 2.12: Perceptron Structure

As seen in the biological neural network, a perceptron has parts that are similar to dendrite, soma, and axon. For the decision process, a mathematical function, which is called an “activation function” is defined. Therefore , the activation function gets the sum of the inputs of the neuron as a parameter, and decides the value of output. This is illustrated by the following equation.

One perceptron can be used to create a linear classifier with two classes. Classes of this classifier are the output value. By changing the value of weight parameters, the decision boundary can be modified, and therefore this classifier can be matched to the training records. To increase the number of classes, more than one perceptron can be integrated.

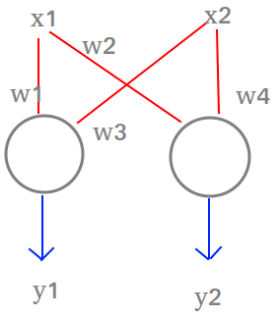


Figure 2.13: Two perceptron network

In Figure 2, two perceptrons are used to create a one layer network. There is still the same number of inputs, but the number of weight parameters are doubled. Also now, because there are two outputs, the number of different classes has increased to 4.

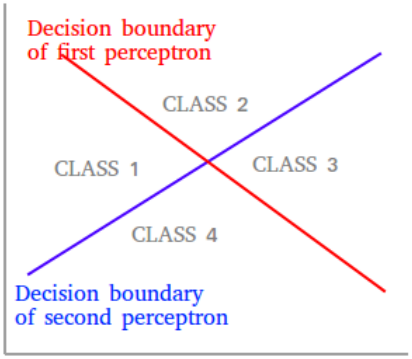


Figure 2.14: 2-Perceptrons Network Classification Surface

By adding more perceptrons, the number of different classes can be increased. Multiple decision boundaries of multiple perceptrons are seen in Figure 2.. This created structure is called an ANN.

One problem of one-layer neural networks is linear separability which is also known as the XOR problem (Kawaguchi, 2000). In this problem, when the total number of classes is two, the one layer neural network cannot define a correct decision boundary.

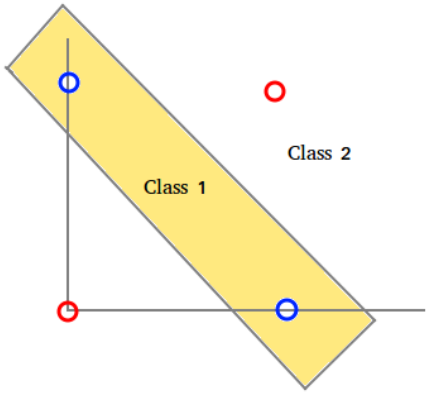


Figure 2.1512: Linear Separability

As shown in Figure 2.1512, there is no way to separate two classes from each other with a linear line with only one perceptron. To be able to solve this problem, neural networks with more than one layer are created. An illustration of a multilayer perceptron network is seen in Figure 2. below.

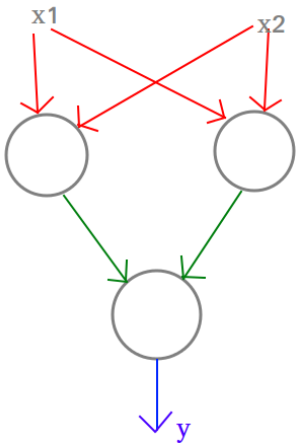


Figure 2.16: Multilayer Perceptron Network

With multilayer networks, nonlinear classification becomes possible, and therefore the XOR problem can be solved. Unfortunately, the more layers are added to network, it becomes more powerful, but it also gets computationally hard to train the network. This problem can be better solved by using different activation functions and modifying the network a little to match the needs.

As in the case of the human brain, the ANN's knowledge comes from examples that that are encountered during use. In the human neural system, the learning process includes the

modifications to the synaptic connections between the neurons. In a similar way, ANNs adjust their structure based on output and input information that flows through the network during the learning phase.

The data processing procedure in any typical neural network has two major steps: the "learning" step and the "application" step. In the learning step, a training database or historical data is needed to train the networks. This dataset includes an input vector and a known output vector. Each one of the inputs and outputs represent a node or neuron. In addition, there are one or more hidden layers. The objective of the learning phase is to adjust the weights of the connections between different layers or nodes. After setting up the learning samples, in an iterative approach, a sample will be fed into the network and the resulting outputs will be compared with the known outputs. If the result and the unknown output are not equal, changing the weights of the connections will be continued until the difference between them is minimized. After acquiring the desired convergence for the networks in the learning process, the validation dataset is applied to the network for the validating step (Shahkarami, et al., 2014)

In the field of neural networks the collection of papers is very good. About 25 years ago, the golden age of neural network research ended. Now the research in this area has been re-energized after the discovery of back propagation described in the following section. Many reviewers have used the "feed-forward" neural network for the interconnection of perceptrons which is also described in the following section.

2.3.5.1 Types of Neural Networks

- **Feed-Forward Neural Network**

The feed-forward neural network is a network of perceptrons with a differentiable squashing function, usually the sigmoidal function. These networks make use of the back propagation algorithm that is applied to adjust the weights by minimizing the error squared. Weights adjustment is done across multiple hidden layers. Figure 2. shows a fully connected feed-forward neural network.

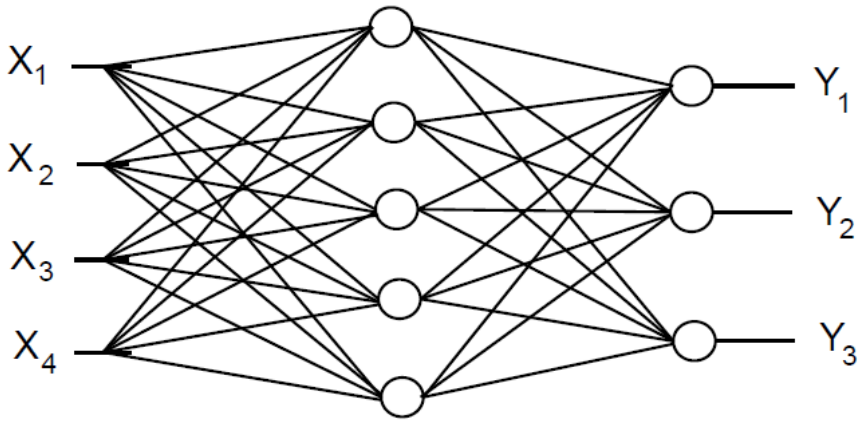


Figure 2.17: Fully-Connected, Feed-Forward Neural Network

Figure 2. presents the mathematical structure of the perceptron. The weights through the network are changing based on the weights changing for each perceptron. For each iteration; the difference between the output and the desired response is calculated. (Rumelhart & McClelland, 1986).

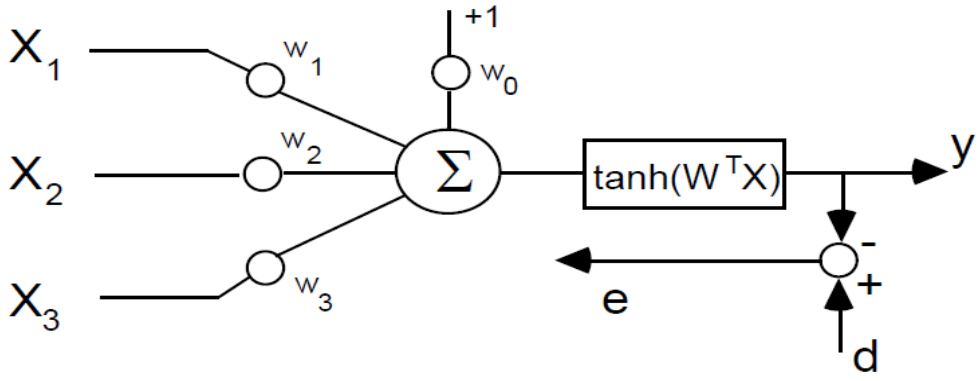


Figure 2.18: Perceptron's Mathematical Structure

During the training process, the inputs enter the neural network and get summed into the first layer of nodes. The outputs from the first layer of nodes get summed into the second layer of nodes. This process continues until the output comes from the neural network. (Smith, 1997).

- **Recurrent Neural Networks**

Unlike feed-forward networks, recurrent neural (RN) networks are those with a bi-directional data flow. While a feed-forward network propagates data linearly from input to output, RN propagates data from later processing stages to earlier stages (Bitzer & Kiebel, 2012).

There are two types of RN networks; the first type is the Simple Recurrent Network (SRN), which is a variation of the Multi-Layer Perceptron (MLP). A three layer network is used, with the addition of a set of context units fixed with weights of one. At each step, the input is propagated in a standard feed-forward fashion, and then a back-propagation learning rule is applied. SRN can be used for sequence-prediction that is beyond the power of a standard MLP. The second type of RN network is the Hopfield Network, which is a recurrent neural network in which all connections are symmetrical. This network was invented by John Hopfield in 1982 (Dong, et al., 2011).

- **Radial Basis Function Networks**

Radial Basis Functions

Radial basis functions are real-valued mathematical functions which have the same value if they are the same distance from the centre point of the function. A centre point and same distance *d* on a 2-dimensional space is illustrated in Figure 2. below.

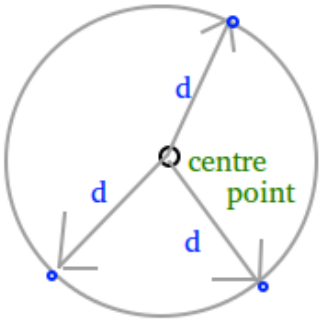


Figure 2.19: Distance from Centre Point

Distance is not vector based, and therefore direction and sign are ignored and only a positive real value is taken as the parameter for the function.

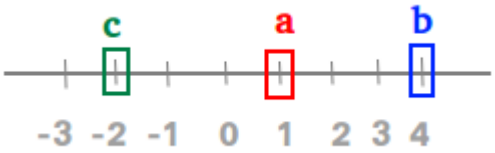


Figure 2.20: Distance in 1-Dimensional Space

In **Figure** 2., three points as *a*, *b*, and *c* are shown in 1-dimensional space. The distance between points *a* and *c*, and *a* and *b* are same as 3. If the centre point of a radial basis function is chosen as the point *a*, then point *c* and *b* will have the same result from the function.

Because of the abstract definition of radial basis functions, new ones can be made very easily. In Figure 2., a linear radial basis function is seen.

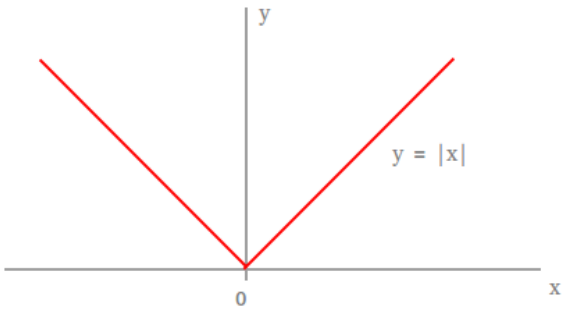


Figure 2.21: Linear Radial Basis Function

Because radial basis functions are like any other mathematical functions, very complex ones can be defined as well. One of them is the gaussian function which is defined in in the following equation:

$$f(x) = ae^{-\frac{(x-b)^2}{2c^2}}$$

As seen in the linear radial basis function in the equation, a linear radial basis function goes to infinity when the parameter x becomes negative or positive infinity. On the other hand, a gaussian function creates a closed curve and goes to zero in infinity. A basic response structure for a gaussian function is shown in Figure 2..

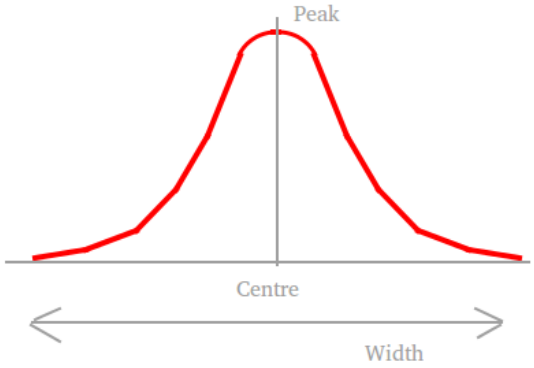


Figure 2.22: Parameters of Gaussian Function

In the above figure, three different parameters of a gaussian function including width, centre point, and peak value are seen. These parameters provide great flexibility on the result of function, and so it can be used in statistics, Mexican-Hat Wavelets and artificial neural networks.

Radial Basis Neural Network

A radial basis function network (RBF) is a subclass of artificial neural networks which uses a radial basis function as the activation functions on its nodes. Figure 21 below illustrates this as a single node to imitate a neuron.

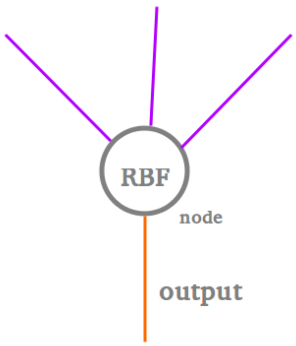


Figure 2.23: A node of RBF Network

In radial basis function network the input consists of two layers. Perceptrons in the first layer use radial basis functions, and in the second layer, linear classification is applied.

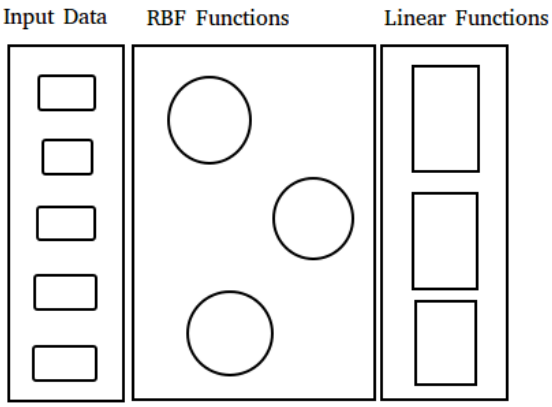


Figure 2.24: RBF Network Classifier Structure

As seen in Figure 2., input data is firstly provided to RBF perceptrons. Outputs that are generated from RBF perceptrons are then provided to the perceptrons that are in the last layer. Finally the classification results are acquired. This process is formulated in the equation below.

Assuming N is a positive integer value, and it is the number of inputs. The input values of the first layer indicate a point in the N-dimensional space. Because it is hard to demonstrate a high dimensional space on a 2-D plane, N will be chosen as 2 here. So the input space can be seen in **Figure** 2..

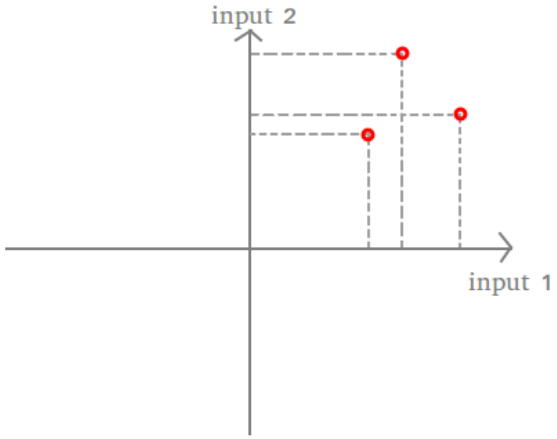


Figure 2.25: Two inputs create 2D space

As explained before, radial basis functions use distance as the parameter, and to be able to measure distance, each radial basis function must define an origin point. This origin point must be defined in the same space of inputs. So, each RBF defines an N dimensional origin for itself. For 2D space, the origin can be defined as in the following equation.

When both inputs and radial basis functions are in the same space, for each radial basis function, a distance value can be calculated. If there are K number of radial basis functions available in the network, distance values can be calculated as below:

In the equation above, j is an integer value between 0 and K. Because all inputs define a single point in the N-dimensional space, it is represented by "x". The RBF_j represents the origin of the j^{th} radial basis function. Lastly, the vertical lines represent the Euclidean distance.

Now, each of the radial basis functions has a distance parameter, and therefore, they can produce an output. This can be represented as v_j, and it is formulated in the equation below.

By the calculation of radial basis functions, the second layer which is also called the "hidden layer" is completed. In the third layer of the network, linear classification by using weights is illustrated in Figure 2..

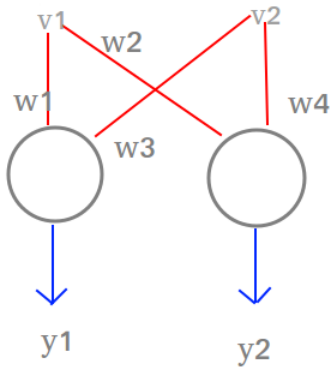


Figure 2.26: Third layer of Network

The number of perceptrons in the third layer is independent from the number of inputs and radial basis functions. For each perceptron, as in the neural networks, the sum of the inputs is calculated and passed to the linear activation function to determine the class. Since inputs of this layer are connected to radial basis functions, outputs can be calculated as in the next equation shown in the following equation.

By following the same steps, the output of each perceptron can be calculated. So the classification by using radial basis function network is completed.

RBF networks and single-layer kernel machines

In recent decades a sufficient number of techniques for different machine learning tasks including classification, regression, function approximation clustering and feature transformation were developed with the help of the class of non-linear functions called radial basis functions (Orr, 1996); (Orr, 1999). One of the interesting ideas is the use of radial basis functions networks and their generalization kernel networks. In this work special emphasis is given to the application of these networks to the problem of data classification.

Radial basis functions are a special kind of functions which have a characteristic feature to monotonically decrease or increase with the increase of the distance from the central

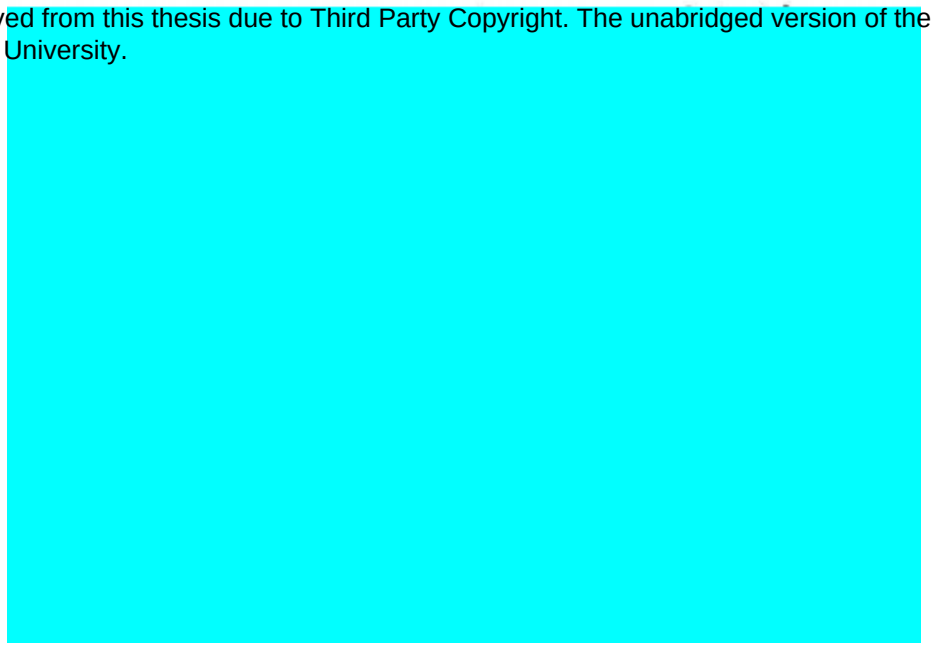
point. The center, the distance scale and shape could vary for different models (Orr, 1996). The most commonly used example is the Gaussian function $f(x) = e^{-\frac{(x-c)^2}{r^2}}$ and multi-quadratic function $f(x) = \frac{\sqrt{r^2+(x-c)^2}}{r}$ while, of course, a lot of variations are possible. The locality of these functions (i.e. given that the input parameters affect result of the function only within predefined local area) and ability to select nonlinear part of input space, led to the idea of the application of these functions to be used as the basis for function approximation methods. Let say that we have the sample pairs of values $\{(x_1,y_1),(x_2,y_2),\dots,(x_m,y_m)\}$ where x_i is called the input or predictor variable and y_i is the target variable. Both could lie in the multidimensional space. The approximation of y can build in the following form: $y(x) = \sum w_i \varphi_i(x,c,r)$ where $\varphi_i(\cdot,c,r)$ is a radial basis function with selected parameters for the center and scale factor. The goal is to find optimal weights w_i which will give the function with the exact values in the selected data points. It has been proven (Broomhead & Lowe, 1998) that such weights can always be found if all the pairs in the dataset are different, by solving a system of linear equations:

$$\begin{matrix} y_1 \\ \vdots \\ y_m \end{matrix} = \begin{pmatrix} A_{1,1} & \cdots & A_{1,m} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,m} \end{pmatrix} \cdot \begin{matrix} x_1 \\ \vdots \\ x_m \end{matrix}, \text{ where } A_{i,j} = \varphi_j(x_i, c_j, r_j).$$

Based on this method ; Broomhead & Lowe(1998) extended this interpolation problem to the function approximation problem where the values in the sample can have the same values x_i while having different output y_i and the aim of the algorithm is to find the approximation of the function with the minimum error on the dataset. The solution to the problem was found with the minimum least square error method. Further relying on the resemblance of the approximation function form, with the function equivalent to the neural network with the single hidden layer, they also proposed the idea of radial basis function networks with each neuron of the hidden layer having an RBF activation function. On Figure 2. scheme of such a neural network can be seen. The nodes in the first input layer which are directly connected to all nodes of the next hidden layer, accept the components of the n-dimensional input and then pass them to the next layer. The

overall input for the j hidden layer node is then formed from these values using formula: $\theta_j = \sum_{i=1}^{n}(x_i - y_{i,j})^2$. The result passed to each output unit is the weighted sum of the rbf functions of each node applied to the corresponding input θ_j.

As stated by Broomhead & Lowe(1998), the main advantage of the proposed neural network over the regular multilayer perceptron is the capability to model and predict the non-linear transformation of the input feature space without the application of iterative non-linear optimization techniques, but instead, to solve a deterministic linear system of equations. While the convergence of the iterative method highly depends on the initial point the optimal point for the RBFN can always be found.



The method proposed by Broomhead & Lowe(1998) was quite straight forward when applied to the fixed set of RBF functions with particular center and scale parameters, however the question of the selection of the optimal RBF functions for each neuron arises in that case. In response to this, the classification based on the number of the training phases for the learning was proposed (Schwenker, et al., 2001). In one-phase training, the learning algorithm only optimizes the weights of the network while the centre of the

functions is selected from the training sample randomly and scaling is fixed by the developer, usually based on the statistical distribution of the data. Two-phase learning assumes that the first the supervised or unsupervised algorithm for centers assignment is used depending on the task. For example, for the classification problem it is suggested to use an adaptation of the Kohonen's learning vector quantization algorithm (LVQ) (Kohonen, 1995). With each training case the prototype center vector c_j is slightly changed according to the current response of the network (correct or incorrect classification) via the formula: $\Delta c_j = \eta(x - c_j)\left(y - \frac{1}{2}z\right)$., Where x is a current input to the network, y is an output result provided by the current state of the network and z is the desired output. Another idea is to use the classification decision tree to divide the feature space into non-intersecting regions then to use the method proposed by Kubat to initialize both centers and scaling factors of the network (Kubat, 1998). The selection of the form for the RBF function also happens on this learning phase. In another paper (Schwenker, et al., 2001) several options for this contours including a symmetric, radially symmetric and arbitrary form of the function are explored. The optimal parameters of the contours could be determined with heuristic algorithm. The different algorithms for center and shape initialization techniques are also proposed in other papers (Wettschereck & Dietterich, 1992); (De Castro L. N., 2001); (Yousef & El Hindi, 2006) and others. Fasshauer & Zhang (2007)_and Mongillo (2011) described methods for the selection of the type of function and scaling factor. Both authors used cross-validation techniques (leave) to select functions and shape parameters from the predefined set which showed better results on the dataset. After the initialization step was finished the same method as in the one-phase learning scheme was applied to determine the optimal weights for the RBF function output.

Some authors (Schwenker, et al., 2001) and some other publishers (Chen, et al., 1991); (Gomm & Yu, 2000) noticed that while two phase learning was simpler for users because the shape of the functions was determined automatically and more computationally effective because the functions and weights can be determined in 2 separate deterministic algorithms, but these procedures can lead to the lack of the accuracy in classifiers. That's

why they proposed the simultaneous optimization of function parameters and weights in the third phase of the learning algorithm via an error back-propagation algorithm which was usual for other kinds of the neural networks. The error function of the network was defined as the squared sum between the produced and expected output of the neural network. In the case of radial-basis networks, activation functions are continuous so the minimum error value is achieved when the derivatives of the error function with respect to the weights, centers and scale parameters to the function are 0. The gradient descent method was applied to these parameters to achieve optimal values. The tests on different kinds of visual datasets have shown that a third learning phase produces a significant improvement in the classification results.

Other methods (Chen, et al., 1991) and (Gomm & Yu, 2000) proposed different approaches to the centre selection. During the iterative training with the least squares algorithm they either include or exclude additional nodes with the centres which will give the best/worst gain in the performance. Effective formulas for system update with any additional or removed hidden nodes are provided.

Another way to think about RBF networks is as kernel machines with specific type of kernel. Kernel machines are special machine learning methods which allow using regular machine learning techniques developed to learn linear functions in the problems with non-linear dependencies. This goal is achieved via transformation (mapping) of input feature space into the Hilbert space. The first kernel machines were a natural extension of the Support Vector Machine proposed by (Cortes & Vapnik, 1995) for classification of the linearly separable data points. The goal of the algorithm was to find the hyperplane which will divide two datasets and will have the maximum distance (margin) between itself and the closest points from the two classes. This hyperplane can be presented as the linear combination of the training samples lying on that margin (support vectors): $H(x) = \sum_i (\alpha_i y_i \langle x_i, x \rangle) + \alpha_0$. The algorithm finds the optimal values for the parameters α. The extension for the non-linear separable case, exploits the so called feature mapping function g with the hyperplane of the form: $H(x) = \sum_i (\alpha_i y_i \langle g(x_i), g(x) \rangle) + \alpha_0$. The function $K(x_i, x)$ which satisfies the

conditions of the Mercer's theorem can be presented in the form $K(x_i, x) = \langle g(x_i), g(x) \rangle$ in the Hilbert space, is called the kernel. If the kernel function is selected appropriately the data points in the new feature space can become separable. This method is usually referred to in the literature as the "kernel trick". In this case the method for linear SVM training could be applied. The Gaussian radial basis function is one of these kernel functions, so the support vector learning could be applied as a learning method for radial basis function network, with the support vectors being the centers of radial basis functions (Schwenker & Kestler, 2001); (Cortes & Vapnik, 1995). Weston & Watkin (1998) provided the idea of the extension of the support vector machine algorithm to the multiclass classification problem with usage of different weights for different outputs and selection of the class which produces the maximum value. That is equivalent to the structure of the neural network described above.

The three methods described above are considered conventional in the field of training radial basis function networks. However, in recent years a lot of algorithms have been developed based on the new computational optimization techniques, for more efficient learning. For example, Vachkov, et al. (2015) proposed the use of a particle swarm optimization algorithm to generate the RBF network with optimal parameters by the sequential addition of the new nodes to the hidden layer. Only the parameters of this new node are optimized at each step. However, all types of parameters including weights, centers and width for the function shape are updated simultaneously for this node. (Billing & Zheng, 1995) use a genetic algorithm to achieve the same goal. De Castro & Von Zuben (2001) exploits another bio-inspired method (e.g. the artificial immune system) to compress the data and select optimal centers for the algorithm. The proposal by Yousef & El Hindi (2006) is based on the same idea but uses conventional algorithms like Edited Nearest Neighbourhood, EXPLORE and DROP for data compression. These methods are more advanced because they not only provide the way to automatically select networks parameters (e.g. weights, centers and function shapes) but also for network structure generation with respect to training time and possible over-fitting. The selection of optimal structure was considered an important problem in the early decade of work with RBF networks. Both the selection of too few and too many nodes have led to

the poor performance of the overall network (Wettschereck & Dietterich, 1992); (Vachkov, et al., 2015); (Billing & Zheng, 1995). This group of methods has solved this problem.

However, while all the papers described above only focus on optimizing parameters of the fixed radial basis function it's important to mention work proposed by Hoffmann (2004) this uses the idea of adaptive transfer functions by the linear combination of multiple types of radial basis functions (for example, Gaussian, multiquadratic, etc.) into a single function which smoothly "morphs" from one shape to another: $\varphi(x,c,r) = \alpha\varphi_1(x,c,r) + (1-\alpha)\varphi_2(x,c,r)$. This new kind of function was named the universal basis function UBF and was found by authors to be more successful than the traditional RBF network in different applications. There is no such previous work found and any related research is not available. Theoretical analysis. But some attempts were made by Duch & Jankowski and Dorffner(1994) to generate the activation/transfer functions which will combine both properties of the functions used in the regular multilayer perceptron's and RBF networks in order to achieve a unified framework for different learning cases. Webb & Shennon (1998) exploited a more complicated use of per point functions adaptation via and normalization with non-linear transformation.

2.3.5 Deep Learning

Deep learning is currently one of the most important active research areas in machine learning. It has attracted extreme attention from researchers due to its potential in wide range of active applications such as object recognition (Zeiler & Fergus, 2014; Simonyan & Zisserman, 2014), speech recognition (Hinton, et al., 2012; Sainath, et al., 2015), natural language processing (Socher, et al., 2012), theoretical science (Kaggle, 2014), medical science (Brebisson & Montana, 2015; Shin, et al., 2013), etc.

Deep learning has also been used commercially by companies like Google, Facebook, Twitter, Instagram, Apple and others to provide better services to their customers. For example, Apple's Siri, the virtual personal assistant in iPhones, provides a large variety of

services including weather reports, sport news, answers to user's questions, and other different services by exploiting the high capability of deep learning approaches (Efrati, 2013). Also, other famous companies make use of deep learning techniques to improve their services, like Microsoft's real-time language translation in Bing voice search (Wang, et al., 2011), IBM's brain-like computer (Jones, 2014; Kirk, 2013), and Google in Google's translator, Android's voice recognition, Google's street view, and image search engine (Jones, 2014).

Deep learning can be defined as a machine learning technique that makes use of neural networks which are linked in a hierarchical architecture that in turn uses several layers to produce an output. Each layer receives input from a layer below, transforms its representations and then propagates it to the layer above (Dong & Wang, 2016; Soniya, et al., 2015).

2.3.5.1 Deep Learning Architectures

- **Deep Neural Networks**

Inspired by the biological nature of human brain mechanisms for processing of natural signals, deep neural networks are representation learning methods with multiple levels of representation (LeCun, et al., 2015; LeCun, 2012).

The expression “deep” is used because the depth of the network is greater when compared to the more conventional neural networks, which are sometimes called shallow networks. In most conventional learning methods, a simple network with one hidden layer may achieve acceptable performance for conducting a specific task, but by applying a deep architecture with more hidden layers higher efficiency can be achieved. This is because each hidden layer extracts more features from the previous layer and creates its own abstract representation. Therefore, to resolve more complicated features, we must add more hidden layers, which make deep learning capable of learning latent information (Soniya, et al., 2015).

This concept of deep networks is like hierarchical neural networks such as the neocognitron model proposed by Fukushima & Miyake (1982), but with some differences in the architecture and learning algorithm. The neocognitron model uses winner-take-all unsupervised learning whereas learning in deep networks is done by the back-propagation algorithm.

The term “deep” learning was coined as a contrast to the “shallow” learning algorithms which have fixed and usually single layer architecture. The “deep” learning architectures are compositions of many layers of adaptive non-linear components (Bengio & LeCun, 2007). It is expected that by analogy with the mammal brain capable of storing information on several layers of different abstractions, these multi-layer architectures will bring improvement to future learning algorithms. However simple training of the neural networks with up to 2 or 3 multiple hidden layers have shown an improvement but further increases in the number of layers did not provide any significant improvement and in some case the results were worse (Bengio, 2009). The existing algorithms have faced the problem of the local minimum and it has been reported that the generalisation of such gradient-based methods have become worse with a larger number of layers. Several papers have also shown that supervised training of each separate layer does not give a significant improvement in results compared to regular multilayer learning. Later development has gone in the direction of the intermediate feature representation for each new layer. Deep learning networks and training algorithms using this approach have achieved significant results in the multiple real-life applications (Bengio, et al., 2013) including computer vision, audio signal processing, and natural language processing and so on. In some fields of study, they are still considered to be among the best available approaches.

The successful examples of the deep neural networks for supervised learning mainly exploit two different approaches and their possible combinations which include both the special structure of the network in terms of neuron connections with hierarchically organized feature transformations applied to their results (i.e. convolutional neural

networks) and also multilayer networks with feature representations for each layer learned with unsupervised learning technique followed by parameter tuning of the network using a regular supervised learning technique.

- **Deep Autoencoders**

The Autoencoder is a simple neural network that is designed for features extraction or dimensionality reduction. It is applied to the unlabelled input with a minimum of restrictions on the number of activated neurons on the hidden layer. As a result, such an encoder can discover any interesting patterns in the data. Specifically, an Autoencoder has the same number of input and output nodes and it is trained in unsupervised manner to recreate the input vector rather than to assign a class label to it. Usually, the number of hidden units is smaller than the input/output layers, which achieve encoding of the data in a lower dimensional space and extract the most discriminative features. If the input data is of high dimensionality, a single hidden layer of an Autoencoder may not be sufficient to represent all the data. Alternatively, many Autoencoders can be stacked on top of each other to create a deep Autoencoder architecture (Hinton & Salakhutdinov, 2006). Deep Autoencoder structures also face the problem of vanishing gradients during training. In this case, the network learns to reconstruct the average of all the training data. A common solution to this problem is to initialize the weights so that the network starts with a good approximation of the final configuration. Finding these initial weights is referred to as pre-training and is usually achieved by training each layer separately in a greedy fashion. After pre-training, the standard back-propagation can be used to fine-tune the parameters.

Miotto, et al. (2016) showed that a stack of denoising autoencoders can be used to automatically infer features from a large-scale electronic health records (HER) database and represent patients without requiring additional human effort. These general features can be used in several scenarios. The authors demonstrated the ability of their system to predict the probability of a patient developing specific diseases, such as diabetes, schizophrenia and cancer.

Various autoencoder-based learning approaches have also been applied to the automatic extraction of biomarkers from brain images and the diagnosis of neurological diseases. These methods often use available public domain brain image databases such as the Alzheimer's disease neuroimaging initiative database. For example, a deep Autoencoder combined with a SoftMax output layer for regression is proposed for the diagnosis of Alzheimer's disease. Hu, et al. (2016) used autoencoders for Alzheimer's disease prediction based on Functional Magnetic Resonance Images (fMRI). The results show that the proposed method achieves much better classification than the traditional means.

- **Deep Belief Network**

Deep belief networks (DBNs), as proposed in the work of Hinton et al. (2006), are based on the Restricted Boltzmann machines (RBMs) which in fact are acyclic graphs which attempt to discover any probability distributions or dependencies in the data. A DBN is initialized through an efficient layer-by-layer greedy learning strategy using unsupervised learning and is then fine-tuned based on the target outputs. Unsupervised training is intended to explain the variations in the input data. After the processing of each layer is completed, networks initialized in this manner are fine-tuned with the regular gradient based learning method. The experiments on the different datasets have shown that unsupervised pre-training provides some sort of regularisation factor which minimises variance and introduces a bias towards the configuration of the input feature space, which is useful for unsupervised learning (Erhan, et al., 2010). This is different from random parameters initialization where the probability of selecting parameters which lead towards the local minimum of the optimization criteria is very high, in which case unsupervised learning provides the algorithm with an insight which leads to better generalization. Figure 2.28 illustrates the framework of a DBN model.

Wulsin, et al. (2010) proposed a DBN approach to detect anomalies in EEG waveforms. EEG is used to record electrical activity of the brain. Interpreting the waveforms from brain activity is challenging due to the high dimensionality of the input signal and the limited understanding of the intrinsic brain operations. Using a large set of training data, DBNs outperform SVM and have a faster query time of around 10s for 50 000 samples.

A DBN was also used for detecting arrhythmias from electrocardiography (ECG) signals. A DBN was also used in monitoring heart rhythm based on ECG data (Y.Yan, et al., 2015). The main purpose of the system is identifying arrhythmias which is a complex pattern recognition problem. Yan et al. attained classification accuracies of 98% using a two-lead ECG dataset. For low-power wearable and implantable EEG sensors, where energy consumption and efficiency are major concerns.

- **Convolutional neural networks**

Convolutional neural networks (CNN) (LeCun & Bengio, 1998) were specially designed for visual object recognition and they were based on the modern ideas concerning the working of human visual perception. Owing to their special structure they are easier to train than conventional fully connected networks. The convolutional network designed by LeCun is currently the best known for character recognition. This network has two types of layers which include convolutional and sub-sampling layers as illustrated in figure 2.29.

Neurons are associated with the fixed two dimensional positions on the input image with the weights assigned to the rectangular patch of the image. This locality principle allows the learning of local features such as edges and shapes. These features are later hierarchically combined by the higher layers of the network. The nodes corresponding to the learned features are later copied with the same weights to different network positions. This is done based on the assumption that the same low level features can be met in the different locations of the image. It also helps to overcome difficulties caused by small input distortions and invariant transformations like shift or rotation. In general weights

sharing decrease learning complexity of the networks requiring optimize less parameters but with more effective representation. The neurons and weights produced by this method constitute a feature map, which is applied to different parts of the image. The convolutional layer is composed of several feature maps which allows the computing of several types of features at each location. It is followed by the sub-sampling (pooling) layer which performs averaging and sub-sampling of the computed features to reduce the overall complexity by the further application of non-linear (sigmoid) transformation. The procedure is repeated hierarchically to the next layer with rectangular patches of the neurons on previous layers being assigned to the neurons on the higher layer. Several groups of these layers are used to constitute the overall network (Bengio, 2009).

CNNs inherently learn a hierarchy of increasingly more complex features and, thus, they can operate directly on a patch of images centred on the abnormal tissue.
Example applications of CNNs in medical imaging include the classification of interstitial lung diseases based on computed tomography (CT) images (Anthimopoulos, et al., 2016), the classification of tuberculosis manifestation based on X-ray images (Cao, 2016) and the organ or body-part-specific anatomical classification of CT images (Roth, 2015).

A more recent deep learning approach is known as convolutional deep belief networks (CDBN) [43]. CDBN maintains structures that are very like a CNN, but is trained differently like DBN. Therefore, it exploits the advantages of CNN whilst making use of pre-training to initialize efficiently the network as a DBN does (Rav`, et al., 2017).

- **Long Short-Term Memory Recurrent Neural Networks**

Although the traditional RNN is a simple and powerful model, it suffers from the vanishing gradient and exploding gradient problems as described in (Bengio, et al., 1994). A variation of RNN called long short-term memory units (LSTMs) was proposed in (Hochreiter & Schmidhuber, 1997) to solve the problem of the vanishing gradient generated by long input sequences. Specifically, LSTM is particularly suitable for

applications where there are very long time lags of unknown sizes between important events. To do so, LSTMs exploit new sources of information so that data can be stored in, written to, or read from a node at each step. During the training, the network learns what to store and when to allow reading/writing to minimize the classification errors. Figure 2.30 illustrates the architecture of this model.

Unlike other types of DNNs, which uses different weights at each layer, LSTM shares the same weights across all steps. This greatly reduces the total number of parameters that the network needs to learn. LSTMs have shown great successes in many natural language processing tasks such as language modelling, bioinformatics, speech recognition, and generating image description.

Lipton, et al.(2015) employed a LSTM RNN to tackle time dependencies in EHR with multivariate time series from intensive care monitoring systems. The reason for using RNNs is that their ability to memorize sequential events could improve the modelling of

the varying time delays between the onsets of emergency clinical events, such as respiratory distress and asthma attack and the appearance of symptoms.

2.3.5.2 Deep Learning and Data Representation

RBF networks and kernel machines in general have proved their effectiveness in different machine learning tasks and there has been extensive development from the theoretical and algorithmic point of view in this field since they were first introduced. However, this method has been seen to have some flaws, which prevent it from being used in some advanced and futuristic applications. Like other methods, for example (kNN) relying on the data smoothness and locality (meaning that similar points should lie close together in the feature space), the kernel machines suffer from the fast growth of the number of learning parameters when predicting data with large number of variations (Bengio, et al., 2005). Another problem is that the kernel machines with a single hidden layer have no mechanisms for features selection in the multidimensional data space and rely completely on the user for this. The optimal selection of features for this method becomes more and more complicated as the data available for analysis increases. To solve this problem common for many machine learning algorithms the paradigm of deep learning has recently emerged. This approach assumes that the learning model should not only provide the prediction results but also give an optimal data representation for the task.

The notion of the good data representation usually includes several points (Bengio, et al., 2013): smoothness and natural clustering – similar data points should lie close to each other in the learned feature space; expressiveness of explanatory factors – the learned feature space should be of reasonable space but still be able to explain multiple variations of data; a hierarchical organization of explanatory factors – it will be useful to have a hierarchical structure of features/ concepts where more abstract features will be defined in the terms of less abstract features located lower in the hierarchy; shared factors across tasks – it is common that the same concepts can be used to explain different events, so it will be useful to be able to use the same features to predict different parameters; sparsity -

only small number of factors should be relevant for each of the particular observations; simplicity – it is desirable for many algorithms to have simple (in the best case linear) dependencies between factors.

2.4 Summary

This chapter presented a background study on different type of heart diseases (mainly CAD, arrhythmia, and sleep apnea) from clinical view. The chapter also discussed classification techniques linear, non-linear, KNN, SVM and deep learning. Also the classification techniques focusing on artificial neural networks and deep learning approaches used in the present research. The next chapter discusses the proposed approaches for heart diseases classifications using data mining algorithms.

3 : Heart Diseases Classifications using Data Mining and Neural Networks

3.1. Overview

Diagnosis of heart disease was traditionally based on medical knowledge obtained from patients, but this manual diagnostic approach is costly and causes delay in the proper treatment. Hence, there is a need to develop a prediction system for heart disease that can provide medical knowledge for diagnosis. Researchers have been using data mining techniques to help health care professionals in the diagnosis of heart disease (Helma, et al., 2000); (Podgorelec, et al., 2002). Several data mining techniques are used in the diagnosis of heart disease such as Naïve Bayes, Decision Tree, Neural Networks, Kernel density, automatically defined groups, bagging algorithm, and support vector machine showing different levels of accuracies (Yan, et al., 2003); (Srinivas, et al., 2010).

This chapter considers the available information on Heart Disease and its detection using artificial intelligence methods. It reviews the use of the data mining technique for the prediction of heart diseases, focusing on neural networks and deep learning methods.

3.2. Heart Diseases Classification

3.2.1. General Heart Disease

Abushariah, et al. (2014) presented a decision support system for heart disease classification using a neural network. The data set used is the Cleveland Heart Database was taken from the UCI learning data set repository (Asuncion, 2007). This research developed two systems based on ANN and Neuro-Fuzzy approaches to develop an automatic heart disease diagnosis system. The experimental results showed that the Neuro-Fuzzy system outperformed the ANN system using the training data set, where the accuracy for each system was 100% and 90.74%, respectively. However, when using the testing data set, the ANN system outperformed the Neuro- Fuzzy system, where the best accuracy for each system was 87.04% and 75.93%, respectively.

Das, et al. (2009) have proposed several tools and various methodologies to develop an effective medical decision support system. A method was introduced which used the Statistical Analysis System (SAS) base software for diagnosing heart disease. For this method the neural networks ensemble model was used, which enabled an increase in generalization performance by combining several individual neural networks trained on the same task. For heart disease diagnosis, the experimental result obtained 89.01% classification accuracy, 80.95% sensitivity and 95.91% specificity values.

Kahramanli & Allahverdi (2008) have developed a hybrid neural network which included an Artificial Neural Network (ANN) and a Fuzzy Neural Network (FNN). The proposed method achieved accuracy values of 84.24% and 86.8% for the Pima Indians diabetes dataset and Cleveland heart disease dataset respectively.

Nawi & Ghazali (2010) proposed a novel method to improve the efficiency of back propagation neural network algorithms. In the proposed Gradient Descent with Momentum and Adaptive Gain (GDM/AG) proposed, for each node the gain value was changed adaptively to modify the initial search direction. The modification enhanced the computational efficiency of the training process and could be implemented in the optimization process. The convergence speed of the proposed algorithm was evaluated using the classification matrix.

Cui, et al. (2012) proposed a training artificial neural network by exploiting Artificial Photosynthesis and Phototropism Mechanism (APPM). They used a stochastic optimization algorithm that stirs the plant growing process. In their algorithm, each entity is called a “branch” and the sampled points are the branch growing trajectory. They applied the APPM algorithm to instruct the connection weights for the artificial neural network. They used two real world issues which are the Cleveland heart disease categorization issue and the sunspot number foreseeing issue to evaluate the performance of their APPM trained ANN. The outcome from the use of the technique showed a significant increase in performance when compared with other sophisticated machine learning techniques.

In many studies; radial basis function networks were performing much better than multilayer perceptron networks. Hannan, et al. (2010) used both a Generalized Regression Neural Network (GRNN) and a Radial Basis Function Network (RBFN) to diagnose heart disease. The data from around 300 patients was collected to be used for training and testing of classifiers. For the experiments, the SVM classification method was used as the GRNN. Results from the study showed that the RBF Network provided much more suitable medicine prescriptions which were verified by the doctor, and SVM did not provide satisfactory results. But the researchers concluded that with more training data a better performance could be expected.

Karpagachelvi, et al. (2011) demonstrated another good example of the application of machine learning in the clinical sector. The authors proposed the use of the Extreme Learning Machine (ELM) classifier in classifying ECG patterns. The researchers tested their ELM classifier on the data obtained from the Physionet arrhythmia database and compared the results with the support vector machine classification. The results showed that the ELM classifier was more sensitive and accurate in classifying five different types of abnormal and normal ECG frequencies, than the support vector machine classification, ELM classifier incorporates the k-nearest neighbour classifier (kNN) and the radial basis function neural network classifier (RBF).

3.2.2. Coronary artery disease

Babaoglu, et al. (2009) identified coronary artery disease existence based on Exercise Stress Testing (EST), using Artificial Neural Networks (ANN). The dataset was performed on 330 patients. The diagnostic accuracy for the Left Main Coronary (LMCA) was 91%, and 69% for the right coronary artery. The elimination of LMCA lesions was confirmed by a 94% negative predictive value.

Atkov, et al. (2012) developed an ANNs based diagnostic model for coronary heart disease (CHD) using a complex range of traditional and genetic factors of this disease.

The obtained dataset was collected from 487 patients. Approaches using different ANN types achieved overall accuracy of 64-94%. The best result (94%) was achieved in a Multilayer Perceptron (MLP) model with two hidden layers and 10 features. On the other hand, the same ANN type with 5 features had a lower accuracy (78%). This suggests that the diagnostic accuracy depends on the ANN type and the number of features.

3.2.3. Arrhythmia

The ANN architectures mostly used for arrhythmia classification are Multilayer Perceptrons (MLP) and Probabilistic Neural Networks (PNN). According to (Yu & Chen, 2007), models constructed with PNN are computationally more robust and efficient than the traditional MLP. However, (Özbay, et al., 2006); (Osowski & Linh, 2001) proposed hybrid neuro-fuzzy network methods in order to minimize the problems of MLP, increasing its generalization and reducing its training time.

Güler & Übeyli (2005) combined neural networks to obtain a more generic method from a more sophisticated form of cross-validation. Mar, et al. (2011) used MLP with a fairer evaluation protocol by applying the patient division scheme proposed by (Chaza, et al., 2004). Thus, by using the reported results in the works of the methods that utilizes ANN as classifier it is impossible to makes a fair comparison. Finally, Mar, et al. (2011) compared MLP with Linear Discriminants and found that MLP was significantly superior. Combining classifiers had been little explored for the task in question. Osowsk, et al. (2008) found that a combination of classifiers not only reduces the overall error in the neural networks, but also reduces the incidence of false negatives.

Clustering techniques are widely used along with Artificial Neural Networks. Özbay, et al. (2006) found that they can improve the generalization capacity of the neural networks and diminish the learning time. Some works used unsupervised clustering techniques to agglomerate all of the heartbeats in the record of a given patient into clusters (Yeh, et al., 2012), and the final classification of each cluster, i.e., the heartbeats of that group, is then defined by a human specialist.

Llamedo & Martinez (2012) represented ANN to classify the ECG arrhythmias. The types of arrhythmias used were normal sinus rhythm, sinus bradycardia, ventricular tachycardia, sinus arrhythmia, atrial premature contraction, paced beat, right bundle branch block, left bundle branch block, atrial fibrillation, and atrial flutter have been as. This data was filtered, the R peaks found, and patterns normalized in the range of $[0,1]$. The patterns, for the training of ANN used the types of arrhythmia separately as well as in differing combinations.

3.2.4. Sleep Apnea

Tagluk, et al. (2010) developed a new method to classify Sleep Apnea Syndrome (SAS) by using wavelet transforms and an artificial neural network (ANN). The network was trained and tested for different momentum coefficients. The abdominal respiration signals were separated into spectral components by using multi-resolution wavelet transforms. These spectral components were applied to the inputs of the artificial neural network. Then the neural network was configured to provide three outputs to classify the SAS situation of the patient. Figure 3.1 shows the proposed methodology of this work.

The input of the neural network was formed by the coefficients of a discrete wavelet decomposition applied to the raw samples of the apnea in the abdominal effort signal. The obtained experimental results, using 360 apneas from twenty-one different patients, have demonstrated the validity of the proposed method. The best global accuracy obtained was 78.85%, which was good enough when compared to the manual scoring.

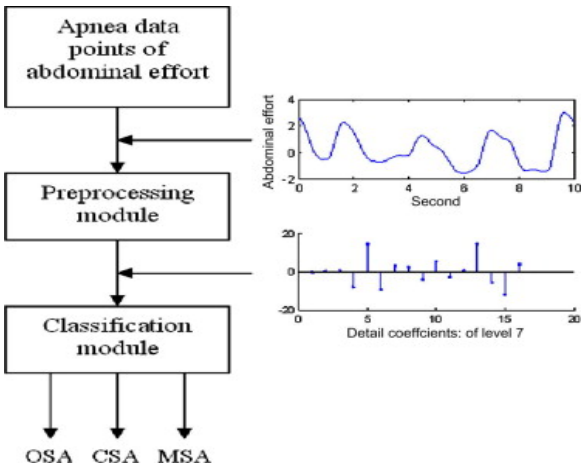


Figure 3.1: The methodology of the proposed algorithm

3.3. Other Classification Methods for Heart Diseases

3.3.1. General Heart Disease

Tan & Teoh (2009) proposed a hybrid approach that consist of Genetic Algorithms (GAs) and Support Vector Machines (SVMs). The genetic algorithm module was responsible for selecting for best attributes in the data set. Then SVM classified the patterns into a reduced data set. They called this approach by wrapper approach. The used datasets were obtained from the UCI machine learning repository. The results showed that the GA-SVM hybrid was a good classifier when the irrelevant attributes were removed. The obtained accuracy for the proposed approach was 76.20%. The proposed approach is also applied to multi-class domain and achieved average accuracy 84.07%.

Luukka & Lampinen (2010) applied a classification method which was based on pre-processing the data with Principal Component Analysis (PCA) and then applying the classification model to the diagnosis of heart disease. Authors used the classical Electronic Medical Record (EMR) heart data sets. Authors argued that the main aspect for improving the results is by applying effective global optimizer, and differential evolution, for fitting the classification model instead of local optimization based

approaches. They found that data pre-processing with PCA achieved higher classification accuracy. The proposed system achieved average accuracy of 82%.

Ansari & Gupta (2011) developed a Decision Support System for Heart Disease Prediction using the Naïve Bayes algorithm. The system extracts useful, previously hidden information from the heart disease database. This model may possibly answer difficult queries, each one with its own potency with respect to ease of model analysis, access to complete information and accurateness. This model can be further enhanced and expanded by incorporating other data mining techniques.

Shouman, et al. (2012) performed a work that applied KNN on a Cleveland Heart Disease dataset to investigate its efficiency for the prediction of heart disease. The author also investigated if the accuracy could be enhanced by integrating voting with KNN. The results showed that applying KNN achieved an accuracy of 97.4%. The results also showed that applying voting could not enhance the KNN accuracy in the diagnosis of heart disease.

Vijiyarani & Sudha (2013) proposed a paper that analyses the classification tree techniques in data mining. The classification tree algorithms used and tested in this work were Decision Stump, Random Forest, and LMT Tree algorithm. The objective of this research was to compare the outcomes of the performance of different classification techniques for a heart disease dataset. This work was done using the Waikato Environment for Knowledge Analysis (WEKA). It is open source software which consisted of a collection of machine learning algorithms for data mining tasks.

Amin, et al. (2013) concluded that there was a large amount of data available in medical institutions, but this data was not properly used. This medical data lacked quality and completeness so highly sophisticated data mining techniques were required to build up an efficient decision support system. The studies revealed the fact that systems should be built which are not only accurate and reliable but also reduce the cost of treatment and increase patient care. The systems should also be easy to understand so as to enhance

human decisions. The author also suggested that work should be done for proposing treatment plans for patients, because data mining techniques have shown significant success in prediction and diagnosis of diseases and especially heart diseases, hence these techniques could also be applied for treatment purposes.

3.3.2. Coronary artery disease

Regarding to the Coronary Artery Disease (CAD) detection; the most commonly used predictive models are the Linear Discriminant Analysis, K-nearest Neighbour Classifier, Artificial Neural Network and Support Vector Machine (Heydari, et al., 2012); (Arif, et al., 2010); (Comak & Arslan, 2010); (Hongzong, et al., 2007). These models have shown to a good extent, that they are of some predictive value. For example, Hongzong, et al. (2007) showed that the prediction accuracy of training and test sets of Linear Discriminant Analysis could be as high as 90.6% and 72.7% respectively, while Heydari, et al. (2012) showed that Artificial Neural Networks could produce accuracy in a test set that is as high as 81.2%.

In spite of these reasonable results; there are apparent limitations in most of these learning algorithms. The Linear Regression and the Linear Discriminant Analysis are both linear techniques and cannot be extended to consider non-linear modalities (variables) which are inevitable in the proper diagnosis of CAD. As a way of circumventing this issue, research in this field has also considered the use of more complex models such as the combination of Support Vector Machine with a Radial Basis Function (RBF) Kernel, Support Vector Machine optimized by particle swarm optimization or other forms of integration of two individual approaches to generate a non-linear technique (Hedeshi & Abadeh, 2014); (Mandal & Sairam, 2012); (Karabulut & Ibrikci, 2012); (Hongzong, et al., 2007). The result of this is an improvement in the prediction accuracy of training and test sets (for example, as high as 96.9% (Hongzong, et al., 2007)).

Ansari & Gupta (2011) offered a Neuro-fuzzy integrated system for the analysis of heart diseases. To show the effectiveness of the projected system, simulation for computerized diagnosis was performed by using realistic causes of coronary heart disease. The author concluded that this kind of system was suitable for the identification of cardiac disease, this includes patients with high or even a low risk. .

Sen, et al. (2013) designed a system which could identify the chances of a coronary heart disease. Authors divided all parameters into two levels, per the criticality of the parameter and assigned each level a separate weightage. Finally, both the levels were taken into consideration so as to arrive at a final decision. The authors implemented a neuro-fuzzy integrated approach at two levels. So that error rate was very low and work efficiency was high. The authors concluded that this same approach could also be used to perform the analysis on some other diseases.

3.3.3. Arrhythmia

Pati, et al. (2012) proposed a system to predict arrhythmia and its types. The system consisted of two parts which included the online and offline processing of data. The offline mode consists of previously formed rules which were obtained from patients having different characteristics such as age, gender or any other symptoms. In online mode, the continuous ECG signal was taken as input and passed to the pre-processing module. In this module, the noise was removed from the ECG signal, then this filtered ECG signal was given to the feature extraction module where all time domain features of the signal were extracted including the R-R interval, the Root Mean Square Successive Difference (RMSSD) in 8 R-R intervals, the Standard Deviation (SDNN) of 8 R-R intervals, the Fractal dimension, the Lyapunov Exponent, and the Hurst Exponent etc.

The system was a real-time application, which used data stream mining algorithms to provide dynamic processing of the data in real time making the system unique. The proposed system was based on Data Stream Mining techniques which consisted of different algorithms for classifying arrhythmia into seven types. They were namely:

Normal Beat(NB), Left Bundle Branch Block Beat(LBBB), Right Bundle Branch Block Beat(RBBB),Premature Ventricular Contraction(PVC),Fusion of ventricular and normal beat(FUSION), Atrial Premature Contraction(APC) and Paced Beat(PACE).

SVM is one of the most popular classifiers found in literature for ECG-based arrhythmia classification methods. Park, et al. (2008) used SVM and validated the method per AAMI standards and the data set split scheme proposed by Chazal, et al. (2004). These same authors used SVM in a mock-hierarchy configuration to resolve the imbalance of the MIT-BIH database, and reported promising values.

Lannoy, et al. (2010) managed to overcome the imbalance of the MIT-BIH database with SVM, alternating the objective function for each class (Weighted SVM). Expressive gains were reported for the SVEB and F classes.

Various approaches with SVM variations have been proposed, such as a combination of the fuzzy theory to refine SVM classification (Özcan & Gurgen, 2010), combined with an ensemble of classifiers (Huang, et al., 2014), genetic algorithms combined with restricted fuzzy SVM (Nasiri, et al., 2009), and least squares SVM (Polat & Günes, 2007). Huang, et al. (2014) used the SVM in a hierarchical manner with a maximum voting strategy and Moavenian & Khorrami (2010) proposed the use of a new kernel function for capturing data from SVM. In that work, same methodology was used for comparing the results obtained from a SVM and a Multilayer Perceptron Artificial Neural Network (MLP-ANN). While SVM was more efficient in execution time, both in the training and in the testing, MLP performed better in terms of accuracy, Sensitivity (Se), positive prediction (+P) and false positive rate (FPR).

Ramli & Ahmad (2003) used a cross correlation analysis technique to extract the important features from 12 lead ECG signal. Using the cross-correlation techniques, the identified values can be used to predict the type of arrhythmias.

Tadejko & Rakowski (2007) introduced an automated classifier with Kohonen self-organizing maps (SOM) and learning vector quantization (LVQ) algorithms. This paper compared the QRS complexes for classification which were based on original ECG morphology features and the proposed new approach based on pre-processed ECG morphology features. The performance of algorithms was assessed to recognize beats either as normal or arrhythmias.

Chazal & Reilly (2006) provides the premature ventricular contraction from the normal beats and other heart diseases. For feature extraction of ECG signal the combination of the morphological based features and timing interval based features were proposed. For ECG signal classification, the MLP with a different number of hidden layers and algorithms, according to the radial basis function and probabilistic neural network was used. The simulation results showed that about 97.14% classification accuracy for ECG beats was achieved. For simulation, the MIT-BIH arrhythmia database was used.

(Castro, et al., 2000) proposed a wavelet transform algorithm for feature extraction from an ECG signal and identification of abnormal heartbeats. This algorithm helped to find out the best correlation with the ECG signal. The ECG signal was first denoised by a soft or hard threshold and then each PQRST cycle was decomposed into a coefficients vector using the optimal wavelet function. The analysed ECG signal coefficients were divided into the P-wave, QRS complex and T-wave, and summed to obtained a features vector of the signal cycles.

Algorithms with a lazy approach, such as the k Nearest Neighbors (kNN), are not much used for the problem of arrhythmia classification, since their efficiency is intimately connected to previous knowledge to perform the classification of each sample that is represented by the complete training set, which leads to a high computational cost during the testing phase. This cost can invalidate its use for diagnosis in real time. Mishra & Raghav (2010) used a classifier based on kNN and reported promising results, however the computational cost was not mentioned. In the literature, no one presented a more fair evaluation protocol for comparison of methods as the one proposed by (Chazal, et al.,

2004), and no one also followed the AAMI recommendations. In addition, the computational cost of these methods was not investigated.

3.3.4. Sleep Apnea

Several algorithms using different methods were developed to identify the apnea class. For example, some authors made use of spectral analysis of heart rate variability (HRV) to identify apnea class and achieved 30 correct score out of 30 (without class B consideration) (De Chazal, et al., 2000; Jarvis & Mitra, 2000; McNames & Fraser, 2000). Other authors used the Hilbert transform to extract frequency information from the heart rate signal and achieved a score of (28/30) (MIETUS, et al., 2000; Schrader, et al., 2000). Some authors achieved the top three ranks in the PhysioNet's challenge about the minute-by-minute quantification (DE CHAZAL, et al., 2000; MCNAMES & FRASER, 2000; RAYMOND, et al., 2000). They reached an accuracy of 89.4%, 92.6%, 92.3%. In addition to HRV, authors made use of different features derived from ECG signals like ECG pulse energy (MCNAMES & FRASER, 2000), R-wave amplitude using power spectral density (PSD) (DE CHAZAL, et al., 2000) and T-wave amplitude using the discrete harmonic wavelet transform (RAYMOND, et al., 2000).

Khandoker, et al. (2009), employed wavelet based features and KNN classifier to achieve an accuracy of 83%. (Xie & Minn, 2012) extracted features from ECG and saturation of peripheral oxygen (SpO2) signals and employed classifier combination such as AdaBoost with Decision Stump and Bagging with REPTree where the classification accuracy was 77.74%.

Many studies show that detection of obstructive sleep apnea can be performed through HRV and the ECG signal. Manrique, et al. (2009) proposed a simple diagnostic tool for OSA with a high accuracy (up to 92.67%) using time-frequency distributions and dynamic features in ECG signal.

In addition, in another paper (Mendez, et al., 2007), a bivariate autoregressive model was used to evaluate beat-by-beat power spectral density of HRV and R peak area, where the classification results showed accuracy higher than 85%. The technique in this work also relies on features of the ECG signal.

In 2012, an automated classification algorithm was proposed based on support vector machine (SVM) using statistical features extracted from ECG signals for both normal and apnea patients based on Heart Rate Variability (HRV), with an accuracy of 96.5 % (Almazaydeh, et al., 2012).

3.4. Summary

This chapter presents some related prior work on different data mining techniques, neural networks and deep learning methods, classification of heart disease using deep learning and neural networks, and classification of heart disease using the traditional data mining methods. The chapter discusses and focuses on the CAD and sleep apnea diseases and its dataset, while also exploring through literature review different machine learning algorithms which can be used for this scenario. This chapter also presents the previous work in machine learning related to the diagnosis of general heart diseases. Many such algorithms like RBF, SVM and its variations, KNN classifiers are discussed with results obtained in previous studies. A comparison of the performance accuracy of these classifiers is also presented, to give an indication on which of these classifiers are more effective to diagnose a certain type of disease out of the four discussed before in the chapter.

The next chapter presents the research methodology used in current research and the details of the datasets used.

Chapter 4 : Research Methodology and Data Collection

4.1. Overview

This chapter presents a description of the methodology used in this research. The block diagram of the overall methodology is shown in Figure 4.1. This chapter builds on the foundation of the Literature review presented in chapter 2 and 3, and addresses some of the weaknesses and gaps found from these reviews. A clear methodology is presented on how the initial ideas have been matured over time, and now provide a concrete objectives and goal for this process. The chapter also summarises how the Data collection and pre-processing activities are performed before the application of classification methods. Brief of different classification methods used on the classification of CAD, Arrhythmia and Apnea are discussed. This includes classification methods such as Radial basis function, deep learning and classic classification techniques. Finally, the comparison and performance measurement of these classification methods are also presented.

4.2. Research Design

Mostly, there are six steps in the process of research proposed by (Creswell, 2012) and they involve: recognising the research problem, review of related literature, specifying a purpose for research, data collection, and analysis and reporting of data, and finally, evaluating the research. Figure 4.1 presents the research design for the development of this research.

In general, the data collection techniques for this research project were as follows:

i. **Internal sources:** obtained by extracting the existing patient's details from the computerized database at King Abdullah Medical City based on agreed data collection form that is provided at Appendix I.
ii. **Online Datasets:** by utilising available datasets from online machine learning repositories and databases.

4.3. Location of Research Study

The primary study area, Mecca city, is one of the most important cities in Saudi Arabia. The Coronary Artery Disease (CAD) data was collected from King Abdullah Medical City (KAMC). It is a 1550-bed quaternary specialized healthcare facility. The medical city consists of three campuses. The specialized hospital campus, a 550-bed quaternary hospital located in The Holy Capital, Makkah. It hosts multiple centres of excellence,

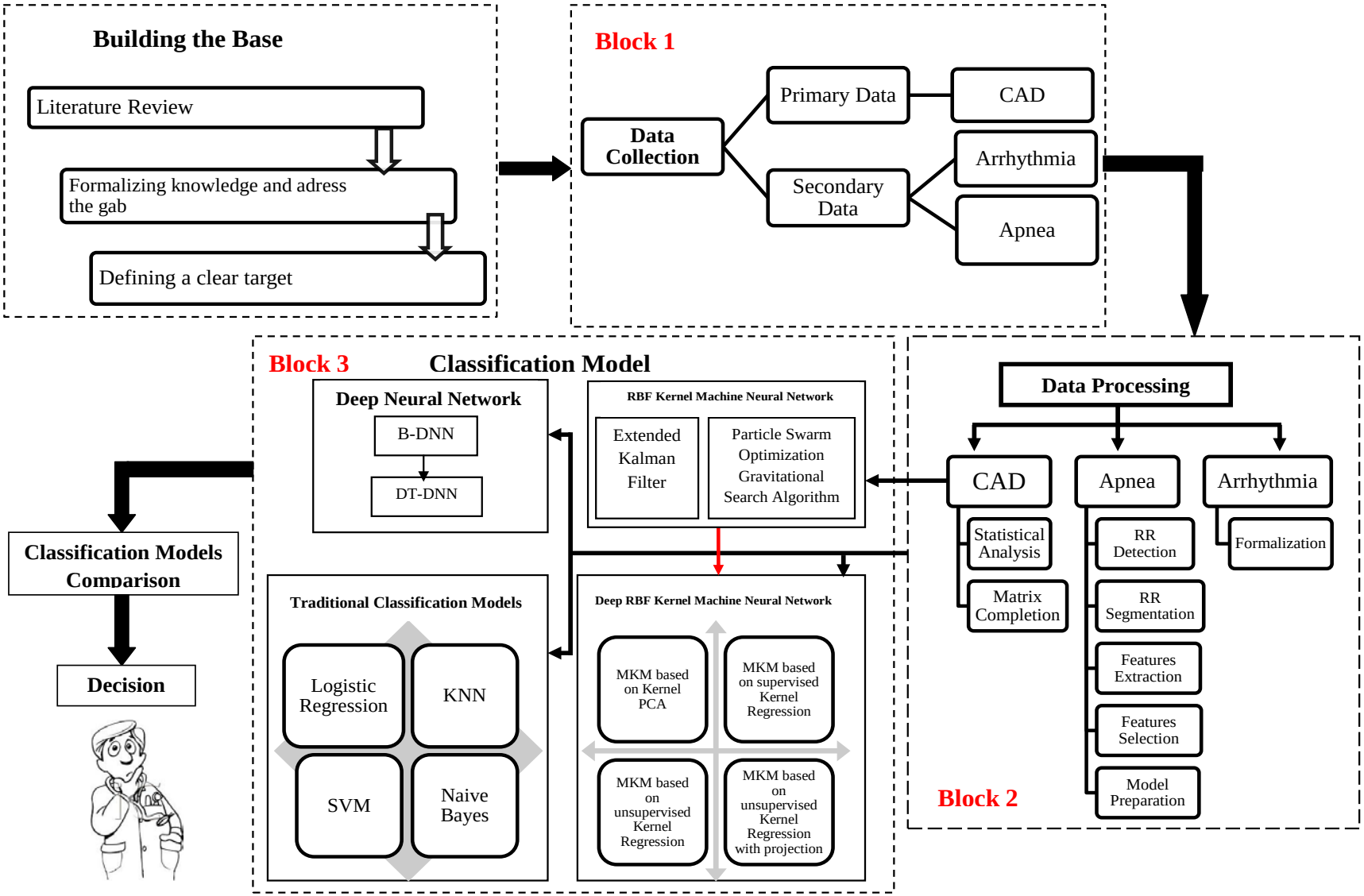


Figure 4.1: Proposed Methodology of the research

such as the Cardiovascular, Neurosicences, Oncology & Specialized Surgery Centres. The second campus is the Oncology Centre, located in Jeddah. The third campus is a 1000-bed quaternary healthcare facility that will be built between Jeddah & Makkah and hosts a specialized hospital, rehabilitation hospital, research centre and an educational center (KAMC, 2017). Figure 4.2 shows the site of the research: at the King Abdullah Medical City, Saudi Arabia.

Figure 4.2: Research study location

4.4. Data Collection

This section introduces the types of data collected in this research project, which were used to answer the research questions and to achieve the objectives. After defining and identifying the research problem and the research design, the task of data collection took place. The method of data collection used for the study dealt with two types of data: primary and secondary, as shown in Figure 4.1, Block 1.

4.4.1. Primary Data

Data which is new, original and collected for the first time is termed as primary data (Kothari, 2004). The Social Dimensions of the Watershed Planning (2006) claimed that

those data types are designed to address an issue or information need in any of the present sources (Malhotra & F., 2006).

To develop the proposed system for CAD diagnosis, primary data which is made up from the details of patients gathered from King Abdullah Medical City electronic resources was collected.

CAD Dataset Description

This dataset is based on the Saudi Arabia population in King Abdullah Medical City collected with the objective of showing the relationship between CAD and other variables. It consists of 688 records with 60 different features demographical like age, gender, occupation and physical height, weight, smoking habits, medical history among others. Each data record is annotated to be either a patient with CAD or without CAD.

4.4.2. Secondary Data

Secondary data is defined as data which has been collected by another party, and could have been subjected to statistical processing/transformation (Kothari, 2004). Organisations collect and store diverse data in order to support their operations (Sounders, et al., 2009). This paper highlighted some of the advantages of using secondary data in a research project, which included having fewer resource requirements, being inconspicuous, being practical for longitudinal studies and offering comparative and contextual data.

The secondary data for this research project, particularly for Arrhythmia and Sleep Apnea Diagnosis, are collected from online databases. The following two dataset are used for this research as secondary data sets:

Arrhythmia Dataset Description

This data set was obtained from the UCI Machine Learning Repository (Lichman, 2013). This data contains 452 records each with 279 attributes. This dataset is used to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. Class 01 refers to 'normal' ECG, classes 02 to 15 refers to different classes of arrhythmia and class 16 refers to those records which cannot be classified.

Sleep Apnea Dataset Description

This data was obtained from the PhysioNet Apnea-ECG database (Goldberger, et al., 2000). It contains ECG recordings for 70 different patients with Obstructive Sleep Apnea (classes a, b, c). Recordings vary in duration from slightly less than 7 hours to nearly 10 hours each. However, only 35 of these recordings contained minute-wise apnea annotations, which indicate the presence or absence of apnea during each minute of ECG data.

4.5. Data Processing

At this stage, data needed to be processed to gather the required features which were then fed into the training model. Processing was required only for CAD and Apnea-ECG datasets, since the Arrhythmia dataset was already pre-processed at the source and obtained as ready features. The processing steps for CAD and Apnea-ECG datasets are presented below and shown in Figure 4.1, Block 2.

4.5.1. CAD Dataset Processing

Before being applied to the classification system, the data set needed to be pre-processed to identify a relationship between existence/non-existence of CAD based on many variables.

Two steps were completed; (1) digitization, at which data was encoded into numerical format, and (2) missing data completion, where the Exact Matrix Completion via Convex Optimization method was selected to improve the data quality and availability, based on its success to evaluate the sparse matrices.

4.5.1.1 Statistical Data Analysis

- *Data Analysis*

There were 60 variables and 688 observations in the data set collected. This was initially analysed by means of frequency distributions and graphs to understand the general nature of the data and to determine the optimal statistical model to test the hypotheses. Logistic regression was the main analysis used to test the hypotheses. As CAD would be the target variable (dependent variable) of the logistic regression model, the observations with missing CAD status data were removed from further analysis. CAD was measured on a dichotomous scale, with the two categories being mutually exclusive - satisfying the prior assumptions of logistic regression.

There were 59 independent variables in the data. The frequency distributions of each of them were studied to discover if further modifications would be required to fit the model, and to eliminate data entry errors. Mismatched entries were found, and were considered as "no information" and tagged as 0 in required cases. The distribution was made of independent categorical variables, after cleaning up the data. Among the 687 observations, 402 came from males and 117 had a history of stroke. 124 people were smokers; whereas, 197 were previous smokers.

There were a few continuous variables in the data. Their range, central tendency and dispersion were studied to ensure proper generation of those variables. These variables were used directly with the logistic model, as they were sufficient for the elementary assumptions of a logistic regression model. However, the model results would only give directional overview – such as "if Blood Urea Nitrogen (BUN) increases, the chances of CAD also increase". On the other hand, if they could be transformed into categorical variables, the model results would provide strategic overview, for example: "People having BUN within 60-80 years have more chances to have CAD". The distribution of the independent variable is shown in Table 4.1.

Table 4.1: Distribution of continuous independent variables

Variable	Mean	Median	Min	Max	Standard deviation
BGC	79.87	0	0	999	122.54
BUN	10.23	0	0	999	39.62
CH	133.60	135	0	340	60.60
ComputeBMI	28.28	27.72	0	114.06	9.93
FBS	37.14	0	0	430	67.35

HB	8.41	11.5	0	26.4	6.57
HDL	35.61	35	0	346	21.35
Hight	159.74	163	0	999	54.94
LDL	87.16	89	0	552	47.58
PPBS	38.84	0	0	1174	97.10
RBC	3.22	4.3	0	85	4.51
TG	121.11	112	0	722	78.31
WBC	5.10	5.6	0	96.5	5.59
Weight	76.05	74	0	999	43.03
Age	51.70	53	0	999	40.08
diastolicHTN	53.09	76	0	114	38.80
systoilcHTN	86.69	120	0	200	63.34
timeofexcercise	14.23	0	0	600	42.56

Considering that most of these continuous variables are very important in the medical context; they were transformed into categorical variables “to be applicable, no medical variable can have a value 0.” Hence, all 0 values were considered as “No Information”. Following is a brief description of how these variables were transformed into categorical variables, after the classification of the distribution of medical variables had been made.

- *Statistical Methodology*

The data was prepared for the model and the Information Value (IV) of each variable was calculated, which helped to eliminate variables from the model. IV is a measure equivalent to correlation analysis, but unlike correlation, it works only for categorical variables. IV indicates the predictive power of the variable. Table 4.2: **IV Values** shows the IV values for variables with its predictive power.

Table 4.2: IV Values

Information Value	Predictive Power
< 0.02	useless for prediction
0.02 to 0.1	Weak predictor
0.1 to 0.3	Medium predictor
0.3 to 0.5	Strong predictor
>0.5	Suspicious or too good to be true

*Note: The variables with low IV will not be used in the model

The second test required for variable elimination is checking multi co-linearity using the Variance Inflation Factor (VIF). If the value of the VIF for any variable is higher than 3, the variable is likely to be correlated with any of the other variables, and will have adverse impact on the model results. The original dataset was used for this operation to extract the affected attributes. VIF of variable is shown in Table 4.3.

Table 4.3: VIF of the variables

Coefficients[a]		
Feature	Collinearity Statistics	
	Tolerance	VIF
Smoking	.563	1.777
age	.761	1.313
gender	.619	1.614
Weight	.415	2.410
ComputeBMI	.180	5.545
BMIGroup	**.166**	**6.026**
Hight	.523	1.914
typeofsmoking	.476	2.102
dursmoking	.763	1.310
Squitting	.479	2.087
NOciggateD	.587	1.704
Exercise	.607	1.648
timeofexcercise	.656	1.524
000000000000	.500	2.000
systoilcHTN	**.051**	**19.633**
diastolicHTN	**.054**	**18.460**
DM	.449	2.229
BGC	.542	1.844
PDM	.523	1.911
PCVD	.553	1.808
ObeseR	.759	1.318
Anemia	.757	1.321
Stroke	.733	1.365
HF	.917	1.090
Amlodipine	.911	1.097
Enoxaparinclexame	.700	1.429
Asprin	.353	2.834
Atrovastatin	.586	1.706
Cerivastatin	.476	2.100
Fluvastatin	**.014**	**72.731**
Pitavastatin	.871	1.147
Pravastatin	**.008**	**126.381**
Rosuvastatin	**.049**	**20.235**
Clopidogreal	**.175**	**5.699**
Pantoprazole	**.190**	**5.251**
Nitroglycerin	**.037**	**26.764**
perindoprilarginine	**.214**	**4.675**
Angiography	**.238**	**4.210**
LDL	.529	1.890
CH	.403	2.482
HDL	.704	1.421

TG	.733	1.365
FBS	.731	1.368
PPBS	.712	1.405
WBC	.314	3.181
HB	.287	3.490
RBC	.723	1.383
BUN	.932	1.073
Albumin	.939	1.065

a. Dependent Variable: CAD

The variables highlighted (in bold font) were not used in the final model. The ComputeBMI and the BMIgroup variables were correlated (r = 0.9).. Similarly, Systolic HTN and Diastolic HTN were correlated (R =0.961). Hence, all of them were kept.
The next step was to build the logistic regression model, with CAD as the target variable.

Table 4.4: Concordance values

Observed		Predicted		
		CAD		Percentage Correct
		0	1	
CAD	0	449	39	92.0
	1	66	133	66.8
Overall Percentage				84.7

As shown in Table 4.4, the overall model concordance was 84.7%, which indicated that the model predicted 84.7% of observations correctly and was statistically good. Any concordance value >60% was considered good.

Table 4.5: Model Summary

Step	Model Summary		
	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	485.719*	.391	.559

Note: The pseudo R-square value that determines the goodness of fit of the logistic model

It is clear from Table 4.5 that the pseudo R-square value of the model is 0.559 which was moderate. The higher the pseudo R-sq the better the model, with R-square ranging from 0 to 1.

Table 4.6: Goodness of fit – Hosmer-Lemeshow

	Hosmer and Lemeshow Test		
Step	Chi-square	df	Sig.
1	4.348	8	.824

Note: This is another test of goodness of fit. The bigger the value of Sig. (Significance), the better the model is. 0< significance <1.

Table 4.6 presents the goodness of fit-Hosmer-Lemeshow test. Results show that this test was satisfied. Any value > 0.05 indicated a good fit of the model. Here the value of HS was 0.824.

The model result provided the following p-values for each variable:

B	S.E.	Wald	Df	Sig.	Exp(B)

- **B** is the coefficient of the variable.
- **SE** is the standard error of the variable.
- **Wald** is the chi-sq value that determines the significance of the variable - a higher chi-sq means a more significant variable.
- **Df** is the degrees of freedom of that variable.
- **Sig**. is the p-value - the lower the p- value, the higher the significance.
- **EXP(B)** is the impact of the variable on the target.

The variables that have p-values < 0.1 were statistically significant at the 10% level of significance.

At the end of the analysis and under the consultant supervision by Dr Osma from King Abdullah Medical City, 21 effected attributes were highlighted and assigned in the diagnosis of CAD.

- *Summary of analysis results*

The variables that significantly impacted CAD diagnosis were as follow:

- Amlodipine: those who have taken Amlodipine have a higher risk of developing CAD
- Enoxaparinclexame: those who have taken Enoxaparinclexame have a risk of developing CAD.

- HF: those who have reported ‘yes’ to HF have a higher risk of developing CAD.
- Rosuvastatin: those who have taken Rosuvastatin have a higher risk of developing CAD.
- Smoking: smokers who have taken Rosuvastatin have a higher risk of developing CAD.
- Stroke: people who have a history of stroke have a higher risk of developing CAD.
- Age: people between 26 – 40 years are in the low-risk zone of CAD
- Weight: weight overall is a significant factor associated with CAD, but no age-group has been identified as more/less risk prone.
- BMI: people who have perfect weight are at much lesser risk than underweight or overweight people.
- HDL: unlike overall cholesterol level and bad cholesterol LDL, good cholesterol HDL is a significant factor of CAD. People having lower HDL, <59 mg have a higher risk of developing CAD.
- FBS: whoever has above normal FBS are in a high risk zone fordeveloping CAD.
- PPBS: people having a slightly higher measurement on PPBS are at higher risk of developing CAD than those who have very high or normal measurements
- BUN: BUN overall is a cause of CAD, but no significant measurement group is identified as being at high risk.

In addition to that, other diagnosis outcomes when CAD is present are listed in Table 4.7. More detailed statistical analysis is provided at Appendix II.

Table 4.7: Top 10 other diagnosis outcomes when CAD is present

Other Diagnosis	Count
MI	140
unstable angina	27
essential primary hypertension	9
atrial fibrillation	10
Cardiomyopathy	33
Angina	22
angina pectoris	13
Arrhythmia	40
Chest pain	43

dilated cardiomyopathy	65

4.5.1.2 Pre-processing

Due to several entry deficiencies in recording, interview or manual entry, patient data can contain some anomalies - in certain cases missing several or one of the major contributing fields (e.g., high level of missing data, which is inherent in the sets of data often used in CAD diagnosis/prediction as these data sets come from multiple sources, e.g., oral interviews, doctors' examinations and technical measurements with different instruments.

Consequently, the estimation process can be disturbed. To avoid this, it would seem the patient's information should be discarded. However, patient information is generally confidential and cannot be discarded by the source. In this case, the use of Matrix Completion technique seems very useful to improve the prediction efficiency. Although this has not provided a high percentage of accuracy, it has helped to use patient information with lower missing field numbers. Thus, there was a higher confidence in the Matrix Completion results in this study.

In this work, the Exact Matrix Completion via Convex Optimization method was selected to improve the data quality and availability, based on its ability to evaluate the sparse matrices using the convex optimization problem.

Since the data obtained is real time information coming out of the hospital data, and the main source of recording are manual and oral interviews and when copying data from several sources there is a high probability of either missing or erroneously entered information.

Since each value in the information set for a patient contributes towards predicting the correct weight and centre of the basis function. Therefore, the correctness of all the available data value is paramount for good performance of the classification algorithm. The first intuition is to remove incorrect or aberrated data, however this means reducing the information to an already scarce data set for the training algorithm to work. Therefore recovery of such data is paramount. In general, it is agreed that recovering a data matrix

from a subset of its entries is impossible. However, if the unknown matrix is known to have a low rank or approximately low rank, then accurate and even exact recovery is possible by nuclear norm minimization. This problem of missing entries is inferred to be matrix completion. A matrix is useful for estimation if it has a high rank, rows and columns has separate ranking. The ranking of the row is determined by the number of linearly independent rows in the matrix, while the limiting factor is the number of dependent rows. For example, if a matrix has three rows, two of them are linearly independent, then it must have a minimum ranking of 2. However, there is a possibility that in combination the three have some linear dependency, therefore this will give the minimum row rank of two however it will be less than 3.

One possible solution, is that the erroneous or missing data does not disturb the estimation process is to discard the patient's information. However, there are ways to overcome this problem by trying to provide an estimate for the missing information. The use of Matrix completion is one of the solutions.

- *Matrix Completion*

"In mathematics, **matrix completion** is the process of adding entries to a matrix which has some unknown or missing values" (Johnson, 1990).

There are few proposed methods which were investigated:

1. Singular value Thresholding based on the rank minimization (Cai, et al., 2010).
2. Augmented Lagrange Multiplier (ALM) Method, this is normally used for inexact or faulty matrices (Lin, et al., 2011).
3. Exact Matrix Completion via Convex Optimization (Candes, 2012)

The last one was selected based on its success to evaluate the sparse matrices using the convex optimization problem and one of the MATLAB toolbox was used for this work.

TFOCS Toolbox

There have been several techniques applied to Matrix completion and one of these is the construction of first-order methods for a variety of convex optimization problems. The

development of this was motivated by the author's interest in compressed sensing, sparse recovery, and low rank matrix completion; see the companion paper (Becker, et al., 2011). In this work the toolbox TFOCS has been used for matrix completion.

The core TFOCS routine tfocs.m supports a particular standard form: the problem is to minimize:

$$\varphi(x), f\,(A(x) + b) + h(x)\,(1)$$

Where *f* and *h* are convex, *A* is a linear operator, and *b* is a vector. The input variable *x* is a real or complex vector, matrix, or element from a composite vector space.

The function *f* must be smooth: its gradient $\nabla f\,(x)$ must be inexpensive to compute at any point in its domain. The function h, on the other hand, must be what we shall henceforth call prox-capable:

$$\emptyset(h) = argmin + \frac{1}{2}t^{-1}\langle z - x, z - x\rangle\,(2)$$

It must be inexpensive to compute its proximity operator for any fixed x and t > 0. In (WHO, 2015), this calculation is referred to as a generalized projection, because it reduces toba projection when h is an indicator function. A variety of useful convex functions are prox-capable, including norms and indicator functions for many common convex sets. Convex constraints are handled by including bin h an appropriate indicator function; unconstrained smooth problems by choosing h(x) ≡ 0; and concave maximizations by minimizing the negative of the objective.

To briefly discuss the explicit inclusion of an affine form $A(x) + b$ into (1). Numerically speaking, it is redundant: the linear operator can instead be incorporated into the smooth function. However, it turns out that with careful accounting, the number of times that A or its ad joint A are called during the evolution of a typical first-order algorithm can be reduced. These savings can be significant when the linear operator is the most expensive part of the objective function, as with many compressed sensing models. Therefore, a separate affine form should be encouraged whenever possible, though it is indeed optional.

A simple example, the LASSO problem was specified by Tibshirani:

$$minimize \quad \frac{1}{2}\|Ax-b\|_2^2$$

$$subject o \|x\|_1 \leq T$$

Where $A \in R^{mXn}, b \in R^m,$ and $\tau > 0$ are given; A can be supplied as a matrix or a function handle implementing the linear operator. This can be rewritten as:

minimize $1 \leq \tau$ and $+\infty$ otherwise.

Because the TFOCS library includes implementations of simple quadratics and 1 norm balls, this model can be translated to a single line of code:

x = tfocs(smooth_quad, { A, -b }, proj_l1(tau));

Sequence for Matrix Completion

Before implementing the MC algorithm, the data file, which is normally an excel sheet with mxn rows and columns, was prepared. The entries which is either missing or erroneous must be empty cells otherwise the algorithm will treat it as a known value. Therefore, the excel file was first pre-processed and erroneous data removed from it. The first step after execution was to enter the value of mu or mean. Once the excel file was read its contents were stored in the appropriate variable, the size of the matrix was determined. Later the known and the empty values were stored in separate variable for processing.

The permutation of the size of the matrix was calculated using the *randperm* MATLAB function, (e.g. if the size of the matrix mxn is 1000 then it will give a permutation of 1:1000 in random numbers). The value stored in the vector with known values was picked up using the column numbers generated by the random permutation. This meant that a randomized set of values were picked up so as not to get a particular biasing when predicting the missing values.

The next step was to find the random permutations of the numbers from 1 to number of rows and 1: number of columns. Permuting the columns to make available entries (columns) randomly spread out (i.e. made the TOFCS algorithm work better).

The main function which calculates the missing values was the solver_NuclearBP. The fitting and rigour of this algorithm depends upon the value of mu which was set in the range 0.01 to 0.0001. The mu value defined the mean error which was tolerable. Although this was an offline algorithm and a minimum value of the mean could have been used. However, this would have caused over fitting and would have tried to fit the noise values spread around the actual values giving a wrong fitting.

The solver_NuclearBP (Nuclear norm basis) function solved the pursuit problem (i.e. matrix completion), by smoothing. The usage of the function was in the format as

[x, out, opts] = solver_sNuclearBP(omega, b, mu, X0, Z0, opts)

Solving the smoothed nuclear norm basis pursuit problem

$$minimize\ norm\_nuc(X) + 0.5 * mu * norm(X - X0, 'fro').\hat{}2$$

$$\text{s.t.} \quad A\_omega * x == b$$

by constructing and solving the composite dual

$$maximize\ -g\_sm(z)$$

where

$$g\_sm(z) = sup\_x < z, Ax - b > -norm(x, 1) - (1/2) * mu * norm(x - x0)$$

A_omega is the restriction to the set omega, and b must be a vector. The initial point x0 and the options structure opts are optional.

The "omega" term may be in one of three forms:

(1) OMEGA, a sparse matrix. Only the nonzero pattern is important.

(2) {n1,n2,omega}, a cell, where [n1,n2] = size(X), and omega is the vector of linear indices of the observed set

(3) {n1,n2,omegaI,omegaJ}, a cell. Similar to (2), except the set omega is now specified by subscripts.

Specifically, omega = sub2ind([n1,n2], omegaI, omegaJ) and

[omegaI,omegaJ] = ind2sub([n1,n2], omega)

If the options field "largeScale" is provided and set to true, then a Lanczos-based SVD is used. In this implementation, we have used the number 1 option as we have non-zero pattern which is important.

4.5.2. Arrhythmia Dataset Processing

This dataset is a complete and error- free dataset. Therefore; no need for any processing technique to be applied other than formalisation. The data is obtained from the source, formalized and then fed to the classification model.

4.5.3. Sleep Apnea Dataset Processing

The obtained dataset was pre-processed through several stages to be fed into the training model. Firstly, the ECG signals were processed to obtain the RR intervals, which were defined as the time between two consecutive R peaks. These RR intervals were the basic metric to obtain data features (11 features were extracted for each data record). Then by filtering the most relevant features were obtained, which in turn were applied to the classification system.

4.5.3.1 RR Intervals Detection

The features used for this research were all metrics based around RR intervals. An RR interval is defined as the time between two consecutive R peaks (Figure 4.3), which in turn are defined as the maximum amplitude of a given QRS complex. QRS is defined as the deflections in an electrocardiogram (EKG) tracing that represent the ventricular activity of the heart. It is the combination of the Q wave, R wave and S wave and represents ventricular depolarization. The normal duration of the QRS complex is 0.08 and 0.10 seconds. When the duration is between 0.10 and 0.12 seconds it is intermediate or slightly prolonged. QRS duration of greater than 0.12 seconds is considered abnormal

(MedicineNet, 2016). These metrics were chosen because RR intervals have been shown to be a telling indicator of HRV, which is a known by-product of sleep apnea (Thuraisingham, 2006).

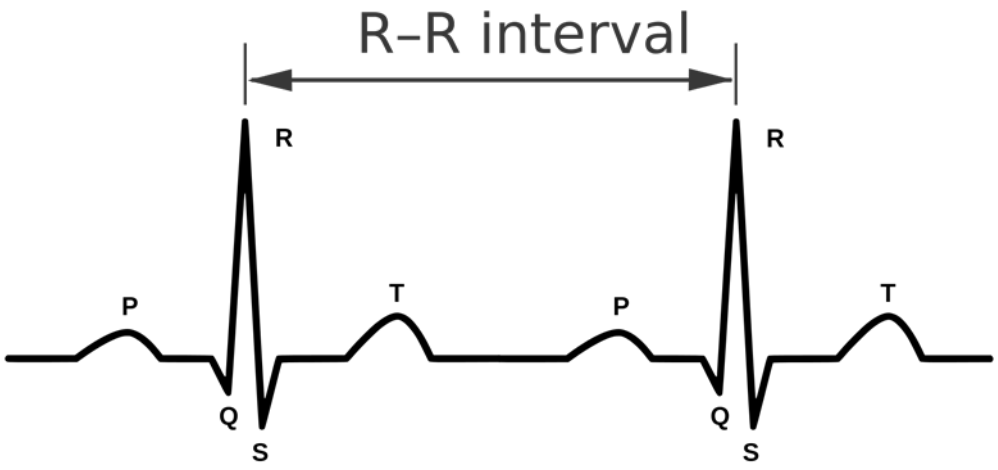


Figure 4.3: Calculation of RR Intervals

The WaveFormDataBase (WFDB) Software Package (WFDB, 2015) was used to extract specific signals and annotations from the ECG recordings files. WFDB is a large collection of software to access PhysioBank and for viewing, annotation, and analysis of signals. .It also included command-line tools for signal processing and automated analysis. Specifically, in this research the WFDB package was used to extract RR intervals using the “ann2rr” command.

4.5.3.2 RR Intervals Segmentation

As each ECG recording in the PhysioNet database was annotated per minute, the extracted RR intervals were segmented on a minute-by-minute basis per the annotations. Therefore, RR intervals were calculated for each minute at each file; which implied that for this research about 17003 RR records (35 file * file length (450-550 minute)) were created. Figure 4.4 presents the extracted RR values for one of the data set files.

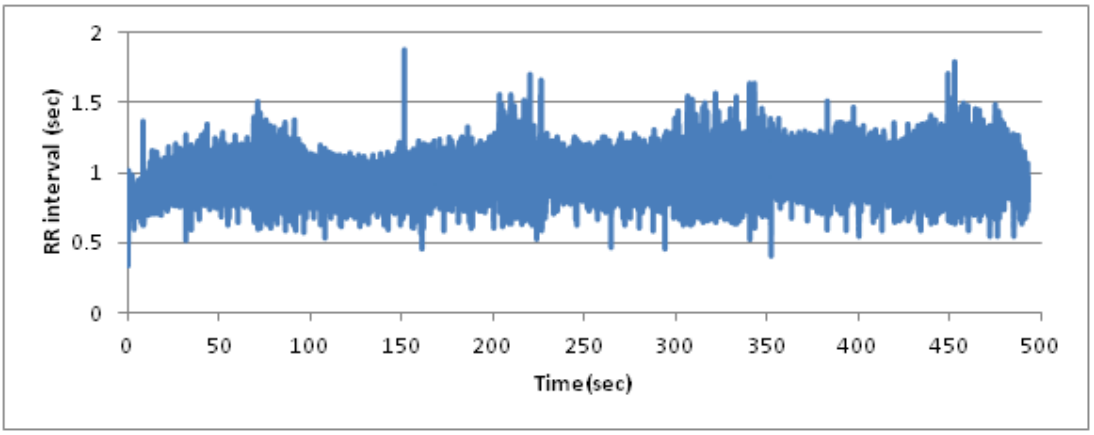


Figure 4.4: Extracted RR Intervals from (a01) file

4.5.3.3 Feature Extraction

Four each segment of the RR intervals obtained from the previous pre-processing phases, statistical features were extracted and fed into the classification model for the possible classification of apnea events. Each feature vector was computed based on 60 seconds of ECG data; as each minute-wise annotation indicated the presence or absence of apnea at the beginning of the following minute. The following ECG features, which are the most common features used in the literature (Chazal, et al., 2004; Kaguara, et al., 2014) for apnea detection, were calculated:

1. Mean of the RR-interval.
2. Median of RR-intervals.
3. Standard deviation SD, of the RR-interval.
4. The NN50 measure (Variant 1), defined as the number of pairs of adjacent RR-intervals where the first RR-interval exceeds the second RR-interval by more than 50ms.
5. The NN50 measure (Variant 2), defined as the number of pairs of adjacent RR-intervals where the second RR-interval exceeds the first RR-interval by more than 50ms.
6. The PNN50_1 measures, defined as NN50 (variant 1) measure divided by the total number of RR intervals.

7. The PNN50_2 measures, defined as NN50 (variant 2) measure divided by the total number of RR intervals.
8. The SDSD measures, defined as the SD of the differences between adjacent RR-intervals.
9. The RMSSD measures, defined as the square root of the mean of the sum of the squares of differences between adjacent RR-intervals.
10. Inter-quartile range, defined as difference between 75th and 25th percentiles of the RR interval value distribution.
11. Mean absolute deviation values, defined as mean of absolute values by the subtraction of the mean RR-interval values from all the RR interval values in an epoch.

Figure 4.5 presents the average value of the extracted features from RR intervals that were extracted from the a01 file of the dataset.

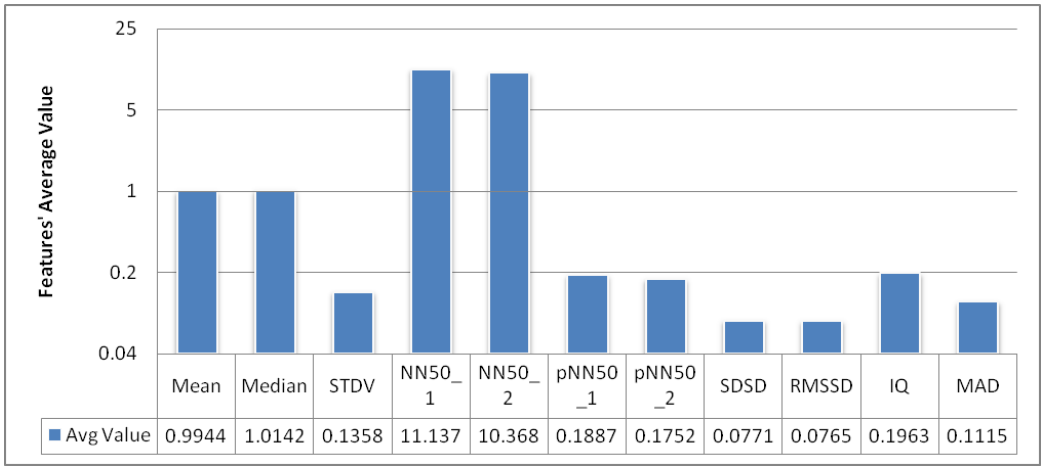


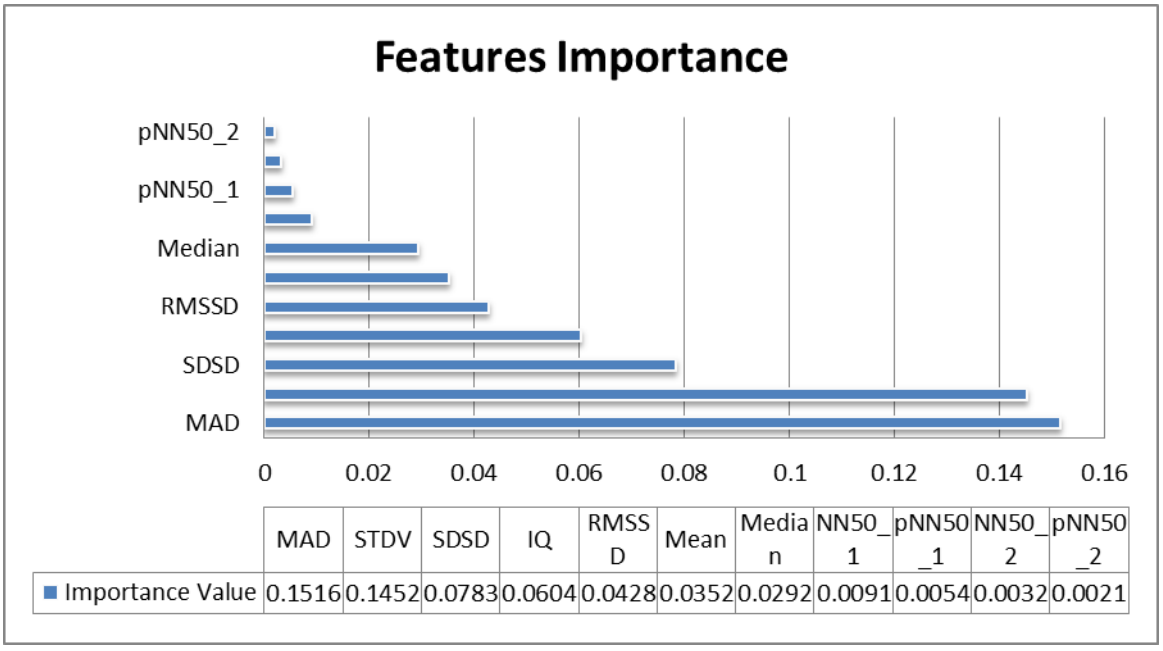
Figure 4.5: Extracted Features from RR Intervals of (a01) file

4.5.3.4 Feature Selection

There exists an important trade-off in performing feature extraction: The more features used will lead to higher levels of classification accuracy, but comes at the price of taking longer to perform apnea detection in real-time. Ranking is crucial for achieving better-quality analysis results. Thus, this research sought to identify the optimal feature subsets from the original features, which was set to minimize the size of the features vector while still being able to classify sleep apnea with high accuracy.

In this phase, the features, that have the strongest effect on prediction, were selected. This stage scored the attributes per their correlation with the classified apnea class. It selected the most informative attributes. In total, 11 features were extracted from each ECG minute. To determine the discriminative power of each feature, the features importance function was adopted. Features importance is a function that ranks features per their significance in predicting the target variable of the classification process. Features with higher values contribute the most in the prediction, while features with values near to Zero do not have high implication on the prediction results. Attributes ranked at zero or less do not contribute to the prediction and should probably be removed from the data (scikit-learn, 2016).

Figure 4.6 presents the results of features importance. The results showed that the Mean absolute deviation (MAD) and Standard Deviation (STDV) were very important features and had the most effect on whether the minute is classified as an Apnea on Non-Apnea minute. While "Median" feature has the less effect on the prediction result, on the other hand the features NN50_1, NN50_2,pNN50_1 and pNN50_2 have an importance value near to zero, which meant that they do not contribute to the prediction so that these four features were eliminated from the features vector so as to have 7 features instead of 11.



	MAD	STDV	SDSD	IQ	RMSSD	Mean	Median	NN50_1	pNN50_1	NN50_2	pNN50_2
Importance Value	0.1516	0.1452	0.0783	0.0604	0.0428	0.0352	0.0292	0.0091	0.0054	0.0032	0.0021

Figure 4.6: Feature Importance

4.5.3.5 Model Preparation

Classification Models were evaluated using stratified 10-fold cross validation. This was a re-sampling technique that provided an estimate of the performance of the model. It did this by splitting the data into 10-parts, training the model on all parts except one which was held out as a test set to evaluate the performance of the model. This process was repeated 10-times and the average score across all constructed models was used as a robust estimate of performance. It was stratified in that it examined the output values and attempt to balance the number of instances that belonged to each class in the 10-splits of the data.

The Pipeline utility was also applied in the model preparation stage. The pipeline is a wrapper that executes one or more models within a pass of the cross-validation procedure. The goal of using pipeline was to ensure that all the steps in the pipeline were constrained to the data available for the evaluation, such as the training dataset or each fold of the cross-validation procedure. To apply pipeline to the model used for this research, the standardization procedure was not only performed on the entire dataset; but was also applied on the training data within the pass of a cross validation run and used to prepare the “unseen” test fold. This made standardization a step-in model preparation in the cross-validation process and prevented the algorithm having knowledge of “unseen” data during evaluation.

4.6. Classification Model

In the classification process, several types of classifiers were used and compared. This process consisted of three parts; the first one used the Radial Basis Function Network as a classification model, the second part used traditional classification algorithms such as; Logistic Regression, KNN, SVM, and Naive Bayes, while the last part applied deep learning techniques in the classification process. A Brief description about these models is given in the following subsections as provided in Figure 4.1, Block 3.

4.6.1. Radial Basis Function Neural Networks

RBF networks are a good candidate for training with non-linear data. An RBF network consists of three layers: namely the input layer, the hidden layer and the output layer. In this part of the research, CAD, Cancer and Simple datasets are used to evaluate the model performance. Two methods were used for training. The first method used the Extended Kalman Filter. The second prediction method was PSOGSA - the Gravitational Search Algorithm (GSA). More details about these two methods are explained below:

Method 1: Extended Kalman Filtering

The Kalman Filter gain is a time-varying gain matrix, which was used for the learning procedure. Different training algorithms were used, such as Quasi Newton and Scaled Conjugate Gradient (SCG).

Method 2: Particle Swarm Optimization Gravitational Search Algorithm (PSOGSA)

To overcome the problem of the local minimum in the Feed-forward Neural Networks (FNN); a hybrid combination of a Particle Swarm Optimization (PSO) (Poly et.al., 2007) and Gravitational Search Algorithm(GSA) (Rashedi et.al., 2009) were applied to combine the global best in PSO with the local search capability of GSA for RBF training.

4.6.2. Traditional Classifiers

As range of well-known data mining algorithms were applied to the processed data to explore the behaviour of these algorithms on the extracted features and evaluate the effectiveness of the processed data in the classification process, namely: Logistic Regression, KNN, SVM and Naïve Bayes Classifier.

4.6.3. Deep Neural Networks

In this part of the research, a deep neural network classifier (DNN) was used. This model consists of 4 layers; one input layer, 2 hidden layers and one output layer. A Pipeline was

used to train the standardization procedure on the training data within the pass of a cross validation run and the trained standardisation was used to prepare the “unseen” test fold.

4.6.4. Multilayer Radial Basis Function Kernal Machine

This section presents how kernel methods were extended to hierarchical structures without requiring complicated machinery and as a deep classification. Four algorithms using RBF kernel were explored. These algorithms are shown in Figure 4.7.

Method 1: Multilayer RBF machine based on Kernel PCA

This algorithm worked in three processes: nonlinear transform by RBF kernel, unsupervised dimensionality reduction by kernel principal component analysis (PCA) and feature section by mutual information and this cycle was repeated multiple times to construct the feature hierarchy of MKM.

Method 2: Multilayer RBF machine based on supervised kernel regression

This algorithm was extended from the first one by applying supervised kernel regression and removing the optional step of feature selection because it was done along with projection. For the latent arable regression, the feature extraction was also incorporated in the regression step but it was based not on the output but on the input. Each hidden layer was learned using the Kernel Partial Least Squares regression (KPLS).

Method 3: Multilayer RBF machine based on unsupervised kernel regression

This algorithm was based on an unsupervised latent space and the motivation behind this claim was that unsupervised methods work well with the regular neural networks and unsupervised learning focusing on important patterns from data, regardless of their labels, as it reduces the input dimensionality of data without losing crucial information. For this algorithm this research used the unsupervised method; Kernel parameters selection and dimensionality selection.

Method 4: Multilayer RBF machine based on unsupervised kernel regression with projection

To improve the accuracy of the previous proposed methods, the unsupervised latent regression with projection method was used. The classifier based on this method was built by adopting the following method: (1) the whole training dataset was subdivided into several groups based on the data class labels, (2) for each group individually; the model was then trained in a way which defined the values of the hyper parameters for each kernel. The optimal values of the particular kernel were defined via eigen-values decomposition problem as was previously done for the unsupervised latent regression.

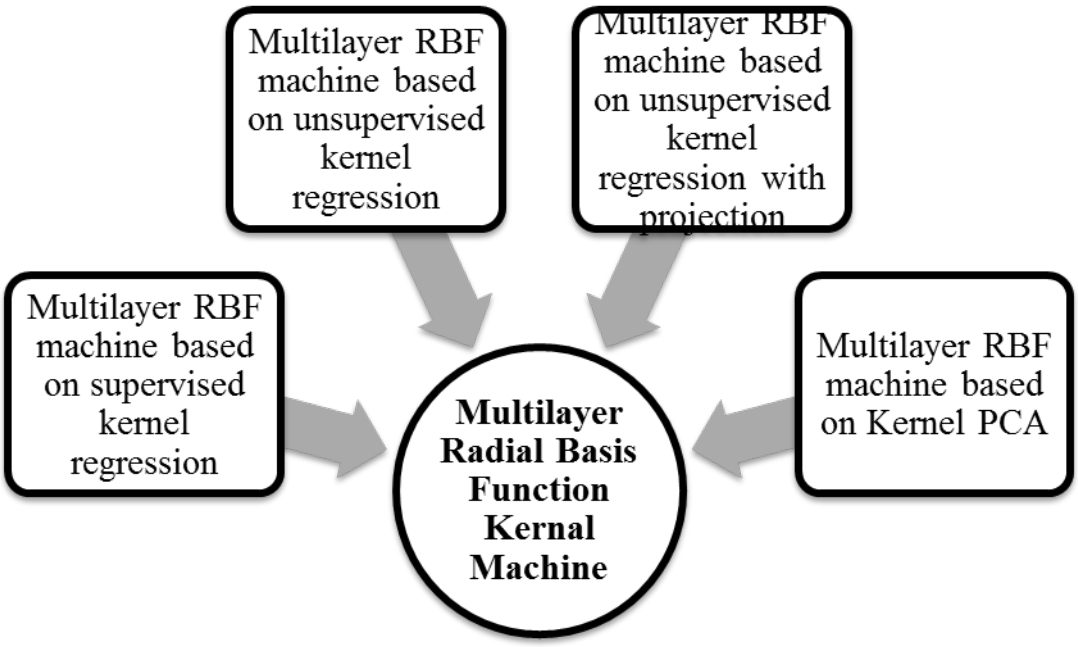


Figure 4.7: Multilayer Radial Basis Function Kernel Machine Algorithms

4.7. Performance Measures

The following metrics (Cornell University, 2003) were measured to evaluate the performance:

- **Accuracy:** the ratio of correctly classified data points to the total number of data points : $\frac{|True\ class\ 1|+|True\ class\ 2|}{|class\ 1|+|class\ 2|}$.

- **Mean squared error (MSE):** the ratio of misclassified data to total number of data points $\frac{|False\ class\ 1|+|False\ class\ 2|}{|class\ 1+class\ 2|}$.

- **Sensitivity:** the ratio of correctly classified elements from class 1 to total number of elements in class 1: $\frac{|True\ class\ 1|}{|class\ 1|}$

- **Specificity:** the ratio of correctly classified elements from class 2 to total number of elements in class 2: $\frac{|True\ class\ 2|}{|class\ 2|}$

- **Cohen's Kappa** is the measure of agreement between the classifier and the ground truth. It is compute by the following formula: $\kappa = \frac{p_o - p_e}{1-p_e}$, where $p_o = \frac{|True\ class\ 1|+|True\ class\ 2|}{|class\ 1|+|class\ 2|}$ - observed agreement which is equivalent to accuracy and $p_e = \frac{|class\ 1|}{|class\ 1|+|class\ 2|} \cdot \frac{|True\ class\ 1|+|False\ class\ 1|}{|class\ 1|+|class\ 2|}$ - expected agreement, the probability of random equivalence.

- **Training time:** time in seconds which were solely used to train the model on specific amount of data with predefined number of layers.

- **Prediction time:** the time in seconds spent solely on the prediction step with pre-trained model containing specific number of steps with fixed amount of training data.
- **Area under ROC**: the area under receiver-operating curve. TP (sensitivity) can then be plotted against FP (1 – specificity) for each threshold used. The resulting graph is called a Receiver Operating Characteristic (ROC) curve

- **Precision**: the proportion of true positives among instances classified as positive. Thus any false positives are also included. A true positive is when the outcome is positive and the actual value is also positive. While the true negative, is when a condition (disease in this case) exists but the classifier outcome is absence. The ratio of the positives gives the performance resolute of a classifier based on the number of correct guesses to the no of guesses it made.

- **Confusion matrix:** is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. This include true positive and negatives. Also to measure the selectivity or recall the ratio of true negative to the false negative and true negative. This provides the fact that how many relevant cases have been detected by the classifier and how many of them are correctly guessed. This ratio of recall gives a measure of inclusiveness.

- **ROC Curve:** a graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as the threshold for assigning observations to a given class and be observed.

4.8. Summary

This chapter presents the research methodology used in current research and the source of dataset used. It also describes the main methodology of data mining and classification process.

The chapter discusses the information pertaining to the Data set obtained from different hospitals located in the city of Riyadh Saudi Arabia. Also, secondary data collected from online sources, are discussed by exploring the advantages they confer; especially giving the fact that other classification methods have been used on them, hence the comparative ease of comparing the performance of the proposed solutions. Besides, the chapter also considers the data pre-processing and feature extraction activities. In case of CAD

missing data, replacement techniques are explored. The discussion on pre-processing includes normalization and thresholding. Additionally, how the Radial basis function and classical thresholding are applied as well as some deep learning algorithms to classify the data set, using training and testing of the neural networks are considered.

The next chapter discusses how the performance measurement techniques such as ROC, confusion matrix and Mean squared error are used to evaluate the performance of the proposed and enhanced algorithms. Furthermore, the following chapters will describe the proposed approaches for diagnosing heart disease based on the combination of Radial basis function neural network and deep learning.

Chapter 5: Prediction of CAD using Radial basis functions in enhancements.

5.1.Overview

In this chapter, the details of the proposed system to automate the diagnostic process of Coronary Artery Disease (CAD) are discussed. Background about the use of Kalman Filtering is presented in Section 2, followed by a discussion of the methodology and the principles of this novel approach of using the Extended Kalman filtering, Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA) for radial basis functions training in section 3. As well, section 3 presents the procedure for testing and verification of the patient dataset. Section 4 presents the results and the analysis of the tests conducted using several combinations of different training algorithms. The final section is a summary of the work and provides some recommendations for advanced work.

5.2. Background

Modern lifestyle habits have significantly increased incidents of cardiovascular disease. Qualified staff available for disease diagnosis in this medical area remains limited and, therefore, are under increased pressure. Fortunately, the diagnosis of complex diseases has become much easier due to progress in computing technologies and artificial intelligence using patient information and the manifestation of their symptoms.

Coronary Artery Disease (CAD) is a complex condition of artery blockage with high mortality figures (WHO, 2015). The prevalence of Coronary Artery Disease (CAD) is increasing across the globe with high costs for governments and other healthcare stakeholders (Karabulut & İbrikçi, 2012). In addition to financial pressures, CAD frequently results in mortality and is one of the world's most prevalent causes of death (Genders, et al., 2012).

A host of factors are used in the diagnosis of CAD, including patient blood pressure, cholesterol level, sugar levels, high BMI (overweight/obesity), physical inactivity, unhealthy eating and smoking (National Institute of Health, 2015). Other factors, such as age, gender, and family history of heart disease, are also likely risk factors for CAD

(British Heart Foundation, 2015). With so many factors involved, detection is challenging because it requires identifying and interpreting the symptoms, risk factors and the patient's medical history. This study was based on real patients in the Saudi Arabia population - in King Abdullah Medical City. Variables are known to have a relationship with CAD were considered and data collected.

When a patient shows symptoms of heart disease, several tests must be immediately done by a doctor (most often an experienced cardiologist) to diagnose for coronary artery disease and prescribe the appropriate treatment regime (National Institute of Health, 2015). This process is generally highly laborious and resource intensive, which makes the diagnosis and treatment very expensive. For this reason, increasing the availability of new smart systems, which facilitate this process, is an urgent priority.

This research investigates the application of a Kalman Filtering (KF) for diagnosing coronary artery disease using two training algorithms for predicting the need for a Coronary Artery Bypass Graft (CABG) in patients identified as having CAD. A second method used is training Radial Basis Function Networks by using a hybrid of Particle Swarm Optimization (PSO) and a Gravitational Search Algorithm (GSA) to solve the CAD prediction problems. Here, the GSA and PSO algorithms are employed as new training methods for a Radial Basis Function Network to investigate the efficiency of these algorithms. The Kalman filter is derived from the data fusion algorithm, which simplifies the recursive calculation of the CAD status and the Coronary Artery Bypass Graft (CABG) requirement (i.e., the two factors of interest). This process uses a combination of knowledge/observations/measurements from patients, predictions from models and considers the inherent noise in the observations/measurements. Non-linear measurements were also involved in the prediction process; thus, an extended variant of the KF (Extended Kalman Filter – EKF) (Ciocoiu, 2002) was also applied in the research. The strength of the EKF is its ability to implement non-linear models (Julier & Uhlmann, 1997), making it an ideal candidate for neural network training.

Most applications of EKF training for neural networks have been for time-series predictions (Okutani & Stephanedes, 1984). Time-series constraints on the data can be

eliminated by using a Radial Basis Function (RBF) neural network architecture designed for classification. The approach used by this research demonstrates the use of EKF with various training algorithms used to train Radial Basis Function Neural Networks for CAD prediction.

Studies show that Coronary Artery Disease (CAD) is increasing across the globe, requiring excessive resources to be used by healthcare stakeholders in trying to manage the disease (WHO, 2015); (Genders, et al., 2012); this research could help develop for more efficient management strategies for the disease.

5.3. Methodology

RBFNs are applicable to different fields such as function approximation, regulation, noisy interpolation, density estimation, optimal classification theory and potential functions (WHO, 2015). RBFNs form a unifying link between the fields above and this causes the training for RBFNs to be substantially faster than the methods used to train Multilayer Perceptron networks (MLPNs). Hence, RBFNs represent an alternative to the widely used MLPNs.

The training in RBFNs can be classified into two stages:

(i) The basis function parameters (corresponding to hidden nodes). Typically, fast and unsupervised clustering methods are used to determine these parameters

(ii) The weights in the final layer. A linear system solution involves in the weight determination.

To design an efficient architecture for the network is a challenging for the neural network community. One of the advantages of RBFNs compared to MLPNs is choosing suitable parameters for the hidden units without involving non-linear optimization of the network parameters. However, the performance of RBFNs depends critically on the number, the position and the shapes of the hidden units (National Heart, 2016). The general way to select the centre of hidden units is to superpose each centre to the data set point. This method incurs a heavy computational cost and leads to poor generalisation of the network (British Heart Foundation, 2015).

The model selection is based on the prediction of how well the trained model will work on the unknown or future values. The network which gives the lowest prediction error will then be recommended for selection.

RBF Neural Network models are a suitable method for classification problems, such as in the detection of diseases (e.g., occurrence of CAD). To address the classification problems, several steps were required. This section provides a brief description of the methodology steps that were followed in the proposed approach, as shown in Figure 5..

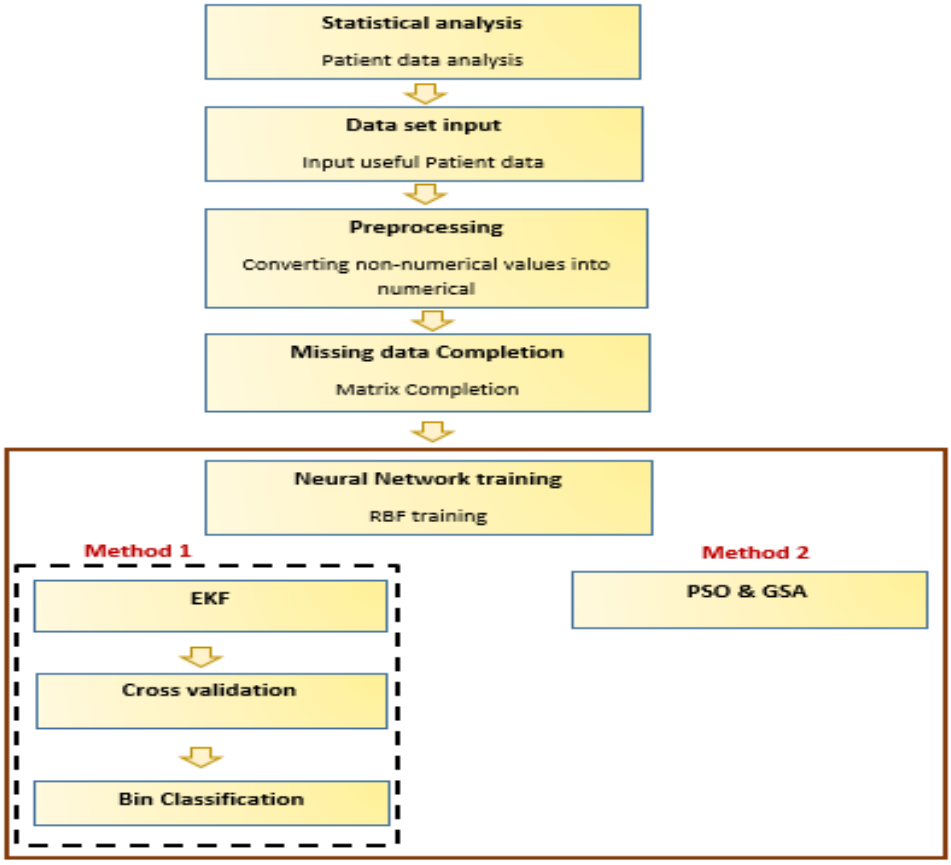


Figure 5.1: Methodology Overview

5.3.1. Data Set Description

There were three datasets used in this part of research. However, the main application was based on the CAD dataset, collected from King Abdullah Medical City. A brief description for each dataset is provided bellow:

- **The King Abdullah Medical City hospital data (CAD)**: details about this dataset were provided in chapter 4 section 4.5.

5.3.2. Training the RBF Neural Network

RBF networks, because of their classification capability, are a good candidate for training with non-linear data. An RBF network consists of three layers: namely the input layer, the hidden layer and the output layer. The input layer broadcasts the coordinates of the input vector to each of the units in the hidden layer. Each unit in the hidden layer then produces an activation based on the associated radial basis function. Finally, each unit in the output layer computes a linear combination of the activations of the hidden units. How a RBF network reacts to a given input stimulus is completely determined by the activation functions associated with the hidden units and the weights associated with the links between the hidden layer and the output layer. In the RBF networks constructed with the proposed learning algorithm, each activation function associated with the hidden unit was built either using TPS or R4RlogR as an activation function.

In this work, two methods were used for training. The first method used the Extended Kalman Filter for the learning procedure, and used different training algorithms, such as the Quasi Newton and Scaled Conjugate Gradient (SCG). The second prediction method is PSOGSA - the Gravitational Search Algorithm (GSA) - which is a novel heuristic optimization method based on the law of gravity and mass interactions.

5.3.2.1 Method 1: Extended Kalman Filtering

The Kalman Filter gain is a time-varying gain matrix. The matrices used were:

- Auto-covariance matrix (for lag zero) for the estimation error of the corrected estimate.
- Auto-covariance matrix (for lag zero) for the estimation error of the predicted estimate.
- The transition matrix A for the linearized model of the original nonlinear model was calculated with the most recent state estimate, which was assumed to be the corrected estimate xc (k).
- Given the continuous-time nonlinear process model. Linearize it at the operating point to obtain A.
- Then calculate A = A discrete as the discretised version of A continuous. Forward method of discretization in manual calculations can be used; however in this case MATLAB is used to discretise this function.

EKF Implementation for current work

The first step was the derivation of the equations on which EKF neural networks training algorithm are based. A neural network (NN) can be described as a non-linear discretised system

$$w_{k+1} .$$

$$w_{k+1} = w_k + \omega_k$$

Where ω_k is the weight vector, and w_{k+1} computation: for k=1,2…, compute state estimate propagation.

The second equation, known as the observation or measurement equation, represents the network's desired response vector y_k as a nonlinear function

$$y_k = h_k(w_k, u_k, v_{k-1}) + v_k$$

Where h_k is the derivative matrix, w_k is the weight vector, u_k is an input training pattern and v_{k-1} is the recurrent node activations v_k from the previous time step for GRBFs.

The input vector u_k, the weight parameter vector w_k, and, for recurrent networks, the recurrent node activations v_k; this equation was augmented by random measurement noise n_k.

The noise (measured) n_k is given as:

$$E[v_k v_l^T] = \delta_{kl} R_k$$

where R_k is the covariance noise matrix.

Similarly, the process noise v_k is given as:

$$E[\omega_k \omega_l^T] = \delta_{kl} Q_k$$

Where Q_k *is the* covariance matrix of the process noise.

The training problem using Kalman filter theory can now be described as finding the minimum mean-squared error estimate of the state 'w' using all observed data so far. Network architecture with M weights and no output nodes and cost function components was assumed.

The EKF solution to the training problem is given by the following recursion:

$$A_k = [R_k + H_k^T P_k H_k]^{-1}$$

$$K_k = P_k H_k A_k$$

$$\hat{w}_{k+1} = \hat{w}_k + K_k \mathcal{E}_k$$

$$P_{k+1} = [P_k - K_k H_k^T P_k + Q_k]^{-1}$$

The vector $\hat{w}_k$ represents the estimate of the state (i.e., weights) of the system at update step k. This estimate is a function of the Kalman gain matrix K_k and the error vector.

$$\mathcal{E}_k = \hat{y}_k + y_k$$

where y_k is the target vector and $\hat{y}_k$ is the network's output vector for the kth presentation of a training pattern.

The Kalman gain matrix is a function of the approximate error covariance matrix P_k, a matrix of derivatives of the network's outputs with respect to all trainable weight parameters H_k, and a global scaling matrix A_k. The matrix H_k may be computed via static backpropagation or backpropagation through time for feedforward and recurrent networks, respectively.

The scaling matrix A_k is a function of the measurement noise covariance matrix $\mathbf{R_k}$, as well as of the matrices H_k and P_k. Finally, the approximate error covariance matrix P_k evolves recursively with the weight vector estimate; this matrix encodes second derivative information about the training problem, and is augmented by the covariance matrix of the process noise$\mathbf{Q_k}$.

This algorithm attempts to find weight values that minimise the sum of squared error.

$$\sum_k \varepsilon_k^T \varepsilon_k$$

Note that the algorithm requires that the measurement and process noise covariance matrices, $\mathbf{R_k}$ and $\mathbf{Q_k}$, be specified for all training instances. Similarly, the approximate error covariance matrix P_k must be initialised at the beginning of training.

Generalised Extend Kalman Filter (GEKF) training was carried out in a sequential fashion as shown in the signal flow diagram below in Figure 5..

One training step is described as follows:

- An input training pattern u_k was propagated through the network to produce an output vectory $\hat{y}_k$. Note that the forward propagation is a function of the recurrent node activations v_k from the previous time step for GRBFs. The error vector j_k is computed in this step as well.

- The derivative matrix H_k was obtained by back propagation. In this case, there was a separate back propagation for each component of the output vector$\hat{y}_k$, and the back-propagation phase involved a time history of recurrent node activations for GRBFs.

- The Kalman gain matrix was computed as a function of the derivative matrixH_k, the approximate error covariance matrix P_k, and the measurement covariance noise matrix $\mathbb{R}_k$. Note that this step included the computation of the global scaling matrixA_k.

- The network weight vector was updated using the Kalman gain matrixK_k, the error vector j_k, and the current values of the weight vector $\hat{w}_k$.

- The approximate error covariance matrix was updated using the Kalman gain matrixk_k, the derivative matrix H_k, and the current values of the approximate error covariance matrix P_k. This step also included the augmentation of the error covariance matrix by the covariance matrix of the process noise Q_k, the procedures shown in block 2, 3 and 4 in Figure 5..

Calculation of Kalman Filter gain

The Kalman filter gain k_k was calculated as follows and is shown in block 2, 3 and 4 in Figure 5.:

- The initial step and the operations here were executed only once. The initial value was set to some guessed value (matrix), e.g., to the identity matrix (of proper dimension)
- Calculation of the Kalman Gain
- Calculation of auto-covariance of corrected state estimate error
- Auto-covariance of corrected state estimate error
- Calculation of auto-covariance of the next time step of predicted state estimate error
- Auto-covariance of predicted state estimate error.

1. **Quasi Newton method**

The Quasi-Newton method is an extension of the Newton optimization algorithm. Unlike the Newton method, which calculates the Hessian (Shanno, 1970) of the function (which is complex, resource heavy and approximation is used between the steps), Quasi-Newton

quickly optimises and is simpler to implement, calculating the minima of the function iteratively. The training of the network was based on different options; these options were adjustable once the network evolved to its optimised set of weights and centres. Quasi-newton not only uses the function but its gradient to find the minima of the function.
One of the optimization goals was to achieve the required performance (in the case of the proposed approach, the MSE) then the training shall stop. However, in certain cases, because of the large amount of variance in the input, achieving this goal was not possible; therefore, there was a way to stop the algorithm at a reasonable time. The conditions on which the training had to be stopped were any one or all of the following occurred:

- The maximum number of set repetitions was reached.
- The maximum amount of time was exceeded.
- Performance was minimised to the goal.
- The performance gradient is below the minimum gradient.
- Validation performance had increased more than the maximum fail times since the last time it decreased (when using validation) and shown in block 3 in Figure 5..

2. Scaled Conjugate Gradient (SCG) method

SCG is a supervised learning algorithm for feed-forward neural networks, which is a member of the class of conjugate gradient methods.

The SCG is one of the conjugate gradient methods; however, it adapts the search direction and step size more carefully, determined by the second order approximation. The other three conjugate gradient algorithms require a line search for every iteration, which is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. The SCG training algorithm was developed to avoid this time-consuming line search, thus significantly reducing the number of computations performed in each iteration, although it may require more iteration to converge than the other conjugate gradient algorithms. The storage requirements for the SCG algorithm are about the same as those of CGF and shown in block 3 in Figure 5..

Procedure

The main theme of this classification solution was to optimise the network and then validate the results.

The available data was divided into training and testing data. This way several models trained on the training set were available to be applied on the test set. The best result on the test set based on these several trained models was then considered as the optimal simulation. This procedure is also detailed in Figure 5..

Cross Validation

The procedure discussed above can introduce a bias towards a particular data set. Therefore, the data set had to be partitioned in smaller sets, these partitioned are then used as testing and training sets interchangeably. This helped to negate the bias, also the average obtained from these different partitions, is then averaged to get a consolidated result. The data set in this system was divided into three sections: training, testing and validation as follows:

- The training set was used to build the model. This contained a set of data, the pre-classified target and predictor variables.
- The testing set was used to evaluate how well the model performed with data outside the training set. The test set contained the pre-classified results data but they were not used when the test set data was run through the model until the end, when the pre-classified data was compared against the model results. The model was adjusted to minimise the error on the training set.
- The validation set was used to evaluate the adjusted model in step 2, where again, the validation set data was run against the adjusted model and results compared to the unseen pre-classified data.

One of the optimization goals was to achieve the required performance (in the case of this research is the Mean Squared Error MSE) then the training was stopped. However, in certain cases because of the large amount of variance in the input it was not possible to

achieve this goal, so there needed to be a way to stop the algorithm at a reasonable time. Therefore, the conditions on which the training must stop was of these cases; either, the performance was minimised to the goal, or the Validation performance had increased more than the maximum fail times since the last time it decreased (when using validation).

Bin Classification

Bin classification was the method used to evaluate the correctness of the algorithm. Since this was a classification problem, regression analysis was not suitable for this work. The error percentage is calculated based on the expected results and the actual results. The expected results, as explained above, were obtained from the current calculations of the feed-forward RBF network.

The calculated values from the network had similar rows and columns as the actual output target data. The requirement for bin classification and error calculation was that there was a minimum of two columns of expected results.

Results and Analysis

Each data set was used for the first method, with Extended Kalman Filtering, and having three different combinations: by swapping the training, testing and validation sets. In this experiment, the two different training algorithms (Quasi-Newton & SCG) were used with the different combinations of the data sets to evaluate the performance of the training algorithms. The accuracy for the three datasets was good. The prediction probability of

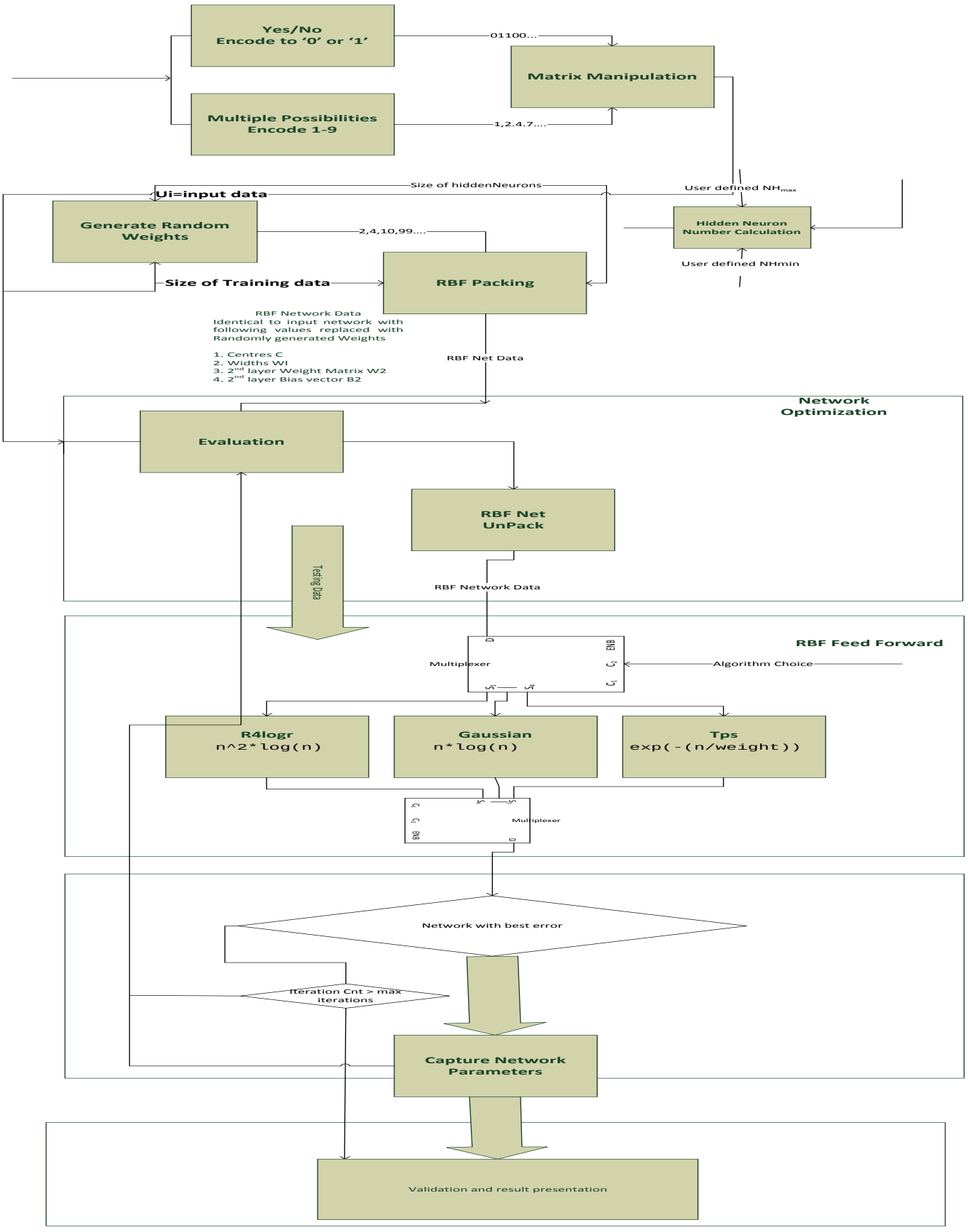

Yes/No
Encode to '0' or '1'
01100...
Matrix Manipulation
Multiple Possibilities
Encode 1-9
1,2.4.7....
Size of hiddenNeurons
User defined NH$_{max}$
Ui=input data
Generate Random Weights
2,4,10,99....
Hidden Neuron Number Calculation
User defined NHmin
Size of Training data
RBF Packing
RBF Network Data
Identical to input network with following values replaced with Randomly generated Weights
1. Centres C
2. Widths WI
3. 2nd layer Weight Matrix W2
4. 2nd layer Bias vector B2
RBF Net Data
Network Optimization
Evaluation
RBF Net UnPack
Testing Data
RBF Network Data
RBF Feed Forward
Multiplexer
Algorithm Choice
R4logr
n^2*log(n)
Gaussian
n*log(n)
Tps
exp(-(n/weight))
Multiplexer
Network with best error
Iteration Cnt > max iterations
Capture Network Parameters
Validation and result presentation

Figure 5.2: Flow of the RBF network build-up to obtain the CAD diagnosis result

this combination resulted in an accuracy of about 92%, using cross validation for the CAD dataset and above 95% in the two other datasets.

1. CAD Data Set

a. Without Matrix completion

The results of the four different combinations are given in Figure 5., and the summary is presented in table 5.1.

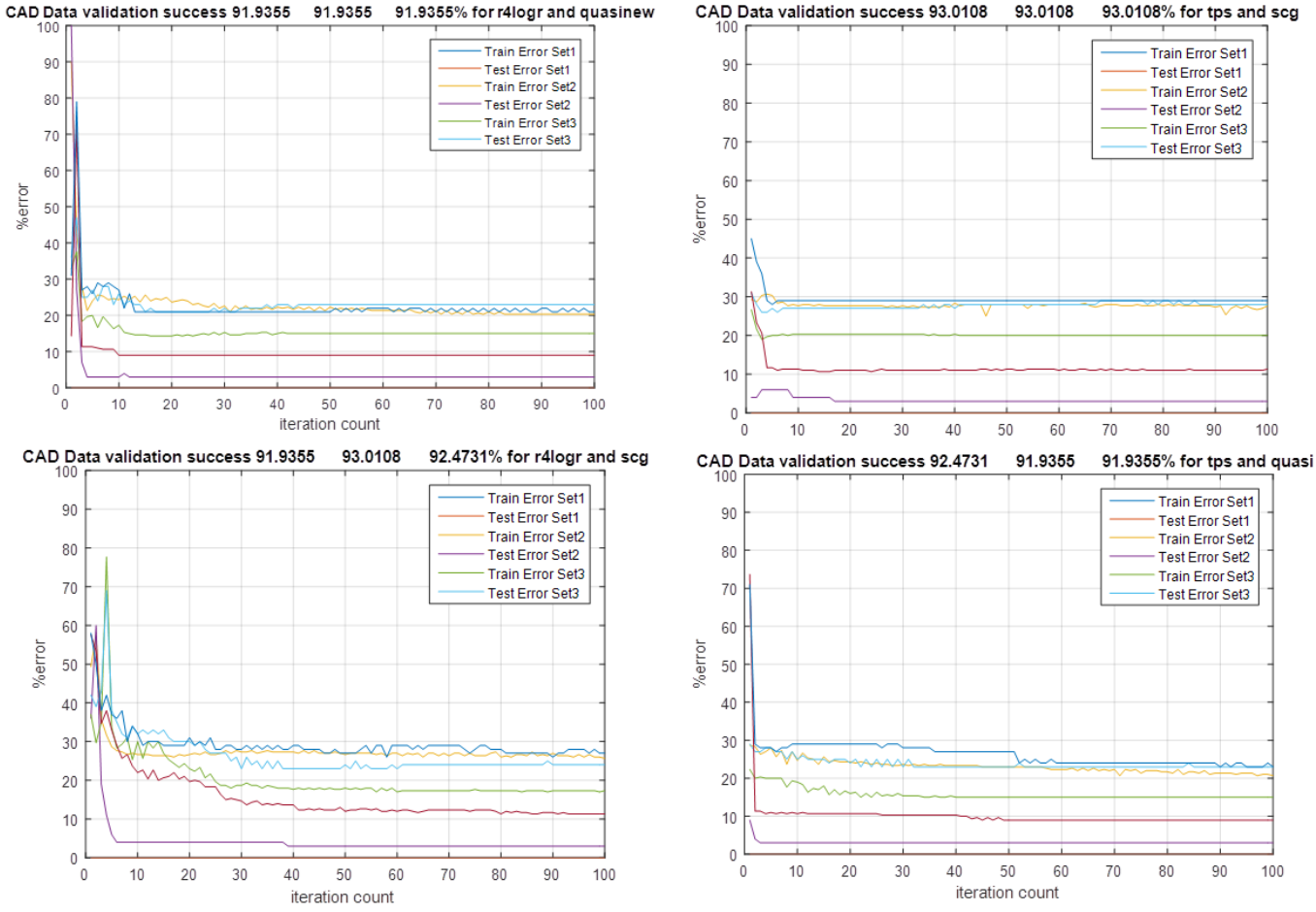


Figure 5.3: Three Combinations of Data sets and 4 different training and Activation functions without Matrix Completion

Table 5.1: Summary of results of the 4 training and activation functions without matrix completion

Training Algo	Act Fcn	Set No 1 Best results					Set No 2 Best results					Set No 3 Best results				
		Train Error %	Test Error%	Valid Error%	Accuracy %	Iteration number	Train Error%	Test Error%	Valid Error%	Accuracy %	Iter num	Train Error %	Test Error %	Valid Error %	Accuracy %	Best iter
Quasi New	R4Rlog	21.33	3.00	8.06	91.9%	4	14.67	21	8.06	91.9%	15	9.00	21.00	8.06	91.9%	13
Quasi New	Tps	26.33	3.00	7.53	92.4%	3	15.00	23.00	8.06	91.9%	22	9.00	23.00	8.06	91.9%	90
SCG	R4Rlog	27.33	3.00	6.99	93.01 %	30	27.33	3.00	8.06	91.9%	39	12.00	26.00	7.53	92.4%	58
SCG	Tps	27.67	3.00	6.99	93.01	17	19.00	26.00	6.99	93.01	3	11.67	28.00	6.99	93.01	5

The following observations have been found

1. Activation algorithm, even with change in data sets the validation error did not increase from 6.99%.
2. The best testing performance was Data set 1, the possible reason was less missing and erroneous data hence keeping the test errors at only 3.00% for all the combinations.
3. The SCG and the R4Rlog combination took the highest number of iterations or processing time to converge to the best result. This means this was computationally slow to converge, however if the validation results are considered, this was the second best out of the four combinations which were used.
4. The training error was still relatively high for all the combinations; however, the Quasi Newton and R4RlogR combination provided the best in terms of training error.
5. Overall the combination of SCG and R4RlogR performed the best, the only downside was that it took far too many iterations to converge, however this was an offline classification application thus in terms of real-time performance this was not in any way a consideration.

b. Matrix Completion

The same data set with similar combinations as above was used except that now the data set was provided after applying Matrix completion and any missing values were predicted and replaced in the data set.

The detailed results of the four different combinations are given in Figure 5. and Table 5., including training errors, testing errors, validation error and the best iteration in each training algorithm for each of the three sets after Matrix completion.

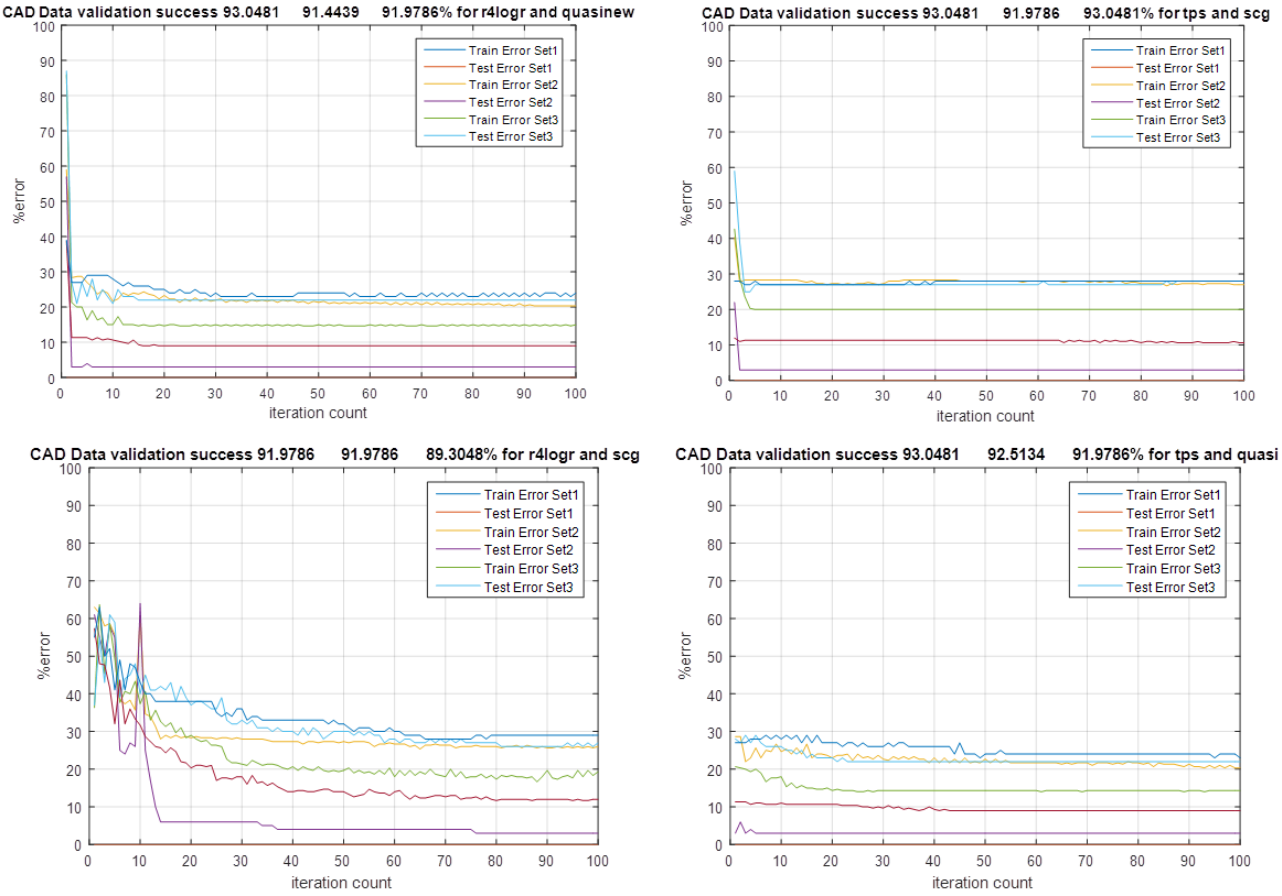


Figure 5.4: Three Combinations of Data sets and 4 different training and Activation functions with Matrix Completion

Table 5.2: Summary of the best result obtained from Figure 5.

		Best results of different data sets														
		Set No 1					Set No 2					Set No 3				
Training Algo	Act Fcn	Train Error %	Test Error %	Valid Error %	Accur %	Iter no	Train Error %	Test Error %	Valid Error %	Accur %	Iter no	Train Error %	Test Error %	Valid Error %	Accur %	Best iter
Quasi New	R4R	28.33	3.00	6.95	93.0	2	20.00	21.00	8.56	91.4	3	9.00	23.0	8.02	91.9	29
Quasi New	Tps	28.67	3.00	10.70	93.0	1	14.67	22.00	7.49	92.5	21	9.00	23.0	8.02	91.9	49
SCG	R4R	26.00	3.00	8.02	91.9	76	11.33	27.00	8.02	91.9	81	12.33	28.0	10.70	89.3	66
SCG	Tps	27.67	3.00	6.95	93.0	2	23.67	25.00	8.02	92.5	3	11.33	27.0	6.95	91.9	3

The following observations were made:

- The accuracy of the combination Quasi New and R4R for the three sub datasets (cross validation) was: 93.0, 91.4 and 91.9, sequentially.
- The accuracy of the combination Quasi New and TPS for the three sub datasets (cross validation) was: 93.0, 92.5 and 91.9, sequentially.
- The accuracy of the combination SCG and R4R for the three sub datasets (cross validation) was: 91.9, 91.9 and 89.3, sequentially.
- The accuracy of the combination SCG and TPS for the three sub datasets (cross validation) was: 93.0, 92.5 and 91.9, sequentially.
- The combination of SCG and TPS produced the best and most consistent set of training and activation algorithms; even with change in the data sets, the validation error did not increase from 6.95%, except the second set, which produced a result of 8.02%.
- The best testing performance is of the Data set 1; the possible reason is fewer missing and erroneous data, hence keeping the test errors at only 3.00% for all the combinations.
- The SCG and the R4Rlog combination produced the highest number of iterations or processing time, with the best result. This combination was computationally slow to converge; however, if the validation results are considered, this was second best out of the four combinations used.

- The training error was still relatively high for all the combinations; this is possibly due to missing information in the data set. However, the Quasi Newton and R4RlogR combination provided the best in terms of training error.
- The highest validation error reached was 10.40% as compared to the non-matrix completed set; however, in general the results found improved by 0.4 to 0.5%, which was quite significant at the top end of predicting a correct result.
- Overall the combination of SCG and R4RlogR performed the best. The only downside was that it took far more iterations to converge; however, this was an offline classification application, thus in terms of real-time performance this was not a consideration.

c. Single Validation Set

The experiment ran for a validation set of size one. Each time a best estimated RBF network was obtained as described in previous experiments, however this time all the data set except one patients information was used for training and testing, once the best network was obtained this was used to estimate the validation error. In this case, it was either a correct or a wrong estimate.

The plot given in Figure 5. shows three hundred patient data sets used for validation. The x-axis is the number of the patient and the y-axis is the error percentage which is either 0 or 100%. Any incorrect estimation is shown as a bar, with the given patient number. The total detected errors are also given on the graph. This will be extended for all the possible patients and with different combinations of training and testing data. Due to the longevity of the run of the test, it was important not to run an exact test for 100 iterations since none of the best-found network is obtained beyond the 40th iteration hence 100 was taken as a pessimistic figure. For this experiment, we limited the experiment to run for 40 iterations.

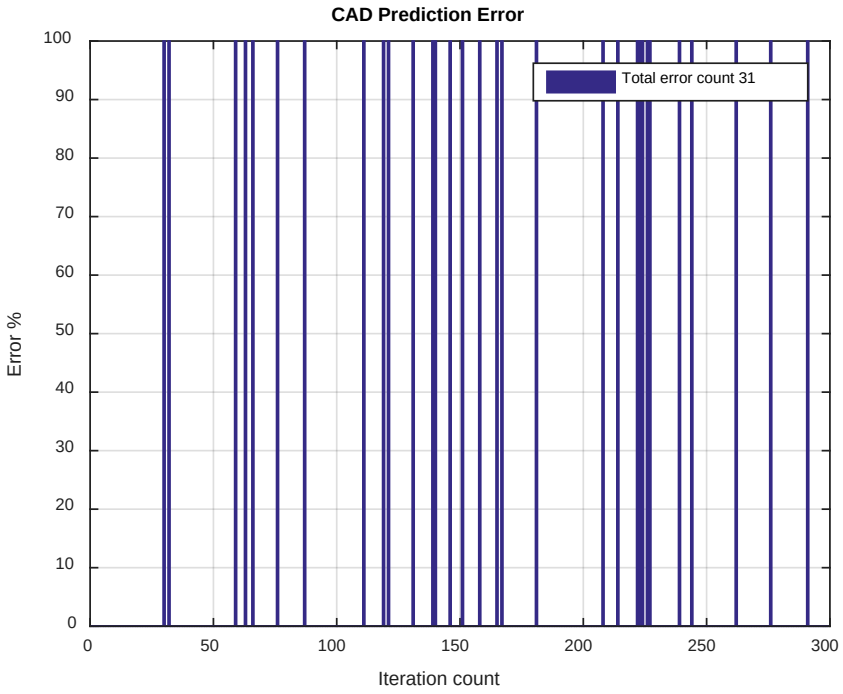


Figure 5.5: CAD Prediction Error

The total number of incorrect predictions was 31 out of 300, which is about 10%.

From the recall point of view this is a good result as plenty of patients data have been used, i.e. with swapping and randomness it has large inclusiveness. However, 90% precision is not good, due to the fact that this gives a 10% True negative cases as per ROC.

This value coincided with the figures which were received during the normal runs where a fixed combination, or three different combinations of training, testing and validation data were used. However, in this case the exclusivity was the fact that only one patient's data was used for the validation each time

d. Expected CAD Statistics

The above error classification method was based on the bin classification, when there are at least two target outputs to be estimated, for example in the previous case CAD and CABG were taken as the output values. However in certain cases when there was only one output then the bin classification method will not work (e.g. as in the last case) if either CAD or CABG needs to be estimated. In case of regression analysis either the

mean square error or the estimation statistics wsas used to determine the correctness of the estimated single output.

In this experiment, the CAD data was divided into three distinct sets of training, testing and validation data. Each set was iterated through the main loop for 100 times to get the best RBF network w.r.t weights and centres of the basis functions. The data sets were then swapped and the iterations are repeated, in this way three distinct sets were obtained and also cross-validated.

For each set (a particular combination) the results (CAD expected values) obtained of the test stage of each iteration were shown as a scatter graph. This helped to understand the convergence and divergence of the data, the best way is to find the three degrees of statistics i.e. mean, standard deviation and the variance. The lower the variance and deviation to the mean indicates that the expected results have less errors and predict the correct values.

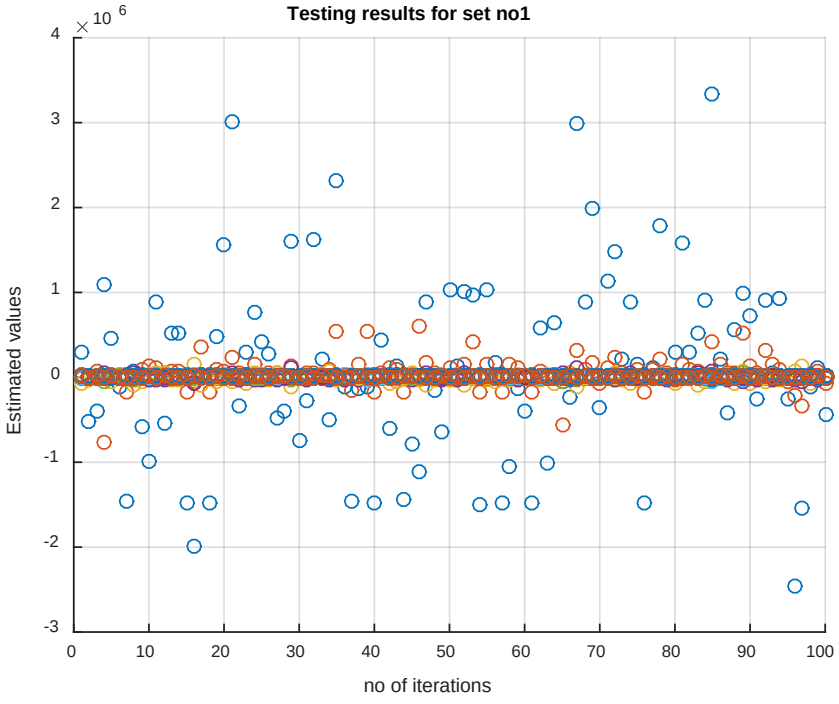


Figure 5.6: Testing results for set no 1

From Figure 5. it is clear that most of the values lie on the zero axis and there are fewer values in the spread and their magnitudes are also not very large.

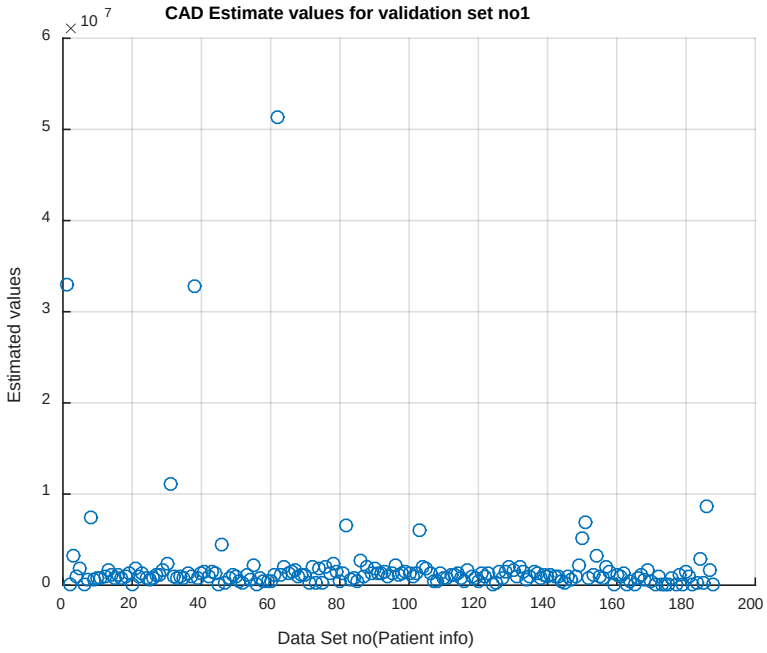


Figure 5.7 CAD estimate values for validation set no 1

Figure 5. above is plotted on the validation target data once using the best network combination obtained from the iterative solution for whose test data values are plotted above. The mean of the values lies near zero with a slight zero bias with a very small amount of spread. There are few odd values however they are in very small numbers, thus giving a good estimate of the predicted values.

The following figures are simply based on dataset which have been partitioned into testing, validation and training data. To provide inclusiveness and indepndence, the

experiments are repeated by swapping the training, testing and validation data sets

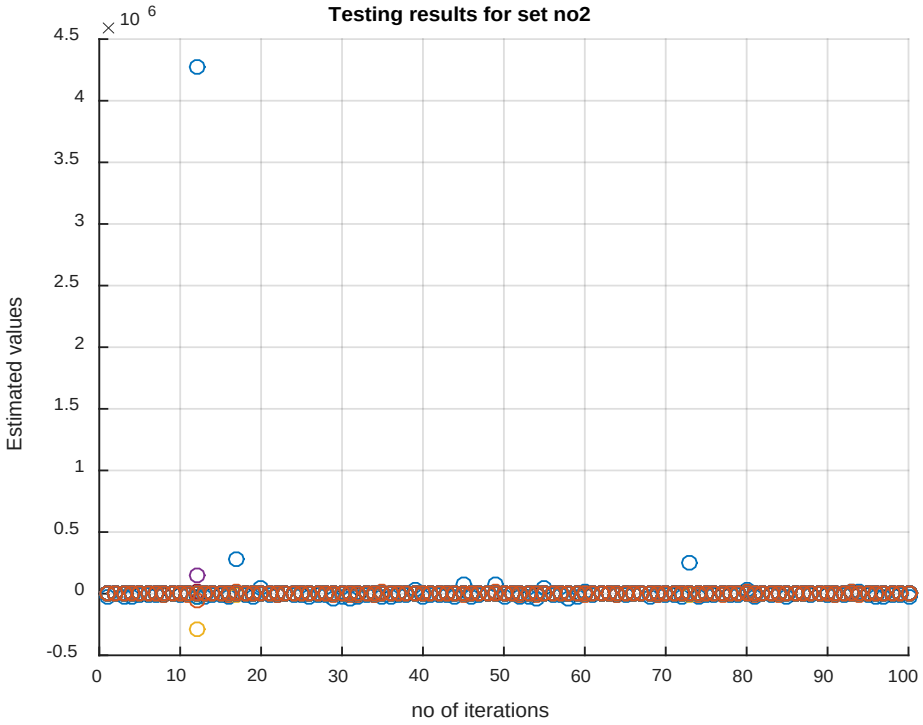


Figure 5.8: Testing results for set no 2

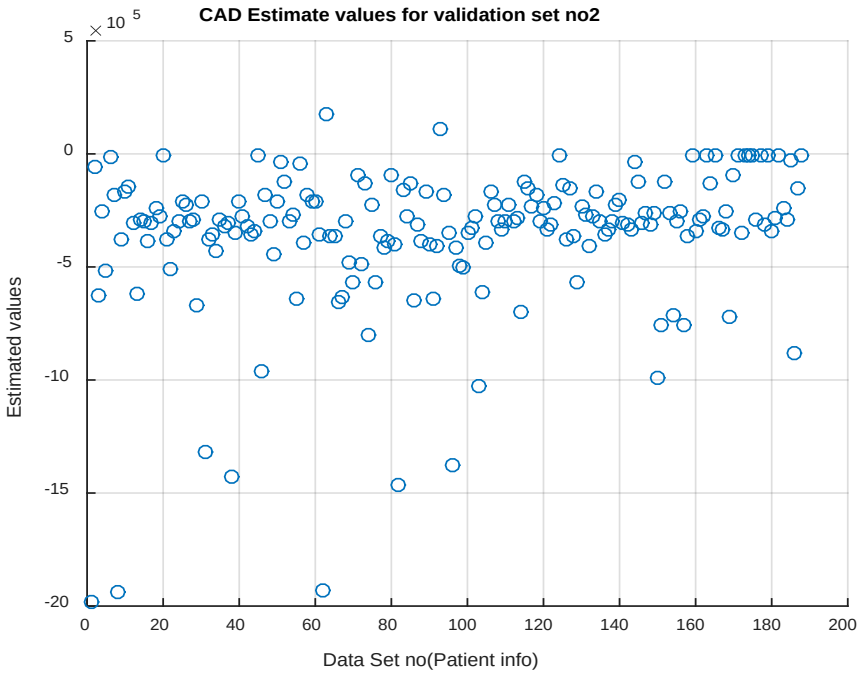


Figure 5.9: CAD estimate values for validation set no 2

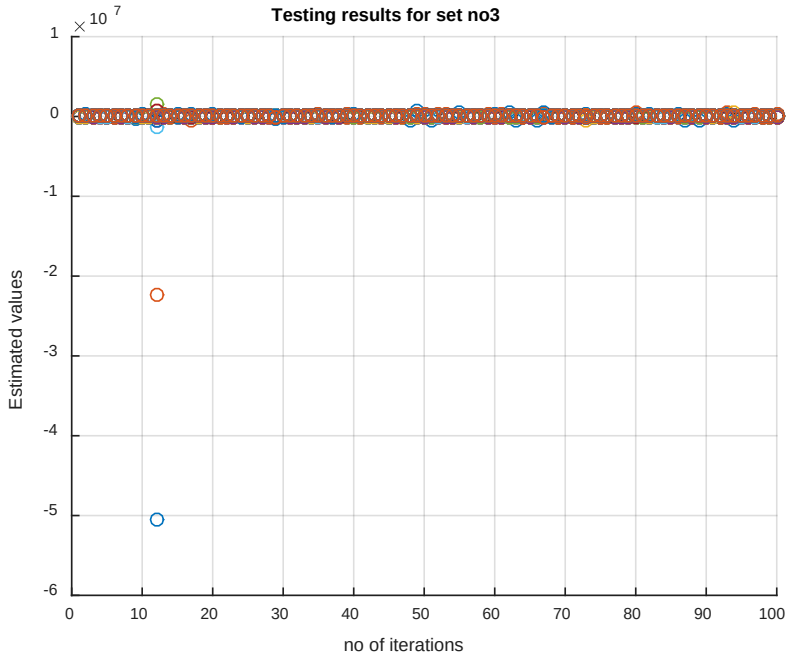


Figure 5.10: Testing results for set no 3

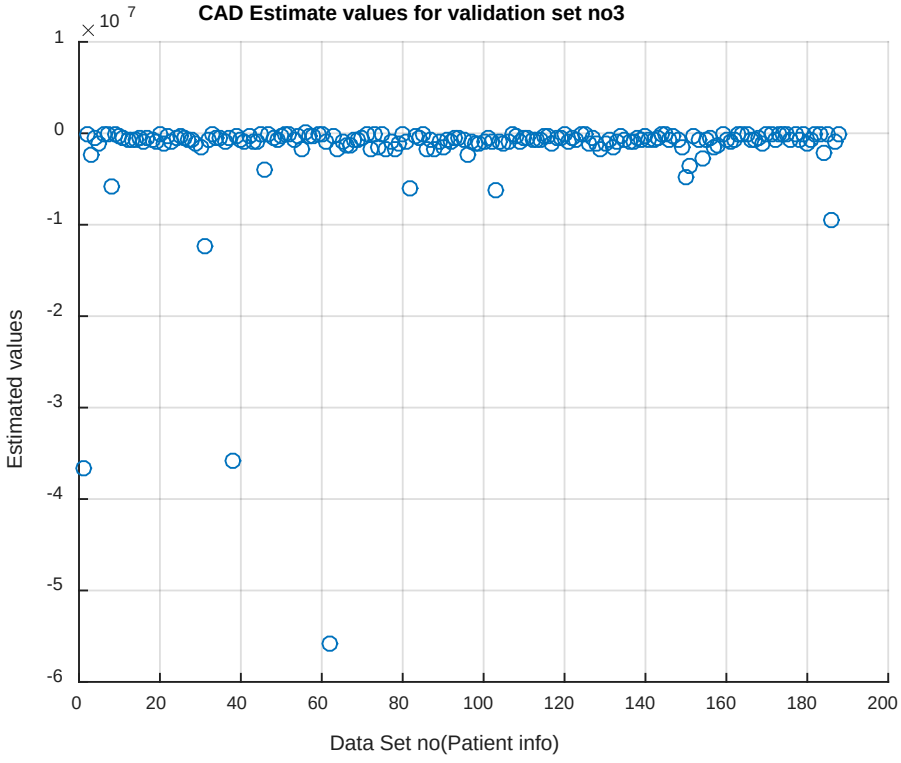


Figure 5.11: CAD estimate values for validation set no 3

All these plots give a specific measure that the estimated values were lying mainly near zero and based on the spread around zero which meant the correctness of the algorithm can be estimated. These statistics need to be investigated further to obtain the mean square error and the PRESS (residual estimated mean square).

5.3.2.2 Method 2: Particle Swarm optimization and Gravitational Search Algorithm

In Feed-forward Neural networks, the minimum error can be found by the best combination of connection weights and biases during the learning process. However, in most cases, the feed forward networks converge to a local minimum, and not to the global minimum, thus learning algorithms force the feed forward networks towards local minima and not global minima. There have been several training algorithms used for FNNs. There have also been several heuristics algorithms used to train FNN's which includes some SA (Simulated Annealing) (Van Laarhoven, 1987), GA (Genetic Algorithms), Particle swarm optimization, Magnetic optimization Algorithms (Tayarani, 2009) etc. The SA and GA have tried to achieve the global minimization but their convergence rate is very slow. However, (Mirjalili, 2010) suggested the use of a hybrid combination of Particle swarm optimization and Gravitational search algorithms over the FNN's. The basic idea of PSOGSA was to combine the ability for getting the global best in PSO with the local search capability of GSA, which can approve to be a very good candidate for FNN training. The outcome of the experiments clearly indicated that the MSE (mean square error) of PSOGSA was better than the other two algorithms. This proved the point that PSOGSA resolved the problem of local minima and also enhanced the convergence speed.

In PSOGSA, an initial population was generated and randomly initialised as in the case of all GA's. Evaluate that each agent in the PSOGSA can be considered as a best solution, at the initial stages. After initialization, the gravitational force, gravitational constant, and resultant forces among agents were calculated using equations given respectively. After that, the accelerations of particles were defined. At each iteration; the best solution so far was updated. After calculating the accelerations and updating the best

solution so far, the velocities of all agents were calculated. Finally, the positions of agents were updated. The process of updating velocities and positions was stopped by meeting an end criterion.

The basic idea of PSOGSA was to combine the ability for social thinking (gbest) in PSO with the local search capability of GSA. To combine these algorithms, by using hybrid models and use of mix characteristics, after this the following was proposed:

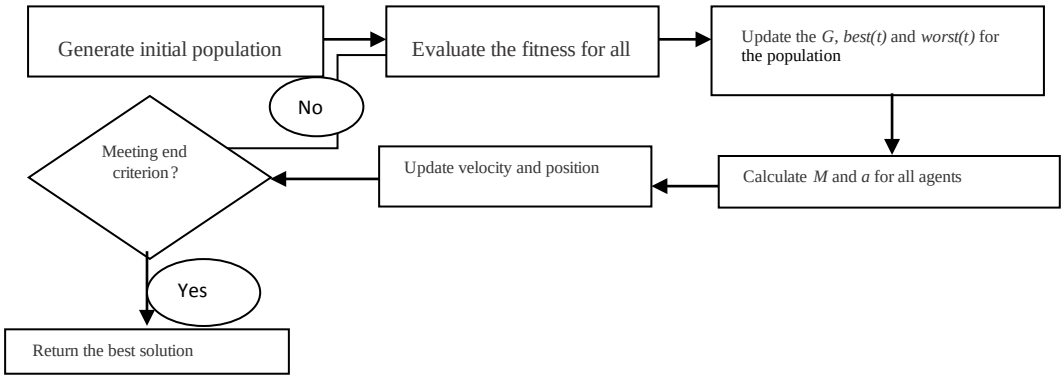


Figure 5.12: General steps of the gravitational search algorithm

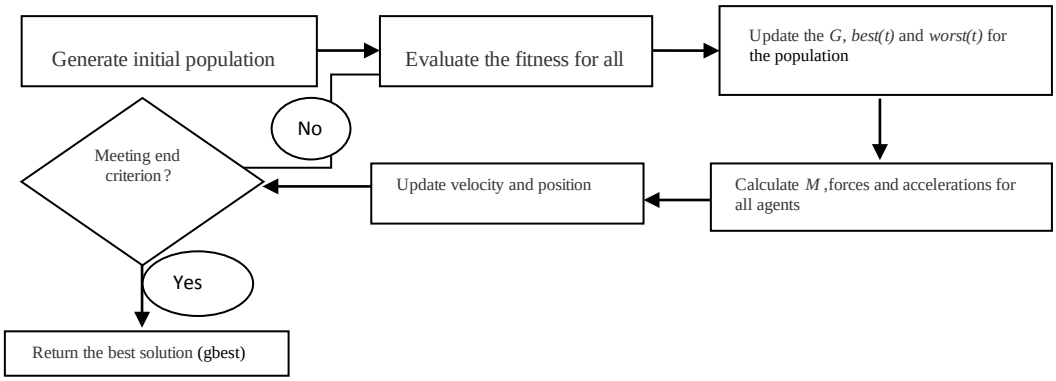


Figure 5.13: Steps of PSOGSA

The algorithm was as follows:

1. Initial Parameters for PSO
    - Number of particles
    - Maximum number of iterations
    - Inirtia weight
    - Max inertia weight
    - Min inirtia weight
    - Velocity vector

- Position vector
- Convergence vector

2. Initialise gBestScore
3. Calculate MSE
    - Calculate value using **RBFN**
    - Update Fitness value
    - Update gbest, pbest
4. Update the w (weight) of pso
5. Update velocity of particles
6. Update position of particles

Repeat until trained.

Experimental Results

Run experiment with a changing number of hidden nodes, the dataset was used; Saudi Arabia hospital data (CAD).

1. Data Set2: CAD_dataset

   Inputs: 19x687

   Targets: 2x687

F

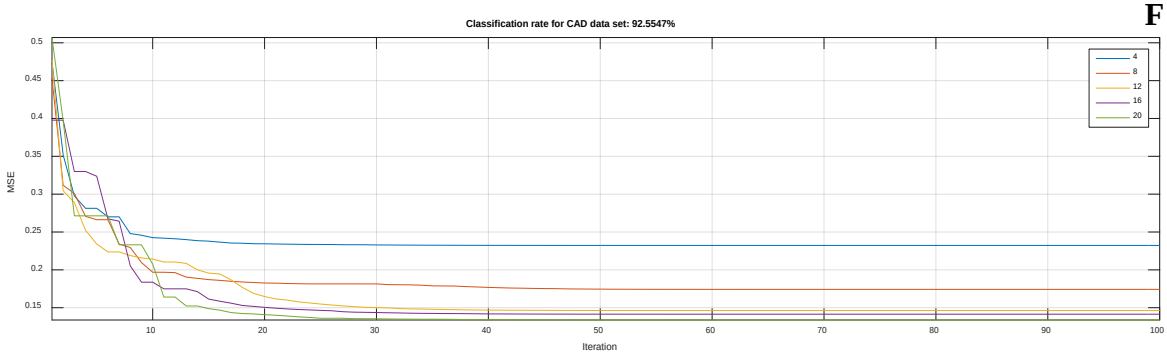


Figure 5.14: Classification rate for CAD data set

Table 5.3: Classification Accuracy for PSOGSA with varying Hidden nodes.

Classification Accuracy	
No of Hidden Nodes	Data Set
4	84.9635
8	86.2774
12	84.0876
16	92.4088 (Best Accuracy)
20	91.8248

Table 5. above give the impact of the number of hidden nodes to the Classification accuracy

In the CAD dataset, Figure 5. shows that the RBFPSOGSA algorithm trained very quickly due to the exponential decrease in the Mean Square Error (MSE) After 40 iterations, the network was well trained (MSE < 0.15) and it continued to improve marginally after that point, yielding a good classification rate of 92.55%

5.4.Discussion

This chapter investigates the diagnosis of CAD using two algorithms for training RBF neural Networks. The RBF neural network model was a neural network model which was superior to the traditional Back-Propagation model on the prediction accuracy and training time (Dong, 2005). RBF neural networks were not widely applied for the prediction of CAD. Limited researches (You, et al., 2009), (Lu, et al., 2012) were performed in this area. However, the research performed demonstrated reasonable performance and recommended the use of RBF in health applications.

RBF was also the preferred network with CAD prediction owing to its high ability to consider non-linear features which were inevitable in the practical diagnosis of CAD. This advantage overcame the limitations of other linear techniques (Heydari, et al., 2012); (Mandal & Sairam, 2012) used in CAD prediction.

Other researches in this field considered the use of more complex models - such as the combination of SVM with RBF Kernel to generate a non-linear technique (Heydari, et al., 2012); (Mandal & Sairam, 2012); (Karabulut & İbrikçi, 2012); (Wu & Lange, 2015).

Evaluation results for the proposed approach showed that RBF neural networks were suitable for diagnosis of heart diseases; since they achieved performance accuracy ranging from 92 % to 99% for the three tested data sets. However, this finding was not consistent with the deduction issued by Davis & Thu (2006) in their work that developed to compare the performance of different neural network techniques to improve the performance of clinical decisions. The authors argued that the RBF neural network model had the worst classification performance when co, as provided in the experimental impared to other models.

As previously mentioned, the proposed RBF model was trained using two training algorithms. The first training algorithm was the Extended Kalman Filter (EKF). The proposed approach used a direct application of the Extended Kalman Filter (EKF) which extended the Kalman filter using linearization to fit non-linear models. The Kalman Filter was generally used in technology for the purpose of guidance, navigation and control but this research proposes using the EKF to train a neural network. This decision was taken owing to its ability to produce optimal estimates of non-linear systems which precisely describes the goal of a neural network.

The Extended Kalman Filter (EKF) learning algorithm was used in diverse research realms, and has shown excellent results in terms of prediction accuracy. Regarding the training of the RBF neural networks models; EKF was used in the previous researches (Ciocoiu, 2002); (Birgmeier, 1995) for different applications such as nonlinear speech modelling (Birgmeier, 1995) and satellite attitude determination (Xinyuan, et al., 2012).

Nonetheless, in this research, EKF was first explored in Coronary Artery Disease prediction. The proposed method used this underlying principle to train RBF neural network in combination with a quasi-newton training and Scaled Conjugate Gradient (SCG) for error minimization and the R4LogR and TPS as the activation functions. This special combination and the way of building the algorithm has never previously been

applied to something in the medical field. The developed method has been tested and proven on data, and showed good performance results.

The second method used for training the RBF model was a hybrid of Particle Swarm Optimization (PSO) and Gravitational Search Algorithm (GSA) to optimise and obtain the CAD prediction correctly. This combination achieves a good classification rate of 83.21% overall (less than EKF approach, but still reasonable). It has proven to be successful in the health applications used in this research.

This finding was consistent with Mirabedini (2014) who proved that PSO as a training algorithm showed the superiority of other traditional algorithms. Also Razmjooy & Ramezani (2016) showed that this method can improve the performance of the wavelet based neural network significantly.

In addition to the proposed CAD prediction systems, this work presented a new dataset. This dataset will be useful to researchers and practitioners working in the heart disease area and would widely encourage comparative studies in the field of such research. It is envisaged that the proposed decision support system for clinical screening would greatly contribute to and assist the management and the detection of CAD at an early stage.

Missing features values are a concern when dealing with datasets, especially, real databases. Therefore, an approach for constructing missing features values based on Matrix Completion has been proposed. Matrix Completion methods were used to enhance the quality of the data sets. The proposed approach is evaluated before and after applying Matrix Completion. The obtained results showed that this method improves the prediction performance. This agreed with other researches, which also showed that matrix completion greatly enhanced the accuracy of prediction (Arif, et al., 2010); (Comak & Arslan, 2010).

5.5.Summary

The work of this chapter addressed the prediction of Coronary Artery Disease by using patient information as a set of data to feed the RBF neural network. Two methods were used for training the RBF neural networks: Extended Kalman Filtering (EKF) and Particle Swarm Optimization and Gravitational Search Algorithm (PSOGSA). Both methods performed significantly differently on different subsets of the training and validation data. The prediction probability of this combination resulted in an accuracy of about 92%, using cross validation. It was also noticed that the classifier had different outcomes for predicting data in different classes. This suggested the possibility that an ensemble of classifiers trained on different parts of the dataset might result in greater performance to meet the need of each sub-dataset.

In supervised machine learning, the aim was to identify the relationship between some inputs and response data for regression or classification problems. For this research, demographic and historical medical data of patients was given as input, together with the diagnosis data for the particular disease. This work attempted to find the underlying functional relationship between the medical observations and diagnosis outcomes to predict the presence or absence of the disease from new medical data.

This chapter considers CAD data as the main dataset to evaluate the two proposed systems to support the clinical heart disease decision. However, the other two sets of heart disease data, sleep apnea and arrhythmia datasets, haven't shown good performance with this particular approach because of their size and this led to the need in this research to explore another health informatics system which can deal with large datasets (i.e. "big data") using a deep learning approach.

Chapter 6 : Heart Disease Detection Using Data Mining Algorithms and Deep Neural Networks

6.1. Overview

Deep learning has shown outstanding results in many applications such as image classification (Hayat, et al., 2015), object recognition (Bai, et al., 2015), face recognition (Huang, et al., 2015), and time series data (Deng, et al., 2014; Hinton, et al., 2012). This chapter proposes a novel scheme based on a deep learning approach for the automatic detection and classification of heart diseases. In order to check the validity of the used datasets; four well-known classification techniques are also applied for heart diseases detection to construct prediction model using Logistic Regression, K-Nearest Neighbours (KNN) , Support Vector Machine (SVM) and Naïve Bayes Classifiers. After constructing the models and the proposed deep approach to detect heart disease. Results of experiments of each model were evaluated, and their performances were compared.

6.2. Datasets Description

In this part of research, three different datasets were used to evaluate the proposed approaches. Details of these datasets were described in chapter 4.

1. Apnea-ECG dataset from PhysioBank Databases.
2. CAD dataset from The King Abdullah Medical City hospital at Saudi Arabia
3. Arrhythmia dataset from UCI Repository

The main application of this part is based on using Apena-ECG dataset, while the other two datasets were applied to enhance the results.

6.3. Heart Diseases Detection Using Traditional Data Mining Algorithms

In this part, four different classification techniques were applied for heart disease detection. An attempt was made to construct prediction model using Logistic Regression, K-Nearest Neighbours (KNN), Support Vector Machine (SVM) and Naïve Bayes Classifiers. Details of the algorithms used to build the models and performance measure and comparison are discussed in this chapter.

6.3.1. Methodology

This section provides the proposed methodology for the detection of the heart diseases using the traditional data mining algorithms. Figure 6.1 shows the block diagram for this methodology.

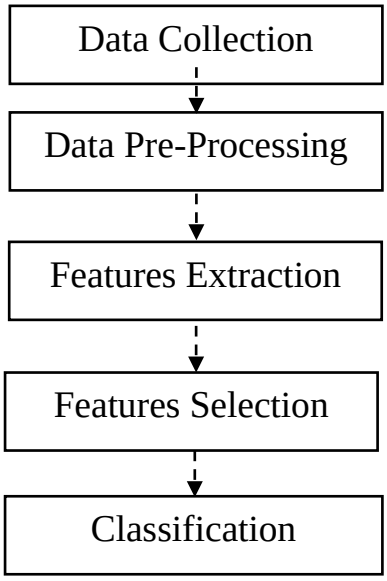


Figure 6.1: Block Diagram of the proposed methodology

6.3.1.1 Data Pre-processing and Features Extraction

Datasets are processed and features are extracted in the same way that was mentioned in chapter 4 for each dataset.

6.3.1.2 Features Selection

As mentioned in the chapter 4; choosing the most effective features on the prediction, achieves a better performance with minimum processing time. In this phase, the most informative features with higher effect on prediction are chosen. To determine the discriminative power of each feature, ANOVA (ANOVA, 2013) statistical tests were adopted. ANOVA is a statistical method that stands for analysis of variance. It is used to test the degree to which two or more groups vary or differ in an experiment. For each derived feature, the ANOVA value was computed in the objective of identifying the significant ones. The lowest ANOVA value was the highest contribution for the feature.

- *Apnea-ECG Dataset*

In this dataset, 11 features were extracted from each ECG minute. After applying the ANOVA test to features vector; it was deduced that NN50_1, NN50_2, pNN50_1, pNN50_2 were the less relevant features and did not contribute highly in the classification results; so they were eliminated from the features set to have 7 features instead of 11. Figure 6.2 presents the ANOVA test value for the features set.

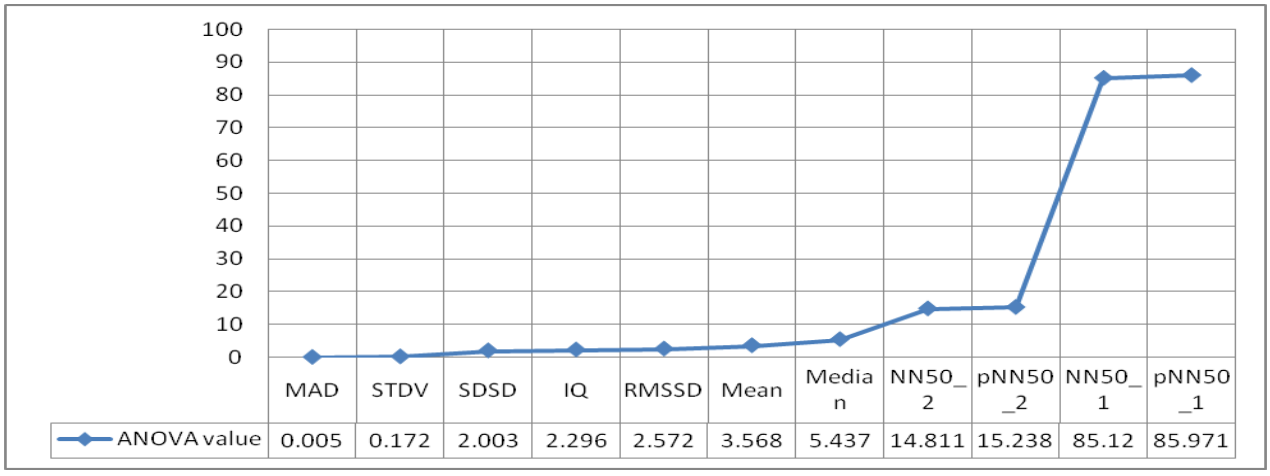
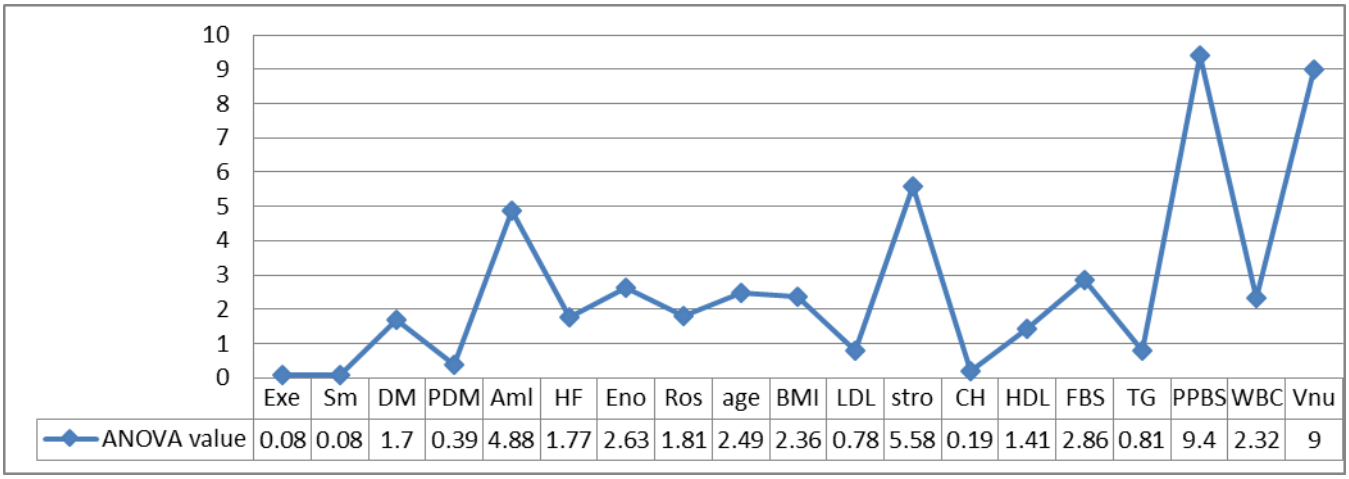


Figure 6.2: Values of ANOVA test for features set of Apnea dataset

- *CAD Dataset*

When applying the ANOVA test to the CAD dataset; the results were obtained and plotted in Figure 6.3. The test's results show that all features had values relatively close to each other, which indicated that all features had a good contribution in the classification result and must be all included in the features vector.



	Exe	Sm	DM	PDM	Aml	HF	Eno	Ros	age	BMI	LDL	stro	CH	HDL	FBS	TG	PPBS	WBC	Vnu
ANOVA value	0.08	0.08	1.7	0.39	4.88	1.77	2.63	1.81	2.49	2.36	0.78	5.58	0.19	1.41	2.86	0.81	9.4	2.32	9

Figure 6.3: Values of ANOVA test for features set of CAD dataset

- *Arrhythmia Dataset*

When applying the ANOVA test to this dataset; it was found also that all features had similar values for this test. Figure 6.4 illustrates the distribution of these results. The values were all close to each other and therefor all features were included in the classification process. Appendix III presents the ANOVA results for all features.

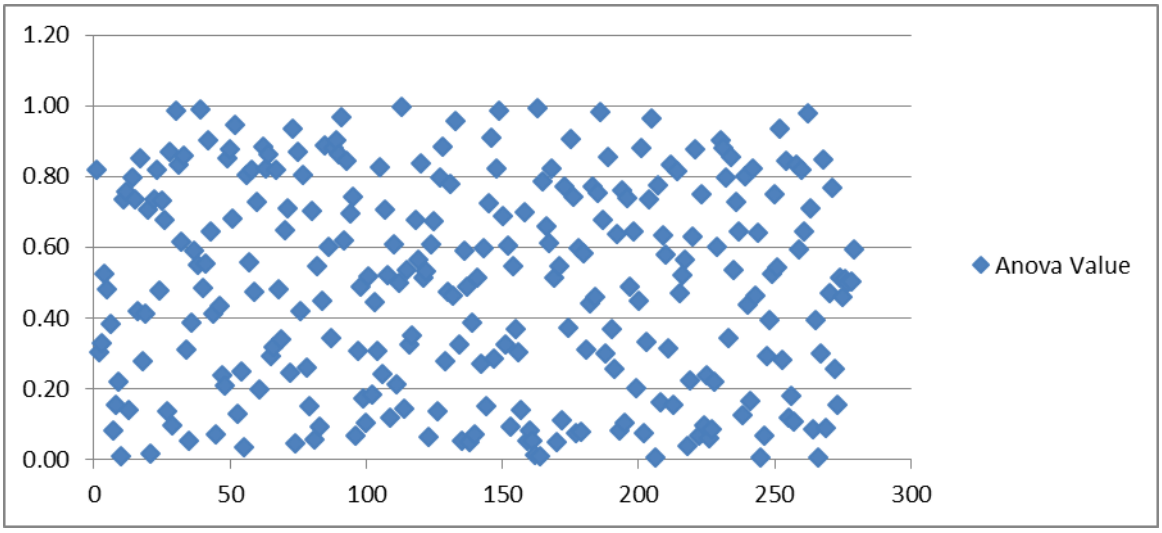


Figure 6.4: Values of ANOVA results for features of Arrhythmia Dataset

6.3.1.3 Classification Model

In the classification process, four different classifiers were applied to detect heart diseases using the three previously mentioned datasets. The classifiers were used: Logistic Regression, KNN, Naïve Bayes and SVM classifier. Data was divided into training and testing data with 80% for training and 20% for testing The sampling method used was cross-validation in which the dataset was split into ten folds. The classifier was trained on nine folds and tested on the remaining one. All the possible combinations of nine folds were explored and the performance metrics were calculated.

6.3.2. Experimental Results

The Orange Data Mining toolset (Orange, 2016) was used to simulate the traditional classifiers and compare results. The performance of the proposed classification models was evaluated for both the minute-based classification and the minute-class-based classification of sleep apnea. Table 6.1 summarises the model parameters for the classifiers. The Euclidean distance function is used to find the nearest neighbours. The model assigns uniform weights to each neighbour. In regards to the SVM model; the classification types is C-SVM, where C is chosen to be 1.0. The model uses the RBF

Kernel with numerical tolerance equals to 0.001 and iterations can reach maximum of 100 iteration. Figure 6.5 illustrates the workflow for the proposed classification scheme.

Table 6.1: Model parameters for the applied classifiers

Logistic Regression	KNN	SVM
Regularization: Ridge (L2), C=1	Number of neighbours: 20 Metric: Euclidean Weight: Uniform	SVM type: C-SVM, C=1.0 Kernel: RBF, exp(-1.0|x-y|²) Numerical tolerance: 0.001 Iteration limit: 100

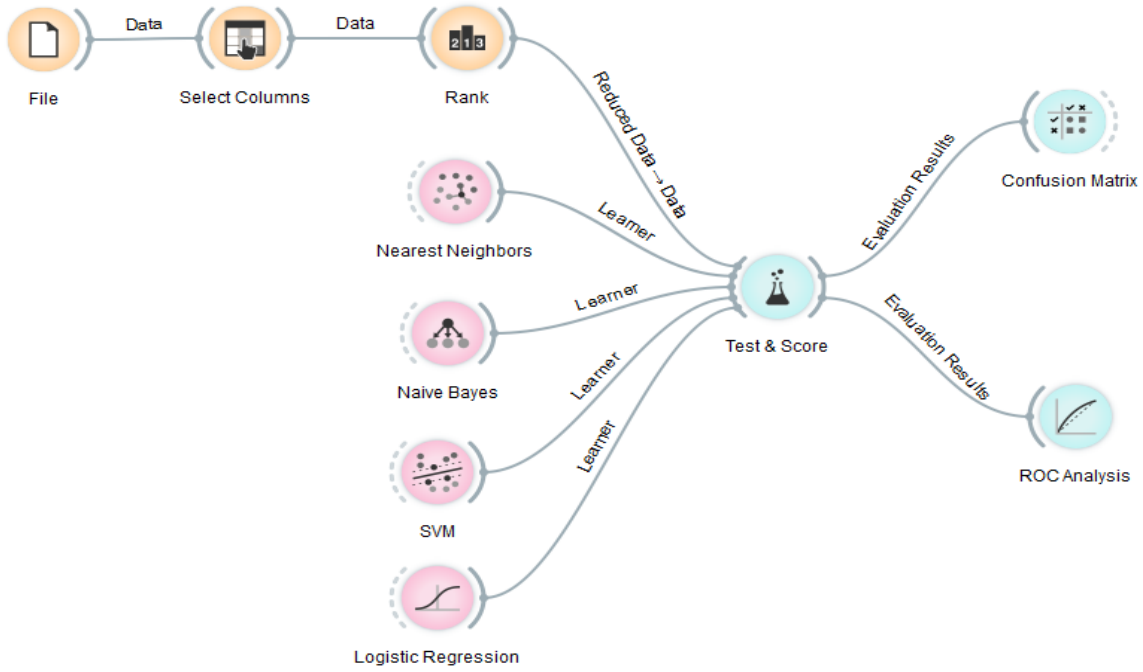


Figure 6.5: Workflow of the proposed classification scheme

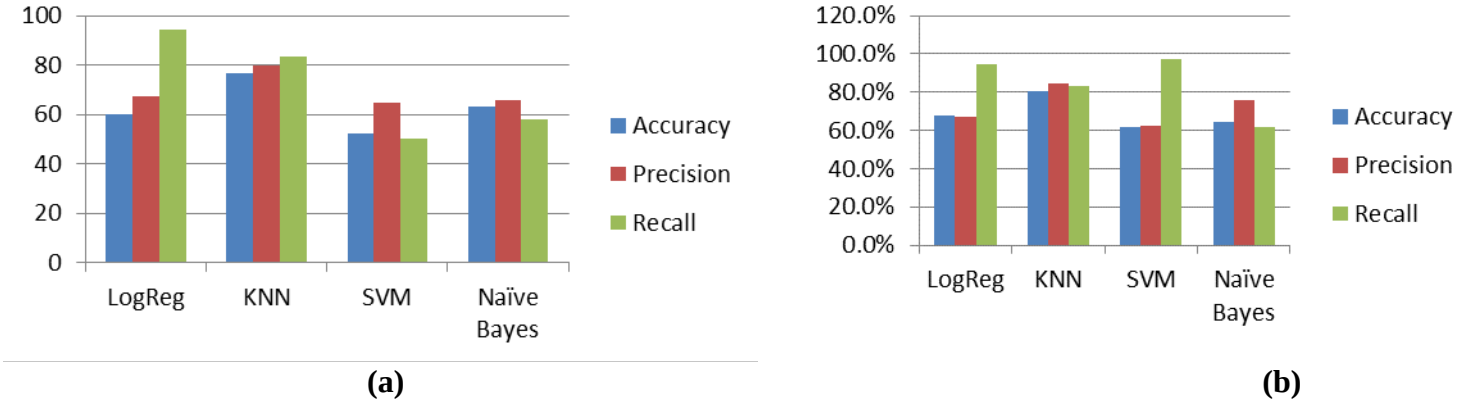
6.3.2.1 Apnea-ECG Dataset Results

Table 6.2 summarises the performance of the four classifies when applied to the Apnea dataset. Figure 6.6 also shows the performance of all the classifiers, before and after features selection. It can be observed that in general, classifiers performed better after selecting the top seven relevant features from the whole number of features (11 features). Figure 6.7 also supports these results and presents the classification accuracy for each classifier before and after the features selection. The results show that the compact

feature subset had good effect on the accuracy of the classification system. Specifically, the accuracy of Logistic regression was increased by 7.6% while that of SVM was increased by more than 10%. The accuracy of KNN and Naïve Bayes was increased by about 4%.

Table 6.2: A summary of the classifiers performance for Apnea Dataset

Classifier	# Features =11					# Features =7				
	Acc.	Prec.	Rec.	Sens.	Spec.	Acc.	Prec.	Rec.	Sens.	Spec.
Logistic Regression	60.3	67.3	94.2	94.2	26.4	67.9	66.9	94.9	94.9	23.9
KNN	76.8	80.0	83.2	83.2	66.4	80.5	84.8	83.3	83.3	76.0
SVM	52.3	64.6	50.2	97.1	5.6	62.0	62.3	97.1	97.1	5.6
Naive Bays	63.2	66.0	57.8	57.8	71.8	67.2	76.0	61.5	61.5	68.7



(a) **(b)**

Figure 6.6: Classification performance obtained : (a) before features selection (b) after features selection

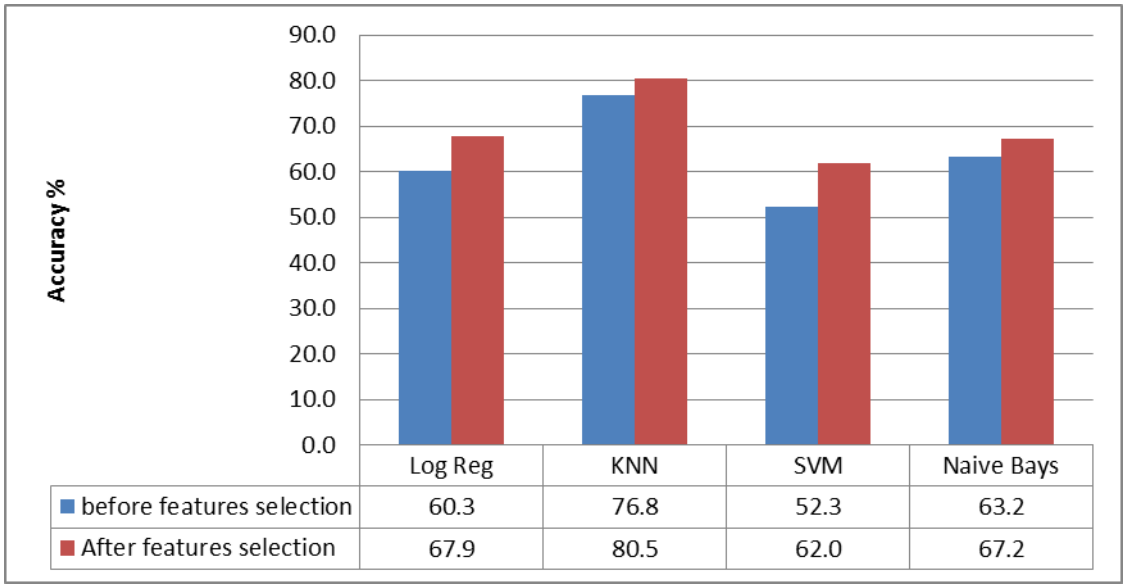
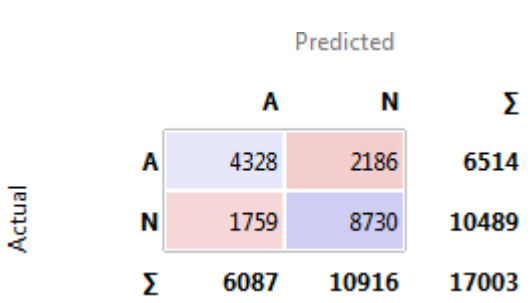
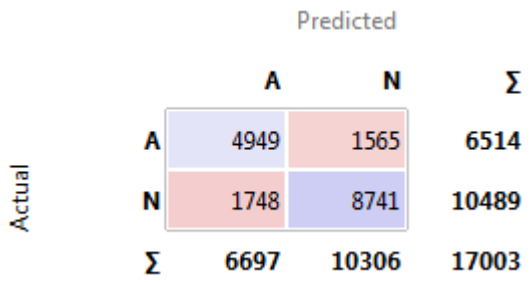
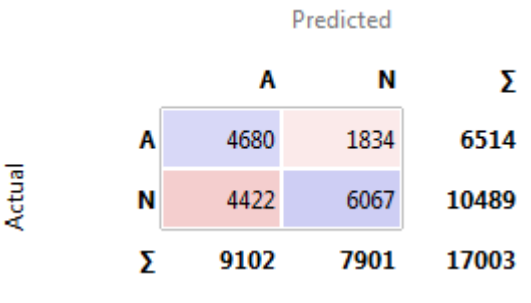
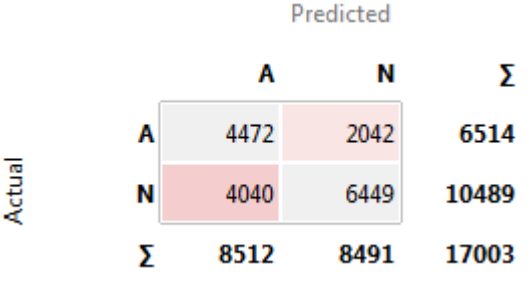
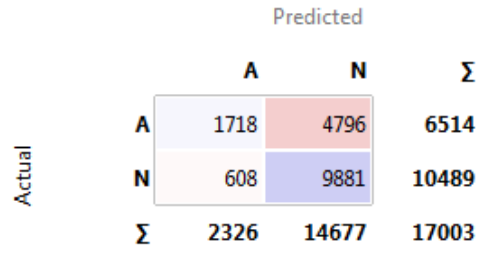
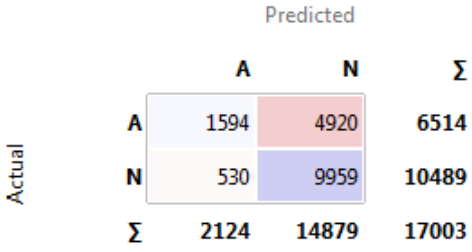
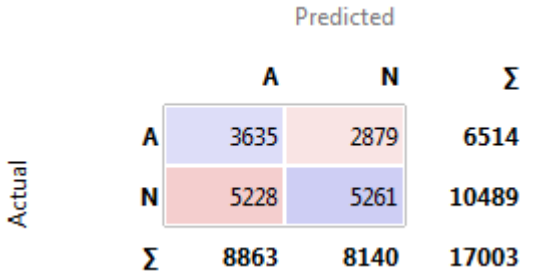
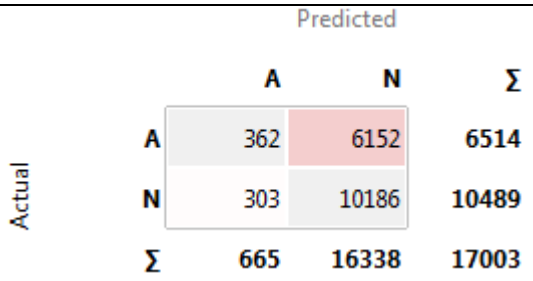


Figure 6.7: Accuracy comparison of classifiers performance before and after features selection

Table 6.4 presents the confusion matrices for the proposed classification models before and after features selection. Regarding the KNN classifier, before features selection, ; it was able to correctly detect 4328 out of 6514 apnea minutes and 8730 out of 10489 non-apnea minutes. But it misclassified 2186 of apnea minutes and considered it as a non-apnea minutes; at the same time it misclassified 8730 of non- apnea minutes and considered it as an apnea minutes. After features selection; the number of correctly classified minutes increased for both apnea and non-apnea minutes, which implicates that the classification is balanced and the results are reasonable. Other classification models also performed better after features selection and the number of correctly classified minutes is increased in the case on non-apnea minutes but decreased with a few rate of the apnea minutes. However; the total number of correctly classified minutes is still higher than before features selection.

Table 6.3: Confusion Matrix for the classifiers for Apnea Dataset

<table>
<tr><td>Before Features Reduction(# features = 11)</td><td>After Features Reduction (# features = 7)</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td colspan="2">KNN Classifier</td></tr>
<tr><td>Before Features Reduction(# features = 11)</td><td>After Features Reduction (# features = 7)</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td colspan="2">Naïve Bayes Classifier</td></tr>
</table>

Before Features Reduction(# features = 11)	After Features Reduction (# features = 7)
	
Logistic Regression Classifier	
Before Features Reduction(# features = 11)	After Features Reduction (# features = 7)
	
SVM Classifier	

ROC Analysis was also obtained to evaluate the classification models performance. This analysis plotted a true positive (TP) rate against a false positive (FP) rate for the test. The curve plotted a FP rate on an x-axis (1-specificity; probability that target=1 when true value=0) against a TP rate on a y-axis (sensitivity; probability that target=1 when true value=1). Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity). Therefore the closer the ROC curve is to the upper left corner, the higher the overall accuracy of the test (Zweig & Campbell, 1993). The results of this analysis are plotted in Figure 6.8 and Figure 6.9. As shown from the figures; KNN, Naive bayes and logistic regression models have reasonable performances, since the area under the ROC curve is higher than 0.5, while the ROC curve area for SVM model revealed a lack of predictive accuracy since it is very close to 0.5, which indicates that it has the poorest accuracy. KNN model performs better that other classification models since the area under curve (AUC) have the highest value comparing with other models .The ROC of SVM model signifies that for a given set of parameters

the area under the curve is comparatively low as compared to other classification techniques. This is may be due to using the model with a RBF kernel which is not sufficient for such type of datasets; while using SVM with linear kernel approved to be more efficient (Khanna et al., 2015).

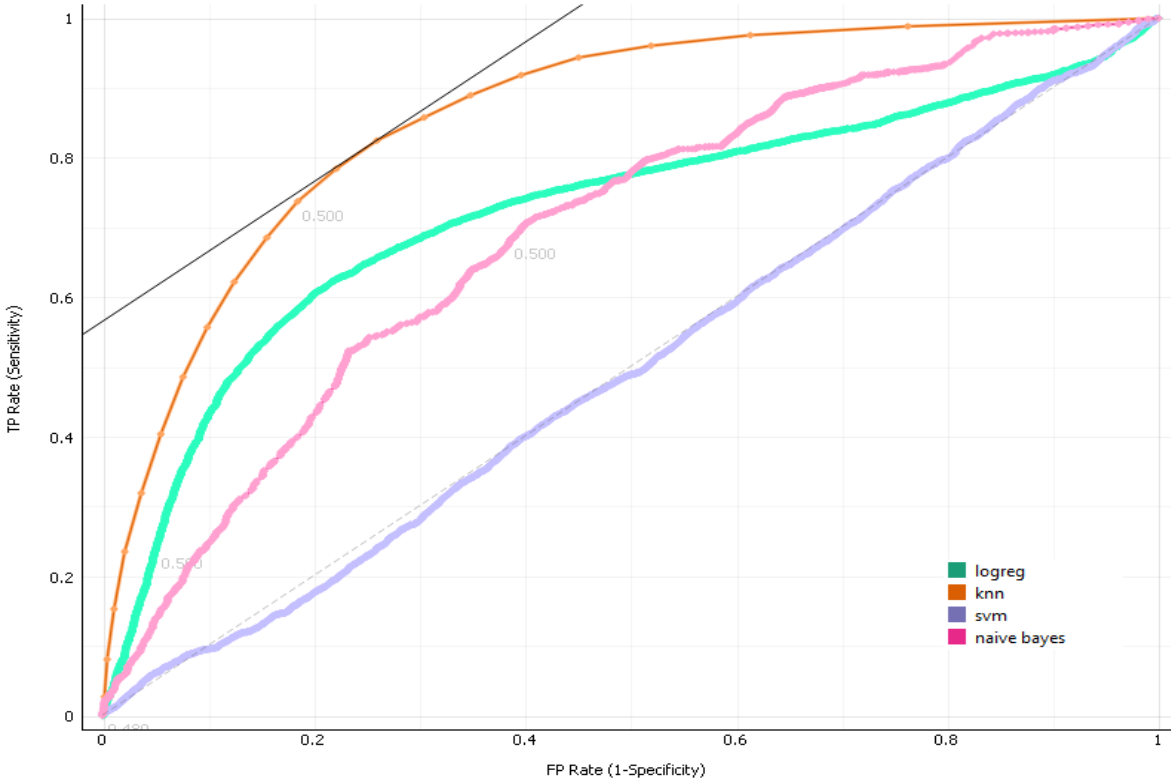


Figure 6.8: ROC Analysis for apnea class

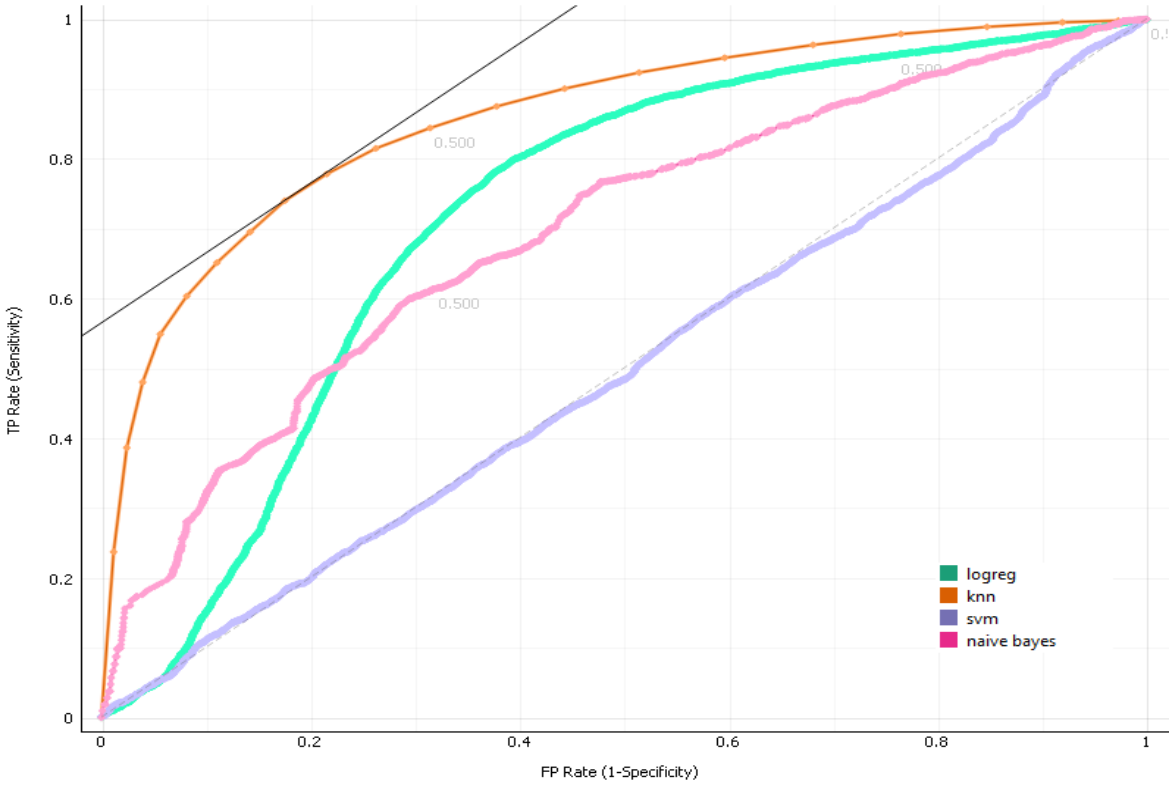


Figure 6.9: ROC Analysis for non-apnea class

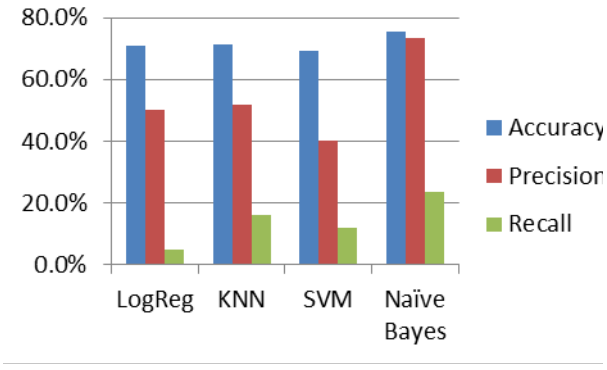
6.3.2.2 CAD Dataset Results

The performance result of applying the proposed classifiers to the CAD dataset are shown in

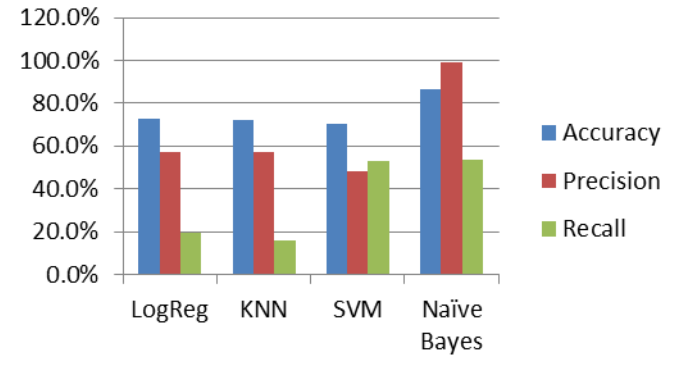
Table 6.4 and Figure 6.10. As a step for validation of the effect of features selection on CAD dataset; the performance was measured before applying features selection and after applying features selection with the best ten features were elected to participate in the classification process. As found from the results, the performance is better when all features were included, which supported the conclusion that was deduced from the result of the ANOVA test. The results also show that Naïve bayes classifier outperforms other models before and after features selection with accuracy of 86.3% and 75.4% respectively. In a similar fashion to the Apnea dataset, SVM has the poorest performance as shown in table 6.4. Based on this particular data set the SVM is therefore not deemed as a suitable model to be used.

Table 6.4: A summary of the classifiers performance for CAD Dataset

Classifier	After features selection			Before features selection		
	Acc.	Prec.	Rec.	Acc.	Prec.	Rec.
Logistic Regression	71.1%	50.0%	5.0%	72.5%	57.4%	19.6%
KNN	71.4%	51.6%	16.1%	72.2%	57.4%	15.6%
SVM	69.3%	40.0%	12.1%	70.1%	48.4%	52.8%
Naive Bays	75.4%	73.4%	23.6%	86.3%	99.1%	53.3%



(a)

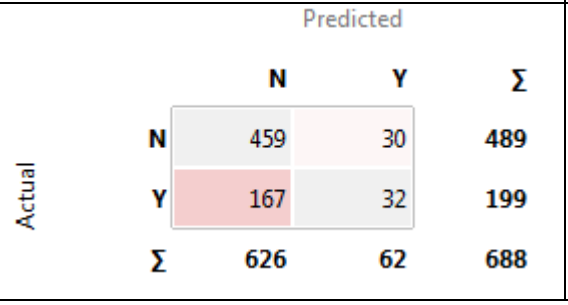
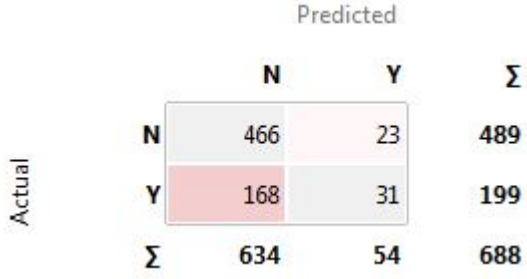
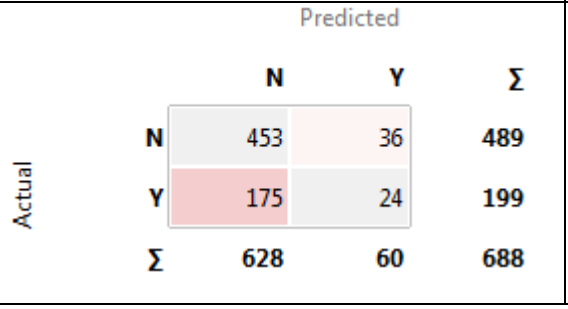
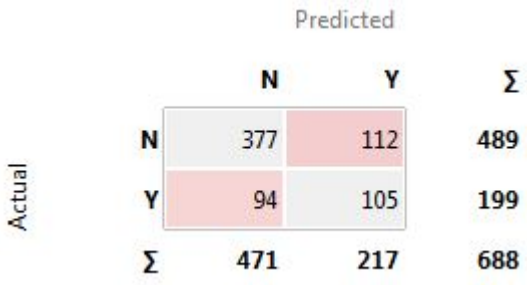


(b)

Figure 6.10: Classification performance obtained: (a) After features selection (b) Before features selection

The confusion matrices for all classification models are presented in Table 6.5. Regarding the KNN classifier, before features selection; it was able to correctly detect 95.3% of non-CAD cases and 15.58% of CAD cases.. But it misclassified 168 of CAD cases and considered it as non-CAD cases; at the same time, it misclassified 23 of non- CAD cases and considered it as CAD case. After features selection; the number of correctly classified minutes' decreases for both non-CAD cases, but still nearly the same in CAD cases. This indicates that the classification model performed better before features selection especially for non-CAD cases. Other classification models also performed better before features selection and the total number of correctly classified cases is higher than after features selection. Logistic regression has the worst number of correct classified CAD cases after features selection; since it could classify only one correct CAD case. However; it could correctly classify a reasonable number of CAD cases before features selection. This ensures that working on CAD dataset is preferable when including all features and not excluding any of them.

Table 6.5: Confusion Matrix for the classifiers for CAD Dataset

<table>
<tr><td colspan="2">After Features Reduction</td><td colspan="2">Before Features Reduction</td></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="4">KNN Classifier</td></tr>
<tr><td colspan="2">After Features Reduction</td><td colspan="2">Before Features Reduction</td></tr>
<tr><td colspan="2"> 
</td><td colspan="2"> 
</td></tr>
<tr><td colspan="4">Naïve Bayes Classifier</td></tr>
<tr><td colspan="2">After Features Reduction</td><td colspan="2">Before Features Reduction</td></tr>
</table>

Predicted Actual N Y Σ N 488 1 489 Y 198 1 199 Σ 686 2 688	Predicted Actual N Y Σ N 460 29 489 Y 160 39 199 Σ 620 68 688
Logistic Regression Classifier	
After Features Reduction	Before Features Reduction
Predicted Actual N Y Σ N 472 17 489 Y 152 47 199 Σ 624 64 688	Predicted Actual N Y Σ N 488 1 489 Y 93 106 199 Σ 581 107 688
SVM Classifier	

The ROC analysis for both CAD and Non-CAD classes are illustrated in Figure 6.11 and Figure 6.12, respectively. The Naïve bayes classifier covered the highest AUC indicate it has the highest accuracy. Other classifiers performance was near each other. They performed well and achieved reasonable AUC value.

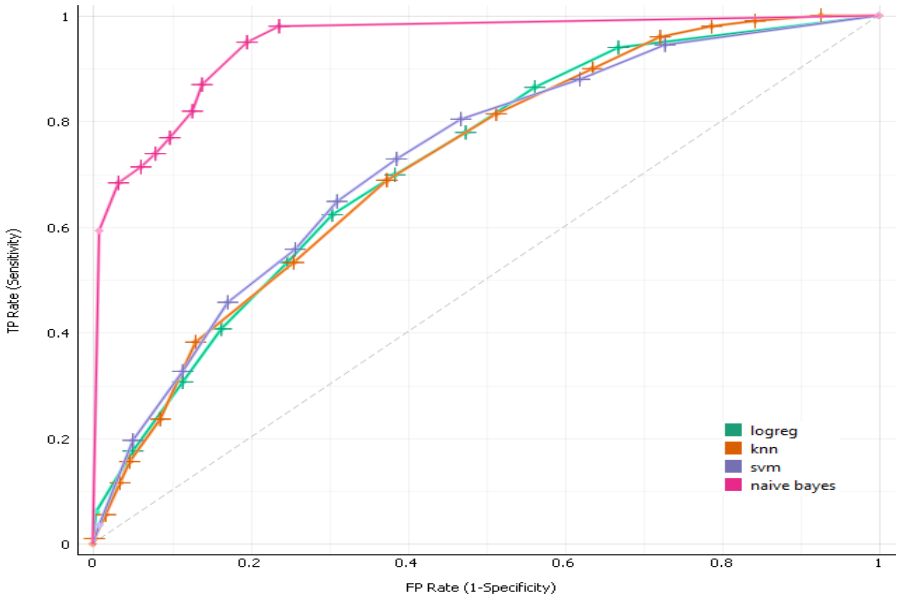


Figure 6.11: ROC Analysis for CAD class

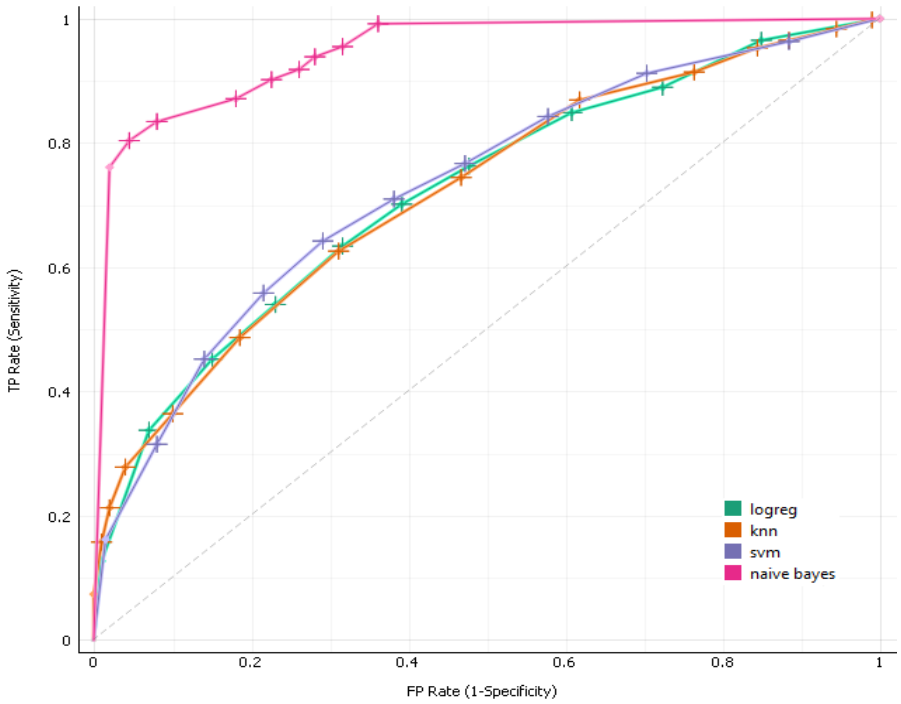
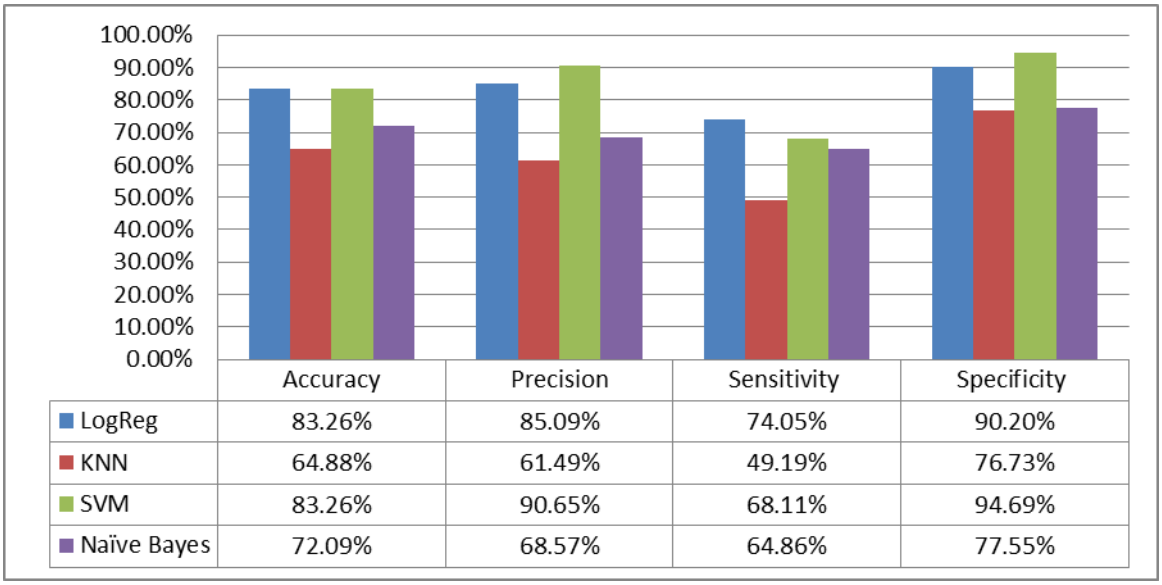


Figure 6.12: ROC Analysis for non-CAD class

6.3.2.3 Arrhythmia Dataset Results

The performance result of applying the proposed classifiers to the Arrhythmia dataset was summarised and plotted in Figure 6.13. As encouraged by the results of the testing of the CAD dataset with features selection having features near the ANOVA values, the performance was then measured by including all features in the classification process. Unlike previous datasets; both SVM and logistic regression models have the best performance; while KNN model has the poorest results.



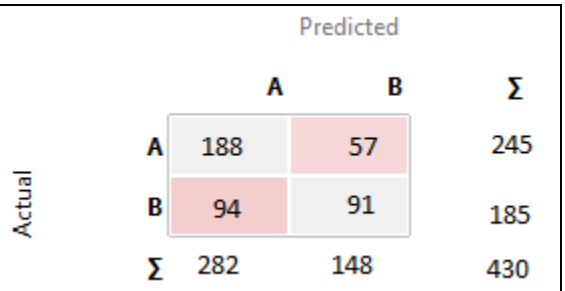
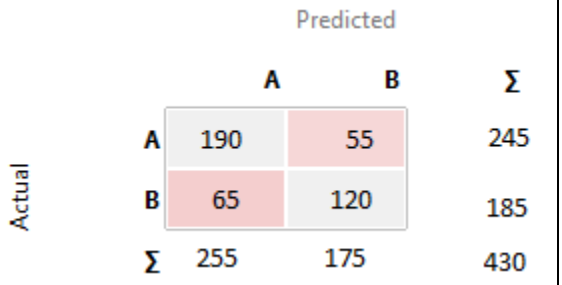
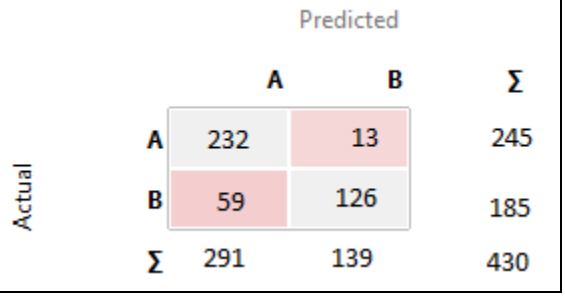
	Accuracy	Precision	Sensitivity	Specificity
LogReg	83.26%	85.09%	74.05%	90.20%
KNN	64.88%	61.49%	49.19%	76.73%
SVM	83.26%	90.65%	68.11%	94.69%
Naïve Bayes	72.09%	68.57%	64.86%	77.55%

Figure 6.13: Classification Performance for Arrhythmia Dataset

Confusion matrices for all classifiers are presented in Table 6.6. Regarding the logistic regression model was able to correctly detect 221 out of 245 arrhythmia cases and 137 out of 185 of non-arrhythmia cases. But it misclassified 24 of arrhythmia cases and considered it as non-arrhythmia cases; at the same time, it misclassified 48 of non-arrhythmia cases and considered it as arrhythmia case. More details about other classification models are available at Table 6.6.

Table 6.6: Confusion Matrix for the classifiers for Arrhythmia Dataset

Classifier Type	Confusion Matrix
Logistic Regression	Predicted / Actual: A: 221, 24, Σ 245; B: 48, 137, 185; Σ: 269, 161, 430

KNN	
Naïve Bayes	
SVM	

ROC analysis for both Arrhythmia and Non- Arrhythmia classes are illustrated in Figure 6.14 and Figure 6.15, respectively. KNN classifier covered the lowest AUC and so achieved the lowest accuracy. SVM and Logistic regression models covered nearly the same area and hence achieved a nearly performance. Naïve bayes model covered middle AUC and achieved fair performance.

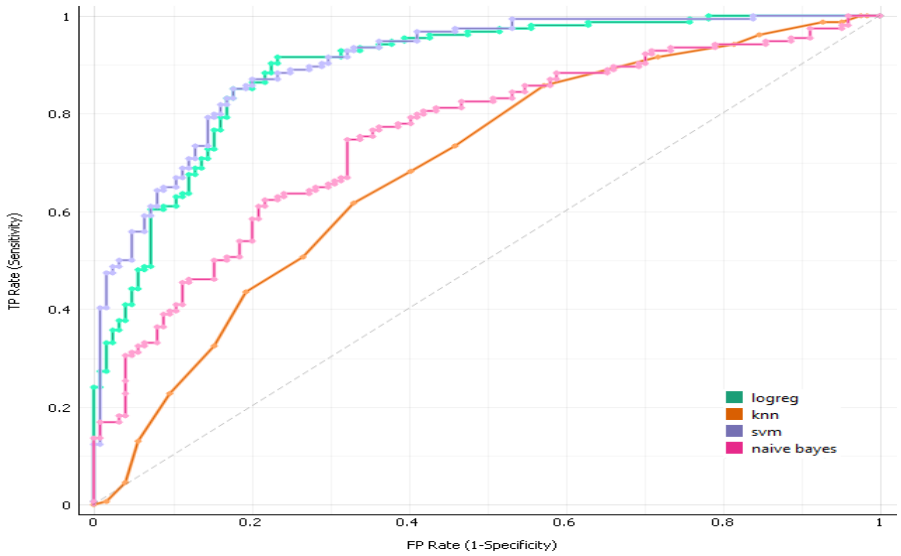


Figure 6.14: ROC Analysis for Arrhythmia class

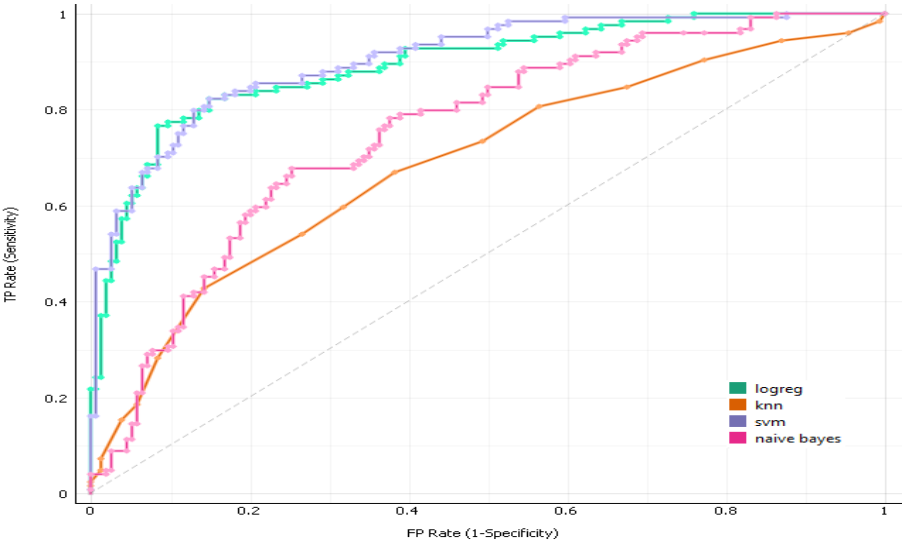


Figure 6.15: ROC Analysis for non- Arrhythmia class

6.4. Heart Disease Detection Using Deep Neural Networks

In this part, the proposed deep learning approach in automatic detection and classification of heart diseases is presented concentrating on Sleep Apnea as the main application. Nevertheless, the proposed studies in this area of research achieved relative satisfactory performance on apnea detection and quantification; there are some important aspects that need be highlighted. First, the proposed approaches either identify the apnea class or detect the presence or absence of each minute of ECG data. To the best of our knowledge, only Babaeizadeh, et al. (2010) and Rachim, et al. (2014) addressed both apnea detection and quantification for each patient recording but both identified only two classes, excluding class B. Secondly, various features were extracted from the RR intervals without careful investigation, causing predictors in the selected classifier to be redundant. At the same time, feature extraction and selection from such high-dimensional feature spaces would require a large amount of computational resource, which is not attainable for most wearable devices and is also inconvenient for their wider application in home-based diagnosis since the modern healthcare system is required to assist physicians to quickly determine the status of subjects for which physicians can provide a quick pre-diagnosis. Hence, to address these issues, this study proposes a novel OSA screening approach to achieve a satisfactory performance using fewer features under limited capacities of wearable devices. The Experimental results of applying this approach to different data sets are provided and discussed.

6.4.1 Methodology

The aim of this study was to propose a novel scheme for OSA detection based on features of ECG signals. This scheme is a hybrid algorithm that combines the Deep Neural Network (DNN) classifier with the Decision Tree classifier. The classification process in this proposed scheme consisted of two phases; the first phase used DNN for minute-based classification, then the output of this phase was fed into a decision tree model to perform the second phase; class identification. This work was based on the ECG signal features to detect sleep apnea. Figure 6.16 represents the block diagram for the methodology proposed in this study.

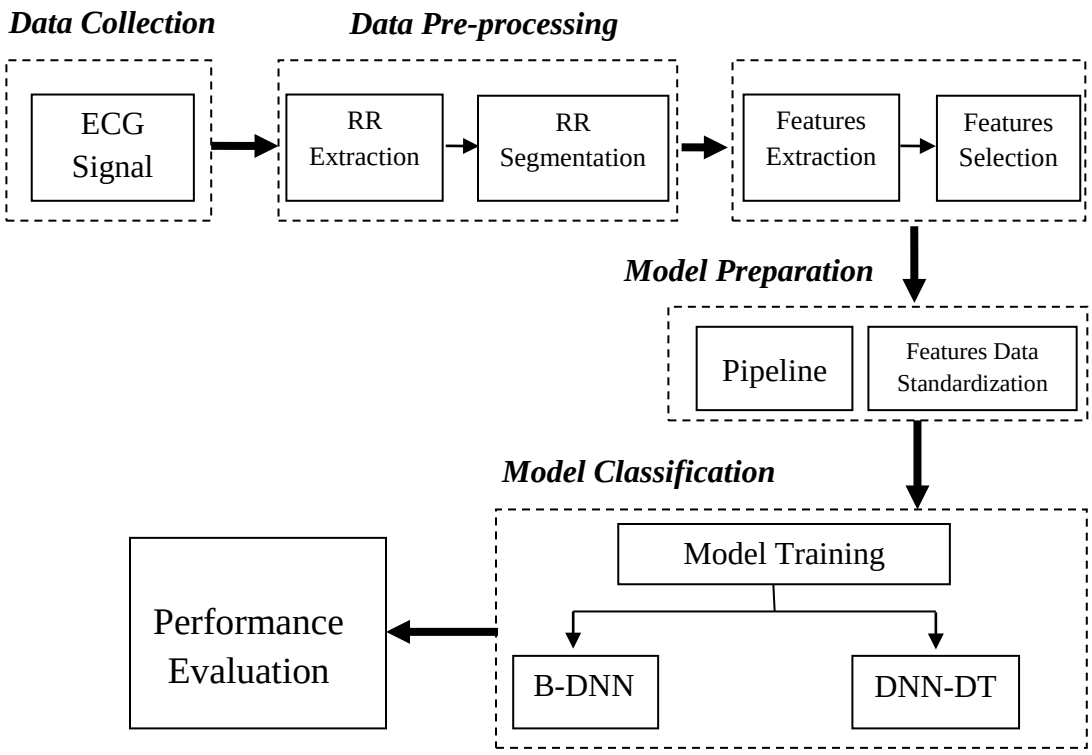


Figure 6.16: Block diagram of the proposed methodology

6.4.2 Model Classification

In the classification process, the extracted and selected features had to be fed into the training model to classify each minute of ECG data.

In the context of the proposed approach, the deep learning model included two phases. In the first phase; a baseline deep neural network (B-DNN) model was used. This model was mainly used for the stage of minute-based classification. While in the second phase; a hybrid model was designed by the combination of the deep neural network model and the decision tree model. This hybrid model was used for the stage of minute-class-based classification.

The Keras (keras, 2016) Library was used for building the proposed deep models. Keras is a highly modular neural networks library, written in Python and capable of running on top of either TensorFlow (tensorflow, 2016) or Theano (LISA Lab., 2016). In the proposed approach, the scikit-learn package was used to evaluate the model using a stratified k-fold cross validation. More details about the proposed deep models are provided in the following subsections.

6.4.2.1. Phase 1 : Baseline Deep Neural Network Model (B-DNN)

- **Training Approach**

Deep learning is built around a hypothesis that a deep, hierarchical model can be exponentially more efficient at representing some functions than a shallow one (Bengio, 2009). Deep learning methods aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features. Multilayer feed forward neural network is an example of deep learning. (Vincent et al., 2008).

Deep Neural Networks (DNN) have the potential to extract better representations from the raw data to create much better models and extracting features that are impossible to construct manually through many layers of nonlinear transforms and features that discriminate classes better. This is shown by higher classification accuracy when using deep networks compared to shallow networks with the same number of parameters (Seide et al., 2011), or manually constructed features (Yu & Deng, 2010).

The emphasis in shallow learning is often (not always) on feature engineering and selection while in deep learning the emphasis is on defining the most useful computational graph topology and optimizing parameters correctly (Kayser, 2016).

Most of the recent experimental results with deep architecture are obtained with models that can be turned into deep supervised neural networks, but with initialization or training schemes different from the classical feed-forward neural networks (Bengio & Glorot, 2010) .

One of the biggest advantages of Deep Learning is enormous flexibility in designing each part of the architecture, resulting in numerous ways of putting priors over data inside the model itself, finding the most efficient activation functions or learning algorithms (Janocha & Czarnecki, 2017).

Regarding the training of deep networks; DNNs can be trained with different approaches. Hinton, et al. (2006) started a revolution in Deep Learning when they gave empirical evidence that if DNNs are initialized properly using unsupervised pre-training (Erhan et al., 2010; Bengio & Glorot, 2010), then good solutions can be found in a reasonable amount of runtime. Recently, there is less of evidence that pre-training actually helps.

Several other solutions have since been put forth to address the issue of efficiently training DNNs with the original training procedure but with different training mechanisms based on topological measures (Glorot et al., 2011). These include heuristics such as dropouts (Srivastava et al., 2014), but also considering alternate deep architectures such as convolutional neural networks (Sermanet et al., 2014), and deep Boltzmann machines (Salakhutdinov & Hinton, 2009). In addition, deep architectures based on new non-saturating activation functions have been suggested to be more effectively trainable. The most successful and widely popular of these is the rectified linear unit (ReLU) activation, which is the focus of this work.

This work proposes a new contribution to the trend of training deep neural network based on topological concepts like weights initialization and activation functions. This is in part inspired by observations obtained from the work proposed by Bianchini & Scarselli (2014) which approved that for those neural networks with the same number of hidden units; deep architectures, with arctangent and polynomial activation functions, can realize maps with a higher complexity with respect to shallow ones.

Also the work proposed by Seide et al. (2011) approved empirically that deep supervised neural networks can reach their best performance without requiring any unsupervised pre-training. This finding was an attempt to close the performance gap between neural networks learnt with and without unsupervised pre-training.

- **Training Measures**
- **Weights Initialization**

There has been lot of research regarding the appropriate values of initial weights, which is really important for an efficient convergence. Furthermore, as shown by Bengio & Glorot (2010); units with more incoming connections should have relatively smaller weights. Weights of The proposed DNN are initialized using a small Gaussian random number.

- **Loss function**

The used loss function during DNN training is Cross Entropy which is logarithmic loss function preferred for binary classification problems.

Cross entropy, also referred to as log loss, is a cost function which in a supervised learning problem measures the compatibility between a prediction (e.g. the class scores in classification) and the ground truth label. Cross entropy is an improvement over the hinge loss function in that it generates scores that are more meaningful and easier to interpret by using probability of the input belonging to each class as the output. For this reason Cross entropy is the most commonly encountered loss function for classification problems (Oruganti, 2016).

Cross entropy is one of the most favourite loss functions that improve the performance of the DNN training. Janocha and Czarnecki (2017) investigated how particular choices of loss functions affect deep models and their learning dynamics. They found that Cross entropy is preferable for deep learning and assists in improving the performance of network training.

Furthermore, Veselý, et al. (2013) used cross entropy loss function in their Sequence-discriminative training of DNN on the 300-hour Switchboard conversational telephone speech task. Authors achieved state-of-the-art results on their task.

- **Activation functions**

In the proposed DNN model, Rectified Linear Unit (RELU) activation is used, which is simply the half-wave rectifier *f(z)= max(z, 0)*. RELU is a non-linear, differentiable activation function that is considered the most popular activation function in deep neural networks due to its ability to circumvent the vanishing gradient problem and inducing the sparsity in the hidden units (Oruganti, 2016). ReLU typically learns much faster in networks with many layers, allowing training of a deep supervised network without unsupervised pre-training (Glorot et al., 2011).

Before ReLU activations were commonly used, deep neural networks were nearly impossible to train since the neurons would get stuck in the upper and lower areas of sigmoidal activation function (Cybenko, 1989) and hyperbolic tangent function (Harmon & Klabjan, 2017).

Dahl et al. (2013) explored the behaviour of deep neural nets using ReLUs on a 50-hour broadcast English Broadcast News task. They proved that the modified DNNs using ReLUs were faster to train than standard sigmoid units and provided a good relative error reduction over standard pre-trained DNNs.

Likewise, Arora et al. (2016) acclaimed that deep architectures based on non-saturating activation functions have been suggested to be more effectively trainable and the most successful and widely popular of these is the rectified linear unit (ReLU) activation.

- **Optimization**

4 Once all the derivatives are computed, parameters are updated using the efficient Adaptive Moment Estimation (Adam) optimization algorithm (Kingma & Ba, 2014) for gradient descent. Adam is a method for efficient stochastic optimization that only requires first-order gradients with little memory requirement. The method computes adaptive learning rates for each parameter.

Ruder (2016) investigated algorithms that are most commonly used for optimizing Stochastic Gradient Descent (SGD) and he showed empirically that Adam works well in practice and compares favourably to other adaptive learning-method algorithms.

Error! Reference source not found. shows the structure of the B-DNN model.

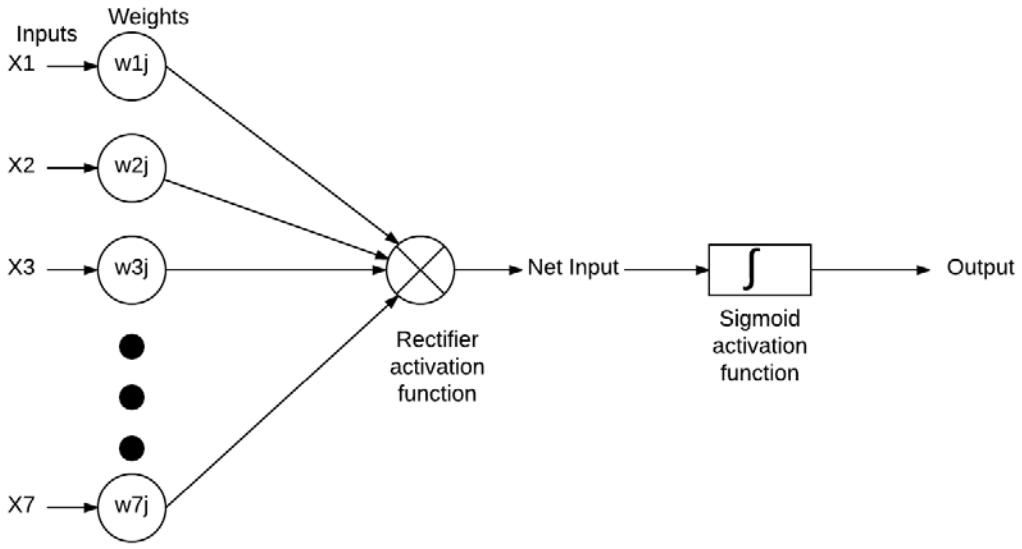


Figure 6.17: Structure of B-DNN Model

- **Network Architecture**

The proposed model uses feed-forward neural network architecture which is called a Multi-Layer Perceptron (MLP). It consists of 4 layers; the input layer, two hidden layers and output layer. Below a brief description of the components of the architecture is given.

- **Input Layer**

The input layer of the proposed model consists of 7 neurons which is the same number as selected features. Previous work has shown that higher-level representations of deep networks tend to be better by applying the standard practice of replacing input with extracted features in order to improve the performance of a machine learning model (Bengio, 2009). Chen and Deng (2013) reported that a better speech recognition performance could be achieved by employing this strategy at their deep RNN. Also, Pascanu et al. (2013) hypothesized that such higher-level representations should make it easier to learn the temporal structure between successive time steps because the relationship between abstract features can generally be expressed more easily.

- **Hidden Layers**

The proposed had two fully connected hidden layers. A neural network topology with more layers offered more opportunity for the network to extract key features and recombine them in useful non-linear ways.

The first hidden layer had the same number of neurons as input variables (7 neurons).While the second one was added to force a type of feature extraction by the network by restricting the representational space, since it took an input of 7 neurons (i.e. same number as of the selected features) and reduced it to 5 (i.e. a new representation of the input features). This put pressure on the network during training to pick out the most important structure in the input data to model.

Both hidden layers are trained using RELU activation function which allows a network to easily obtain sparse representations that leads to mathematical advantages (Glorot et al., 2011).

The proposed MLP model with two hidden layers seems to be shallow network. However simple training of the neural networks with up to 2 or 3 multiple hidden layers have shown an improvement but further increases in the number of layers did not provide any significant improvement and in some case the results were worse (Bengio, 2009). The existing algorithms have faced the problem of the local minimum and it has been reported that the generalisation of such gradient-based methods have become worse with a larger number of layers.

The proposed approach is inspired by the recent attempt of Arora, et al. (2016) that proposed algorithm to train a ReLU DNN with one hidden layer and approved its efficiency. Furthermore, Ba and Caruana (2014) demonstrated empirically that when Single-layer fully-connected feed-forward nets trained to mimic deep models can perform similarly to well-engineered complex deep convolutional architectures. The results suggest that the strength of deep learning may arise in part from a good match between deep architectures and current training procedures, and that it may be possible to devise better learning algorithms to train more accurate shallow feed-forward nets. For a given number of parameters, depth may make learning easier, but may not always be essential.

- **Output Layer**

This layer is composed of one neuron that is responsible for producing and presenting the final network outputs, which result from the processing performed by the neurons in the previous layers. This layer used the sigmoid activation function in order to produce a probability output in the range of 0 to 1 that could easily and automatically be converted to crisp class values.

Figure 6.18 presents the architecture of the proposed B-DNN model. It shows that the model consisted of 4 layers; the first was the input layer with 7 neurons (i.e. again, the same number as selected features), the second in the first hidden layer with 7 neurons which in turn passed the values to the second hidden layer that squeezed the representational space of the network to have only 5 neurons, that was then fed to the output layer which had only one neuron that presented the prediction result (Apnea or Non- apnea).

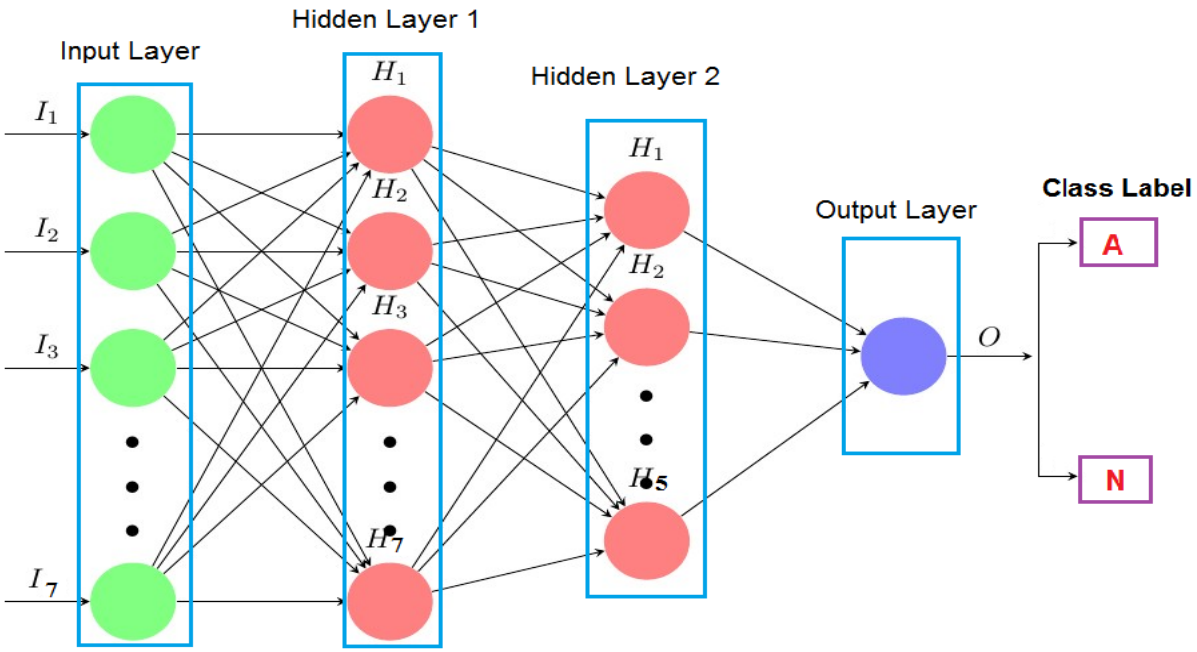


Figure 6.18: DNN Model Architecture

As previously mentioned, this work proposed a scheme for minute-based classification and minute-class-based classification of sleep apnea. This B-DNN model was used for achieving the first phase, which was minute-based classification.

6.4.2.2. Phase 2: Hybrid Deep Neural Network and Decision Tree Model (DNN-DT)

This model was a hybrid algorithm that combined the Deep Neural Network (DNN) classifier with the Decision Tree classifier. Each file of the data set passed through two stages of classification; the first stage was for quantification of apnea minutes; where each minute was classified as Apnea or non-apnea using the D-BNN classifier. Then the output of the first phase that was performed using the B-DNN model (i.e. which classified minutes as apnea or non-apnea) was fed into a decision tree model to perform class identification (Class A, B or C). Totally, the result was used for the full minute-class-based classification phase. Figure 6.19 shows the architecture of the proposed DNN-DT.

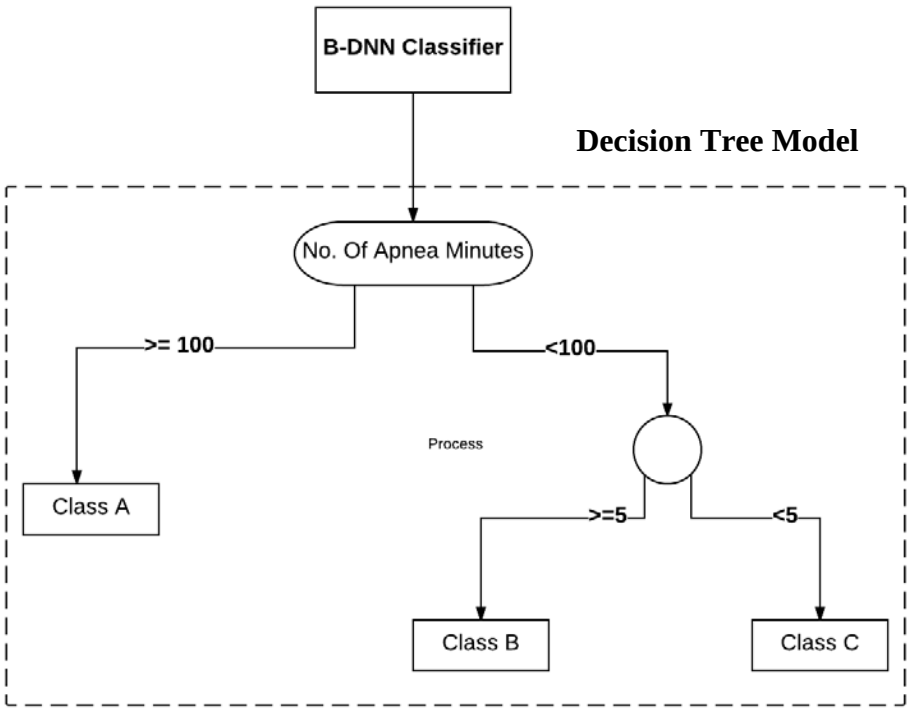


Figure 6.19: DNN-DT Classifier Architecture

6.4.2 Experimental Results

The proposed deep learning approach was evaluated for the three previously mentioned datasets; Apnea-ECG, CAD and Arrhythmia. The Performance results are presented in the next subsections.

6.4.3.1 Apnea-ECG Dataset

The performance of the proposed classification models was evaluated for both the minute-based classification and minute-class-based classification of sleep apnea.

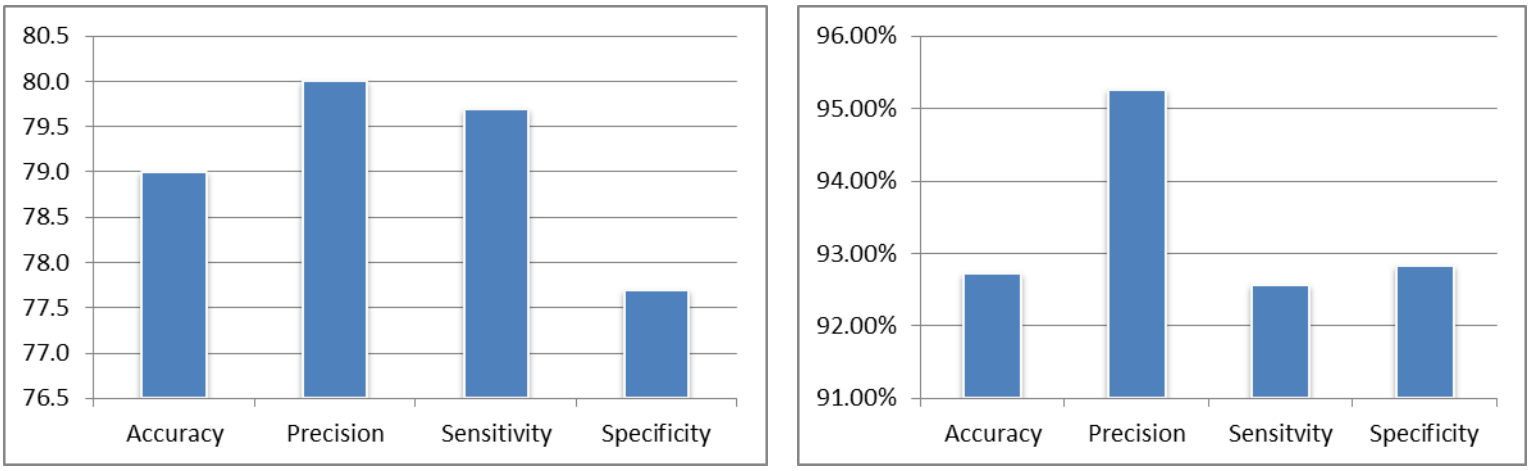
- Minute-based Classification

Since only the training set (35 ECG recording) of PhysioNet Apnea-ECG data base contained minute-wise apnea annotations, given the necessity of annotated test data to evaluate the classifier's performance, only these 35 recordings were used in the experiment. As previously mentioned, the proposed approach was evaluated using the 10-fold cross validation technique. The performance of the proposed classifier is presented in Table 6.7 and plotted in Figure 6.20. Figure 6.20 also shows the performance of the

classifier, before and after features selection. It can be observed that the classifier performed better after selecting the top seven relevant features from the whole number of features (11 features). The results show that the compact feature subset had a good effect on the accuracy of the classification system. The proposed B-DNN achieved a 13.7 % increase in accuracy.

Table 6.7: A summary of classifier performance for minute-based Classification

Classifier	# Features =11					# Features =7				
	Acc.	Prec.	Rec.	Sens.	Spec.	Acc.	Prec.	Rec.	Sens.	Spec.
Proposed B-DNN	79.0	80.0	79.7	79.7	77.7	92.7	95.3	92.6	92.6	92.8



(a) **(b)**

Figure 6.20: B-DNN Model performance obtained : (a) before features selection (b) after features selection

Table 6.8 presents the confusion matrices for the proposed classification models before and after features selection. It is clear from the results that the performance is improved after features selection. Before features selection; the model correctly classify 5290 out of 6514 apnea minutes; while after features selection, it correctly detect 6009 apnea minutes. Also, the number of non-apnea minutes in increased from 8150 out of 10489 to 9031 after features selection. This indicates that the model performed better with less number of features.

Table 6.8: Confusion Matrix before and after features selection for Apnea dataset

# Features =11				# Features =7			
	A	**N**	**∑**		**A**	**N**	**∑**
A	5290	2339	**7629**	**A**	6009	1458	**7467**
N	1224	8150	**9374**	**N**	505	9031	**9536**
∑	**6514**	**10489**	**17003**	**∑**	**6514**	**10489**	**17003**

- Minute-Class- based Classification

The second phase of the proposed approach was for both the detection of the apnea class and the quantification of apnea minutes. The number of the classified minutes for each recording was used to determine whether a patient recording belonged to class A, B or C unlike the state-of-art methods that were only able to classify two classes instead of three. As mentioned previously, the PhysioNet database for the training set contained 20 recordings of class A, 5 of class B and 10 of class C.

Table 6.9 provides a summary of the performance of the DNN-DT scheme for the first five files of the data set. The table provides the actual class, the predicted class, Accuracy, Precision, Recall and finally the confusion matrix for each file. The performance summary of other files of the dataset is available at Appendix IV.

Table 6.9: Performance Summary for minute-class-based model of DNN-DT scheme for minute's quantification and class detection for each file of the Apnea dataset

File Name	Act. Class	Pred. Class	Acc.	Prec.	Rec.	Confusion Matrix
a01	A	A	97.55%	98%	98 %	**A** 469 8 **N** 1 11 **∑** **470** **19** **489**
a02	A	A	82.78%	85%	86 %	**A** 408 64 **N** 12 44 **∑** **420** **108** **528**
a03	A	A	90.18%	92 %	92%	**A** 236 31 **N** 10 242 **∑** **246** **273** **519**
a04	A	A	98.38%	99%	99%	**A** **N**

						A	452	3	
						N	1	36	
							453	**39**	**492**
a05	A	A	81.29%	83%	83		**A**	**N**	
						A	242	43	
						N	34	135	
							276	**178**	**454**

Based on the results available at Table 6.9; Figure 6.21 summarises the Accuracy, Precision, and Recall results for the quantification of the minutes stage of the DNN-DT scheme. Results show that minutes of all files are classified correctly with high accuracy ranges from 80 % to 100% which indicated that the proposed model performs very well and effectively applied to the Apnea dataset.

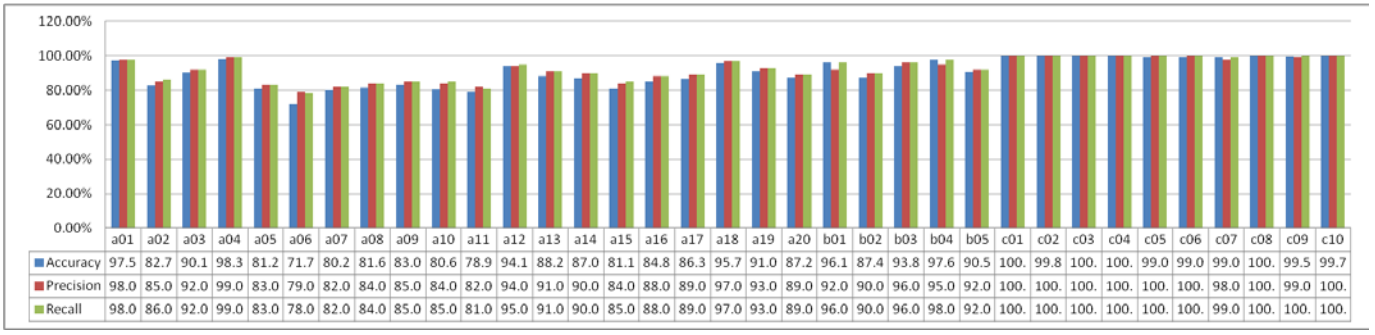


Figure 6.21: Performance of the minute-class-based classification at DNN-DT scheme

The results for the class detection stage of the DNN-DT scheme is summarised in plotted in Figure 6.22 ; where the average values for accuracy, sensitivity, specificity and precision are provided for the three apnea classes. The results show that the model achieved high performance regarding all metrics and approves its effectiveness and robustness.

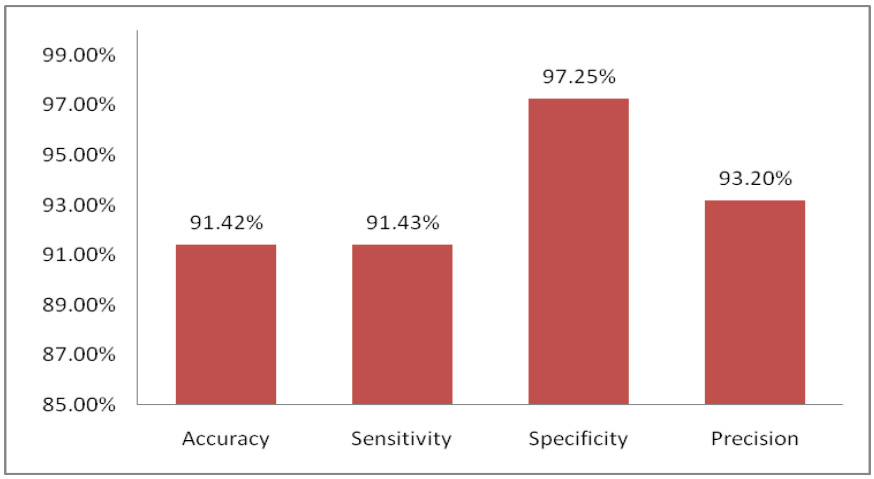


Figure 6.22: Average Performance of the class-based classification stage at DNN-DT scheme

Figure 6.23 presents the confusion matrix for the class detection stage of the scheme. The model was able to correctly classify 19 files of class A and misclassified only one file to consider it from class B. At the same time it misclassified 2 files of class B and considered them from class C. Regarding the files from class C; they are all correctly classified. It can be deuced from this that the most misclassified classes are from class B which is considered by the state-of-art work as a misleading class due to its characteristics that is very close to the characteristics of Class C.

	A	B	C	∑
A	19	1	0	20
B	0	3	2	5
C	0	0	10	10
∑				35

Figure 6.23: Confusion matrix for the class-based classification at DNN-DT scheme

6.4.3.2 CAD Dataset

The proposed deep learning approach was also applied to the CAD dataset in order to evaluate the model behaviour for such small dataset. Figure 6.24 below shows the performance results after applying B-DNN to CAD dataset. The proposed model does not

achieved high accuracy comparing with the results of the apnea dataset. This is due to the nature and the size of the dataset.

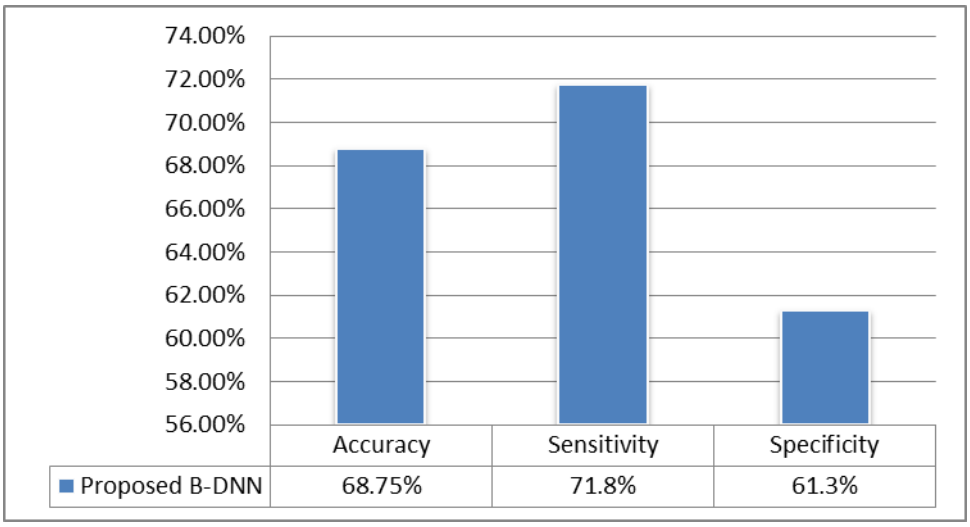


Figure 6.24: Performance of B- DNN-DT at CAD dataset

The confusion matrix is presented in Figure 6.25. The proposed model was able to correctly classify 71.77% of CAD cases and 61.3 % of non-CAD cases. This is reasonable values relative to the overall accuracy.

	C	**N**	**∑**
C	351	77	**490**
N	138	122	**198**
∑	**489**	**199**	**688**

Figure 6.25: Confusion matrix of B- DNN scheme at CAD dataset

6.4.3.3 Arrhythmia Dataset

The proposed B-DNN was also evaluated using the Arrhythmia dataset. Figure 6.26 below shows the performance results. It achieved good performance measures; better than the results of CAD dataset. The confusion matrix is presented in Figure 6.27; where it shows that the proposed model was able to correctly classify 200 arrhythmia cases out of 245 and 152 non-arrhythmia cases out of 185.

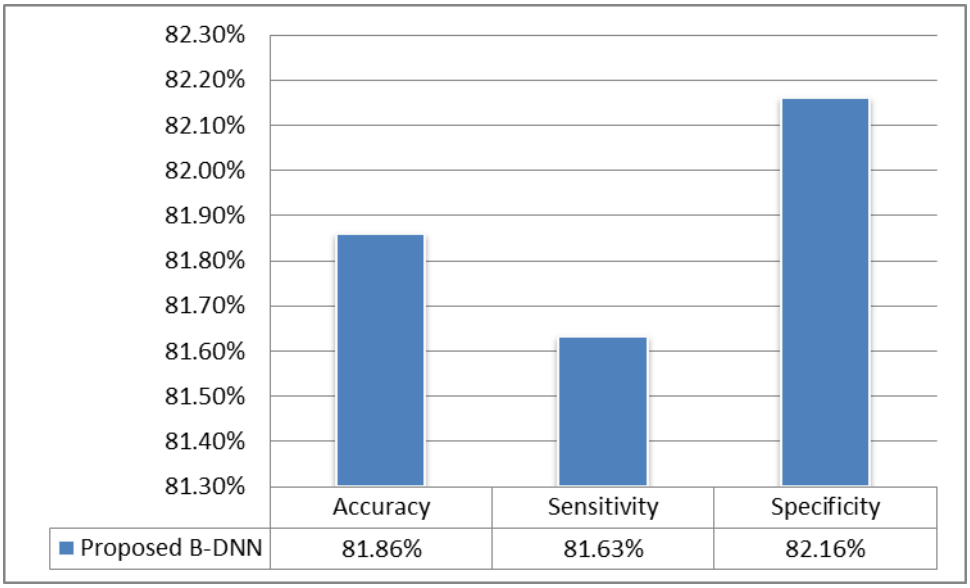


Figure 6.26: Performance of B- DNN-DT at Arrhythmia dataset

	A	N	∑
A	200	33	**233**
N	45	152	**197**
∑	**245**	**185**	**430**

Figure 6.27: Confusion matrix of B- DNN scheme at Arrhythmia dataset

6.5. Discussion

In this chapter, two data mining approaches were evaluated. The first one used several well-known data mining techniques for heart disease diagnosis focusing on three famous types namely CAD, Arrhythmia and Sleep Apnea. The work in this part was introduced to test the used datasets and validate its effectiveness and completeness, and therefore find a starting point for improvement in advanced data mining approaches.

The models were built on the three previously mentioned datasets with four well-known data mining algorithms i.e. Logistic Regression, KNN, SVM and Naïve Bayes using Orange data mining software. The performances of the models were evaluated using the standard metrics

of accuracy, precision, sensitivity and specificity. 10-Fold Cross Validation was adopted for random sampling of the training and test data samples.

For each technique, the performance obtained for each disease was evaluated. For example, the maximum accuracy obtained to diagnose Sleep apnea was 80.5% achieved using the KNN algorithm. Many studies in the literature were proposed to detect Sleep apnea using these well-known algorithms. For example, Mendez, et al. (2007) employed KNN for categorizing events into normal and apnea with an average accuracy of 85% in both training and testing. Jezzini, et al. (2015) achieved a classification accuracy of 98.7 % using KNN, 97.5% using SVM and 96.25% using Naïve Bays. Correspondingly; Almazaydeh, et al.(2012) achieved an accuracy of 96.5 % using SVM. while; it is not simple to give a logical explanation for such a high difference in the obtained results when compared with the state-of-the-art methods; but the same features were extracted for the same dataset and applied to reliable data mining software; and got such results. In addition, the authors of these papers were contacted to get their experimental work to test but could not gain their cooperation. But, the work was just to validate and test the datasets.

The performance of classification methods to detect CAD diseases was not encouraging (i.e. between 72.5%-86.3%). The highest performance results were obtained by the Naïve Bayes Algorithm. Other research works explored CAD diagnosis using the well-known data mining algorithm and achieved reasonable performance (Alizadehsani, et al., 2012); (Babaoglu, et al., 2010). They achieved an accuracy of 79.17 using SVM, whereas other researchers (Setiawan, et al., 2009) achieved an accuracy of 81.5 using KNN algorithm. Also, Babaoglu, et al. (2009) reported the use of exercise test data which was applied to the SVM algorithm to detect CAD. They finally achieved 81.46% accuracy.

Because such low results were achieved by the classifying methods in our proposed approach, it is strongly recommended that further research be carried out using different algorithms to improve the prediction of CAD diseases.

Regarding the Arrhythmia detection; the best performance results were achieved by both Logistic Regression and SVM with an accuracy of 83.26%, whereas the Naïve Bayes classifier achieved an accuracy of 72% and the KNN classifier achieved an accuracy of 64.88%. Other researches in the literature for arrhythmia combined the traditional algorithms with more advanced approaches to improve the classification performance. (Özbay, et al., 2006), (Nasiri, et al., 2009); (Huang, et al., 2014).

Another important issue when dealing with health datasets is features selection techniques and how to determine the most important features that lead to more accurate diagnosis. In this work, the ANOVA test was applied to determine the most relevant feature in the classification process. Results showed that in the case of the Apnea-ECG data set, features selection improved the results. This finding was consistent with results obtained from other researches (TİMUŞ & KIYAK, 2015); (Noviyanto, et al., 2011); (Ravelo-Garcia, et al., 2013). While in the case of CAD and Arrhythmia datasets, the classification results were better when all features were involved in the classification process.

The second part of this chapter investigated the detection of heart disease based on the deep learning approach. As indicated before; the major application for this part was the Apnea-ECG data set. The first data mining goal for this part was to classify each minute in all the patient's ECG recordings as either containing Apnea or not, using a deep learning technique. The proposed deep model was able to achieve this goal by classifying 92.7% of the minutes correctly. As the classification accuracy value was high, cardiologists could rely on it for assisting Sleep Apnea diagnosis.

The deep learning approach was not widely used in sleep apnea detection. As far as it was possible to establish; only one other research addressed sleep apnea detection using deep learning (stacked auto-encoder with deep neural network) on smartphones and small embedded systems with an accuracy range of 87-90%.

This gave an assurance that the proposed approach was a novel approach for sleep apnea detection, which outperformed all other available approaches.

The second data mining goal regarding this data set was both the quantification of apnea minutes and detection of the apnea class (class A (apnea), class B (borderline) and class C (control)). As mentioned before; this goal was achieved by using a novel approach that was a hybrid algorithm which combined the Deep Neural Network classifier with the Decision Tree classifier. The proposed approach classified apnea classes with an average accuracy of 91.4%. Again, as far as it is possible to know, this combination has not been used before in sleep apnea detection. This was considered a good contribution for this research. Another combination had been developed Tagluk, et al. (2010) using wavelet transforms and an artificial neural network (ANN) to classify sleep apnea and achieved an accuracy of 78.85% which was relatively low when compared with the proposed approach in this research.

Another contribution in this area was that the proposed approach identified the three apnea classes (A, B and C) while other papers only identified two classes (A and C). As far as it is possible to know, other authors (Babaeizadeh, et al., 2010; Rachim, et al., 2014) have addressed both apnea detection and quantification for each patient recording, but both identify only two classes not three (i.e. class B is excluded).

Regarding the third data mining goal which was exploring the effect of features selection on the classification performance; it was found that reducing the number of features (from 11 to 7) enhanced the classification accuracy. The results showed that the proposed approach performed better after reducing the features set indicating that the selected features were in fact highly relevant in classifying sleep apnea. The selected statistical features set was a hybrid combination of the features proposed by other researchers (Chazal, et al., 2004) and (Yilmaz, et al., 2010). (Isa, et al., 2010) also examined these two approaches to select the optimal features set. The number of features on Chazal et al. features more than Yilmaz et al. (8 versus 3). It was found that the classification results using only 3 features, as proposed by Yilmaz et al. gives about a 3.59% gain on the

overall classification accuracy (CA) compared with the classification using 8 features as proposed by Chazal et al. The proposed approach in this study with its selected hybrid features outperformed these previous studies and achieved a gain of a 13.7 % on the overall accuracy using the features selection technique.

Features selection played a key role in the performance measure by removing irrelevant attributes from the dataset and so increasing classification accuracy. This finding is consistent with results obtained from other researchers (TİMUŞ & KIYAK, 2015); (Noviyanto, et al., 2011); (Ravelo-Garcia, et al., 2013).

Deep learning approaches are often said to require enormous amounts of data to work well, but recently this notion has been challenged, and different researchers argued this matter. (DIELEMAN, 2015) developed a deep convolutional neural network as a solution to the data science competition held by the National Data Science Bowl (Kaggle, 2014). Their results in this competition proved that it was not necessary for deep approaches to require large datasets, since they only used a relatively small dataset. They indicated that judicious use of techniques to prevent overfitting in such small datasets such as dropout, weight decay, data augmentation, pre-training, pseudo-labeling and parameter sharing, had enabled them to train very large models with up to 27 million parameters on a small dataset. Also, (Menkovski, et al., 2015) discussed the challenge of the deep learning approach in medical imaging where annotated data was usually very scarce. They demonstrated that a deep neural network model can be trained even on relatively small number of annotations if proper augmentation and regularisation was implemented.

In this context; the proposed deep approach was applied to CAD and Arrhythmia data sets to explore the behaviour of this approach with such a small-size dataset. It achieved an accuracy of 68.8% with the CAD dataset and 81.9% with the Arrhythmia dataset. This performance is acceptable with such small datasets. It can be claimed that deep neural network model scales if the network was complex enough so that the model will not over-fit. Increasing the number of features overcomes the small size of the dataset, so that

the model can compensate the gap of knowledge obtained from the training examples by the prepared extracted features. This finding was consistent with the conclusions demonstrated by the previously mentioned studies (Dieleman, 2015) ; (Menkovski, et al., 2015); (Zhu, et al., 2006). Deep learning with a small dataset is a challengeable issue; and the work in the proposed approach here is a starting point for future improvements.

Most previous researchers that worked on CAD did not apply deep learning approach with its special techniques. They just applied Multilayer Perceptron and considered it as a deep learning approach which was effectively a semi-deep learning approach. One researcher (Babaoglu, et al., 2009) applied Artificial Neural Networks (ANN) to determine CAD. The diagnostic accuracy for the Left Main Coronary (LMCA) left anterior descending and left circumflex coronary arteries using ANN were 91%, 73% and 65%. And 69% for the right coronary artery was also predicted. Other researchers (Atkov, et al., 2012) also developed an artificial neural networks-based (ANNs) diagnostic model for coronary heart disease (CHD) using a complex of traditional and genetic factors of this disease. The best result (94%) was achieved in a Multilayer Perceptron (MLP) model with two buried layers and 10 factors (profession, LDL, HDL, triglycerides, cholesterol rate, SCORE index, left ventricular ejection fraction, family CHD history, coronary anterior.

The detection of arrhythmia is similar to CAD detection using deep learning. It is limited to the use of multilayer neural network classifier as in the research done by Yu & Chen (2007). This approach combined the Multilayer Perceptrons (MLP) and Probabilistic Neural Networks (PNN). In another work, Güler & Übeyli (2005) used a similar approach by combing the neural networks to classify the ECG arrhythmias and achieved reasonable performance accuracy of 96.94%.

6.6. Summary

In this chapter, a hybrid approach that included deep neural networks and decision trees, for the detection and quantification of sleep apnea using features of ECG signals was proposed. Statistical features were extracted from the RR interval and served as training and testing data for the applied classifiers. The proposed approach treated, with novelty,

the following points: (i) identified both apnea classes and detected minute-by-minute classification unlike the state-of-the-art methods which either identified apnea class or detected its presence, (ii) identified the three apnea classes (A, B and C) while other papers only identified two classes (A and C), (iii) makes a comparative study of the most used classification methods adopted in the literature but using the same features and the same dataset. The experimental results showed that this approach was robust and computationally efficient and clearly outperformed the state-of-the-art methods.

Regarding network training, the proposed deep approach went in the direction of improving the training procedure of the network instead of the unsupervised pre-training process which recently has been largely obsoleted. Instead of pre-training, the difference is now in the activation functions, learning algorithm and other topological measures.
The proposed approach used RELU as an activation function which make training faster because it is faster to compute and because the optimisation is easier and also it suffers significantly less from the vanishing gradient problem. Furthermore, Cross Entropy was used as a loss function which in approved its efficiency in deep networks for binary classification problems. Additionally, Adam was used for optimization. The obtained results showed that deep learning can be applied by optimizing the training procedures and learning algorithms to train feed-forward neural networks for a given number of parameters. Also the depth of the neural network is not the concern to obtain good performance. Training measures is the most important.

Deep architectures have benefited much more from the pre-training stage in terms of training efficiency and test performance. Deep learning involves creating complex, hierarchical representations from simple building blocks to solve high-level problems. The network learns something simple at the initial level in the hierarchy, then sends the information to the next level where the information is combined into something more complex. The process continues, with each level building from the input it received from the previous level.

Chapter 7 : Heart Disease Diagnosis Using Deep Radial Basis Function Kernel Machine

7.1. Overview

Over the years Radial Basis Function (RBF) Kernel Machines have been used in Machine Learning tasks, but there are certain flaws that prevent their usage in some up-to-date applications (e.g., some Kernel Machines suffer from fast growth number of learning parameters whilst predicting data with many variations). Besides, Kernel Machines with single hidden layer have no mechanisms for features selection in multidimensional data space, and the machine-learning task becomes intractable with a large amount of data available for analysis. To address these issues, this chapter investigates the usage of a framework for a "deep learning" architecture composed of multilayered adaptive non-linear components – Multilayer RBF Kernel Machine. To be precise, three different approaches of features selection and dimensionality reduction to train RBF based on Multilayer Kernel Learning are explored, and comparisons made between them in terms of accuracy, performance and computational complexity. As opposed to the "shallow learning" algorithm with usually single layer architecture, results show that the multilayered system produces better results with large and highly varied data. In particular, features selection and dimensionality reduction, as a class of the multilayer method, shows results that are more accurate.

7.2. Background

7.2.1. Deep Kernel Methods

Several attempts to combine the deep-learning approach with the kernel based method were made in the past. For instance, a method was proposed by Cho & Saul (2009) and Cho & Saul (2010) which mimics the behavior of the multilayer neural network via a single layer kernel network with specifically generated kernel functions (i.e. arc-cosine kernels). Arc-cosine kernels are produced with recursive substitutions of the output of the

kernel function. This action was equivalent to the non-linear transformation of the feature maps. (Bouvrie, et al., 2009) proposed the notion of the hierarchical feature invariance. Other papers proposed special kernels for visual feature recognition (Bo, et al., 2010); (Bo, et al., 2011). They are designed to recognize local compact shapes, intensity/color features as well as orientation features common in image processing tasks (SIFT, HOG). The compact representation for these descriptors is learned via projections with kernel principal component analysis. The learned kernel descriptors aggregate information for image patches. Then similar descriptors are applied on the next level to aggregated learned information which is later used to yield final output. In another paper (Mairal, et al., 2014) this idea was generalized and instead of ad-hoc kernels for intensity, color and shape a single representation was provided. These works are specially designed to work with images and their 2D representation. They try to mimic the behavior of the Convolutional Neural Network but instead of the neural nodes they use a kernel feature mapping and kernel descriptors.

A more general idea is to use multiple kernels connected on several levels but not to restrict them to the specific domain or locality principle. Cho & Saul (2009) proposed the first deep research work on this subject. They did not only propose the idea of the recurrent arc-cosine kernels which mimic the behavior of the deep neural network via such operations as multi-layered kernel composition, kernel multiplication and averaging, but also introduced multi-layer multi-kernel machines. The data processing in these machines was conducted in several steps. Firstly, the supervised filtering was performed, and then unsupervised dimension reduction was applied to the results. Finally, the obtained representation was fed to the kernel (i.e. in this particular case, the arc-cosine kernel). The process was repeated several times with the output of the first layer passed as the input to the next layer. Unlike single layer kernel compositions kernel machines allowed data filtering and pruning on each layer of the network which led to better classification performance.

Bach, et al. (2004) proposed the method of multi-kernel. It was intended to find the best linear combination of the basis kernel functions which in turn was used as the main

kernel for the SVM classification method. Other researchers (Zhuang, et al., 2011) combined this idea with the multi-layered kernel machines proposed by Cho & Sal (2009) and provided the method for classification with a two-layer multi-kernel machine. For that research, they introduced a function g which combined the lower level kernel in the kernel of the next layer based on the vector of parameters. The produced kernel was then used to make a final decision. Both types of coefficients (i.e. for the first and second layer) were solved in a single optimization problem. This approach helped to overcome the drawback of the multi-kernels machines from Cho & Saul (2009) because it provided the method of selection for the kernel combination, giving non-static adoptable kernels to the method and generalising for different types of kernels (i.e. the method can work with different kernels than arc-cosine).

Strobl & Visweswaran (2013) extended the results to multilayer kernel learning with sufficiently good generalisation on each additional level. The kernels on each layer were combined then the resulting system was applied to make a decision with the SVM classifier and finally depending on the error in the leave-out cross-validation the coefficients of kernel combination were updated.

Rebail et al (2016) used the same concept of the multi-layer kernel learning but replaced the optimising algorithm with an adaptive back-propagation algorithm which was adopted from the regular neural network learning procedure. The gradient descent and ascent methods were used to optimise weights on each kernel layer.

Huang, et al. (2014) tested the idea of ensemble kernel learning for faster training of deep kernel structures and successfully applied the algorithm in the field of sound processing. Weak learners adaptively select the part of the feature space according to the overall error level and generated feature mapping with the coefficients of the Fourier transform. Weak solvers were applied to the different parts of the dataset and together generated a feature map which further used for overall classification. The main advantage of this solver was that it was efficiently scalable and so could be easily applied to large datasets.

Jose, et al. (2013) used a slightly different approach. Instead of applying a global combination of kernels to the whole feature space it combined the localized –kernels, which was the combination of basis feature mappings with assigned probability distributions, showing which probability mapping was used for the particular data point. These localized kernels were further combined with global kernels which were applied for any data point. The main achievement of this method was to speed up the non-linear SVM learning.

Yger, et al. (2011) state that while a lot of studies are concentrated on the kernel selection, little attention has been given to the feature selection stage of the multilayer kernel learning. In the original paper (Cho & Saul, 2009), this was achieved by supervised filtering and unsupervised kernel principal component analysis. This procedure requires optimization of multiple parameters and therefore is computationally consuming. They proposed to overcome this difficulty by training each hidden layer via supervised partial least squared regression. The regression algorithm required the computation of several parameters which were used on the later learning stage and thus saved time.

It is interesting that most of the papers have adopted the approach used in the Convolutional Neural Networks. This approach includes feature extraction with further nonlinear transformation and filtering of the results. Almost all the papers described above try to mimic this behaviour in the kernel feature space. We have been able to find only a few works which use the second approach, which is unsupervised layer pre-training. For example, one researcher (Wang, 2011) described a method of kernel combination which required unsupervised kernel probability estimation and the results obtained were used as input for the next kernel level. Another researcher (Cho & Saul, 2009) proposed the use of kernel principal component analysis to select the features for each kernel machine layer. Thus, it seems that it could be a promising task to compare how different unsupervised techniques for feature selection could affect performance of the classification method and if it is possible to smoothly incorporate these methods in the kernel machine. Another opportunity is to check whether it is possible to build some

variation of the stacked auto-encoder based only on the kernel methods and also how well it will perform.

7.2.2. Unsupervised Kernel Techniques

Kernel principal component analysis (Welling, 2005) is an extension of the principal component analysis used for dimension reduction and feature extraction but with additional feature mapping into the Hilbert space with special kernel function.

Kernel canonical correlation analysis (Welling, 2005) is a special technique to extract into the kernel features space those features which allow common, correlated features for two datasets to be found. Projections which will transform similar data points from two data sets to close points in the projection space can then take place.

Spectral clustering (Ng, et al., 2001) is a clustering technique which combines data clustering with dimension reduction and performs data division in the lower dimension space.
Some authors (Wiering, et al., 2013); (Takeda, et al., 2007) have described techniques for supervised and unsupervised regression in the kernel space. Others (Wang, et al., 2010) have provided an extensive description of the kernel dimension reduction methods.

7.3. The proposed Deep RBF kernel machine

This thesis proposes a novel approach using Multilayer Radial Basis Function Kernel Machine based on deep learning. In this approach, some claims to knowledge are listed below;

- Three different approaches of features selection and dimensionality reduction to train the RBF kernel based on the idea of MLK were examined. MLKM approach has been suggested because the kernel machine was associated with a single layer that has no mechanism for feature selection, as evident in previous works.

- A multilayer radial basis for machine learning tasks was proposed because some kernel machines caused fast growth learning parameters which was the main cause of a huge number of variations.

- In reference to (Chatzis, et al., 2011) based on single hidden layer of kernel machines, which however no mechanism for multidimensional data space selection features, therefore a multilayer Radial Basis Function Kernel should be exploited

- A more flexible machine learning task with a more realistic sized dataset has been used for analysis.

- Although many machine learning algorithms have been developed for deep learning, the approach in this work is that the learning model as well as providing the prediction results, also learns an optimal data representation, which is achieved through a simple algorithm. Moreover, the new and innovative algorithm dealt with the problem identified with existing algorithms facing the local minimisation and generalisation of gradient-based methods, which have less potential as the number of data layers increases.

- The thesis has also examined and proposed the possibility of adopting a framework for a “deep learning” architecture composed of a multilayered adaptive non-linear component called a Multilayer RBF Kernel Machine.

- Most importantly, extending Cho & Saul(2009) algorithm for nonlinear transform by RBF kernel, unsupervised dimensionality reduction by kernel principal component analysis (PCA), and applying unsupervised kernel regression and removing the optional step of feature selection. With higher features selections, better computational time can be achieved
- The proposed framework explores three different approaches of feature selection and dimensionality reduction to train the RBF based on Multilayer Kernel

Learning. Based on these three selected approaches, a comparison was made in terms of accuracy, performance and computational complexity.

- The result of the proposed method provided better results with large datasets, which was justified using multi-layers as they required more data and better computational time suggested by Yger, et al. (2011).

- This proposed algorithm based on unsupervised latent space and the motivation behind this proposed system is that unsupervised methods work well with the regular neural networks and unsupervised learning focusing on important patterns from data regardless of their labels as it reduces the input dimensionality of data without losing crucial information

This work therefore provides three different approaches of features selection and dimensionality reduction to train RBF kernel based on the idea of MLK, and makes a comparison between them in terms of accuracy, the effect of changing the number of layers on the performance and computational complexity.

7.4. Methodology

This work shows how kernel methods can be extended to hierarchical structures without requiring complicated processing and computations. Three algorithms using RBF kernel are explored, and the main differences between them in terms of how to define the transformation through the combination of the linear mapping and the nonlinear activation function are studied after training and testing. In this section, a brief on the methodological steps is provided. Figure 7.1 presents the overview of the proposed methodology.

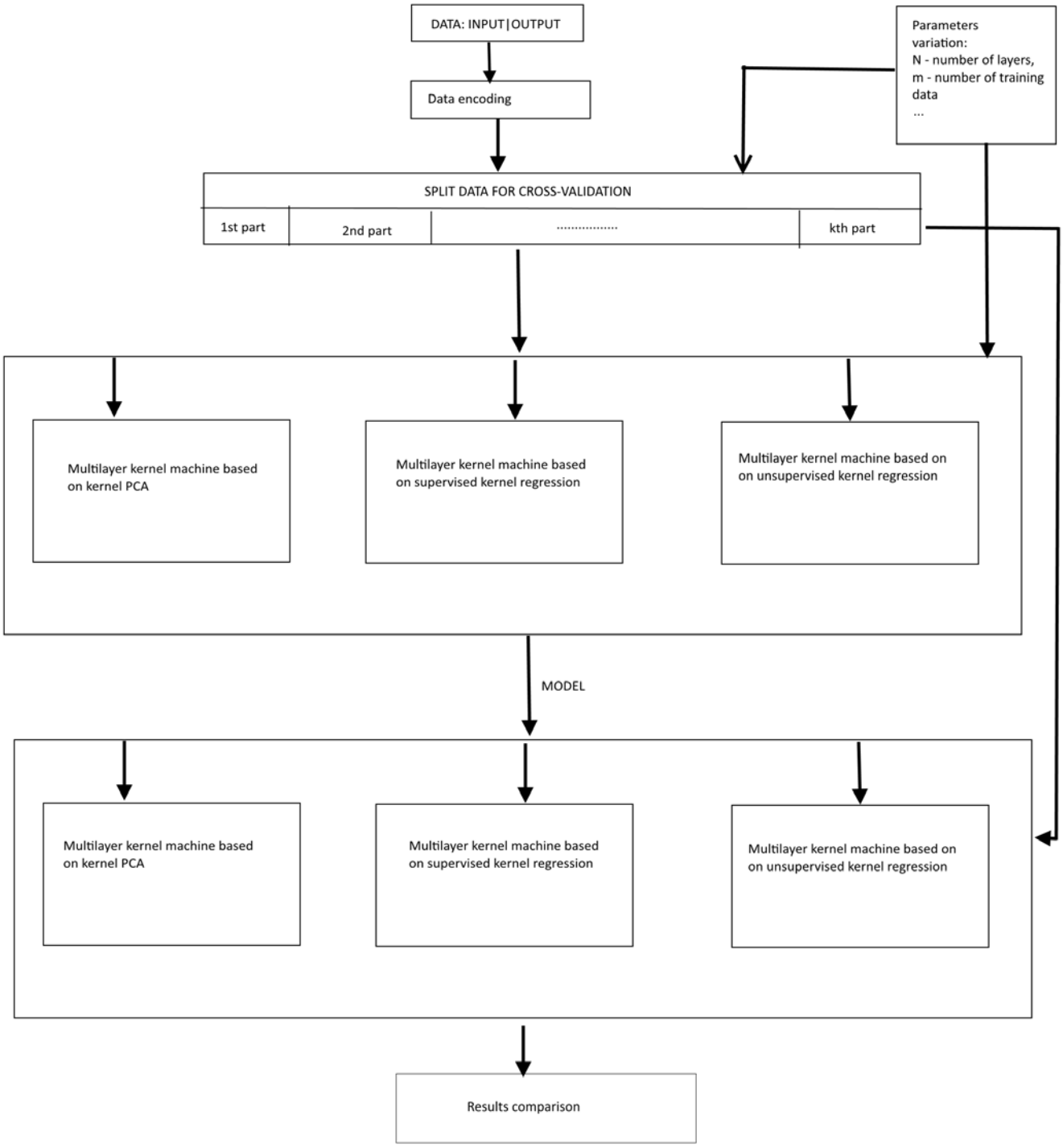


Figure 7.1: Methodology Overview

A description of the Multilayer kernels machine (MKMs) for the three different transformations including a multilayer RBF machine based on Kernel PCA, together with supervised and unsupervised regressions is shown in Figure 7.2.

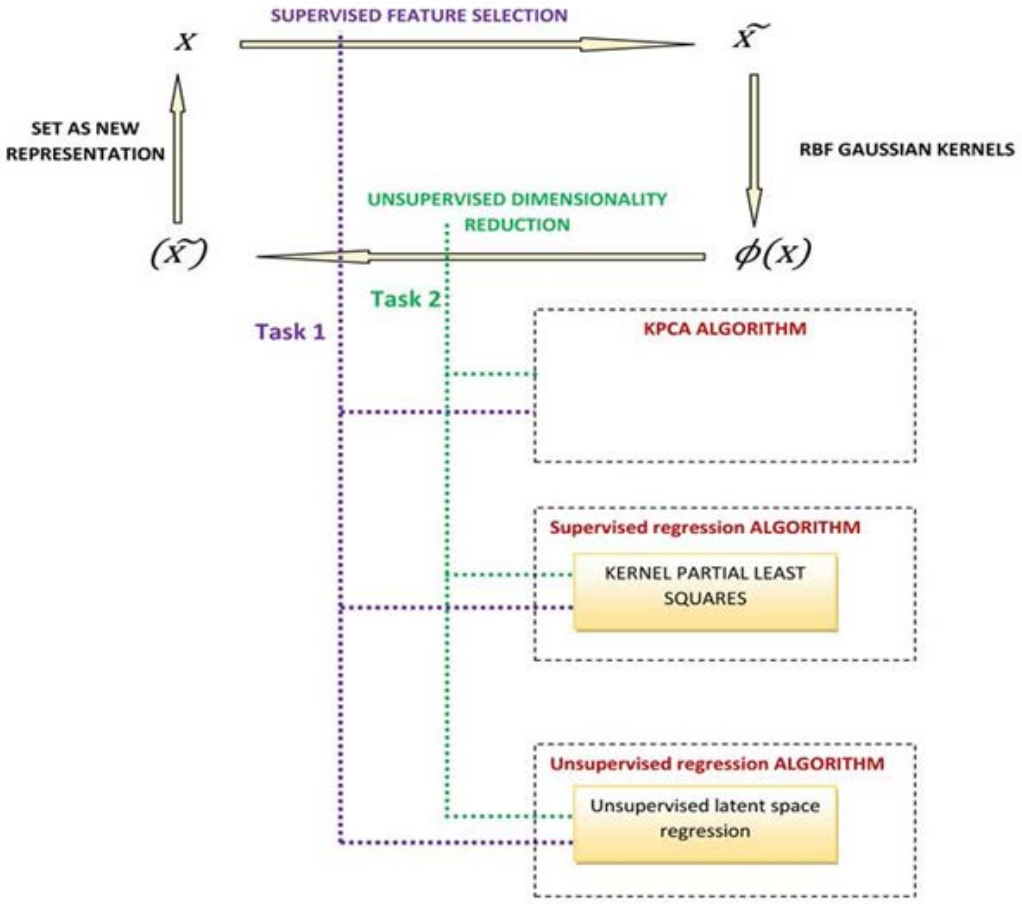


Figure 7.2: Multilayer kernels machine (MKMs) for the three different transformations

7.4.1. Multilayer RBF kernel machine based on Kernel PCA

The main concept in MKMs is the sequential transformation of input information using supervised feature selection, kernel PCA and unsupervised dimensionality reduction. The cycle shown in the figure 7.2 above depicts the steps for the implementation of the combined algorithm with an unsupervised dimensionality reduction, which will iterate

many times to formulate resulting multilayer kernel machines which incorporate an unsupervised regression algorithm into a kernel PCA to discard unwanted features.

Cho & Saul (2009) was the first to describe a multilayer kernel machine when he developed an arc-cosine kernel that mimicked the projections of a randomly initialized neural network.
This algorithm consisted of three including a nonlinear transform by the RBF kernel, unsupervised dimensionality reduction by kernel principal component analysis (PCA) and feature section by mutual information and this cycle was repeated multiple times to construct the feature hierarchy of MKM.
Below is a summary on how the Multilayer RBF machine based on Kernel PCA was implemented in this research which is also shown in Figure 7.2.

1. Let N be the number of layers to be used.
2. Using the ranking method, prune the features by removing redundant features.
3. Select appropriate kernels and kernel parameters (cross-validation or otherwise).
4. Apply the kernel PCA algorithm and use to extract the set of features in the next layer.
5. If number of iterations exceeds N go to step 6, otherwise go to step 2.
6. Feed the feature representations to the classifier to make the final decision.

The steps in the above procedures were typical of the standard approach. The promising outcome of this procedure was the justification to establish its implementation. A detailed discussion of the steps is presented below.

In Kernel PCA, iterative application is employed to realize deep learning in MKMs. Kernel PCA has been in existence since over ten years but has now been improved by adopting an unsupervised approach be pre-training with deep belief nets (Cho & Saul, 2009). The kernel PCA features in MKMs was used as input for features in the next layer. In this case appropriate kernels and their parameters were selected and applied from one layer to another. However, while nonlinear transform can be utilized by kernel PCA in

MKMs, an RBF kernel that mimics the projections of a randomly initialized neural network, regarded an alternative approach, was used in this work.

Kernel PCA features are best selected using ranking method in which redundant features are discarded. The ranking method was used to prune away inappropriate or unwanted features at each layer in the MKM. Naturally, the method helped to focus on the kernel PCA for deep learning. Features were pruned in just one-step, and then the kernel PCA algorithm was applied to produce a result that could be used as an input for the next layer. An optional option to prune selected features was employed to further minimize feature redundancy. Steps 2 to 5 were then repeated according to the number of layers set by the N value in step 1. The result of the algorithm was then computed and the feature representations fed to the classifier to make the final decision.

7.4.2. Multilayer RBF kernel machine based on supervised kernel regression

This algorithm was extended from the first one by applying supervised kernel regression and removing the optional step of feature selection because it was done along with the projection. Yger et al (2011) assumed that this would give a better computation time. For the latent variable regression, the feature extraction will also be incorporated in the regression step but it would be based not on the output but on the input. The author claimed to overcome the drawback which was the step of feature selection, by learning each hidden layer using the Kernel Partial Least Squares regression (KPLS).

Below is a summary on how the Multilayer RBF machine was implemented based on supervised kernel regression and is shown in Figure 7.2.

1. Let N be the number of layers to use.
2. Select the appropriate kernels and kernel parameters (cross-validation or otherwise).
3. Apply supervised regression to extract the next feature value and corresponding Eigen value.

4. If the Eigen value is greater than selected threshold go to step 3 otherwise, use all the extracted features as input to the next layer.
5. If number of iterations exceeds N go to step 6, otherwise go to step 2.
6. Feed the feature representations to the classifier to make the final decision.

In this algorithm, feature selection methods and unsupervised dimensionality reduction was incorporated into the supervised regression algorithm. In MKMs, deep learning approach is achieved through repetitive iteration of supervised regression algorithm sequentially as listed above. The use of this however was not new in this context. What is new here is how the supervised regression with Eigen value was applied to extract the appropriate feature representative. Using the first algorithm, a major contribution was made by replacing kernel PCA with supervised regression. The selection of appropriate kernel and kernel parameters was retained, as they were already in place from the existing algorithm. Since supervised training only occurred in the last layer for MKMs, this made the feature selection method very important. From the first step, the N number of layers is determined for this process. As in the case of the first algorithm for kernel PCA, the ranking method was used again to prune features for appropriate selection. Supervised regression was then applied to the pruned features. In supervised regression, the use of LMKMs was used to inspire deep learning training architectures. This was a more important achievement of the supervised regression algorithm used to extract next feature value and corresponding Eigen value.

The Eigen value was computed and determined through the algorithm steps with any value greater than the threshold being discarded. While appropriate values are all extracted as an input for the next layer. Steps 2-4 was then repeated according to the number of layers set by the N value in step 1. Then, the feature representations are fed to implement the Multilayer RBF machine based on supervised kernel regression.

7.4.3. Multilayer RBF machine based on unsupervised kernel regression

This proposed algorithm based on unsupervised latent space and the motivation behind this claim is that unsupervised methods work well with the regular neural networks and

unsupervised learning focusing on important patterns from data, regardless of their labels, as it reduces the input dimensionality of data without losing crucial information. In this algorithm unsupervised methods were used, as described in a paper by (Memisevic, 2003), which were Kernel parameters selection and dimensionality selection as shown in Figure 7.3.

Letdata in q-dimensional space is represented by: $Y \in R^q$ (observable space), the representation of this data in the d-dimensional space q>d: $X \in R^d$ (latent space) is required. Let us also say there are N data points.

There are 2 types of unsupervised kernel regression for this purpose:

1. **Optimize an error in the observable space.**

$$E^{obs}(X) = \sum_{i=1}^{N} \left\| y_i - \frac{\sum_{j=1}^{N} K(x_i, x_j) y_j}{\sum_{k=1}^{N} K(x_i, x_k)} \right\|^2$$

The advantages of this method is that any kernel bandwidth h can be used because it is easily replaced by the scale of the X. The disadvantage is that there is a very computationally consuming optimisation problem with many parameters (N). Also, it is not an explicit representation of x in terms of y which will be needed to proceed with the classification of new points.

2. **Optimize an error in the latent space.**

$$E^{lat}(X) = \sum_{i=1}^{N} \left\| x_i - \frac{\sum_{j=1}^{N} K(y_i, y_j) x_j}{\sum_{k=1}^{N} K(y_i, y_k)} \right\|^2$$

- <u>Advantages:</u>

– There is an efficient way to solve this problem via eigen value decomposition.

– There is an explicit representation of x in terms of y: $x = \frac{\sum_{j=1}^{N} K(y,y_j) x_j}{\sum_{k=1}^{N} K(y,y_k)}$, the solution depends on the selected kernel bandwidth which can be explained in the following steps :

1. For train set Y, find the solution X which optimizes the error in the latent space.

2. For particular X and Y solve the optimization problem to find optimal X scaling S (X:=X*S) which optimizes the error in observable space.
3. Identify the optimal range for the h (kernel bandwidth) based on the graph connectivity algorithm.
4. Perform the algorithm of traversing through different values of h to identify the optimal value.
5. Select an appropriate number of parameters d.
6. Write the code to incorporate the method into the classification.
7. Run the tests.
8. Add the special constraints concerning the distances between different classes in the optimization problem to better fit the data for further classification. Adapt the optimization solution.
9. If results are unsatisfactory, try the optimization in the observable space:
    a. Determine X for train Y
    b. Find the model for finding X for new points Y.

The idea of unsupervised kernel dimension reduction has been applied focusing on both linear and nonlinear unsupervised kernel dimension reduction (Wang, et al., 2010). However, this current work has considered non-linear unsupervised kernel dimension reduction inspired by work by another researcher (Wang, et al., 2010).

A kernel based model (Cho & Saul, 2009) which was a Multilayer kernel machine (MKM), was adopted for the three algorithms used for experiments in this work. Particularly, MKM was introduced in the first algorithm to integrate unsupervised dimensionality reduction with supervised feature selection methods into a kernel PCA algorithm.

Below is a summary of how in this research the Multilayer RBF machine was implemented based on an unsupervised kernel regression as shown in Figure 7.2:

1. Let N be the number of layers to be used.
2. Apply unsupervised regression to extract latent variables which better represent the input parameters. (Kernel parameters selection and dimensionality selection is embedded in this step) based on the ideas described in the work of Memisevic (2003)

- Learning of optimal latent space representation with input data
- Learning of transformation from observable to latent space.
- Selection of the kernel parameters and optimal dimensionality of the latent space

3. Use extracted latent variables as input to the next step.
4. If number of iterations exceeds N go to step 6, otherwise go to step 2.
5. Feed the feature representations to the classifier to make the final decision.

This algorithm combined both a KPCA and a supervised regression algorithm. This was done to achieve a more reliable input and consequently, more reliable results. An unsupervised regression algorithm was embedded with supervised feature selection and unsupervised dimensionality reduction.

The idea of multilayer kernel machines (MKMS) implemented in this work was to help filter only feature relevant input, to be fed into a developed unsupervised regression algorithm, and to construct an infinite dimensional representation. Additionally, to help obtain a result, unsupervised dimensionality reduction was implemented with feature space.

The attempt to adopt this approach is a high-level implementation concept of MKMs through the use of 2 different machine learning techniques, which were used to develop another.

The implementation of the multilayer RBF machine based on kernel PCA was not new. However, in this algorithm, PCA was replaced by unsupervised kernel regression. The idea of unsupervised regression application was suggested by Memisevic (2003) which was used as the basis for this research which combined feature selection with unsupervised regression. Unlike supervised regression procedure, latent variables were extracted instead using the unsupervised regression method. In this work, these variables were extracted by applying the input to obtain even better input parameters. This is equally based on three key steps from Memsevic work. The N number of layers required was set. Because there were several layers, previous output of the extracted latent variables was used as input to next layer. This process continued until the N number was

reached. The feature representations were then fed to implement the multilayer RBF machine based on this algorithm for unsupervised kernel regression. The procedure for the unsupervised regression method is provided below in Figure 7.3.

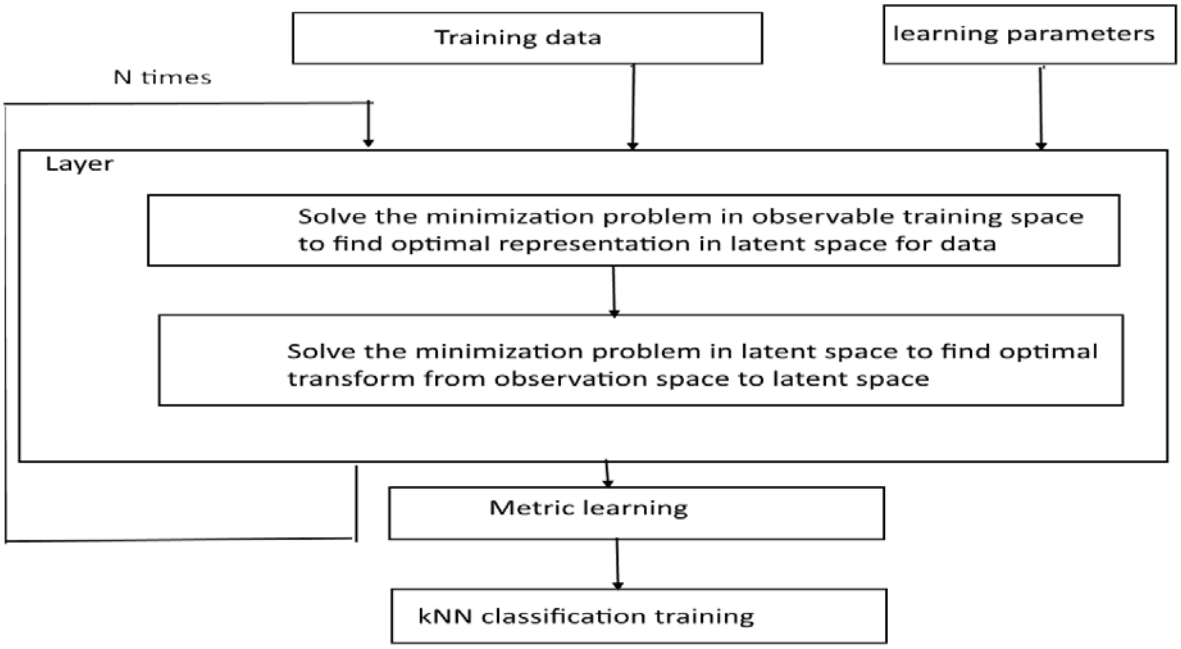


Figure 7.3: The procedure of unsupervised regression

After model training and evaluation of the three algorithms; the unsupervised method did not give a good accuracy as was expected. To improve performance, the unsupervised latent regression with projection method was used.

The classifier based on this method is built by the following steps:

1. The whole training dataset is subdivided into several groups based on the data class labels.
2. For each group, individually the following model is trained:
    a. $x = g(y) = \frac{\sum_{j=1}^{N} K(y,y_j) \cdot x_j}{\sum_{k=1}^{N} K(y,y_k)}$; $= f(x) = \frac{\sum_{j=1}^{N} K(x,x_j) \cdot y_j}{\sum_{k=1}^{N} K(x,x_k)}$, where $K(a,b)$ is a kernel function.

b. The kernel is selected from the class of the multilayer RBF kernels.

c. The 1-layer RBF kernel is: $K(x,y) = e^{-\frac{1}{2h^2}\|x-y\|^2}$, the combination of the RBF kernel K$(x,y)$with kernel $K_1(x,y)$ will be $K(x,y) = e^{1-\frac{1}{h^2}K_1(x,y)}$. The goal of the training is to define the values of the hyper parameters for each kernel.

d. The optimal values of the kernel are defined via eigenvalues decomposition problem as previously was done for the unsupervised latent regression. Then for these computed values the observable space error is defined as: $E = \sum_{i=1}^{N}\|y_i - f(g(y_i))\|$, the built in cross-validation is used, meaning that y_i and x_i are excluded from the prediction stage. $f(x_i) = \frac{\sum_{j=1,j\neq i}^{N} K(x_i,x_j)\cdot y_j}{\sum_{k=1,k\neq i}^{N} K(x_i,x_k)}$, $g(y_i) = \frac{\sum_{j=1,j\neq i}^{N} K(y_i,y_j)\cdot x_j}{\sum_{k=1,k\neq i}^{N} K(y_i,y_k)}$.

3. The points for which we want to predict the class label are processed via model $f(g(y))$ to find the projection error for each group. The class giving minimum projection error is selected as a class label.

7.5. Implementation

In this section, multilayer kernel machines will be compared with different approaches for feature selection and dimensionality reduction. That means steps 3 and 4 of the described method (supervised feature selection and unsupervised dimensionality reduction) will be changed and how these changes affect the performance of the classification will be considered.

The major points for consideration are:

1. Determine overall classification accuracy on a different dataset by comparing machines with the same number of layers and equivalent kernel parameters.
2. Compare change of the classification performance with increased number of layers to answer these questions:
    - How much the performance improves with addition of each new layer?

3. What are the number of layers, after which an increase in number doesn't improve the performance further. Compare how performance varies depending on the number of training examples.
4. Compare computational complexity.

To achieve these goals; the following requirements have been followed:

1. The same classification algorithm for all 3 was used which was the k-nearest neighbourhood (KNN) classification algorithm with distance metric.
2. K-fold cross-validation was performed: the input data was subdivided into k pieces, then on each iteration a single piece of data was used for validation while other data was used for training in all 3 algorithms. As a result, a precision and recall was computed for all methods and a number of selected features.
3. Different parameters of the method we changed to see how the results of the classification changed
   - The number of layers including such questions as: how much did the performance improve with addition of each new layer and if there is any limit for the number of layers also, after how many layers did the performance stops improving?
   - Number of training examples

7.6.Experimental Results

7.6.1. Datasets Description

In this part of research, three different datasets were used to evaluate the proposed approaches. Details of these datasets were mentioned in chapter 4.

1. CAD dataset from The King Abdullah Medical City hospital at Saudi Arabia
2. Arrhythmia dataset from UCI Repository
3. Apnea-ECG dataset from PhysioBank Databases

In order to achieve the robustness of the results 10-fold cross validation was used. All the points from the dataset selected for evaluation were:

1. Randomly permutated
2. Subdivided into 10 groups of equal size. The prediction algorithms were sequentially applied to each of these 10 groups while the others 9 were used to train the prediction model. The quality metrics were computed for each iteration and the averaged in order to achieve final results.

7.6.2. Results Discussion and Analysis

7.6.2.1 CAD dataset

For the hospital dataset, the results are presented by considering five hidden layers for determining accuracy, MSE (mean squared error), sensitivity, specificity, Cohen's Kappa, training rimes and validation times respectively as shown in the following tables.

First of all the kPCA algorithm is evaluated. The performances are evaluated between one to five hidden layers. The iteration or processing time increases with almost a factor of two with each additional hidden layer. This is understandable as the number of iterations increase exponentially and thus processing times. shows the worst time results in terms of training and validation phases. This was however a result of the large number of features selected at the projection step transferred to the feature selection algorithm and KNN-classifier. Also, Cohen's Kappa gave no value after layer 3, which suggested it was non-supportive of multiple layers beyond 3 layers.

As Table 7.1 shows below the behaviour. The accuracy, sensitivity and specificity almost remain consistent and is not impacted much by the increase of hidden layers, hence can easily be concluded that kPCA algorithm is not effected by the number of hidden layers. However, looking at the results it can be concluded that the best performance is obtained for no of hidden layers to be three. The obtained accuracy, sensitivity and specificity is maximum in the case of three hidden layers.

Table 7.1: Results Summary of kPCA algorithm for CAD dataset

Layer No / Metric	1	2	3	4	5
Accuracy,%	65.07	63.92	61.59	61.44	63.76
MSE,%	34.92	36.07	38.4	38.55	36.23
Sensitivity	78.57	77.56	67.75	71.63	77.34
Specificity	32	32	46.5	36.5	30.5
Cohen's kappa	0.39	0.39	0.44	0.4	0.38
Training time, s	245.62	506.62	828.29	1166.99	1119.56
Validation time,s	43.7	101.01	263.67	397.79	464.26

In Table 7.2 the conduct of the supervised regression based on deep RBF kernel machine has been presented. This is in terms of the mentioned matrices and can be seen that the accuracy of the classification increases with increased number of hidden layers. Another note worthy outcome is the training and validation times, which are far less than obtained by the KPCA method. However, the Accuracy and specificity is not as good as KPCA method however the Sensitivity is better in case of Supervised regression. In this case also the values obtained with three hidden layers are the best and based on number of iterations and processing time, this can be said to provide best overall results.

Table 7.2: Results Summary of Supervised Regression algorithm for CAD dataset

No of hidden layers / Metric	1	2	3	4	5
Accuracy,%	62.6	59.56	60	54.92	60.28
MSE,%	37.39	40.43	40	45.07	39.71
Sensitivity	72.44	70.81	70	63.46	70.81
Specificity	38.5	32	35.5	34	34.5

Cohen's kappa	0.41	0.37	0.39	0.36	0.39
Training time, s	610.37	1313.61	2098.83	3106.95	2733.81
Validation time, s	11.63	15.54	18.67	21.96	17.25

Table 7.3 provides how the unsupervised regression work based on deep RBF kernel machine using the hospital data set with extracted and normalised features. The major finding is that the accuracy is decreasing and MSE increasing with increasing number of layers. The sensitivity figures are also better for a single hidden layer, so does the specificity. The training and validation processing times are quite less as compared to other algorithms used.

Table 7.3: Results Summary of Unsupervised Latent Regression algorithm for CAD dataset

	1	2	3	4	5
Accuracy,%	48.98	49.56	48.55	52.6	44.05
MSE,%	51.01	50.43	51.44	47.39	55.94
Sensitivity	46.93	47.34	45.71	55.71	38.36
Specificity	54	55	55.5	45	58.0
Cohen's kappa	0.39	0.4	0.39	0.39	0.37
Training time, s	626.72	1307.40	2108.91	2508.64	2589.73
Validation time, s	31.62	36.42	45.38	43.18	39.41

As Table 7.4 shows below the behave Unsupervised Latent Regression with projection algorithm based on deep RBF kernel learning method. Importantly the accuracy, MSE sensitivity and all the performance criterions are consistent for the number of hidden layers. This clearly shows that this algorithm is not good for deep learning and the very basis of deep learning is not satisfied as number of hidden layers have no impact on the performance and outcome.

Table 7.4: Results Summary of Unsupervised Latent Regression with projection algorithm for CAD dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	61.30	61.30	61.30	61.30	61.30
MSE,%	38.69	38.69	38.69	38.69	38.69
Sensitivity	71.83	71.83	71.83	71.83	71.83
Specificity	35.5	35.5	35.5	35.5	35.5
Cohen's kappa	0.39	0.39	0.39	0.39	0.39
Training time, s	192.40	486.44	1318.25	3719.59	9958.89
Validation time, s	11.64	11.39	11.03	11.15	11.13

Figure 7.4 presents the accuracy values for the four algorithms according to variation in number of layers.

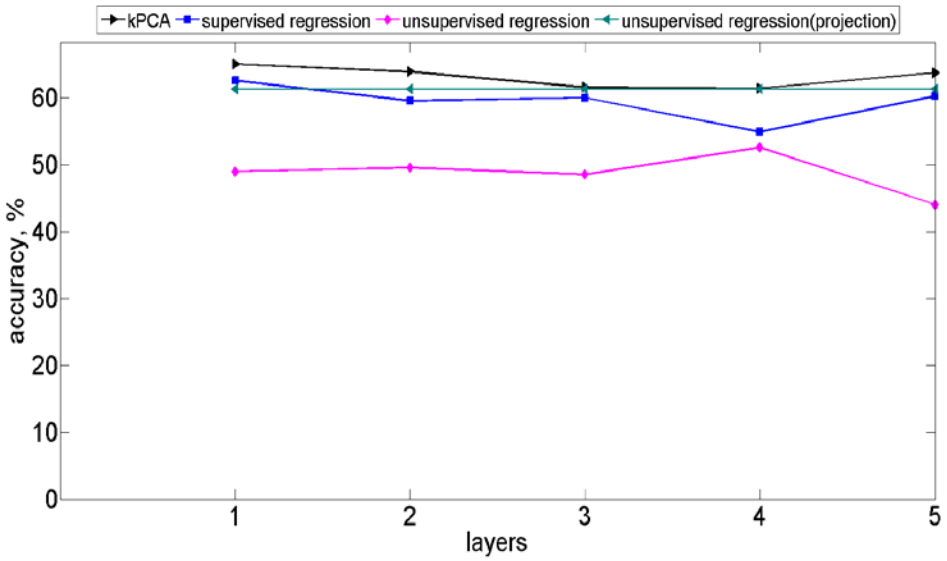


Figure 7.4: Accuracy vs. number of layers when applying the four algorithms for CAD dataset

The other figures of MSE, sensitivity, specificity, Cohen's Kappa, Training and validation time for CAD can be found in Appendix V.

Table 7.5 presents the confusion matrix for CAD dataset using KPCA algorithm for each layer and describe the performance on a set of actual data for prediction, i.e.the existence and nonexistence of CAD in a patient from the given data set.

Table 7.5: Summary of Confusion Matrix for kPCA algorithm for CAD dataset

Number of Layers	Confusion Matrix			
1		**C**	**N**	**∑**
	C	280	84	**364**
	N	209	115	**324**
	∑	**489**	**199**	**688**
2		**C**	**N**	**∑**
	C	261	80	**341**
	N	228	119	**347**
	∑	**489**	**199**	**688**
3		**C**	**N**	**∑**
	C	474	191	**665**
	N	15	8	**23**
	∑	**489**	**199**	**688**
4		**C**	**N**	**∑**
	C	489	197	**686**
	N	0	2	**2**
	∑	**489**	**199**	**688**
5		**C**	**N**	**∑**
	C	489	196	**685**
	N	0	3	**3**
	∑	**489**	**199**	**68**

The Summary of Confusion Matrix for Supervised Regression algorithm, Unsupervised Latent Regression and Unsupervised Latent Regression with projection algorithm for CAD dataset have been shown in Appendix V.

7.6.2.2 Arrhythmia dataset

For the Arrhythmia dataset, the results are presented by considering 5 different layers for determining accuracy, MSE, sensitivity, specificity, Cohen's Kappa, training rimes and validation times respectively as shown in the following tables. Unlike the hospital dataset, the kPCA algorithm showed much improved time results especially in the training phases. Likewise, the Cohen's Kappa values are all considerable stable when compared to the hospital dataset.

As Table 7.6 shows below the behaviour of KPCA based on deep RBF kernel machine across 5 layers with respect to known matrices which are accuracy, MSE, sensitivity, specificity, Cohen's Kappa, training rimes and validation times. The accuracy became a bit higher by increasing the number of layers as this means more complex information is been processed at each forward layer. However, at a stage even adding further layers and additional computations doesn't help to improve the classification performance. At this point the number of hidden layers to be used is not increased further and is taken as the optimal results obtained from the given network and algorithm.

Table 7.6: Results Summary of kPCA algorithm for Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	65.34	64.41	66.04	64.18	65.81
MSE,%	34.65	35.58	33.95	35.81	34.18
Sensitivity	78.77	78.77	78.36	81.22	82.44
Specificity	47.56	45.40	49.72	41.62	43.78
Cohen's kappa	0.5	0.48	0.51	0.46	0.48
Training time, s	161.85	290.41	439.78	617.78	716.48
Validation time,s	647.15	659.92	744.39	969.24	1014.18

In Table 7.7 the conduct of the supervised regression based on deep RBF kernel machine in term of the mentioned matrices and can be seen the accuracy getting low with greater number of layers and gave the best performance in layer 1 with 69.53 %. This means the supervised regression obtains good level of classification even with fewer hidden layers for this particular type of data set. Since the classification is a binary type hence some weighted and normalised features have more impact on the output and even adding further layers and features may not help to improve efficiency.

Table 7.7: Results Summary of Supervised Regression algorithm for Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	69.53	65.81	61.62	63.48	59.06
MSE,%	30.46	34.18	38.37	36.51	40.93
Sensitivity	73.87	70.20	64.08	66.12	63.26
Specificity	63.78	60	58.37	60	53.51
Cohen's kappa	0.59	0.55	0.52	0.53	0.48
Training time, s	447.63	700.93	1040.29	1348.98	1605.86
Validation time, s	6.58	7.27	9.33	9.43	10.18

Table 7.8 provides how the unsupervised regression work based on deep RBF kernel machine in term of the mentioned matrices and can be seen the accuracy decrease by increasing the number of layers and took less time in the training and validation.

Table 7.8: Results Summary of Unsupervised Latent Regression algorithm for Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	57.67	56.27	56.27	53.95	54.65
MSE,%	42.32	43.72	43.72	46.04	45.34
Sensitivity	100	90.61	90.61	71.02	80.40
Specificity	1.62	10.81	10.81	31.35	20.54
Cohen's kappa	0.02	0.18	0.18	0.35	0.28
Training time, s	329.044	635.73	968.47	1189.06	1487.87
Validation time, s	15.96	18.15	20.52	20.75	22.81

As Table 7.9 shows below the behaviour of Unsupervised Latent Regression with projection algorithm based on deep RBF kernel machine across 5 layers with respect to known matrices which are accuracy, MSE, sensitivity, specificity, Cohen's Kappa,

training rimes and validation times. All of the matrices (except training and validation time) keep the same level with the 5 layers.

Table 7.9: Results Summary of Unsupervised Latent Regression with projection algorithm for Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	68.0	68.0	68.0	68.0	68.0
MSE,%	32.0	32.0	32.0	32.0	32.0
Sensitivity	72.8	72.8	72.8	72.8	72.8
Specificity	61.6	61.6	61.6	61.6	61.6
Cohen's kappa	0.57	0.57	0.57	0.57	0.57
Training time, s	182	431	744	1200	1520
Validation time, s	15.4	11.1	74.9	54.7	47.4

Figure 7.5 below presents the accuracy values for the four algorithms according to the variation in the number of layers. The other figures of MSE, sensitivity, specificity, Cohen's Kappa, Training and validation time for Arrhythmia can be found in Appendix V

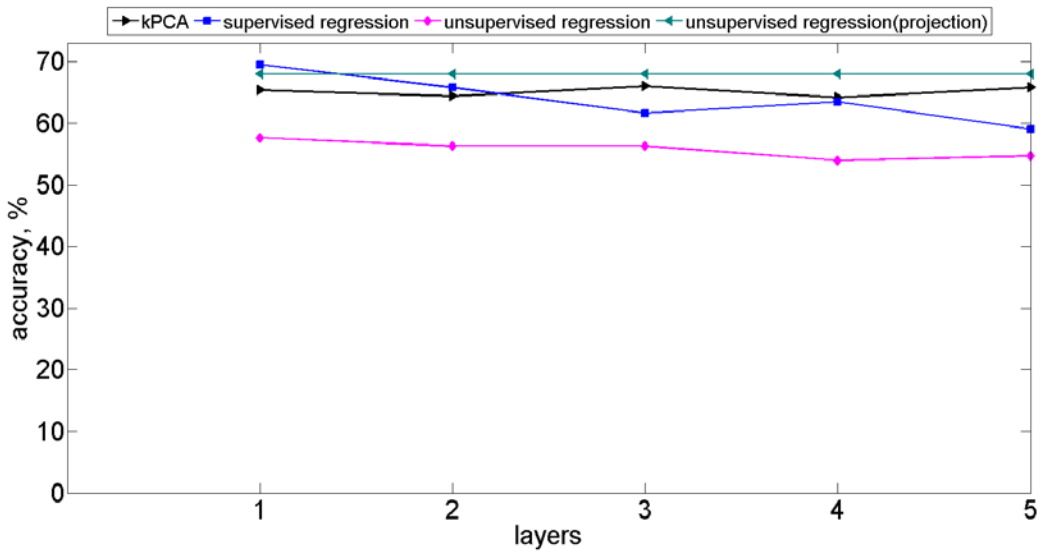


Figure 7.5: Accuracy vs. number of layers when applying the four algorithms for Arrhythmia dataset

Table 7.10 presents the confusion matrix for Arrhythmia dataset using Unsupervised Latent Regression with projection algorithm for each layer and describe the performance on a set of actual data for prediction the existence and nonexistence Arrhythmia.

Table 7.10: Summary of Confusion Matrix for Unsupervised Latent Regression with projection algorithm for Arrhythmia dataset

Number of Layers	Confusion Matrix			
1		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
2		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
3		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
4		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
5		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**

The Summary of Confusion Matrix for KPCA, Supervised Regression algorithm, Unsupervised Latent Regression algorithm for Arrhythmia dataset have been shown in Appendix V.

7.6.2.3 The Apnea-ECG data set

For the Apnea-ECG database, no significant change was observed in both specificity and sensitivity values for all 5 layers. The impressive time results shown for training and validation times means this appears to be the best algorithm for best result in this respect.

The improvement in the time parameters was achieved by limiting the number of features selected after projection. Unlike the previous two datasets, the Cohen kappa showed unaffected values through the 5 layers.

As Table 7.11 shows below the behave of KPCA based on deep RBF kernel machine across 5 layers with respect to known matrices which are accuracy, MSE, sensitivity, specificity, Cohen's Kappa, training rimes and validation times. The accuracy obtained is highest with number of hidden layers confined to two, even the specificity and sensitivity obtained is really good for 2 hidden layers but overall classification metrics are really good as compared to the CAD cases as was evaluated in the last sections.

Table 7.11: Results Summary of kPCA algorithm for Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	96.7	98.7	98.5	96.25	96.8
MSE,%	3.3	1.3	1.5	3.75	3.2
Sensitivity	98.97%	99.74%	99.43%	97.93%	99.19%
Specificity	95.23%	98.09%	97.86%	95.21%	95.25%
Cohen's kappa	0.95	0.95	0.94	0.94	0.95
Training time, s	70	234.5	368.2	422.8	702
Validation time,s	69.2	103.6	226.9	535.2	688.9

In Table 7.12 the conduct of the supervised regression based on deep RBF kernel machine in term of the mentioned matrices and can be seen the accuracy increase in layer 3.

Table 7.12: Results Summary of Supervised Regression algorithm for Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	93.4	93.6	94.5	92.7	93.5
MSE,%	6.6	6.4	5.5	7.3	6.5

Sensitivity	93.95%	94.11%	97.16%	93.63%	96.71%
Specificity	93.05%	93.24%	92.86%	92.11%	91.53%
Cohen's kappa	0.89	0.89	0.91	0.88	0.90
Training time, s	125.4	448.3	1076.7	893.6	1568.7
Validation time, s	36.8	30.3	47.9	41.45	61.2

Table 7.13 provides how the unsupervised regression work based on deep RBF kernel machine in term of the mentioned matrices and can be seen the accuracy highly decrease by increasing the number of layers.

Table 7.13: Results Summary of Unsupervised Latent Regression algorithm for Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	99.65	75.1	69.3	88.5	58.4
MSE,%	0.35	24.9	30.7	11.5	41.6
Sensitivity	99.74%	90.11%	90.90%	92.25%	74.78%
Specificity	99.62%	65.82%	55.81%	86.10%	48.18%
Cohen's kappa	0.99	0.84	0.12	0.84	0.76
Training time, s	299	413.7	858.3	784.1	1604
Validation time, s	108	92.2	125.8	96.7	146.06

As Table 7.14 shows below the behave Unsupervised Latent Regression with projection algorithm based on deep RBF kernel machine across 5 layers with respect to known matrices which are accuracy, MSE, sensitivity, specificity, Cohen's Kappa, training rimes and validation times. All the matrices (except training time) keep the same level with very slightly difference a cross the 5 layers.

Table 7.14: Results Summary of Unsupervised Latent Regression with Projection algorithm for Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	95.3	95.3	95.4	95.4	95.3
MSE,%	4.7	4.7	4.6	4.6	4.7
Sensitivity	100	100	99.9	99.8	100
Specificity	93.2	93.2	93.4	93.2	93.2
Cohen's kappa	0.92	0.92	0.92	0.92	0.92
Training time, s	31.8	83.9	202.9	860.6	2262.3
Validation time, s	13.3	11.9	10.8	13.36	11.05

Figure 7.6 below presents the accuracy values for the four algorithms according to the variation in the number of layers.

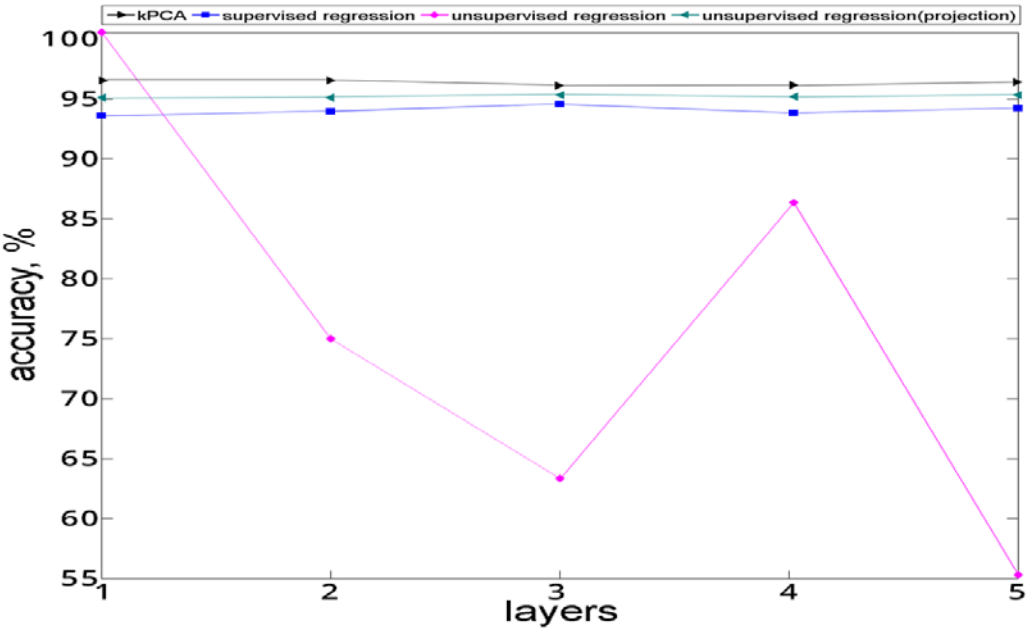


Figure 7.6: Accuracy vs. number of layers when applying the four algorithms for Apnea dataset

The other figures of MSE, sensitivity, specificity, Cohen's Kappa, Training and validation time for Apnea can be found in Appendix V.

Table 7.15 presents the confusion matrix for Apnea dataset using Supervised Latent Regression algorithm for each layer and describe the performance on a set of actual data for prediction the Apnea and non Apnea minute.

Table 7.15: Summary of Confusion Matrix for Supervised Regression algorithm for Apnea dataset

Number of Layers	Confusion Matrix			
1		**A**	**N**	**∑**
	A	6120	729	**6849**
	N	394	9760	**10154**
	∑	**6514**	**10489**	**17003**
2		**A**	**N**	**∑**
	A	6130	709	**6839**
	N	384	9780	**10164**
	∑	**6514**	**10489**	**17003**
3		**A**	**N**	**∑**
	A	6329	749	**7078**
	N	185	9740	**9925**
	∑	**651**	**10489**	**17003**
4		**A**	**N**	**∑**
	A	6099	828	**927**
	N	415	9661	**1076**
	∑	**6514**	**10489**	**17003**
5		**A**	**N**	**∑**
	A	6300	888	**7188**
	N	214	9601	**9815**
	∑	**6514**	**10489**	**17003**

The Summary of Confusion Matrix for KPCA, Unsupervised Regression algorithm, and Unsupervised Latent Regression with projection algorithm for Apnea dataset have been shown in Appendix V.

7.6.2.4 Results Analysis

Comparing the Arrthymia dataset results for both supervised and unsupervised regression, a much more reduced accuracy percentage was experienced in supervised regression ,and betetr sensitivity values for unsupervised learning is obtained as compared to supervised regression. for unsupervised regression.

In Apnea-ECG database supervised regression showed a relatively stable accuracy percentage and improved validation times when compared to the unsupervised regression.

The most stable results were shown by the supervised regression method for both balanced and unbalanced datasets. This was the only algorithm which showed continuous improvement in sensitivity and specificity with the increasing number of layers. Other algorithms showed the tendency to stabilize results very quickly. That meant that in practice it made sense to test these algorithms only on small number of layers to understand whether they were sufficient for the current dataset or not. The worst time results were shown for the KPCA algorithms for both validation and training phases. This was caused by the large number of features selected at the projection step which were then transferred to the feature selection algorithm and KNN-classifier. The time parameters could be improved by the limit on the number of features selected after projection, although, that might affect the accuracy in some cases. As expected all the methods showed better results with an increasing amount of training data. Nonetheless, the biggest improvement in the time taken was shown by supervised regression. Unsupervised regression showed better results with a certain level of accuracy.

7.7. Discussion

In this chapter; a novel approach using Multilayer Radial Basis Function (RBF) Kernel Machine based on deep learning was proposed. This work was inspired by the multilayer kernel machines (MKM) proposed by Cho & Saul (2009). The proposed work in this chapter explored training the MKM with RBF kernels instead of the arc-cosine kernels used by Cho & Saul. A major drawback of the Cho & Saul model was that the hidden layers are greedily trained by first applying a kernel PCA projection followed by a biased features selection based on mutual information with class labels and supervised cross-validation. This inefficient and complicated criterion tends to increase the time wasted in the training and validation phases. The main contribution of this work was to overcome this drawback by training deep RBF kernel-based architectures for three different

transformations; Multilayer machine based on Kernel PCA (inspired from Cho & Saul (2009)), Supervised Regressions (inspired from Yger et al (2011)) and Unsupervised Regressions (inspired from Memsevic (2003)).

This work therefore provided three different approaches for features selection and dimensionality reduction to train the RBF kernel based on the idea of MLK. These approaches were all implemented to make a good environment for comparison, since the documentation of the previously mentioned work (Cho & Saul, 2009) and (Yger, et al., 2011) did not provide sufficient details about performance measures and approach evaluation. Therefore, it was necessary to implement the proposed approaches despite the drawbacks and try to alleviate them by exploring the network parameters that could affect the performance. A comparison was performed in terms of several matrices, the effect of changing the number of layers on the performance and computational complexity.

The proposed approach was evaluated based on the three datasets, Arrhythmia, CAD and Apnea-ECG, to investigate the behaviour of MLK when utilised on health data, such that a system could be developed using a novel machine learning approach which had not used before with heart disease data.

In Cho & Saul (2009) approach; the feature selection criteria was inefficient. This shortcoming was solved by using as a feature selection strategy. When compared with the KPCA method proposed by Cho & Saul, the unsupervised kernel regression with projection showed good performance in terms of accuracy, sensitivity and specificity. In addition, reduction in wasted time was remarkable. This was the aim and the key role in the health applications.

Another drawback of the Cho & Saul's approach was that the greedy choice of hyper-parameters (e.g., number of hidden layers) resulted in over fitting of the training set. To obtain better results there is a need for this to be regularised as the performance evaluation may be misleading and give too many true positives to obtain better sensitivity as the criterion due to overfitting is too soft. This was not consistent with the results

obtained from the evaluation of the proposed RBF kernel approach with its three transformations. It was deduced from the results, that the number of layers affected the performance per the nature of training data and the feature selection algorithm. The behaviour of the approach to use number of hidden layers till it does not overfit the training data, verify the approach taken by the author was reasonable and comparable to the performance criterion of Cho & Saul's approach (Cho & Saul, 2009).

The last thing to highlight was the effect of the data set size on the performance measure. As previously mentioned; the proposed approach was applied to three data sets which were of different sizes, where CAD and Arrhythmia data sets tend to be small data sets, the Apnea-ECG data set was a large data set. The results showed that the proposed approach on the Apnea-ECG data set achieved a higher performance than on the CAD and Arrhythmia data sets. Therefore, it can be concluded that the deep MLK performs better on large data sets. This finding is inconsistent with the results obtained by some previous researchers (Strobl & Visweswaran, 2013). They argued that deep MLK methods performed well on a limited samples size. This was not consistent with the idea of deep MLK methods that have mostly been applied to huge datasets. However, the improvement proposed in this work has actually nullified the previous concept and has provided for the basis that if the data set is pre-processed and missing information is some how retrieved, the Multilayer kernel can perform well and obtain really good results as in the case of Apnea-ECG data set, which is not only large but had no missing and incorrect information. However, normalisation of the data is required to obtain the optimal results.

7.8. Summary

In conclusion, the multilayered systems showed improvement in results as shown clearly in the tables and have achieved a high accuracy rates for unsupervised and supervised learning, even with large and highly varied data set, as compared with "shallow learning" algorithms with single layer architecture, as shown in chapter 6 and other results discussed in the literature. Moreover, amongst the multilayer algorithms, supervised

regression tends to produce more stable and accurate results. Naturally supervised learning provides the labelling parameters for feature extraction and matching and provides better results. In the case of binary classification as done in this work it makes it simpler for the ML method as labels are simple between existence of a disease or absence. As such, the usage of this supervised regression algorithm should be given greater consideration when conducting machine learning tasks involving large data sets and non-linear functions.

Chapter 8 : Conclusion and Future Work

8.1 Overview

The main purpose of this current research is to participate in the global efforts to enhance the quality of healthcare services, including proposing technology as one of solutions for the early and automated detection of fatal diseases using Machine learning and effective use of Data sciences. Saudi Arabia is one of the developing countries that has committed to utilising technology to help solve the growing demand on proactive and proven healthcare services. Providing decent healthcare services through technology globally and in Saudi Arabia in particular will deliver a huge repository of patient information. This will create a new challenge that needs more investigations in order to achieve the desired goals of improved healthcare quality and access. Usage of advanced data analytic procedures will be at the forefront of activities to address the aforementioned issue.

As part of the efforts to apply advanced data analytics in addressing healthcare issues, the findings of this study revealed that the proposed models have high classification performance in detecting common heart disease such as CAD, Arrhythmia and sleep-apnea and have either comparable or have exceeded the performances of existing classification methods. RBF networks have been used to predict CAD with back propagation and data set training techniques such as extended Kalman filters to improve the confidence of detecting accurately the existence and absence of the disease in patients.

Furthermore, Deep learning techniques have been enhanced such as Multilayer Kernel and supervised and unsupervised regression, to bring in better results for the disease classification problem. The balance between selection of the right number of hidden layers have been found to be a key to not only getting better performance but also defying the fact that such large data set does not bear good results for techniques such as supervised regression.

In addition, this research attests to the fact that data and information is valuable and can be used to effectively discover new trends, learn better methods to treat health

complications, and find useful patterns that solve many healthcare problems in relation to cardiac health. This chapter discusses the findings of this research and recommendations for future works, on how the learnings form this work can be used and likely enhancements are proposed.

8.2 Review of the Work and Contributions

In this study, the aim was to design a health informatics system for heart disease detection using data mining techniques, from obtaining and pre-processing datasets that can enhance the reliability of heart disease diagnosis using echocardiography. The achievement of this research can be summarised as follows:

- The development of an automatic Coronary Artery Disease prediction using a Combination of Methods for Training Radial Basis Function Networks. The study assessed the value and speed of training Radial Basis Function Networks (RBFN) instead of Multilayer Perceptron Networks (MLPN) to improve treating patients with Coronary Artery Disease (CAD). Additionally, it was assumed this would more effectively identify those at risk of CAD for early treatment. Exact Matrix Completion via Convex Optimization method was selected to improve the data quality and availability, based on its success;

- The development of an automatic Heart Disease detection system using deep Neural Networks. The proposed approach included two phases; in the first phase; a baseline deep neural network (B-DNN) model was designed. This model was mainly used for the stage of minute-based classification of apnea dataset in addition to CAD and Arrhythmia dataset. While in the second phase; a novel hybrid model was designed by the combination of the baseline deep neural network and the decision tree models to be used for the stage of minute-class-based classification of apnea dataset.

- The development of an automatic Heart Disease detection system using Multilayer Radial Basis Function Kernel Machine based on deep learning approach. Multilayer Radial Basis Function Kernel Machine approach combined the first RBFN proposed system and the second-deep learning concept in order to develop an automatic heart disease detection system using Deep Radial Basis Function Kernel Machine. Moreover, different approaches of features selection and dimensionality reduction to train the RBF kernel based on the idea of Multilayer Kernel Machine were developed. Furthermore, this thesis explored how to train deep kernel-based architectures by combination of supervised and unsupervised latent regression methods and a comparison was made in terms of accuracy, performance and computational complexity.

Different experiments were conducted and classification performance has been compared to determine optimal statistical algorithms for predicting different types of heart disease (CAD, Arrhythmia and Sleep Apnea) using different approaches. Based on the objectives previously presented, the following represent the undertakings and the main contributions of this research:

- Dealing with the important features in the health databases and developments of novel techniques of statistical processing for the heart disease data

The use and processing of a new CAD dataset is another contribution to the research. There was no such data set available before for the evaluation of CAD with such huge list of attributes (approximately 50 attributes). The data was collected from a famous hospital in Saudi Arabia and was analysed and processed in order to effectively and efficiently adapt for the classification process. In particular, the data from Saudi Arabia was a major part of this research work. The compiled dataset could also be useful for other researchers in promoting research on the area of heart disease, particularly in the field of CAD prediction.

Missing features values are a concern when developing datasets from real data. Therefore, an approach for constructing missing features values based on Matrix Completion has been proposed. The proposed approach showed improvement of classification accuracy on the constructed dataset than the original dataset.
Another important issue when dealing the health databases are features selections techniques and how to determine the most important features that lead to more accurate diagnosis. Thus, this research sought to identify the optimal feature subsets from the complete feature list, which was set to minimize the size of the features vector while still being able to classify the data with high accuracy.

Therefore, the features, that have the strongest effect on prediction, were selected. The normalization and adaptation stage scored the attributes according to their correlation with the classified class. To determine the discriminative power of each feature, the features importance function was adopted. Features importance is a function that ranks features according to their significance in predicting the target variable of the classification process. Features with higher values contribute the most in the prediction, while features with values near to Zero do not have high implication on the prediction results.

- Investigating the effect of hybridization of algorithms on system performance

Based on the experiments on different machine learning algorithms and with the use of different heart diseases datasets, the study concludes that hybridization of the existing machine learning algorithms have produced better approaches for clinic support system. When combining different Radial basis functions and other deep learning algorithms to get a hybrid scheme, there was a significant increase in the performance. The current idea is concentrated on combining the advantages of different algorithms in one approach.

To conclude the above, since the health data is not easily available and in some cases, non-accessible because of confidentiality issues. Manual recording or misinterpretation also causes variability in captured data. This is quite a challenging issue while dealing

with health informatics systems, the proposed approaches in this study can deal with different nature of health datasets with varying feature sets and get adapted with data availability.

This study showed that data mining techniques can be used efficiently to model and predict heart disease cases. The outcome of this study can be used as an assistant tool by cardiologists to help them to make more consistent diagnosis of heart disease.

8.3 Future Work

All heart diseases datasets used in this study have binary outcomes. Clinical practice however, is often more complex and outcomes may be in different format. It is envisaged that future work can contribute to the knowledge base and improve the accuracy and reliability of established systems by broadening the databases and expanding the criteria for measuring the performance of established systems.

For the future work, the goal can be to implement the proposed systems to be a single comprehensive system that can help the clinical support decision in the best possible way for prediction of cardiovascular diseases. A single package which can deal with different data set and provide instance comparison between different ML algorithms performance will be very useful. Although the obtained results have shown lot of promise and improvement from previous work, still there is room for further improvement.

First of all, it was detected that classifier performs in a significantly varying way on different datasets. This suggests that it is possible that ensemble of the classifiers trained on different part of the dataset might have a greater performance. Based on this, the following are important suggestions for further investigation of the method:

- Analyse the performance improvements with additional pre-processing steps.
- Apply ensemble learning techniques with same classifier type (RBF).

- Apply ensemble learning technique for different classifiers types.
- Integration of proposed systems with eHealth system

- To contact and coordinate with the King Abdullah Medical City Authority to discuss the integration of current proposed approaches with their eHealth system. This will provide a platform to test the performance on a real-world problem and further performance enhancement measures can be taken based on the outcomes of the testing.
- Finally, the control of the quality of obtained data using Machine learning and statistical approaches is also imperative, and Proper formatting of patients Echocardiogram can improve the retrieval of information in a fast and effective manner. These factors could be taken as important aspects of future work as well.

REFERENCES

American Heart Association, 2015. *Heart Disease and Stroke Statistics—2015 Update.* [Online] Available at: http://circ.ahajournals.org/content/circulationaha/131/4/e29.full.pdf [Accessed October 2016].

Australian Bureau of Statistics, 2010. *Causes of Death.* [Online] Available at: http://www.ausstats.abs.gov.au/Ausstats/subscriber.nsf/0/E8510D1C8DC1AE1CCA2576F600139288/$File/33030_2008.pdf [Accessed October 2016].

Abushariah, M., Alqudah, A., Adwan, O. & Yousef, R., 2014. Automatic Heart Disease Diagnosis System Based on Artificial Neural Network (ANN) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) Approaches. *Journal of Software Engineering and Applications,* Volume 7, pp. 1055-1064.

Alizadehsani, R. et al., 2012. *Diagnosis of Coronary Artery Disease Using Cost-Sensitive Algorithms.* Brussels, Belgium, IEEE.

Almazaydeh, L., Elleithy, K. & Faezipour, M., 2012. *Obstructive Sleep Apnea Detection*.

American Academy of Sleep Medicine, 2016. *Sleep Apnea - Symptoms & Risk Factors.* [Online] Available at: http://www.sleepeducation.org/essentials-in-sleep/sleep-apnea/symptoms-risk-factors [Accessed 16 October 2016].

American Heart Association, 2016. *Coronary Artery Disease - Coronary Heart Disease*. [Online] Available at: http://www.heart.org/HEARTORG/Conditions/More/MyHeartandStrokeNews/Coronary-Artery-Disease---Coronary-Heart-Disease_UCM_436416_Article.jsp#.WANbfeV97IV [Accessed 16 October 2016].

Amin, S. U., Agarwal, K. & Beg, D. R., 2013. Data Mining in Clinical Decision Support Systems for Diagnosis, Prediction and Treatment of Heart Disease. *International Journal of Advanced Research in Computer Engineering & Technology,* 2(1), pp. 218-223.

ANOVA, 2013. *ANOVA (Analysis of Variance).* [Online] Available at: http://www.statisticssolutions.com/manova-analysis-anova/ [Accessed 24 August 2016].

Ansari, A. Q. & Gupta, N. K., 2011. Automated Diagnosis of Coronary Heart Disease Using Neuro-Fuzzy Integrated System. *2011 World Congress on Information and Communication Technologies*, pp. 1379 - 1384.

Apoor Gami, M. & Neil Sanghvi, M., 2013. *Journal of the American College of Cardiology.*

Arif, M., Malagore, I. & Afsar, F., 2010. Detection and Localization of Myocardial Infarction using K-nearest Neighbor Classifier. *Journal of Medical Systems,* Volume 36, pp. 279-289.

Atkov, O. Y. et al., 2012. Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters. *Japanese College of Cardiology Journal of Cardiology*, 59(2), p. 190–194.

Babaeizadeh, S., White, D., Pittman, S. & Zhou, S., 2010. Automatic detection and quantification of sleep apnea using heart rate variability. *Journal of Electrocardiology,* Volume 43, p. 535–541.

Babaoglu, I., Baykan, O. K., Aygül, N. & Bayrak, M., 2009. Assessment of exercise stress testing with artificial neural network in determining coronary artery disease and predicting lesion localization. *Journal of Expert System With Applications,* 36(2), pp. 2562-2566.

Babaoglu, I., Findik, O. & Bayrak, M., 2010. Effects of principle component analysis on assessment of coronary artery diseases using support vector machine. *Expert Systems with Applications,* 37(3), pp. 2182-2185.

Bach, F. R., Lanckriet, G. R. G. & Jordan, M. I., 2004. *Multi-kernel learning, conic duality and the SMO algorithm.* Alberta, Canada.

Bai, J., Wu, Y., Zhang, J. & Chen, F., 2015. Subset based deep learning for RGB-D object recognition. *Neurocomputing,* Volume 165, p. 280–292.

BBC, 2013. *What causes coronary heart disease?*. [Online] Available at: http://www.bbc.co.uk/science/0/21686950 [Accessed 17 October 2016].

Becker, S. R., Cand`es, E. J. & Grant, M., 2011. Templates for Convex Cone Problems with Applications to Sparse Signal Recovery. *Mathematical Programming Computation,* 3(3), pp. 165-218.

Bengio, Y., 2009. Learning deep architecture for AI. *Foundations and Trends in Machine Learning,* 2(1), pp. 1-127.

Bengio, Y., Courville, A. & Vincent, P., 2013. Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 35(8), pp. 1798 - 1828.

Bengio, Y., Delalleau, O. & Le Roux, N., 2005. *The curse of highly variable functions for local kernel machines.* British Columbia, Canada, NIPS'05 Proceedings of the 18th International Conference on Neural Information Processing Systems.

Bengio, Y. & LeCun, Y., 2007. Scaling learning algorithms towards AI. In: O. C. D. D. J. W. L. Bottou:MIT Press.

Billing, A. & Zheng, G., 1995. Radial basis function network configuration using genetic algorithms. *Neural Networks,* 8(6), p. 877–890.

Birgmeier, M., 1995. *A fully Kalman-trained radial basis function network for nonlinear speech modeling.* s.l., IEEE International Conference.

Bitzer, S. & Kiebel, S. J., 2012. *Recognizing recurrent neural networks (rRNN): Bayesian inference for recurrent neural networks.* s.l., s.n., pp. 201-217.

Bo, L., Lai, K., Ren, X. & Fox, D., 2011. Object recognition with hierarchical kernel descriptors. *CVPR '11 Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition,* pp. 1729-1736.

Bo, L., Ren, X. & Fox, D., 2010. Kernel descriptors for visual recognition. *NIPS'10 Proceedings of the 23rd International Conference on Neural Information Processing Systems,* pp. 244-252.

Bouvrie, J., Rosasco, L. & Poggio, T., 2009. On invariance in hierarchical models. *22nd International Conference on Neural Information Processing Systems,* pp. 162-170.

Brebisson, A. D. & Montana, G., 2015. Deep neural networks for anatomical brain segmentation. *Computing Research Repository (CoRR),* Volume abs/1502.02445.

British Heart Foundation, 2015. *Risk factors.* [Online] Available at: https://www.bhf.org.uk/heart-health/risk-factors [Accessed 6 March 2016].

Broomhead, D. & Lowe, D., 1998. *Multivariable functional interpolation and adaptive networks*.

Cai, J.-F., Candès, E. J. & Shen, Z., 2010. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization ,* 20(4), pp. 1956-1982.

Cambridge University Press, 2009. *Linear versus nonlinear classifiers.* [Online] Available at: http://nlp.stanford.edu/IR-book/html/htmledition/linear-versus-nonlinear-classifiers-1.html [Accessed 11 January 2017].

Caples, S. M., 2007. Sleep-disordered breathing and cardiovascular risk. *Sleep,* 30(3), pp. 291-303.

Carnegie Mellon University , 2009. *Linear Classifiers and the Perceptron Algorithm.* [Online] Available at: http://www.stat.cmu.edu/~cshalizi/350/lectures/25/lecture-25.pdf [Accessed 11 January 2017].

Castro, B., Kogan, D. & Geva, A. B., 2000. ECG feature extraction using optimal mother wavelet. *The 21st IEEE Convention of the Electrical and Electronic in Isreal,* pp. 346-350.

Chatzis, S. P., Korkinof, D. & and Demiris, Y., 2011. *The One-Hidden Layer Non-parametic Bayesian Kernel Machine.* s.l., IEEE, pp. 825 -831.

Chazal, F. d. & Reilly, R. B., 2006. A patient adapting heart beat classifier using ECG morphology and heartbeat interval features. *IEEE Trans. Biomed. Eng.,* 53(12), p. 2535–2543.

Chazal, P., Penzel, T. & Heneghan, C., 2004. Automated detection of obstructive sleep apnoea at different time scales using the electrocardiogram. *Physiological Measurement,* Volume 25, pp. 967-983.

Chazal, P., Penzel, T. & Heneghan, C., 2004. Automated Detection of Obstructive Sleep Apnoeaa at Different Time Scales Using the Electrocardiogram. *Institute of Physics Publishing,* August, 25(4), pp. 967-983.

Chaza, P. d., O'Dwyer, M. & Reilly, R. B., 2004. Automatic classification of heartbeats using ECG morphology and heartbeat interval features. *IEEE Transactions on Biomedical Engineering,* 51(7), p. 1196–1206.

Chen, S., Grant, P. & Cowan, C. F. N., 1991. *Orthogonal least-squares algorithm for training multi-output radial basis function networks.* Bournemouth, IET, pp. 336 - 339.

Cho, Y. & Saul, L., 2009. Kernel methods for deep learning. *Advances in Neural Information Processing Systems,* Volume 22, p. 342–350.

Cho, Y. & Saul, L., 2010. Large margin classification in infinite neural networks. *Neural Computing,* 22(10), pp. 2678-2697.

Ciocoiu, I. B., 2002. RBF Networks Training Using a Dual Extended Kalman Filter. *Neurocomputing,* 48(1-4), pp. 609-622.

Clifton, D., Gibbons, J., Davies, J. & Tarassenko, L., 2012. *Machine Learning and Software Engineering in Health Informatics.* Zurich, s.n., p. 37 – 41.

Comak, E. & Arslan, A., 2010. A Biomedical Decision Support System Using LS-SVM Classifier with an Efficient and New Parameter Regularization Procedure for Diagnosis of Heart Valve Diseases. *Journal of Medical Systems,* Volume 36, p. 549 – 556.

Corbett, F., Michel, O. & Herrmann, A., 1943. *History of the Perceptron.* [Online] Available at: http://web.csulb.edu/~cwallis/artificialn/History.htm [Accessed 26 October 2016].

Cornell University, 2003. *Performance Measures for Machine Learning.* [Online] Available at: https://www.cs.cornell.edu/courses/cs578/2003fa/performance_measures.pdf [Accessed 20 January 2017].

Cortes, C. & Vapnik, V., 1995. Support-vector networks. *Machine Learning,* 20(3), pp. 273-297.

Creswell, J. W., 2012. Conducting and Evaluating Quantitative and Qualitative Research. *Educational Research Planning.*

Cui, Z., Yang, C. & Sanyal, S., 2012. Training artificial neural networks using APPM. *International Journal of wireless and mobile computing,* 5(2), pp. 168-174.

Dalrymple, P., 2011. Data, Information, Knowledge: The Emerging Field of Health Informatics. *Bulletin of the American Society for Information Science and Technology,* 37(5), pp. 41-44.

Dalrymple, P. W., 2011. Data, Information, Knowledge:The Emerging Field of Health Informatics. *Bulletin of the American Society for Information Science and Technology*, 37(5), pp. 41-44.

Damtew, A., 2011. *Designing A Predictive Model For Heart Disease Detection Using Data Mining Techniques,* Addis Ababa: School of Graduate Studies of Addis Ababa University.

Das, R., Turkoglu, I. & al, e., 2009. Effective diagnosis of heart disease through neural networks ensembles. *Journal of expert system with applications,* Volume 93, p. 7675–7680.

Davis, W. & Thu, T. N. T., 2006. *Predicting Cardiovascular Risks Using Possum-Ppossum And Neural Net Techniques.* s.l., Proceedings of the Eighth International Conference on Enterprise Information Systems.

De Castro L. N., V. Z. F. J., 2001. *An immunological approach to initialize centers of radial basis function neural networks.* Brazil, s.n., pp. 79-84.

De Castro, L. N. & Von Zuben, F. J., 2001. *An immunological approach to initialize centers of radial basis function neural networks.* Brazil, s.n., pp. 79-84.

DE CHAZAL, P. et al., 2000. Automatic classification of sleep apnea epochs using the electrocardiogram. *Computers in Cardiology,* Volume 27, p. 745–748.

Deng, J., Zhang, Z., Eyben, F. & Schuller, B., 2014. Autoencoder-based unsupervised domain adaptation for speech emotion recognition. *IEEE Signal Process. Lett,* Volume 21, p. 1068–1072.

Derrer, D., 2014. *Sleep Apnea.* [Online] Available at: http://www.webmd.com/ [Accessed 14 August 2016].

DIELEMAN, S., 2015. *CLASSIFYING PLANKTON WITH DEEP NEURAL NETWORKS.* [Online] Available at: http://benanne.github.io/2015/03/17/plankton.html [Accessed 5 Januray 2017].

Din, S. & Rabbi, F., 2006. Statistical Analysis of Risk Factors for Cardiovascular disease in Malakand Division. *Pakistan Journal of Statistics and Operation Research,* Januray.2(1).

Dong, B. & Wang, X., 2016. *Comparison Deep Learning Method to Traditional Methods Using for Network Intrusion Detection.* s.l., s.n., p. 581 – 585.

Dong, C., 2005. MATLAB Neural Network and Its Applications. In: *National Defense Industry Press.* Beijing: s.n., p. 121.

Dong, Z., Hao, Y. & Song, R., 2011. Injection Material Selection Method based on Optimizing Neural Network. *Advances in Intelligent and Soft Computing,* Volume 104, pp. 339-344.

Dorffner, G., 1994. A unified framework for MLPs and RBFNs: introducing conic section functions networks. *Cybernetics and Systems,* Volume 4.

Duch, W. & Jankowski, N., n.d. *Transfer functions: hidden probabilities for better neural networks.* [Online] Available at: http://www.fizyka.umk.pl/publications/kmk/01Esann-intro.pdf [Accessed 11 January 2017].

Duna, i. A., Mucsi, I., Juhász, J. & Novák, M., 2006. Obstructive sleep apnea and cardiovascular disease. *Orv Hetil.,* 147(48), pp. 2303-2311.

Efrati, A., 2013. *How 'Deep Learning' Works at Apple, Beyond.* [Online] Available at: https://www.theinformation.com/How-Deep-Learning-Works-at-Apple-Beyond [Accessed 26 October 2016].

Erhan, D., Courville, A., Bengio, Y. & Vincent, P., 2010. Why does unsupervised pre-training help deep learning?. *Journal of Machine Learning Research,* Volume 11, p. 625−660.

Eric Cohen, M., 2014. *Can Sleep Apnea Predict a Heart Attack?.* [Online] Available at: http://www.everydayhealth.com/columns/eric-cohen-breathe-well-sleep-well/can-sleep-apnea-predict-a-heart-attack/ [Accessed 20 January 2017].

ESCAP, 2010. *Statistical Yearbook for Asia and the Pacific 2009.* [Online] Available at: http://www.unescap.org/stat/data/syb2009/9.Health-risks-causes-of-death.asp [Accessed October 2016].

Foundation, B. H., 2015. *Risk Factors of Coronary Heart Disease.* [Online] Available at: https://www.bhf.org.uk/heart-health/risk-factors [Accessed 17 October 2016].

Fukushima, K. & Miyake, S., 1982. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition,* 15(6), pp. 455-469.

G.E., H., S., O. & Y.-W, T., 2006. Teh Y.-W. A fast learning algorithm for deep-belief nets. *Neural Computation,* 18(27), pp. 1527 - 1554.

Gale Nutrition Encyclopedia, 2011. *Heart Disease.* [Online] Available at: http://www.answers.com/topic/ischaemic-heart-disease [Accessed January 2017].

Genders, T. S. S., Steyerberg, E. W., Hunink, M. & Laule, M., 2012. Prediction model to estimate presence of coronary artery disease: retrospective pooled analysis of existing cohorts. *British Medical Journal,* Volume 344, pp. 1-13.

Genders, T., Steyerberg, E., Hunink, M. & Nieman, K., 2012. Prediction model to estimate presence of coronary artery disease: retrospective pooled analysis of existing cohorts. *British Medical Journal,* Volume 344, pp. 1-13.

Gillies, A. & Sterratt, D., 2012 . *Neuron Tutotial.* [Online] Available at: http://www.anc.ed.ac.uk/school/neuron/ [Accessed 26 October 2016].

Goldberger, A. et al., 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation,* 101(23), p. e215–e220.

Gomm, J. & Yu, D., 2000. Selecting radial basis function network centers with recursive orthogonal least-squares training. *IEEE Transactions on Neural Networks,* 11(2), pp. 306-314.

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning.* s.l.:MIT Press.

Güler, I. & Übeyli, E. D., 2005. ECG beat classifier designed by combined neural network model. *Pattern Recognition,* 38(2), p. 199–208.

Gunter, D. & Terry, P., 2005. Emergence of National Electronic Health Record Architectures in the United States and Australia: Models, Costs, and Questions. *Journal of Medical Internet Research,* 7(1), pp. 1-13.

Hannan, S. A., Manza, R. R. & Ramteke, R. J., 2010. Generalized Regression Neural Network and Radial Basis Function for Heart Disease Diagnosis. *International Journal of Computer Applications,* 7(13), p. 7–13.

Hayat, M., Bennamoun, M. & An, S., 2015. Deep reconstruction models for image set classification. *IEEE Trans. Pattern Anal. Mach. Intell,* Volume 37, p. 713–727.

Hedeshi, N. & Abadeh, M., 2014. Coronary Artery Disease Detection Using a Fuzzy-Boosting PSO Approach. *Computational Intelligence and Neuroscience,* 2014(6).

Heller, R. F., Chinn, S., Tunstall Pedoe, H. D. & Rose, G., 1984. How well can we predict coronary heart disease? Findings in the United Kingdom Heart Disease Prevention Project. *British Medical Journal(Clinical Research Edition),* 12 May, 288(6428), pp. 1409-1411.

Helma, C., Gottmann, E. & Kramer, S., 2000. *Knowledge discovery and data mining in toxicology.*

Heydari, S. T., Ayatollahi, S. M. & Zare, N., 2012. Comparison of Artificial Neural Networks with Logistic Regression for Detection of Obesity. *Journal of Medical Systems,* 36(4), pp. 2449-2454.

Higuera, V., 2014. *Healthline Media Overview of Basics of Heart Disease.* [Online] Available at: http://www.healthline.com/health/heart-disease/types#Overview1 [Accessed 09 October 2016].

Hinton, G. et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag,* Volume 29, p. 82–97.

Hinton, G. E. et al., 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magzine,* 29(6), p. 82–97.

Hoffmann, G. A., 2004. Adaptive transfer function in radial basis function (RBF) networks. In: *Computational Science - ICCS 2004.* s.l.:Springer Link, pp. 682-686.

Hongzong, S. et al., 2007. Support vector machines classification for discriminating coronary heart disease patients from non-coronary heart disease. *West Indian Medical Journal,* 56(5), p. 451 – 457.

Huang, H., J. Liu, Q. Z. & R. Wang, G. H., 2014. A new hierarchical method for inter-patient heartbeat classification using random projections and RR intervals. *Biomedical Engineering,* pp. 1-26.

Huang, P.-S.et al., 2014. *Kernel methods match deep neural networks on TIMIT*. Florence, Italy , IEEE.

Huang, Z., Wang, R., Shan, S. & Chen, X., 2015. Face recognition on large-scale video in the wild with hybrid Euclidean-and-Riemannian metric learning. *Pattern Recognition,* Volume 48, p. 3113–3124.

Isa, S. M., Fanany, M. I., Jatmiko, W. & Murni, A., 2010. *Feature and Model Selection on Automatic Sleep Apnea Detection using ECG.* s.l., International Conferences on Advanced Computer Science and Information Systems.

JARVIS, M. & MITRA, P., 2000. Apnea patients characterized by 0.02 Hz peak in the multitaper spectrogram of electrocardiogram signals. *Computers in Cardiology,* Volume 27, p. 769–772.

Jezzini, A., Ayache, M., Elkhansa, L. & Ibrahim, Z. a. a., 2015. *ECG classification for sleep apnea detection.* Beirut, IEEE.

Johnson, C. R., 1990. *Matrix Completion Problems: A Survey.* s.l., Matrix Theory and Applications .

Jones, N., 2014. Computer science: The learning machines. *Nature,* 505(7482), pp. 146-148.

Jose, C., Goyal, P., Aggrwal, P. & Varma, M., 2013. Local deep kernel learning for efficient non-linear SVM prediction. *The Journal of Machine Learning Research (JMLR),* 28(3), p. 486–494.

Julier, S. J. & Uhlmann, J. K., 1997. New extension of the Kalman filter to nonlinear systems. *International Society for Optics and Photonics,* Volume 3068.

Kaggle, 2014. *Higgs boson machine learning challenge.* [Online] Available at: http://www.kaggle.com/c/higgs-boson [Accessed 26 October 2016].

Kaggle, 2014. *National Data Science Bowl.* [Online] Available at: https://www.kaggle.com/c/datasciencebowl [Accessed 5 Janurary 2017].

Kaguara, A., Myoung Nam, K. & Reddy, S., 2014. A deep neural network classifier for diagnosing sleep apnea from ECG data on smartphones and small embedded systems. In: *Thesis.* s.l.:s.n., p. December.

Kahramanli, H. & Allahverdi, N., 2008. Design of a hybrid system for the diabetes and heart diseases. *Journal of Expert Systems with Applications,* Volume 35, pp. 82-89.

KAMC, 2017. *King Abdullah Medical City.* [Online] Available at: http://www.kamc.med.sa/index.php/en/ [Accessed 20 January 2017].

Karabulut, E. & Ibrikci, T., 2012. Effective Diagnosis of Coronary Artery Disease Using The Rotation Forest Ensemble Method. *Journal of Medical Systems,* Volume 36, pp. 3011-3018.

Karabulut, E. M. & İbrikçi, T., 2012. Effective Diagnosis of Coronary Artery Disease Using The Rotation Forest Ensemble Method. *Journal of Medical Systems,* 36(5), p. 3011 – 3018 .

Karpagachelvi, S., Arthanari, M. & Sivakumar, M., 2011. Classification of ECG Signals Using Extreme Learning Machine. *Computer and Information Science,* 4(1).

Kawaguchi, K., 2000. *Linear Separability and the XOR Problem.* [Online] Available at: http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node19.html [Accessed 26 October 2016].

keras, 2016. *Keras Documentation.* [Online] Available at: https://keras.io/ [Accessed August 2016].

Khandoker, A., Karmakar, C. & Palaniswami, M., 2009. Automated recognition of patients with obstructive sleep apnoea using waveletbased features of electrocardiogram recordings. *Computers in Biology and Medicine,* 39(3), pp. 88-96.

Kirk, J., 2013. *Universities, IBM join forces to build a brain-like computer.* [Online] Available at: http://www.pcworld.com/article/2051501/universities-join-ibm-in-cognitive-computing-research-project.html [Accessed 26 October 2016].

Kohonen, T., 1995. Learning vector quantization. In: *Self-Organizing Maps.* Cambridge: Springer Series in Information Sciences , pp. 175-189.

Kothari, C. R., 2004. *Research Methodology Methods and Techniques.* New Delhi, NewAge International Publishers.

Kubat, M., 1998. Decision trees can initialize radial basis function networks. *IEEE TRANSACTIONS ON NEURAL NETWORKS,* 9(5), pp. 813 - 821.

Kuulasmaa, K. et al., 2000. Estimation of contribution of changes in classic risk factors to trends in coronary-event rates across the WHO MONICA project populations. *Lancet,* 26 February, 355(9205), pp. 675-687.

Lannoy, G. d., François, D., Delbeke, J. & Verleysen, M., 2010. Weighted SVMs and feature relevance assessment in supervised heart beat classification. *Biomedical Engineering Systems and Technologies,* p. 212–223.

LeCun, Y., 2012. *Learning invariant feature hierarchies.* Florence, p. 496–505.

LeCun, Y. & Bengio, Y., 1998. Convolutional networks for images, speech and time-series. In: *The handbook of brain theory and neural networks.* Cambridge: MIT Press , pp. 255-258 .

LeCun, Y., Y. Bengio & Hinton, G. E., 2015. Deep learning. *Nature,* Volume 521, p. 436–444.

Lichman, M., 2013. *UCI Machine Learning Repository.* [Online] Available at: http://archive.ics.uci.edu/ml [Accessed 1 November 2016].

Lin, Z., Chen, M. & Ma, Y., 2011. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *NIPS 2011.*

Llamedo, M. & Martinez, J. P., 2012. An automatic patient-adapted ECG heartbeat classifier allowing expert assistance. *IEEE Transactions on Biomedical Engineering,* 59(8), p. 2312–2320.

Louridas, P. & Ebert, C., 2016. Machine Learning. *IEEE Software,* 33(5), pp. 110-115.

Lu, P., Chen, J., Zhao, H. & Y.Gao, 2012. In Silico Syndrome Prediction for Coronary Artery Disease in Traditional Chinese Medicine. *Evidence-Based Complementary and Alternative Medicine,* Volume 2012, p. 11.

Luukka, P. & 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. *Journal of Computer Intelligence in Optimization Adaption, Learning and Optimization,* Volume 7, pp. 263-283.

Luz', E. J. d. S., Schwartz, W. R., Cháveza, G. C. & Menotti, D., 2015. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine,* December, Volume 127, p. 144–164.

Mairal, J., Koniusz, P., Harchaoui, Z. & Schmid, C., 2014. *Convolutional kernel networks.* Montreal, Canada.

Malhotra, N. K. & F., B. D., 2006. *Marketing Research An Applied Approach.* 2nd ed. England: Pearson Education Limited.

Mandal, I. & Sairam, N., 2012. SVM-PSO based Feature Selection for Improving Medical Diagnosis Reliability using Machine Learning Ensembles. *Computer Science & Information Technology,* Volume 6, p. 267 – 276.

Manrique, Q. et al., 2009. *Detection of Obstructive Sleep Apnea in ECG Recordings Using Time-Frequency Distributions and Dynamic Features.* s.l., s.n., pp. 5559-5562.

Mar, T. et al., 2011. Optimization of ECG classification by means of feature selection. *IEEE Transactions on Biomedical Engineering,* 58(8), p. 2168–2177.

MCNAMES, J. & FRASER, A., 2000. Obstructive sleep apnea classification based on spectrogram patterns in the electrocardiogram. *Computers in Cardiology,* Volume 27, p. 749–752.

MedicineNet, 2016. [Online] Available at: http://www.medicinenet.com/script/main/art.asp?articlekey=5160

Medicine, N. L. o., 2016. *Arrhythmia.* [Online] Available at: https://medlineplus.gov/arrhythmia.html [Accessed October 2016].

Memisevic, R., 2003. *Unsupervised kernel regression for non-linear dimensionality reduction.* [Online] Available at: https://www.iro.umontreal.ca/~memisevr/pubs/ukr.pdf [Accessed 11 January 2017].

Mendez, M. et al., 2007. *Detection of Sleep Apnea from surface ECG based on features extracted by an Autoregressive Model.* IEEE.

Mendez, M. et al., 2007. *Detection of Sleep Apnea from Surface ECG Based on Features Extracted by an Autoregressive Model.* pp. 6105-6108.

Menkovski, V., Aleksovski, Z., Saalbach, A. & Nickisch, H., 2015. *Can Pretrained Neural Networks Detect Anatomy?.* NIPS 2015 Workshop on Machine Learning in Healthcar.

MIETUS, J., PENG, C., IVANOV, P. & GOLDBERGER, A., 2000. Detection of obstructive sleep apnea from cardiac interbeat interval time series. *Computers in Cardiology,* Volume 27, p. 753–756.

Mirjalili, S., 2010. *A new Hybrid PSOGSA Algorithm For Function Optimization.* s.l., s.n.

Mishra, A. K. & Raghav, S., 2010. Local fractal dimension based ECG arrhythmia classification. *Biomedical Signal Processing and Control,* 5(2), p. 114–123.

Mitchell, T., 1999. Machine learning and data mining. *Communication ACM,* 42(11), pp. 30-36.

Moavenian, M. & Khorrami, H., 2010. A qualitative comparison of artificial neural networks and support vector machines in ECG arrhythmias classification. *Expert Systems with Applications,* 37(4), p. 3088–3093.

Najafabadi, M. M. et al., 2015. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1).

Nasiri, J. A., Naghibzadeh, M., Yazdi, H. S. & Naghibzadeh, B., 2009. ECG arrhythmia classification with support vector machines and genetic algorithm. *IEEE European Symposium on Computer Modeling and Simulation,* p. 187–192.

National Heart Foundation of Australia, 2016. *Heart arrhythmias,* s.l.: National Heart Foundation of Australia.

National Heart, L. a. B. I., 2016. *What Is Coronary Heart Disease?.* [Online] Available at: http://www.nhlbi.nih.gov/health/health-topics/topics/cad [Accessed 17 October 2016].

National Institute of Health, 2015. *What is Coronary Heart Disease.* [Online] Available at: http://www.nhlbi.nih.gov/health/health-topics/topics/cad [Accessed 6 March 2016].

Nawi, N. M. & Ghazali, R., 2010. *The Development of Improved Back-Propagation Neural Networks Algorithm for Predicting Patients with Heart Disease.* s.l., the first international conference ICICA, pp. 317-324.

NCBI, 2014. *Echocardiography*. [Online] Available at: https://www.ncbi.nlm.nih.gov/pubmedhealth/PMH0063035/ [Accessed 15 January 2017].

Ng, A. Y., Jordan, M. & Weiss, Y., 2001. On spectral clustering analysis and an algorithm. *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pp. 849--856.

NHS, 2015. *Coronary heart disease - Diagnosis.* [Online] Available at: http://www.nhs.uk/Conditions/coronary-heart-disease/Pages/diagnosis.aspx [Accessed 17 October 2016].

Noviyanto, A., Isa, S. M., Wasito, I. & Arymurthy, A. M., 2011. Selecting Features of Single Lead ECG Signal for Automatic Sleep Stages Classification using Correlation-based Feature Subset Selection. *IJCSI International Journal of Computer Science Issues*, 8(5), p. 139.

Okutani, I. & Stephanedes, Y. J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. *Transportation Research Part B: Methodological*, 18(1), pp. 1-11.

Orange, 2016. [Online] Available at: http://orange.biolab.si/

Orr, M. J., 1996. *Introduction to radial basis function network.* [Online] Available at: http://www.cc.gatech.edu/~isbell/tutorials/rbf-intro.pdf [Accessed 11 January 2017].

Orr, M. J., 1999. *Recent advances in radial basis function networks*.

Osowski, S. & Linh, T. H., 2001. ECG beat recognition using fuzzy hybrid neural network. *IEEE Transactions on Biomedical Engineering*, 48(11), p. 1265–1271.

Osowsk, S., Markiewicz, T. & Hoai, L. T., 2008. Recognition and classification system of arrhythmia using ensemble of neural networks. *Measurement*, 41(6), pp. 610-617.

Özbay, Y., Ceylan, R. & Karlik, B., 2006. A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Computers in Biology and Medicine*, 36(4), p. 376–388.

Özcan, N. & Gurgen, F., 2010. Fuzzy support vector machines for ECG arrhythmia detection. *IEEE International Conference on Pattern Recognition,* p. 2973–2976.

Park, K. S. et al., 2008. Hierarchical support vector machine based heartbeat classification using higher order statistics and hermite basis function. *Computers in Cardiology,* pp. 229-232.

Pati, D. D. et al., 2012. Intelligent Arrhythmia Diagnostics System. *International Journal of Computer Science Issues,* November, 9(6), pp. 408- 413.

Payam Mohebbi, P. J., 2016. Survey Data Mining and its Application in Industrial Engineering. *Nature and Science,* 14(9), pp. 70-75.

PhysioNet, 2012. *Detecting and quantifying apnea based on the ECG.* [Online] Available at: https://www.physionet.org [Accessed 18 August 2016].

Podgorelec, V., Kokol, P., Stiglic, B. & Rozman, I., 2002. Decision Trees: An Overview and Their Use in Medicine. *Journal of Medical Systems,* 26(5), pp. 445-463.

Polat, K. & Günes, S., 2007. Detection of ECG arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine. *Applied Mathematics and Computation,* 186(1), p. 898–906.

PRO SLEEP CARE, 2015. *SLEEP APNEA.* [Online] Available at: http://prosleepcare.weebly.com/sleep-apnea.html [Accessed 20 January 2016].

Rachim, V., Li, G. & Chung, W., 2014. Sleep apnea classification using ECG-signal wavelet-PCA features. *Bio-Medical Materials and Engineering,* Volume 24, p. 2875–2882.

Ramli, A. B. & Ahmad, P. A., 2003. Correlation analysis for abnormal ECG signal features extraction. *4th National Conference on Telecommunication Technology,* pp. 232-237.

Ravelo-Garcia, A. et al., 2013. *Cepstrum feature selection for the classification of Sleep Apnea-Hypopnea Syndrome based on heart rate variability.* Zaragoza, Computing in Cardiology Conference (CinC).

RAYMOND, B., CAYTON, R., BATES, R. & CHAPPELL, M., 2000. Screening for obstructive sleep apnoea based on the electrocardiogram – the Computers in Cardiology Challenge. *Computers in Cardiology,* Volume 27, pp. 267-270.

Razmjooy, N. & Ramezani, M., 2016. Training Wavelet Neural Networks Using Hybrid Particle Swarm Optimization and Gravitational Search Algorithm for System Identification. *International Journal of Mechatronics, Elictrical and Computer Technology,* 6(21), pp. 2987-2997.

Rebai, I., BenAyed, Y. & Mahdi, W., 2016. Deep multilayer multiple kernel learning. *Neural Computing and Applications,* 27(8), p. 2305–2314.

Rouse, M., 2016. *machine learning.* [Online] Available at: http://whatis.techtarget.com/definition/machine-learning [Accessed 28 October 2016].

Rumelhart, D. E. & McClelland, J. L., 1986. *Parallel Distributed Processing:Explorations in the Microstructure of Cognition.* Cambridge, s.n.

Sainath, T. N. et al., 2015. Deep convolutional neural networks for large-scale speech tasks. *Neural Networks,* Volume 64, p. 39–48.

Schrader, M. et al., 2000. Detection of sleep apnea in single channel ECGs from the PhysioNet data base. *Computers in Cardiology,* Volume 27, p. 263–266.

Schwenker, F. & Kestler, H. A., 2001. 3-D Visual Object Classification with Hierarchical Radial Basis Function Networks. In: *Radial Basis Function Networks 2.* s.l.: Studies in Fuzziness and Soft Computing, pp. 269-293.

Schwenker, F., Kestler, H. A. & Palm, G., 2001. *Three learning phases for radial-basis-function networks.*

scikit-learn, 2016. *scikit-learn.* [Online] Available at: http://scikit-learn.org/stable/modules/feature_selection.html [Accessed 22 November 2016].

Sen, A. K., Patel, S. B. & Shukla, D. D. P., 2013. A Data Mining Technique for Prediction of Coronary Heart Disease Using Neuro-Fuzzy Integrated Approach Two Level. *International Journal Of Engineering And Computer Science,* 2(9), pp. 2663-2671.

Setiawan, N. A., Venkatachalam, P. A. & Fadzil, M. H. A., 2009. Rule selection for coronary artery disease diagnosis based on rough set. *International Journal of Recent Trends in Engineering,* 2(5), pp. 198-202.

Shahkarami, A., Mohaghegh, S. D., Gholami, V. & Haghighat, S. A., 2014. *Artificial Intelligence (AI) Assisted History Matching.* s.l., s.n.

Shahwan-Akl, L., 2010. Cardiovascular Disease Risk Factors among Adult Australian-Lebanese in Melbourne. *International Journal of Research in Nursing,* 1(1), pp. 1-7.

Shin, H. et al., 2013. tacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4d patient data. *IEEE Trans. Pattern Anal. Mach. Intell.,* 35(8), p. 1930–1943.

Shouman, M., Turner, T. & Stocker, R., 2012. Applying k-Nearest Neighbour in Diagnosing Heart Disease Patients. *International Journal of Information and Education Technology,* 2(3), pp. 220-223.

Simons, L. et al., 2003. Risk functions for prediction of cardiovascular disease in elderly Australians: the Dubbo Study.. *Medical Journal of Australia,* February, 178(3), pp. 113-116.

Simonyan, K. & Zisserman, A., 2014. Very deep convolutional networks for large-sacle image recognition. *Computing Research Repository (CoRR),* Volume abs/1409.1556.

Smith, B. R., 1997. *Neural Network Enhancement of Closed-Loop Controllers for Ill-Modeled Systems with Unknown Nonlinearities.* PhD. Dissertation ed. s.l.:ETD.

Smith, M., Robinson, L. & Segal, R., 2016. *Sleep Apnea.* [Online] Available at: http://www.helpguide.org/articles/sleep/sleep-apnea.htm [Accessed October 2016].

Socher, R., Bengio, Y. & Manning, C. D., 2012. *Deep learning for NLP (without magic).* Jeju Island, Korea, s.n.

Soniya, Paul, S. & Singh, L., 2015. *A review on advances in deep learning.* s.l., s.n., pp. 1 - 6.

Sounders, M., Lewis, P. & Thornhill, A., 2009. *Research Methods for Business Students.* 5th ed. England:: Pearson Education Limited.

Srinivas, K., B.Kavihta, R. & Govardhan, A., 2010. Applications of Data Mining Techniques in Healthcare and Prediction of Heart Attacks. *International Journal on Computer Science and Engineering (IJCSE)*, March, 02(02), pp. 250-255.

Strobl, E. & Visweswaran, S., 2013. Deep Multiple kernel learning. *ICMLA '13 Proceedings of the 2013 12th International Conference on Machine Learning and Applications*, 27(8), pp. 414-417.

Strobl, E. V. & Visweswaran, S., 2013. *Deep Multiple Kernel Learning.* s.l., Proceedings of the 2013 12th International Conference on Machine Learning and Applications, pp. 414-417.

Tadejko, P. & Rakowski, W., 2007. Mathematical Morphology Based ECG Feature Extraction for the Purpose of Heartbeat Classification. *6th International Conference on Computer Information Systems and Industrial Management Applications*, pp. 322-327.

Tagluk, M. E., Akin, M. & Sezgin, N., 2010. Classıfication of sleep apnea by using wavelet transform and artificial classification. *Expert Systems with Applications*, March, 37(2), pp. 1600-1607.

Takeda, H., Farsiu, S. & Milanfar, P., 2007. Kernel regression for image processing and reconstruction. *IEEE Trans Image Process.* , 16(2), pp. 349-366.

Tan, K. & Teoh, E., 2009. A hybrid evolutionary algorithm for attribute selection in data mining. *Journal of Expert system with applications*, Volume 36, pp. 8616-8630.

tensorflow, 2016. [Online] Available at: https://www.tensorflow.org/

Thuraisingham, R., 2006. Preprocessing RR interval time series for heart rate variability analysis and estimates of standard deviation of RR intervals. *Computer Methods and Programs in Biomedicine*, 83(1), pp. 78-82.

TİMUŞ, O. H. & KIYAK, E., 2015. Optimizing MLP Classifier and ECG Features for Sleep Apnea Detection. *Journal of Naval Science and Engineering* , 11(1), pp. 1-18.

Tsirogiannis, G. et al., 2004. *Classification of Medical Data with a Robust Multi-Level Combination Scheme.* Budapest, s.n.

University of Toronto, 2013. *What is Health Informatics.* [Online] Available at: http://ihpme.utoronto.ca/academics/pp/mhi/faq/ [Accessed 11 January 2017].

Vachkov, G., Stoyanov, V. & Christova, N., 2015. *Growing RBF network models for solving nonlinear approximation and classification problems.* Albena, Bulgaria, European Conference on Modeling and Simulation.

Vijiyarani, S. & Sudha, S., 2013. An Efficient Classification Tree Technique for Heart Disease Prediction. *IJCA Proceedings on International Conference on Research Trends in Computer Technologies 2013,* ICRTCT(3), pp. 6-9.

Vincent, P. et al., 2010. Stacked denoising auto-encoders: learning useful representations in a deep network using local denoising criteria. *The Journal of Machine Learning Research,* Volume 11, pp. 3371-3408.

Wang, M., Sha, F. & Jordan, M. I., 2010. *Unsupervised kernel dimension reduction.* s.l., s.n., pp. 2379--2387.

Wang, X., 2011. *The application of deep kernel machines to various types of data.* [Online]
Available at: https://uwaterloo.ca/computational-mathematics/sites/ca.computational-mathematics/files/uploads/files/shirly_project.pdf

Wang, Y.-Y., Yu, D., Ju, Y.-C. & Acero, A., 2011. *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech.* s.l.:Voice search.

Webb, R. & Shennon, S., 1998. Shape-adaptive radial basis functions. *IEEE Transactions on Neural Networks,* 9(6), pp. 1155 - 1166.

Welling, M., 2005. Kernel principal component analysis. *Advances in neural information processing systems,* Volume 15, pp. 70-72.

Wettschereck, D. & Dietterich, D., 1992. *Improving the performance of the radial basis function networks by learning center locations.* s.l., Morgan Kaufmann, pp. 1133-1140.

WFDB, 2015. *The WFDB Software Package.* [Online] Available at: https://www.physionet.org/physiotools/wfdb.shtml [Accessed 25 August 2016].

WHO, 2007. *World Health Organization.* [Online] Available at: http://www.who.int/mediacentre/factsheets/fs310.pdf [Accessed October 2016].

WHO, 2015. *WHO World Health Organization.* [Online] Available at: http://www.who.int/mediacentre/factsheets/fs317/en [Accessed 6 March 2016].

WHO, 2016. *Cardiovascular Disease (CVDs).* [Online] Available at: http://www.who.int/mediacentre/factsheets/fs317/en/ [Accessed 17 October 2016].

Wiering, M. et al., 2013. *Deep support vector machines for regression problems.* s.l., s.n.

Wilson, P. et al., 1998. Prediction of Coronary Heart Disease Using Risk Factor Categories. *American Heart Association Journal.*

Wu, T. T. & Lange, K., 2015. Matrix completion discriminant analysis. *Computational Statistics & Data Analysis,* Volume 92, p. 115 – 125.

Xie, B. & Minn, H., 2012. Real-time sleep apnea detection by classifier combination. *Information Technology in Biomedicine,* 16(3), pp. 469-477.

Xinyuan, D., Jinjie, W., Sufeng, W. & Ting, C., 2012. *An improved CDKF algorithm based on RBF neural network for satellite attitude determination.* s.l., 2012 International Conference on Image Analysis and Signal Processing, pp. 1-7.

Yan, H., Zheng, J., Jiang, Y. & Li, Q., 2003. *Development of a decision support system for heart disease diagnosis using multilayer perceptron.* s.l., s.n., pp. 709-712.

Yeh, Y.-C., Chiou, C. W. & Lin, H.-J., 2012. Analyzing ECG for cardiac arrhythmia using cluster analysis. *Expert Systems with Applications,* 39(1), p. 1000–1010.

Yger, F., Berar, M., Gasso, G. & Rakotomamonjy, A., 2011. A supervised strategy for deep kernel machine. *19th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning,* pp. 501-506.

Yilmaz, B. et al., 2010. Sleep stage and obstructive apneaic epoch classification using single-lead ECG. *Biomed Eng Online,* 9(1), p. 39.

Yousef, R. & El Hindi, K., 2006. Training radial basis function networks using reduced sets as center points. *International Journal of Information Technology,* 2(1), p. 21.

You, W. et al., 2009. Recognition of Coronary Heart Disease Patients by RBF Neural Network Basing on Contents of Microelements in Human Blood. *2009 Second International Symposium on Computational Intelligence and Design*, Volume 2, pp. 409 - 412.

Yu, S.-N. & Chen, Y.-H., 2007. Electrocardiogram beat classification based on wavelet transformation and probabilistic neural network. *Pattern Recognition Letters*, 28(10), p. 1142–1150.

Zarb, J., 2016. [Online] Available at: http://ihpme.utoronto.ca/academics/pp/mhi/

Zeiler, M. D. & Fergus, R., 2014. *Visualizing and understanding convolutional networks.* s.l., s.n., p. 818–833.

Zhuang, J., Tsang, I. W. & Ho, i. S. C. H., 2011. Two-layer multiple kernel learning. *The Journal of Machine Learning Research (JMLR)*, Volume 15, pp. 909-917.

Zhu, R. M. H., Zhang, L. & Chen, A., 2006. *A New Method to Assist Small Data Set Neural Network Learning.* Jinan, Intelligent Systems Design and Applications, 2006.

Zweig, M. & Campbell, G., 1993. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, Volume 39, pp. 561-577.

Arora, R., Basu, A., Mianjy, P. & Mukherjee, A., 2016. Understanding Deep Neural Networks with Rectified Linear Units. *arXiv preprint arXiv:1611.01491*.

Bengio, Y., 2009. Learning deep architectures for AI. *Foundation and Trends in Machine Learning*, 2(1), pp. 1-127.

Bengio, Y. & Glorot, X., 2010. *Understanding the difficulty of training deep feedforward neural networks.* s.l., s.n., pp. 249-256.

Bianchini, M. & Scarselli, F., 2014. On the Complexity of Neural Network Classifiers: A Comparison Between Shallow and Deep Architectures. *IEEE Transactions on Neural Networks and Learning Systems* , 25(8), pp. 1553 - 1565.

Chen, J. & Deng, L., 2013. A new method for learning deep recurrent neural networks. *arXiv:1311.6091 [cs.LG]*.

Cybenko, G., 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of control,signals and systems*, 2(4), pp. 303-314.

Dahl, G. E., Sainath, T. N. & Hinton., G. E., 2013. *Improving deep neural networks for LVCSR using rectified linear units and dropout.* s.l., IEEE, pp. 8609-8613.

Erhan, D. et al., 2010. *Why does unsupervised pre-training help deep learning.* s.l., s.n., pp. 625-660.

Glorot, X., Bordes, A. & Bengio, Y., 2011. *Deep Sparse Rectifier Neural Networks.* Fort Lauderdale, FL, USA, s.n.

Harmon, M. & Klabjan, D., 2017. Activation Ensembles for Deep Neural Networks. *arXiv preprint arXiv:1702.07790*.

Hinton, G. E., Osindero, S. & Teh, Y., 2006. A fast learning algorithm for deep belief nets. *Neural Computation,* Volume 18, pp. 1527-1554.

Janocha, K. & Czarnecki, W. M., 2017. On Loss Functions for Deep Neural Networks in Classification. *Theoretical Foundations of Machine Learning 2017.*

Kayser, M., 2016. *Quora.* [Online]
Available at: https://www.quora.com/What-is-the-difference-between-deep-learning-and-shallow-learning
[Accessed 15 June 2017].

Kingma, D. & Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

Oruganti, R. M., 2016. *Image Description using Deep Neural Networks,* s.l.: s.n.

Pascanu, R., Gulcehre, C., Cho, K. & Bengio, Y., 2013. How to Construct Deep Recurrent Neural Networks. *arXiv preprint arXiv:1312.6026.*.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. *arXiv:1609.04747.*

Salakhutdinov, R. & Hinton, G. E., 2009. *Deep boltzmann machines.* s.l., s.n., p. 3.

Seide, F., Li, G., Chen, X. & Yu, D., 2011. *Feature engineering in Context-Dependent Deep Neural Networks for conversational speech transcription.* Waikoloa, HI, USA, IEEE.

Seide, F., Li, G. & Yu, D., 2011. *Conversational Speech Transcription Using Context-Dependent Deep Neural Networks.* s.l., s.n.

Sermanet, P. et al., 2014. *Overfeat: Integrated recognition, localization and detection using convolutional networks.* s.l., s.n.

Srivastava, N. et al., 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), p. 1929–1958.

Veselý, K., Ghoshal, A., Burget, L. & Povey, D., 2013. *Sequence-discriminative training of deep neural networks.* s.l., Interspeech, pp. 2345-2349.

Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A., 2008. Extracting and composing robust features with denoising autoencoders. *ICML 2008.*

Yu, D. & Deng, L., 2010. *Deep-Structured Hidden Conditional Random Fields for Phonetic Recognition.* s.l., s.n.

(Anon., n.d.)
(Tayarani-N, 2009)

APPENDICES

Appendix I: Data Collection Form

Data Collection Form

Title

Prediction of Coronary Artery Disease Using a Combination of Methods for Training Radial Basis Function Networks

Principal investigator:

Dr. Osama Elkhateeb

Inclusion criteria of case group	**Yes** ☐ **1**	**No** ☐ **2**	**Exclusion criteria of case group**	**Yes** ☐ **1**	**No** ☐ **2**
1- More than 50% stenosis (if more mention it …..%)	**Yes** ☐ **1**	**No** ☐ **2**	**1- Patient refuse**	**Yes** ☐ **1**	**No** ☐ **2**
2- Diagnosed As CHD*	**Yes** ☐ **1**	**No** ☐ **2**	**2- Uncertain Diagnosis**	**Yes** ☐ **1**	**No** ☐ **2**
3- Age 30 - 70 Years Old	**Yes** ☐ **1**	**No** ☐ **2**			
4- vessels occlusion (No#)	**Yes** ☐ **1**	**No** ☐ **2**			

***Diagnosed As CHD By .....**

Inclusion criteria of control group	**Yes** ☐ **1**	**No** ☐ **2**	**Exclusion criteria of control group**	**Yes** ☐ **1**	**No** ☐ **2**
1- Volunteering from other hospital departments	**Yes** ☐ **1**	**No** ☐ **2**	**1- stenosis**	**Yes** ☐ **1**	**No** ☐ **2**
			2- Diabetes	**Yes** ☐ **1**	**No** ☐ **2**
			3- Hypertension	**Yes** ☐ **1**	**No** ☐ **2**

(e.g. Electrocardiogram (ECG), Catheterization, Chest X-Ray (CXR), Computed Tomography Scan (CT), Magnetic Resonance Imaging (MRI))

Patient serial code	☐☐☐	**Patient initials:**	☐☐☐

Date of consent	☐☐ ☐☐ ☐☐☐☐ *Day Month Year*	**Age at enrolment**	☐☐☐ **Years (e.g. 045)**
Investigator's name		**Investigator's signature**	

Name			
Age	☐☐☐ **Years (e.g. 045)**	**Gender**	**Male** ☐ **1** **Female** ☐ **2**
Nationality	**Saudi** ☐ **1** **Non Saudi** ☐ **2**	**Area of residence**	**Makkah** ☐ **1** **Outside Makkah** ☐ **2**
Marital status		**Number of Children**	
Weight		**Height**	
BMI(kg/m^2)	**= .....** **Underweight <18.50** ☐ **1** **Normal range 18.50 - 24.99** ☐ **2** **Overweight ≥25.00** ☐ **3** **Obese ≥30.00** ☐ **4**	**Level of educatio-n**	**Illaterate** ☐ **1** **Primary** ☐ **4** **Average** ☐ **2** **Secondary** ☐ **5** **University** ☐ **3** **Graduate** ☐ **6**
Smoking	**No** ☐ **1** **Previous** ☐ **2** **Current** ☐ **3**	***If current smoker, did you try quitting**	**Yes** ☐ **1** **No** ☐ **2**

Patient serial code	☐☐☐	**Patient initials:**	☐☐☐

***Number of Cigarette / Year**		***Daily cigarettes consumption**	**No of Cigarette or Pachkets ( )** **Light smoker(less than 10 cig/day)** ☐ **1** **intermitted smoker(10-20 cig/day)** ☐ **2** **Heavy smoker (more than 20 cig/day)** ☐ **3**
***Is it difficult for you Cessation of smoking**	**Not Difficult** ☐ **1** **Some Difficulty** ☐ **2** **Difficlt** ☐ **3**	**Exercise, sport or Walking :**	**Yes** ☐ **1** **No** ☐ **2**
***If yes, type of sport/ min**		**Work**	

Medical History and Co-Morbidities

Hypertension	**Yes** ☐ **1** **No** ☐ **2** **Since (years):**	**Prehypertension (120–139 over 80–89)** ☐ **1** **Stage 1 hypertension (140–159 over 90–99)** ☐ **2** **Stage 2 hypertension (≥160 over ≥100)** ☐ **3** **Isolated systoliv hypertension (≥140 over <90)** ☐ **4**
Diabetes	**Yes** ☐ **1** **No** ☐ **2**	**Blood glucose concentration:**
Parents have hypertension	**Yes** ☐ **1** **No** ☐ **2**	
Parents have Diabetes	**Yes** ☐ **1** **No** ☐ **2**	
Parents have cardiovascular disease	**Yes** ☐ **1** **No** ☐ **2**	
Obese Relative	**Yes** ☐ **1** **No** ☐ **2**	

Patient serial code	☐☐☐	**Patient initials:**	☐☐☐

Anemia	**Yes** ☐ **1** **No** ☐ **2**	**Type**
Stroke	**Yes** ☐ **1** **No** ☐ **2**	
Heart failure	**Yes** ☐ **1** **No** ☐ **2**	

Medications:

Any medications used regularly	**Yes** ☐ **1** **No** ☐ **2**

Any medication used for more than 1 month should be mentioned below:

Name of medication	Period by weeks	Dose	Comment

Diagnostic tests and test results :

Clinical diagnosis		
Angiography	**Yes** ☐ **1** **No** ☐ **2**	**Result :**

Patient serial code		**Patient initials:**	

The results of lipid profile	**LDL**	**CH**	**HDL**	**TG**
The results of other tests	**FBS**		**PPBS**	**Others**
The results of CBC	**WBC**		**HB**	**RBC**
Others				

Appendix II: The Statistic analysis Study of Coronary-Artery Disease Data Based on King Abdullah Medical City in Saudi Arabia (KAMC-CAD)

The study is based on Saudi Arabia population, in King Abdullah Medical City with the objective to find the relationship between CAD and other variables; namely, demographic variables like age, gender, occupation, physical variables like height, weight, smoking habit, medical history among others

Data Analysis

There were 60 variables and 688 observations in the data, which was at first, analyzed by means of frequency distributions and graphs, to understand the general nature of the data and to determine the optimal statistical model to test the hypotheses. Logistic regression was the main analysis used to test the hypotheses.

Table 1: Distribution of CAD

Row Labels	# Population	% Population
0 (No)	488	70.9%
1 (Yes)	199	28.9%
(blank)	1	0.02%
Grand Total	688	100.0%

As CAD would be the target variable of the logistic regression model, the one missing observation was removed from further analysis. CAD is measured in dichotomous scale and the two categories are mutually exclusive, satisfying the prior assumptions of logistic regression.

There were 59 independent variables in the data. The frequency distribution of each of them were studied to ensure further modifications, if any, to fit in the model and to eliminate data entry errors. Mismatched entries were found and were considered as “no information” and tagged as 0 in required cases.

Following is the distribution of independent categorical variables, after cleaning up the data. Among the 687 observations, 402 came from males and 117 had a history of stroke. 124 people were smokers whereas 197 used to smoke earlier.

Table 2: Distribution of independent variables (Categorical)

Variable	Category	category description	Frequency	%
Albumin	0	No	679	98.80%
	1	Yes	8	1.16%
Amlodipine	2	No	602	87.60%
	1	Yes	76	11.10%
	0	No Information	9	1.31%
Asprin	2	No	296	43.10%
	1	Yes	292	42.50%
	0	No Information	99	14.40%
Atrovastatin	2	No	314	45.70%
	1	Yes	273	39.70%
	0	No Information	100	14.60%
BMIGroup	3		212	30.90%
	2		178	25.90%
	4		129	18.80%
	5		66	9.61%
	6		53	7.71%
	0		25	3.64%
	1		24	3.49%
CABG	2	No	568	82.70%
	0	No Information	95	13.80%
	1	Yes	24	3.49%
CaseControl	1		436	63.50%
	2		251	36.50%
Cerivastatin	2	No	588	85.60%
	0	No Information	99	14.40%
Clopidogreal	2	No	409	59.50%
	1	Yes	180	26.20%
	0	No Information	98	14.30%
DM	2	No	394	57.40%
	1	Yes	286	41.60%
	0	No Information	7	1.02%
Enoxaparinclexame	2	No	627	91.30%
	1	Yes	52	7.57%
	0	No Information	8	1.16%
Fluvastatin	2	No	587	85.40%
	0	No Information	99	14.40%
	1	Yes	1	0.15%
HF	2	No	646	94.00%
	1	Yes	28	4.08%
	0	No Information	13	1.89%
HTN	1	Yes	344	50.10%

	2	No	337	49.10%
	0	No Information	6	0.87%
Lovastatin	2	No	589	85.70%
	0	No Information	98	14.30%
Mevastatin	2	No	589	85.70%
	0	No Information	98	14.30%
ObeseR	2	No	397	57.80%
	1	Yes	280	40.80%
	0	No Information	10	1.46%
PCVD	2	No	396	57.60%
	1	Yes	159	23.10%
	3		126	18.30%
	0	No Information	6	0.87%
PDM	1	Yes	301	43.80%
	2	No	254	37.00%
	3		126	18.30%
	0	No Information	6	0.87%
Pitavastatin	2	No	589	85.70%
	0	No Information	98	14.30%
Pravastatin	2	No	589	85.70%
	0	No Information	98	14.30%
Rosuvastatin	2	No	570	83.00%
	0	No Information	98	14.30%
	1	Yes	19	2.77%
Squitting	0	No Information	461	67.10%
	1	Yes	150	21.80%
	2	No	76	11.10%
VNumber	0		511	74.40%
	1		95	13.80%
	2		54	7.86%
	3		18	2.62%
	4		9	1.31%
anemia	2	No	585	85.20%
	1	Yes	95	13.80%
	0	No Information	7	1.02%
angiography	1	Yes	334	48.60%
	2	No	258	37.60%
	0	No Information	95	13.80%
clinicaldiagnose	9		323	47.00%
	1		179	26.10%
	3		115	16.70%
	2		54	7.86%

	4		11	1.60%
	6		2	0.29%
	5		1	0.15%
	7		1	0.15%
	8		1	0.15%
nitroglycerin	2	No	574	83.60%
	0	No Information	98	14.30%
	1	Yes	15	2.18%
pantoprazole	2	No	375	54.60%
	1	Yes	214	31.10%
	0	No Information	98	14.30%
perindoprilarginine	2	No	429	62.40%
	1	Yes	159	23.10%
	0	No Information	99	14.40%
stroke	2	No	561	81.70%
	1	Yes	117	17.00%
	0	No Information	9	1.31%
typeofsmoking	0		369	53.70%
	1		237	34.50%
	2		44	6.40%
	3		27	3.93%
	4		8	1.16%
	999		2	0.29%

Figure 1: Gender distribution

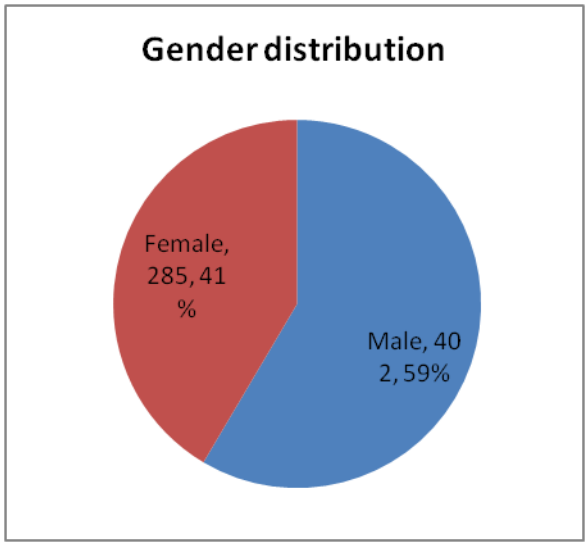


Figure 2: Smoking habit

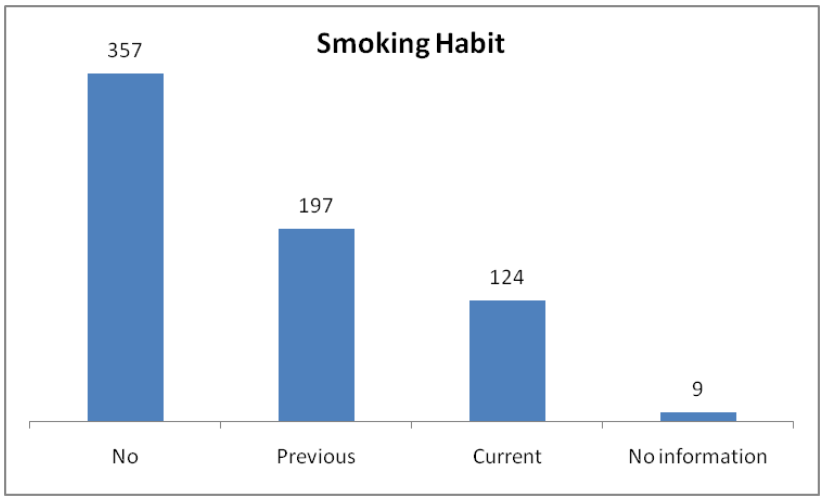
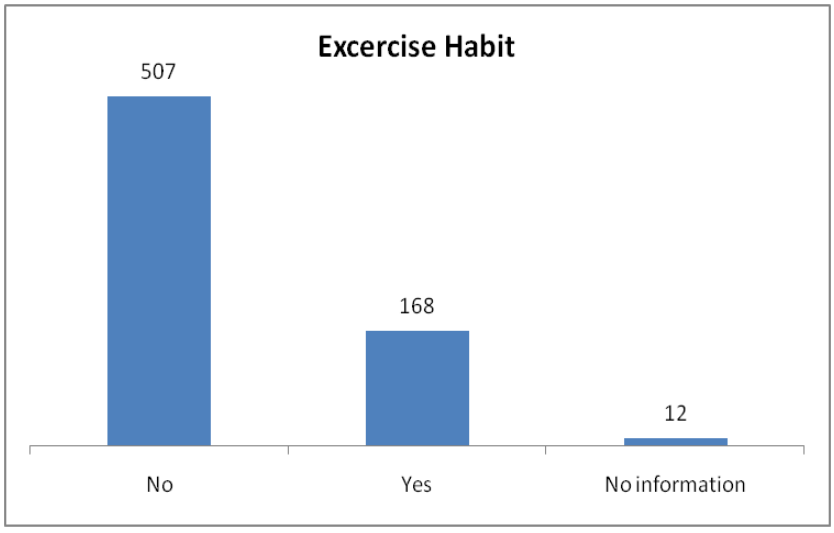
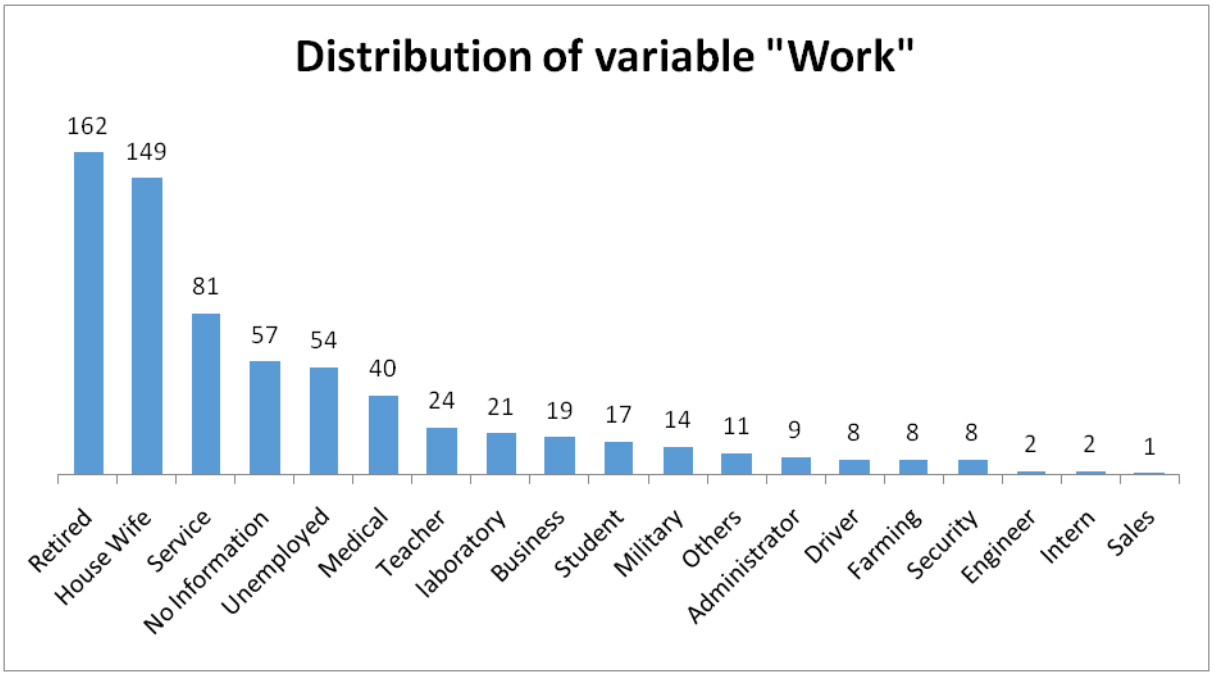


Figure3: Exercise habit



The next concern was the variable named “work”. As it had too many categories to impact a statistical model, logical grouping was done to reduce the number of categories. Observations generated with numbers 999 and 0, were considered as “No information”. The job types have been classified under several broader categories as follows.

Figure 4: Broad categories of the variable 'Work'



There were a few continuous variables in the data. Their range, central tendency and dispersion have been studied to ensure proper generation of those variables. These variables could be used directly into the logistic model as they suffice the elementary assumption of a logistic regression model. But, the model result would only give directional overview like “if BUN increases, the chances of CAD also increases”. On the other hand, if they could be transformed into categorical variables, the model result would provide strategic overview like “People having BUNwithin 60-80 have more chances to have CAD”.

Table 3: Distribution of independent variables (Continuous)

variable	Mean	Median	Min	Max	Standard deviation
BGC	79.87	0	0	999	122.54
BUN	10.23	0	0	999	39.62
CH	133.60	135	0	340	60.60
ComputeBMI	28.28	27.72	0	114.06	9.93
FBS	37.14	0	0	430	67.35
HB	8.41	11.5	0	26.4	6.57
HDL	35.61	35	0	346	21.35
Hight	159.74	163	0	999	54.94

LDL	87.16	89	0	552	47.58
PPBS	38.84	0	0	1174	97.10
RBC	3.22	4.3	0	85	4.51
TG	121.11	112	0	722	78.31
WBC	5.10	5.6	0	96.5	5.59
Weight	76.05	74	0	999	43.03
Age	51.70	53	0	999	40.08
diastolicHTN	53.09	76	0	114	38.80
systoilcHTN	86.69	120	0	200	63.34
timeofexcercise	14.23	0	0	600	42.56

Most of these continuous variables are very important in medical context and they were transformed into categorical variables, as per their medical definition. From the research, it was understood that no medical variable can have a value 0. Hence, all 0 values were considered as "No Information". Following is a brief description of how these variables are transformed into categorical variables

BCG: BCG is the indicator of down-syndrome and it is kept as a continuous variable.

BUN: Blood Urea Nitrogen test. In general, 7 to 20 mg/dL (2.5 to 7.1 mmol/L) is considered normal. We clubbed this variable as "Normal", "Less than Normal" and "More than Normal". (source: http://www.mayoclinic.org/tests-procedures/blood-urea-nitrogen/basics/results/prc-20020239)

CH, LDL and HDL: These are total cholesterol, bad and good cholesterol respectively. These variables are clubbed with the help of the following tables. (source: https://www.nlm.nih.gov/medlineplus/magazine/issues/summer12/articles/summer12pg6-7.html)

Table 4: CH, LDL and HDL Limits

CH Level	Category
Less than 200mg/dL	Desirable
200-239 mg/dL	Borderline High
240mg/dL and above	High
LDL Level	Category
Less than 100mg/dL	Optimal
100-129mg/dL	Near optimal/ above optimal
130-159 mg/dL	Borderline high
160-189 mg/dL	High
190 mg/dL and above	Very High
HDL Level	Category

Less than 40 mg/dL	A mojaro risk factor for the heart disease
40—59 mg/dL	The higher, the better
60 mg/dL and higher	Considered protective against heart disease

Compute BMI: People with BMI values ranging from 18.5 to 24.9 are categorized as normal, from 25.0 to 29.9 as overweight, and 30 and above as obese. People with BMI values of 40 and above are categorized as morbidly obese.(source: http://www.no-obesity-epidemic.org/bmi-obesity.html)

FBS & PPBS: The following table helped to classify both Fasting Blood Sugar and Post Prandial Blood Sugar. (source: http://www.medindia.net/patients/calculators/bloodsugar_chart.asp)

Table 5: FBS and PPBS limits

Category	Fasting Value (mg/dl)		Post Prandial (mg/dl)
	Minimum Value	Maximum Value	Value 2 hours after consuming glucose
Normal	70	100	Less than 140
Early Diabetes	101	126	140 to 200
Established Diabetes	More than 126	-	More than 200

RBC & WBC: Normal RBC range is, for male: 4.7 to 6.1 million cells per microliter (cells/mcL) and for female: 4.2 to 5.4 million cells/mcL. The normal number of WBCs in the blood is 4,500-10,000 white blood cells per microliter (mcL).Both of these variables are clubbed as "Normal", "Less than Normal" and "More than Normal". (source: https://www.nlm.nih.gov/medlineplus/ency/article/003644.htm and https://www.nlm.nih.gov/medlineplus/ency/article/003643.htm respectively)

HB: The reference ranges for haemoglobin concentration in adults are as follows: Men: 14.0-17.5 (mean 15.7) g/dL, Women: 12.3-15.3 (mean 13.8) g/dL (source: http://emedicine.medscape.com/article/2085614-overview)

TG: Triglycerides is a type of Fat which is categorised as follows:

Table 6: TG Levels

Triglycerides level	
Normal	Less than 150 mg/dL
Borderline High	150 - 199 mg/dL
High	200 - 499 mg/dL
Very High	500 mg/dL or above

Source: https://www.nlm.nih.gov/medlineplus/ency/article/003493.htm

Diastolic HTN & Systolic HTN:

Source: http://www.heart.org/HEARTORG/Conditions/HighBloodPressure/AboutHighBloodPressure/Understanding-Blood-Pressure-Readings_UCM_301764_Article.jsp

Table 7: Diastolic and Systolic HTN limits

Blood Pressure Category	Systolic mm Hg (upper #)	Diastolic mm Hg (lower #)
Normal	less than 120	less than 80
High Normal	120 – 139	80 – 89
(Hypertension) Stage 1	140 – 159	90 – 99
(Hypertension) Stage 2	160 or higher	100 or higher
(Hypertension) Stage 3	Higher than 180	Higher than 110

Table 8: distribution of medical variables- after classification

	Category	Frequency	Percent
CH	Desirable	532	77.33%
	Borderline High	65	9.45%
	High	21	3.05%
	No information	70	10.17%
LDL	Optimal	339	49.27%
	Near Optimal	173	25.15%
	Borderline High	75	10.90%
	High	21	3.05%
	Very High	7	1.02%

	No information	73	10.61%
HDL	Protective	38	5.52%
	Major risk	347	50.44%
	Moderate risk	231	33.58%
	No Information	72	10.47%
BMI	Underweight	49	7.12%
	Normal Weight	179	26.02%
	Overweight	212	30.81%
	Obese	195	28.34%
	Morbidly Obese	53	7.70%
RBC	Less than normal	278	40.41%
	Normal	240	34.88%
	More than normal	44	6.40%
	No Information	126	18.31%
HB	Less than normal	231	33.58%
	Normal	187	27.18%
	More than normal	22	3.20%
	No Information	248	36.05%
TG	Normal	479	69.62%
	Mildly High	83	12.06%
	High	73	10.61%
	Very High	3	0.44%
	No information	50	7.27%
Diastolic HTN	<60	30	4.36%
	Normal	108	15.70%
	High Normal	221	32.12%
	Stage1 Hypertension	80	11.63%
	Stage2 Hypertension	18	2.62%
	Stage3 Hypertension	1	0.15%
	No Information	230	33.43%
Systolic HTN	<90	2	0.29%
	Normal	66	9.59%
	High Normal	263	38.23%
	Stage1 Hypertension	78	11.34%
	Stage2 Hypertension	41	5.96%
	Stage3 Hypertension	8	1.16%
	No Information	230	33.43%

Height, Weight, Age& Time of exercise: These four variables are transformed as per their distribution and logical grouping of the variables.

Table 9: Distribution of Height, Weight, Age and Time of exercise- after classification

	Category	Frequency	Percent
Height	<=122	2	0.29%
	122 – 152	67	9.74%
	152 – 182	585	85.03%
	>=182	11	1.60%
	No information	23	3.34%
Weight	<=40	30	4.36%
	40 – 50	36	5.23%
	50 – 60	92	13.37%
	60-70	140	20.35%
	70 – 80	139	20.20%
	80 – 90	114	16.57%
	90 – 100	56	8.14%
	>=100	81	11.77%
Age	<=25	53	7.70%
	26-40	151	21.95%
	40 – 55	184	26.74%
	56 – 75	270	39.24%
	>=76	30	4.36%
Time of exercise	0	533	77.47%
	1' – 30	86	12.50%
	121 - 180	13	1.89%
	31 – 60	32	4.65%
	61 – 120	18	2.62%
	>=180	6	0.87%

Most of the population (85%) has height between 152 – 182 cm and 71% population has weight between 50 – 90 kgs. The study population is aged, with the 39% population between 56 – 75 years followed by 27% between 40 - 55 years.

Figure 5: Distribution of Height

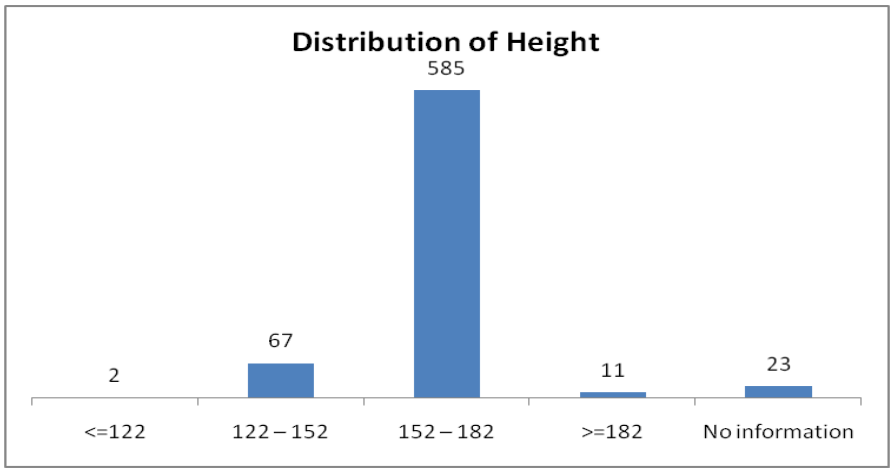


Figure 6: Distribution of Weight

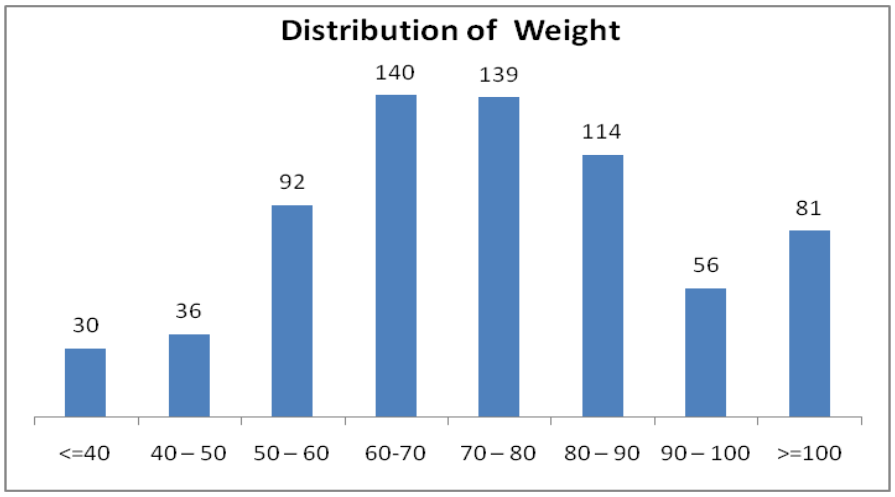
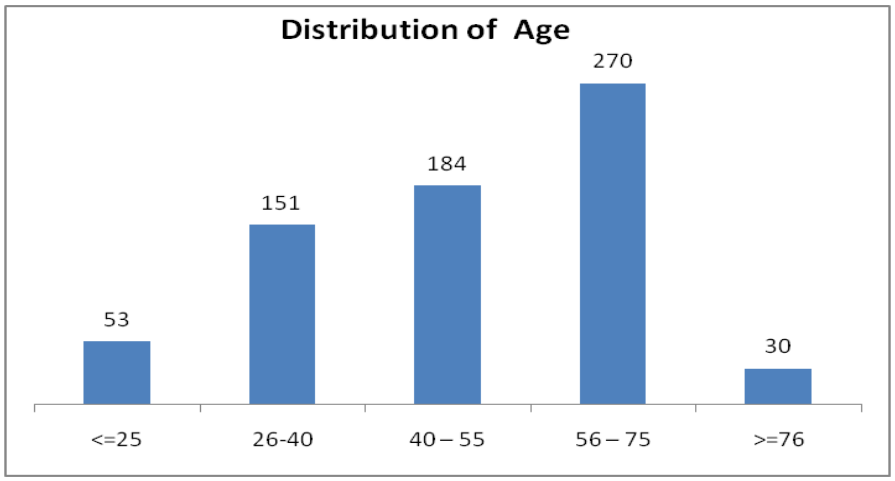


Figure 7: Distribution of Age



Statistical Methodology

The data is now prepared for the model and the Information Value (IV) of each variable is calculated, which will help decide on variable elimination from the model. IV is a measure equivalent to correlation analysis and unlike Correlation; it works for only categorical variables. IV indicates the predictive power of the variable.

Table 10: IV Values

Information Value	Predictive Power
< 0.02	useless for prediction
0.02 to 0.1	Weak predictor
0.1 to 0.3	Medium predictor
0.3 to 0.5	Strong predictor
>0.5	Suspicious or too good to be true

Table 11: IV values of the variables

Var	Iv	Decision
Albumin	0.01	Useless for Prediction
Amlodipine	0.05	Weak Predictor
Asprin	0.31	Strong Predictor
Atrovastatin	0.31	Strong Predictor
BGC	0.11	Medium Predictor
BMIGroup	0.04	Weak Predictor
CABG	0.01	Useless for Prediction
CaseControl	0.67	Too good to be true
Cerivastatin	0.00	Useless for Prediction
Clopidogreal	0.19	Medium Predictor
DM	0.13	Medium Predictor
Enoxaparinclexame	0.03	Weak Predictor
Exercise	0.00	Useless for Prediction
Fluvastatin	0.00	Useless for Prediction
HF	0.03	Weak Predictor
HTN	0.13	Medium Predictor
Lovastatin	0.00	Useless for

		Prediction
Mevastatin	0.00	Useless for Prediction
NOciggateD	0.13	Medium Predictor
ObeseR	0.04	Weak Predictor
PCVD	0.07	Weak Predictor
PDM	0.05	Weak Predictor
Pitavastatin	0.00	Useless for Prediction
Pravastatin	0.00	Useless for Prediction
Rosuvastatin	0.03	Weak Predictor
Smoking	0.09	Weak Predictor
Squitting	0.04	Weak Predictor
VNumber	1.91	Too good to be true
Anemia	0.00	Useless for Prediction
angiography	0.36	Strong Predictor
clinicaldiagnose	0.33	Strong Predictor
dursmoking	0.00	Useless for Prediction
Gender	0.04	Weak Predictor
nitroglycerin	0.01	Useless for Prediction
pantoprazole	0.17	Medium Predictor
perindoprilarginine	0.13	Medium Predictor
Stroke	0.11	Medium Predictor
typeofsmoking	0.10	Weak Predictor
BUN	0.38	Strong Predictor
CH	0.03	Weak Predictor
ComputeBMI	0.04	Weak Predictor
FBS	0.38	Strong Predictor
HB	0.65	Too good to be true
HDL	0.05	Weak Predictor
Hight	0.04	Weak Predictor
LDL	0.05	Weak Predictor
PPBS	0.31	Strong Predictor
RBC	0.46	Strong Predictor
TG	0.07	Weak Predictor
WBC	0.71	Too good to be true
Weight	0.14	Medium Predictor
Work	0.34	Strong Predictor
Age	0.51	Too good to be true

diastolicHTN	0.10	Weak Predictor
systoilcHTN	0.06	Weak Predictor
timeofexcercise	0.05	Weak Predictor

The variables with low IV will not be used in the model.

The second test required for variable elimination is checking multicollinearity using Variance Inflation Factor. If the value of VIF for any variable is higher than 3, the variable is likely to be correlated with any of the other variables and to impact the model result wrongly.The original dataset was used for this operation.

Table 12: VIF of the variables

Coefficients[a]		
Model	Collinearity Statistics	
	Tolerance	VIF
Smoking	.563	1.777
age	.761	1.313
gender	.619	1.614
Weight	.415	2.410
ComputeBMI	.180	5.545
BMIGroup	**.166**	**6.026**
Hight	.523	1.914
typeofsmoking	.476	2.102
dursmoking	.763	1.310
Squitting	.479	2.087
NOciggateD	.587	1.704
Exercise	.607	1.648
timeofexcercise	.656	1.524
000000000000	.500	2.000
systoilcHTN	**.051**	**19.633**
diastolicHTN	**.054**	**18.460**
DM	.449	2.229
BGC	.542	1.844
PDM	.523	1.911
PCVD	.553	1.808
ObeseR	.759	1.318
Anemia	.757	1.321
Stroke	.733	1.365
HF	.917	1.090
Amlodipine	.911	1.097

	Enoxaparinclexame	.700	1.429
	Asprin	.353	2.834
	Atrovastatin	.586	1.706
	Cerivastatin	.476	2.100
	Fluvastatin	**.014**	**72.731**
	Pitavastatin	.871	1.147
	Pravastatin	**.008**	**126.381**
	Rosuvastatin	**.049**	**20.235**
	Clopidogreal	**.175**	**5.699**
	Pantoprazole	**.190**	**5.251**
	Nitroglycerin	**.037**	**26.764**
	perindoprilarginine	**.214**	**4.675**
	Angiography	**.238**	**4.210**
	LDL	.529	1.890
	CH	.403	2.482
	HDL	.704	1.421
	TG	.733	1.365
	FBS	.731	1.368
	PPBS	.712	1.405
	WBC	.314	3.181
	HB	.287	3.490
	RBC	.723	1.383
	BUN	.932	1.073
	Albumin	.939	1.065

a. Dependent Variable: CAD

The variables highlighted will not be used in the final model. ComputeBMI and BMIgroup are correlated (r = 0.9), removing any one of them will help. Similarly, systolicHTN and Diastolic HTN are correlated (R =0.961), hence, we can keep any one of them.

The next step is to build the logistic regression model, with CAD as target variable.

Table 13: Table of concordance

Observed		Predicted		
		CAD		Percentage
		0	1	Correct
CAD	0	449	39	92.0
	1	66	133	66.8

Overall Percentage	84.7

The overall model concordance is 84.7% which says that the model has predicted 84.7% observations correctly and is statistically good. Any concordance value >60% is considered good.

Table 14: Model Summary

the pseudo R-square values that determines the goodness of fit of the logistic model,

Model Summary			
Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	485.719*	.391	.559

The pseudo R-square value of the model is 0.559 which is moderate. The higher pseudo R-sq is, the better the model, R-square ranges from 0 to 1.

Table 15: Goodness of fit – Hosmer-Lomeshow test

This is another test of goodness of fit. The bigger the value of Sig. (Significance), the better the model is. 0< significance <1.

Hosmer and Lemeshow Test			
Step	Chi-square	df	Sig.
1	4.348	8	.824

The Hosmer-lomeshow test is satisfied. Any value greater than 0.05 indicates good fit of the model. Here value of HS is 0.824.

Table 16: Model results

Variables in the Equation						
	B	S.E.	Wald	df	Sig.	Exp(B)
Amlodipine			**4.985**	**2**	**.083**	
Amlodipine(1)	-3.693	3.004	1.511	1	.219	.025
Amlodipine(2)	**.727**	**.390**	**3.477**	**1**	**.062**	**2.069**
Asprin			.038	2	.981	
Asprin(1)	-21.092	40193.222	.000	1	1.000	.000
Asprin(2)	-.078	.398	.038	1	.845	.925
Atrovastatin			2.219	2	.330	
Atrovastatin(1)	-23.036	28175.929	.000	1	.999	.000
Atrovastatin(2)	.620	.416	2.219	1	.136	1.859

DM			2.142	2	.343	
DM(1)	22.162	40192.980	.000	1	1.000	4217398237.353
DM(2)	-.551	.377	2.142	1	.143	.576
Enoxaparinclexame			**5.028**	**2**	**.081**	
Enoxaparinclexame(1)	1.545	3.052	.256	1	.613	4.688
Enoxaparinclexame(2)	**1.067**	**.483**	**4.884**	**1**	**.027**	**2.906**
HF			**5.161**	**2**	**.076**	
HF(1)	-17.185	18045.622	.000	1	.999	.000
HF(2)	**1.296**	**.570**	**5.161**	**1**	**.023**	**3.653**
HTN			.000	2	1.000	
HTN(1)	.763	46135.515	.000	1	1.000	2.144
HTN(2)	-.002	.352	.000	1	.996	.998
ObeseR			.854	2	.652	
ObeseR(1)	1.287	1.423	.817	1	.366	3.620
ObeseR(2)	.073	.298	.060	1	.806	1.076
PCVD			.364	2	.834	
PCVD(2)	.332	.679	.240	1	.625	1.394
PCVD(3)	.396	.661	.359	1	.549	1.486
PDM			1.288	2	.525	
PDM(2)	-.075	.663	.013	1	.909	.927
PDM(3)	-.418	.671	.388	1	.533	.659
Rosuvastatin			**5.173**	**2**	**.075**	
Rosuvastatin(1)	43.765	49085.625	.000	1	.999	10155789155238700000.000
Rosuvastatin(2)	**1.735**	**.763**	**5.173**	**1**	**.023**	**5.667**
Smoking			3.595	3	.309	
Smoking(1)	1.801	2.168	.690	1	.406	6.054
Smoking(2)	.417	1.206	.120	1	.729	1.518
Smoking(3)	**.822**	**.478**	**2.956**	**1**	**.086**	**2.276**
NOciggateD	.007	.011	.410	1	.522	1.007
Squitting			2.783	2	.249	
Squitting(1)	.063	.609	.011	1	.917	1.065
Squitting(2)	.766	.532	2.069	1	.150	2.150
clinicaldiagnose			8.056	8	.428	
clinicaldiagnose(1)	-.491	.391	1.583	1	.208	.612
clinicaldiagnose(2)	.223	.505	.194	1	.659	1.249
clinicaldiagnose(3)	.616	.407	2.284	1	.131	1.851
clinicaldiagnose(4)	.010	.969	.000	1	.992	1.010
clinicaldiagnose(5)	-20.346	40192.970	.000	1	1.000	.000

clinicaldiagnose(6)	.241	1.867	.017	1	.897	1.272
clinicaldiagnose(7)	-24.280	40192.970	.000	1	1.000	.000
clinicaldiagnose(8)	21.915	40192.970	.000	1	1.000	3293921037.999
gender(1)	.383	.629	.370	1	.543	1.466
stroke			**5.031**	**2**	**.081**	
stroke(1)	-4.772	28959.524	.000	1	1.000	.008
stroke(2)	**.734**	**.327**	**5.031**	**1**	**.025**	**2.082**
typeofsmoking			4.497	5	.480	
typeofsmoking(1)	21.265	23574.137	.000	1	.999	1719404318.172
typeofsmoking(2)	20.921	23574.137	.000	1	.999	1219107448.297
typeofsmoking(3)	19.703	23574.137	.000	1	.999	360665931.618
typeofsmoking(4)	21.043	23574.137	.000	1	.999	1376256357.025
typeofsmoking(5)	22.501	23574.137	.000	1	.999	5916508799.880
Work1			20.185	18	.322	
Work1(1)	1.519	1.451	1.097	1	.295	4.570
Work1(2)	1.184	.904	1.713	1	.191	3.266
Work1(3)	.153	1.119	.019	1	.891	1.166
Work1(4)	-19.173	27739.910	.000	1	.999	.000
Work1(5)	-1.401	1.483	.893	1	.345	.246
Work1(6)	.063	.657	.009	1	.923	1.066
Work1(7)	-16.922	27675.948	.000	1	1.000	.000
Work1(8)	.175	1.310	.018	1	.894	1.191
Work1(9)	1.375	.945	2.116	1	.146	3.954
Work1(10)	1.819	.960	3.591	1	.058	6.167
Work1(11)	-.997	.790	1.594	1	.207	.369
Work1(12)	-.336	1.302	.067	1	.796	.714
Work1(13)	.720	.601	1.437	1	.231	2.054
Work1(14)	-19.140	40192.970	.000	1	1.000	.000
Work1(15)	-18.829	12462.949	.000	1	.999	.000
Work1(16)	-.921	.775	1.411	1	.235	.398
Work1(17)	1.166	1.233	.894	1	.344	3.208
Work1(18)	.508	.842	.363	1	.547	1.661
age			**11.296**	**4**	**.023**	
age(1)	-.778	.808	.926	1	.336	.459
age(2)	-.697	.570	1.498	1	.221	.498
age(3)	**-1.752**	**.621**	**7.969**	**1**	**.005**	**.173**
age(4)	.066	.316	.044	1	.834	1.068

Weight			**13.764**	**7**	**.056**	
Weight(1)	1.619	1.977	.671	1	.413	5.049
Weight(2)	.807	.661	1.491	1	.222	2.242
Weight(3)	-.242	1.246	.038	1	.846	.785
Weight(4)	.355	.823	.186	1	.666	1.426
Weight(5)	-.442	.719	.378	1	.538	.643
Weight(6)	-.629	.640	.965	1	.326	.533
Weight(7)	.732	.548	1.783	1	.182	2.079
ComputeBMI			3.188	4	.527	
ComputeBMI(1)	-1.548	1.561	.984	1	.321	.213
ComputeBMI(2)	**-1.926**	**1.141**	**2.848**	**1**	**.092**	**.146**
ComputeBMI(3)	-1.821	1.345	1.832	1	.176	.162
ComputeBMI(4)	-1.771	1.244	2.028	1	.154	.170
Hight			2.373	4	.668	
Hight(1)	-17.810	23325.933	.000	1	.999	.000
Hight(2)	-16.012	10599.581	.000	1	.999	.000
Hight(3)	2.429	1.857	1.712	1	.191	11.349
Hight(4)	2.750	1.854	2.201	1	.138	15.642
timeofexcercise			**12.369**	**5**	**.030**	
timeofexcercise(1)	**2.639**	**1.511**	**3.049**	**1**	**.081**	**14.000**
timeofexcercise(2)	-.659	.888	.551	1	.458	.517
timeofexcercise(3)	-.044	.935	.002	1	.963	.957
timeofexcercise(4)	.453	1.266	.128	1	.720	1.573
timeofexcercise(5)	-1.926	1.184	2.646	1	.104	.146
systoilcHTN			3.002	6	.809	
systoilcHTN(1)	-20.178	24758.773	.000	1	.999	.000
systoilcHTN(2)	.775	1.197	.419	1	.517	2.171
systoilcHTN(3)	1.285	1.215	1.118	1	.290	3.614
systoilcHTN(4)	.607	1.234	.242	1	.623	1.835
systoilcHTN(5)	.886	1.217	.531	1	.466	2.427
systoilcHTN(6)	.707	1.253	.319	1	.572	2.029
BGC	.001	.001	.648	1	.421	1.001
LDL			6.683	5	.245	
LDL(1)	-.449	1.368	.108	1	.742	.638
LDL(2)	-1.824	1.448	1.586	1	.208	.161
LDL(3)	-1.526	1.376	1.231	1	.267	.217
LDL(4)	-.563	1.427	.156	1	.693	.569
LDL(5)	-1.311	1.351	.943	1	.332	.269
CH			1.021	3	.796	
CH(1)	.207	1.024	.041	1	.840	1.230
CH(2)	-.290	.962	.091	1	.763	.748
CH(3)	.090	1.236	.005	1	.942	1.094
HDL			**9.137**	**3**	**.028**	

HDL(1)	**1.867**	**.687**	**7.384**	**1**	**.007**	**6.468**
HDL(2)	**1.202**	**.666**	**3.253**	**1**	**.071**	**3.326**
HDL(3)	1.275	1.347	.895	1	.344	3.578
TG			2.304	4	.680	
TG(1)	1.157	1.841	.395	1	.530	3.182
TG(2)	.562	1.799	.098	1	.755	1.755
TG(3)	.224	1.939	.013	1	.908	1.251
TG(4)	.628	1.806	.121	1	.728	1.873
FBS			**22.003**	**3**	**.000**	
FBS(1)	**1.312**	**.668**	**3.857**	**1**	**.050**	**3.712**
FBS(2)	**2.893**	**.638**	**20.575**	**1**	**.000**	**18.055**
FBS(3)	**1.493**	**.555**	**7.227**	**1**	**.007**	**4.451**
PPBS			**12.626**	**3**	**.006**	
PPBS(1)	**-2.012**	**.718**	**7.842**	**1**	**.005**	**.134**
PPBS(2)	-.246	.691	.127	1	.722	.782
PPBS(3)	**-1.193**	**.531**	**5.047**	**1**	**.025**	**.303**
WBC			3.082	3	.379	
WBC(1)	-.421	.632	.444	1	.505	.656
WBC(2)	.299	.380	.617	1	.432	1.348
WBC(3)	-1.444	1.098	1.729	1	.189	.236
HB			.728	3	.867	
HB(1)	-.118	.374	.100	1	.752	.889
HB(2)	.471	.696	.458	1	.499	1.601
HB(3)	-.395	1.060	.139	1	.710	.674
RBC			.259	3	.968	
RBC(1)	.133	.397	.113	1	.737	1.142
RBC(2)	-.173	.534	.104	1	.747	.842
RBC(3)	.207	.776	.071	1	.789	1.230
BUN			**7.875**	**3**	**.049**	
BUN(1)	-.752	.585	1.651	1	.199	.472
BUN(2)	.543	.530	1.049	1	.306	1.721
BUN(3)	-.360	.504	.511	1	.475	.697
						.000
Constant	-25.539	23574.138	.000	1	.999	

a. Variable(s) entered on step 1: Amlodipine, Asprin, Atrovastatin, DM, Enoxaparinclexame, HF, HTN, ObeseR, PCVD, PDM, Rosuvastatin, Smoking, NOciggateD, Squitting, clinicaldiagnose, gender, stroke, typeofsmoking, Work1, age, Weight, ComputeBMI, Hight, timeofexcercise, systoilcHTN, BGC, LDL, CH, HDL, TG, FBS, PPBS, WBC, HB, RBC, BUN.
b. This is the model result. B is the coefficient of the variable. SE is the standard error of the variable. Wald is the chi-sq value that determines the significance of the variable, higher chi-sq means more significant variable. df is the degrees of freedom of that variable. Sig. is the p-value, the lower the p- value, the higher the significance is. EXP(B) is the impact of the variable on the target

The variables that has p-value<0.1 are statistically significant at 10% level of significance.

Summary of Results and Conclusions

The variables that have significantly impacted CAD are:

1. Amlodipine: Those who have taken Amlodipine have a higher chance to develop CAD
2. Enoxaparinclexame: Those who have taken Enoxaparinclexamehave a higher chance to develop CAD.
3. HF: Those who have reported 'yes' to HF has a higher chance to develop CAD.
4. Rosuvastatin: Those who have taken Rosuvastatin have a higher chance to develop CAD.
5. Smoking: Smokers have a higher risk of developing CAD
6. Stroke: people had a history of stroke have a higher chance to develop CAD
7. Age: People between 26 – 40 years are in low risk zone of CAD
8. Weight: Weight overall has come out to be a significant factor behind CAD, but no particular age-group has been identified as more/less risk prone.
9. BMI: People who have perfect weight are in much lesser risk than underweight or overweight people.
10. Time of exercise: People who exercised with time of exercise>=180 are in much higher risk than others
11. HDL: Unlike overall cholesterol level and bad cholesterol LDL, good cholesterol HDL is a significant factor of CAD. People having lower HDL, <59 mg are in high risk of CAD.
12. FBS: whoever having above normal FBS are in high risk zone
13. PPBS: People having slightly higher measurement on PPBS are in higher risk than those who have very high or normal measurements
14. BUN: BUN overall is a cause of CAD but no significant measurement group is identified as highly risky.

Additional analysis:

Additionally, the following cross-tabs were created to study the distribution of CAD against various variables.

Table 17: CAD vs VN Number

		VN Number					
		0	1	2	3	4	Grand Total
CAD	0	488					488
	1	23	95	54	18	9	199
	(blank)	1					1
	Grand Total	512	95	54	18	9	688

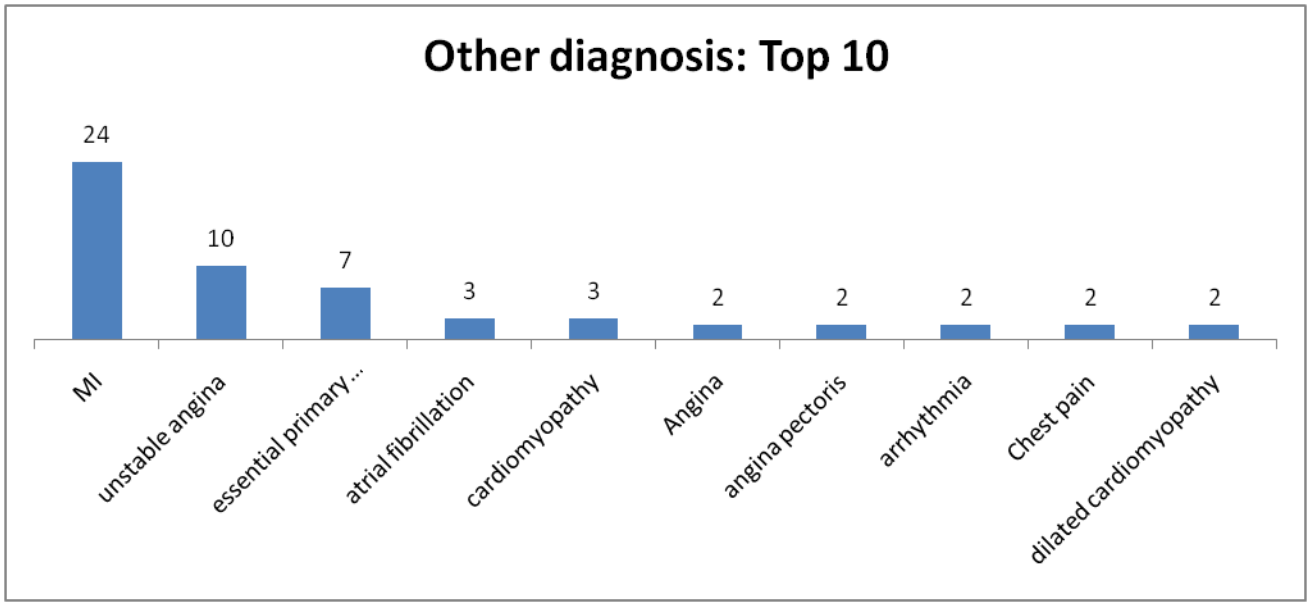
Table 19: CAD vs CABG

		CABG				
		0	1	2	Grand Total	
CAD	0	66	14	408		488
	1	29	10	160		199
	(blank)			1		1
	Grand Total	512	95	54		18

Table 20: Top 10 other diagnosis when CAD is present

Other Diagnosis	Count
MI	24
unstable angina	10
essential primary hypertension	7
atrial fibrillation	3
cardiomyopathy	3
Angina	2
angina pectoris	2
arrhythmia	2
Chest pain	2
dilated cardiomyopathy	2

Figure 8: Top 10 other diagnosis when CAD is present



Appendix III: ANOVA test results for arrhythmia data set

Features	Anova Value
Age	0.98
Sex	0.29
Height	0.45
Weight	0.45
QRS duration	0.37
P-R interval	0.42
Q-T interval	0.84
T interval	0.53
P interval	0.08
QRS	0.96
T	0.32
P	0.22
QRST	0.93
J	0.81
Heart	0.50
DI_Q wave	0.21
DI_R wave	0.78
DI_S wave	0.55
DI_R' wave,	0.63
DI_S' wave	0.56
DI_Number of intrinsic deflections	0.31
DI_Existence of ragged R wave	0.61
DI_Existence of diphasic derivation of R wave	0.01
DI_Existence of ragged P wave	0.66
DI_Existence of diphasic derivation of P wavel	0.59
DI_Existence of ragged T wave	0.39
DI_Existence of diphasic derivation of T wave	0.67
DII_Q wave	0.58
DII_R wave	0.59
DII_S wave	0.14
DII_R' wave,	0.78
DII_S' wave	0.78
DII_Number of intrinsic deflections	0.99
DII_Existence of ragged R wave	0.62
DII_Existence of diphasic derivation of R wave	0.61
DII_Existence of ragged P wave	0.95
DII_Existence of diphasic derivation of P wavel	0.46
DII_Existence of ragged T wave	0.33

DII_Existence of diphasic derivation of T wave	0.02
DIII_Q wave	0.89
DIII_R wave	0.04
DIII_S wave	0.26
DIII_R' wave,	0.58
DIII_S' wave	0.97
DIII_Number of intrinsic deflections	0.92
DIII_Existence of ragged R wave	0.16
DIII_Existence of diphasic derivation of R wave	0.10
DIII_Existence of ragged P wave	0.92
DIII_Existence of diphasic derivation of P wavel	0.89
DIII_Existence of ragged T wave	0.50
DIII_Existence of diphasic derivation of T wave	0.24
AVR_Q wave	0.04
AVR_R wave	0.69
AVR_S wave	0.81
AVR_R' wave,	0.40
AVR_S' wave	0.56
AVR_Number of intrinsic deflections	0.95
AVR_Existence of ragged R wave	0.90
AVR_Existence of diphasic derivation of R wave	0.84
AVR_Existence of ragged P wave	0.11
AVR_Existence of diphasic derivation of P wavel	0.38
AVR_Existence of ragged T wave	0.77
AVR_Existence of diphasic derivation of T wave	0.94
AVL_Q wave	0.23
AVL_R wave	0.71
AVL_S wave	0.80
AVL_R' wave,	0.74
AVL_S' wave	0.64
AVL_Number of intrinsic deflections	0.88
AVL_Existence of ragged R wave	0.20
AVL_Existence of diphasic derivation of R wave	0.20
AVL_Existence of ragged P wave	0.90
AVL_Existence of diphasic derivation of P wavel	0.59
AVL_Existence of ragged T wave	0.28
AVL_Existence of diphasic derivation of T wave	0.55

AVF_Q wave	0.40
AVF_R wave	0.35
AVF_S wave	0.34
AVF_R' wave,	0.31
AVF_S' wave	0.98
AVF_Number of intrinsic deflections	0.94
AVF_Existence of ragged R wave	0.94
AVF_Existence of diphasic derivation of R wave	0.77
AVF_Existence of ragged P wave	0.92
AVF_Existence of diphasic derivation of P wavel	0.93
AVF_Existence of ragged T wave	0.82
AVF_Existence of diphasic derivation of T wave	0.61
V1_Q wave	0.41
V1_R wave	0.80
V1_S wave	0.24
V1_R' wave,	0.67
V1_S' wave	0.14
V1_Number of intrinsic deflections	0.60
V1_Existence of ragged R wave	0.82
V1_Existence of diphasic derivation of R wave	0.57
V1_Existence of ragged P wave	0.37
V1_Existence of diphasic derivation of P wavel	0.63
V1_Existence of ragged T wave	0.27
V1_Existence of diphasic derivation of T wave	0.22
V2_Q wave	0.05
V2_R wave	0.16
V2_S wave	0.05
V2_R' wave,	0.19
V2_S' wave	0.32
V2_Number of intrinsic deflections	0.43
V2_Existence of ragged R wave	0.09
V2_Existence of diphasic derivation of R wave	0.61
V2_Existence of ragged P wave	0.51
V2_Existence of diphasic derivation of P wavel	0.23
V2_Existence of ragged T wave	0.68
V2_Existence of diphasic derivation of T wave	0.96
V3_Q wave	0.19
V3_R wave	0.93
V3_S wave	0.15
V3_R' wave,	0.14

V3_S' wave	0.20
V3_Number of intrinsic deflections	0.44
V3_Existence of ragged R wave	0.14
V3_Existence of diphasic derivation of R wave	0.75
V3_Existence of ragged P wave	0.44
V3_Existence of diphasic derivation of P wavel	0.64
V3_Existence of ragged T wave	0.40
V3_Existence of diphasic derivation of T wave	0.61
V4_Q wave	0.30
V4_R wave	0.77
V4_S wave	0.36
V4_R' wave,	0.30
V4_S' wave	0.60
V4_Number of intrinsic deflections	0.54
V4_Existence of ragged R wave	0.92
V4_Existence of diphasic derivation of R wave	0.27
V4_Existence of ragged P wave	0.20
V4_Existence of diphasic derivation of P wavel	0.20
V4_Existence of ragged T wave	0.70
V4_Existence of diphasic derivation of T wave	0.97
V5_Q wave	0.17
V5_R wave	0.27
V5_S wave	0.06
V5_R' wave,	0.18
V5_S' wave	0.29
V5_Number of intrinsic deflections	0.88
V5_Existence of ragged R wave	0.32
V5_Existence of diphasic derivation of R wave	0.42
V5_Existence of ragged P wave	0.41
V5_Existence of diphasic derivation of P wavel	0.60
V5_Existence of ragged T wave	0.74
V5_Existence of diphasic derivation of T wave	0.63
V6_Q wave	0.30
V6_R wave	0.72
V6_S wave	0.49
V6_R' wave,	0.01
V6_S' wave	0.70
V6_Number of intrinsic deflections	0.40
V6_Existence of ragged R wave	0.77
V6_Existence of diphasic derivation of R wave	0.28
V6_Existence of ragged P wave	0.95
V6_Existence of diphasic derivation of P wavel	0.74

V6_Existence of ragged T wave	0.37
V6_Existence of diphasic derivation of T wave	0.94
DI_JJ wave,	0.05
DI_Q wave,	0.49
DI_R wave,	0.53
DI_S wave,	0.40
DI_R' wave,	0.44
DI_S' wave,	0.36
DI_P wave,	0.73
DI_T wave,	0.08
DI_QRSA	0.99
DI_QRSTA	0.08
DII_JJ wave,	0.88
DII_Q wave,	0.93
DII_R wave,	0.51
DII_S wave,	0.99
DII_R' wave,	0.45
DII_S' wave,	0.77
DII_P wave,	0.88
DII_T wave,	0.99
DII_QRSA	0.86
DII_QRSTA	0.03
DIII_JJ wave,	0.39
DIII_Q wave,	0.62
DIII_R wave,	0.45
DIII_S wave,	0.02
DIII_R' wave,	0.18
DIII_S' wave,	0.23
DIII_P wave,	0.22
DIII_T wave,	0.38
DIII_QRSA	0.13
DIII_QRSTA	0.14
AVR_JJ wave,	0.92
AVR_Q wave,	0.74
AVR_R wave,	0.57
AVR_S wave,	0.54
AVR_R' wave,	0.87
AVR_S' wave,	0.36
AVR_P wave,	0.63
AVR_T wave,	0.20
AVR_QRSA	0.58
AVR_QRSTA	0.63

AVL_JJ wave,	0.77
AVL_Q wave,	0.42
AVL_R wave,	0.49
AVL_S wave,	0.58
AVL_R' wave,	0.21
AVL_S' wave,	0.93
AVL_P wave,	0.36
AVL_T wave,	0.95
AVL_QRSA	0.48
AVL_QRSTA	0.12
AVF_JJ wave,	0.31
AVF_Q wave,	0.32
AVF_R wave,	0.42
AVF_S wave,	0.78
AVF_R' wave,	0.02
AVF_S' wave,	0.03
AVF_P wave,	0.30
AVF_T wave,	0.32
AVF_QRSA	0.86
AVF_QRSTA	0.82
V1_JJ wave,	0.15
V1_Q wave,	0.47
V1_R wave,	0.01
V1_S wave,	0.88
V1_R' wave,	0.97
V1_S' wave,	0.50
V1_P wave,	0.21
V1_T wave,	0.33
V1_QRSA	0.37
V1_QRSTA	0.41
V2_JJ wave,	0.34
V2_Q wave,	0.85
V2_R wave,	0.67
V2_S wave,	0.33
V2_R' wave,	0.18
V2_S' wave,	0.15
V2_P wave,	0.31
V2_T wave,	0.02
V2_QRSA	0.22
V2_QRSTA	0.70
V3_JJ wave,	0.50
V3_Q wave,	0.60

V3_R wave,	0.62
V3_S wave,	0.15
V3_R' wave,	0.76
V3_S' wave,	0.46
V3_P wave,	0.14
V3_T wave,	0.05
V3_QRSA	0.79
V3_QRSTA	0.10
V4_JJ wave,	0.51
V4_Q wave,	0.12
V4_R wave,	0.13
V4_S wave,	0.05
V4_R' wave,	0.08
V4_S' wave,	0.57
V4_P wave,	0.05
V4_T wave,	0.79
V4_QRSA	0.25
V4_QRSTA	0.30
V5_JJ wave,	0.37
V5_Q wave,	0.22
V5_R wave,	0.95
V5_S wave,	0.28
V5_R' wave,	0.08
V5_S' wave,	0.07
V5_P wave,	0.11
V5_T wave,	0.48
V5_QRSA	0.99
V5_QRSTA	0.74
V6_JJ wave,	0.61
V6_Q wave,	0.03
V6_R wave,	0.58
V6_S wave,	0.01
V6_R' wave,	0.20
V6_S' wave,	0.94
V6_P wave,	0.11
V6_T wave,	0.53
V6_QRSA	0.97
V6_QRSTA	0.55

Appendix IV: Performance Summary for minute-class-based model of DNN-DT scheme for each file of Apnea data set

File Name	Act. Class	Pred. Class	Acc.	Prec.	Rec.	F1	Confusion Matrix
a01	A	A	97.55%	98%	98 %	98%	A N A 469 8 N 1 11 ∑ 470 19 489
a02	A	A	82.78%	85%	86 %	84%	A N A 408 64 N 12 44 ∑ 420 108 528
a03	A	A	90.18%	92 %	92%	92 %	A N A 236 31 N 10 242 ∑ 246 273 519
a04	A	A	98.38%	99%	99%	99%	A N A 452 3 N 1 36 453 39 492
a05	A	A	81.29%	83%	83	83	A N A 242 43 N 34 135 276 178 454
a06	A	A	71.77%	79%	78%	77%	A N A 116 23 N 90 281 206 304 492
a07	A	A	80.24%	82%	82%	82%	A N A 293 62 N 29 127 322 189 511
a08	A	A	81.62%	84%	84%	84%	A N A 146 35 N 43 277 189 312 501
a09	A	A	83.05%	85%	85%	85%	A N A 357 49 N 24 65 381 114 495
a10	A	B	80.65%	84%	85%	84%	A N A 44 20 N 56 397 100 417 517
a11	A	A	78.98%	82%	81%	81%	A N A 159 26 N 63 218

								222	**244**	**466**
a12	A	A	94.12%	94%	95%	94%		**A**	**N**	
							A	529	25	
							N	5	18	
								534	**43**	**577**
a13	A	A	88.27%	91%	91%	91%		**A**	**N**	
							A	222	25	
							N	22	226	
								244	**251**	**495**
a14	A	A	87.07%	90%	90%	90%		**A**	**N**	
							A	367	36	
							N	16	90	
								283	**126**	**509**
a15	A	A	81.15%	84%	85%	84%		**A**	**N**	
							A	336	46	
							N	32	96	
								368	**142**	**520**
a16	A	A	84.86%	88%	88%	88%		**A**	**N**	∑
							A	297	35	
							N	23	127	
								320	**162**	**482**
a17	A	A	86.35%	89%	89%	89%		**A**	**N**	
							A	138	33	
							N	20	292	
								158	**325**	**483**
a18	A	A	95.70%	97%	97%	97%		**A**	**N**	
							A	433	10	
							N	5	41	
								38	**51**	**489**
a19	A	A	91.05%	93%	93%	93%		**A**	**N**	
							A	195	25	
							N	10	272	
								205	**297**	**502**
a20	A	A	87.21%	89%	89%	89%		**A**	**N**	
							A	298	39	
							N	17	156	
								315	**195**	**510**
b01	B	C	96.11%	92%	96%	94%		**A**	**N**	
							A	0	0	
							N	19	468	
								19	**468**	**487**
b02	B	B	87.44%	90%	90%	90%		**A**	**N**	
							A	70	31	
							N	23	393	
								93	**424**	**517**
b03	B	B	93.89%	96%	96%	96%		**A**	**N**	
							A	62	8	
							N	11	360	
								73	**368**	**441**
b04	B	C	97.65%	95%	98%	96%		**A**	**N**	
							A	0	0	
							N	10	416	

								10	**416**	**426**
b05	B	B	90.51%	92%	92%	91%		**A**	**N**	
							A	25	2	
							N	32	374	
								57	**376**	**433**
c01	C	C	100%	100%	100%	100%		**A**	**N**	
							A	0	0	
							N	0	478	
								0	**478**	**478**
c02	C	C	99.80%	100%	100%	100%		**A**	**N**	
							A	0	0	
							N	1	493	
								1	**493**	**494**
c03	C	C	100%	100%	100%	100%		**A**	**N**	∑
							A	0	0	
							N	0	454	
								0	**454**	**454**
c04	C	C	100%	100%	100%	100%		**A**	**N**	
							A	0	0	**0**
							N	0	476	
									476	**476**
c05	C	C	99.37	100%	100%	100%		**A**	**N**	
							A	2	0	
							N	1	463	
								3	**463**	**466**
c06	C	C	99.79	100%	100%	100%		**A**	**N**	
							A	0	0	
							N	1	467	
								1	**467**	**468**
c07	C	C	99.09%	98%	99%	99%		**A**	**N**	
							A	0	0	
							N	4	425	
								4	**425**	**429**
c08	C	C	100%	100%	100%	100%		**A**	**N**	
							A	0	0	
							N	0	513	
								0	**513**	**513**
c09	C	C	99.57%	99%	100%	99%		**A**	**N**	
							A	0	0	**2**
							N	2	455	
								2	**455**	**457**
c10	C	C	99.77%	100%	100%	100%		**A**	**N**	
							A	0	0	
							N	1	424	
								1	**424**	**425**

Appendix V: results of multi-layer kernel RBF model

1. **loadArythmiaData**

Input:	N – number of required data points
Description:	Loads the data from the built in arrhythmia dataset. We consider only 2 classes 1 – has arrhythmia, 0 – has no arrhythmia. The data are loaded uniformly from each class if possible: N/2 points from the apnea class and N/2 points from the no-apnea class. If uniform division is not possible the additional points are taken from the dataset with larger number of points
Output:	input – matrix with N rows where each row represents a single entry of the input to the classifier, each column represent a single feature. For the arrhythmia dataset we have 200 features. output - the class labels corresponding to the data entries from input. 1 – means arrhythmia, 0 – no arrhythmia

2. **loadHospitalData**

Input:	N – number of required data points
Description:	Loads the data from the provided dataset MC_20. Points are uniformly sampled from each of possible 2 possible classes if possible. If that's not possible the additional points are taken from the CAD=1 class.
Output:	input – matrix with N rows where each row represents a single entry of the input to the classifier, each column represent a single feature. output - the class labels corresponding to the data entries from input (CAD=0, CAD =1).

3. shuffleData

Input:	input – data entries for classifier output – class labels corresponding to the data entries
Description:	Randomly shuffles input data and correspondingly shuffles output class labels. So we can be sure that data for training and validation will be selected uniformly from both classes.
Output:	input – permutated data entries output – corresponding class labels for permutated data entries

4. kFoldValidation

Input:	input – array of the data we need to split k – number of folds for k-fold validation
Description:	Splits data into k groups equal groups, each time one of the groups is used for validation, the others – for training
Output:	indV – cell array, each cell contains indexes in the original data elements which will be used for validation indT – cell array, each cell contains indexes in the original data elements which will be used for training name – string, name of the validation algorithm

5. randomValidation

Input:	input – array of the data we need to split rate – proportion of the validation data to all data (value from 0 to 1) times – number of groups/resamplings: different validation sets
Description:	Randomly selects a predefined proportion of data which will be used for

	validation, the rest of the data will be used for training. The operation is repeted selected number of times
Output:	indV – cell array, each cell contains indexes in the original data elements which will be used for validation indT – cell array, each cell contains indexes in the original data elements which will be used for training name – string, name of the validation algorithm

6. generalValidation

Input:	input – input data entries for classification output – output labels corresponding to the data entries indT – cell array with indices corresponding to training data for each fold indV – cell array with indices corresponding to validation data for each fold trainFunc – the pointer to the function (classifier = trainFunc(input,output)) which will perform training on the data according to selected algorithm and will return the object which can be used for prediction predictFunc – the pointer to the function (predictedLabels = predictFunc(classifier, input)) which can classify data based on the information learned previously during training fileName – name of the file to save the results fileFolder – name of the folder where to save results titlePlot – title for the plot which is the result of the function
Description:	According to predefined partitions of the data performs training and
Output:	mseV – mean squared error for all validation data trainTime – total time required for training for all the folds testTime – total time required for validation for all the folds

7. **oversample**

Input:	inp – training data outp – labels corresponding to the training data
Description:	The training dataset is supplied with additional repetitive points in order to have equal number of data entries in all the classes of the dataset
Output:	inp – oversampled training set outp – corresponding labels for the oversampled dataset

8. **getHRange**

Input:	inputX – training dataset
Description:	Selects maximum value for the bandwidth in such way that all the points will have at least one neighbor within that bandwidth Selects minimum value for the bandwidth in such way that all the points will form a connected graph. The 2 points in the graph will be connected if the value of the kernel with the selected is far from 0
Output:	hmin – minimum value in the range for kernel bandwidth hmax – maximum value in the range for kernel bandwidth

9. **connectivityComputations**

Input:	adjacencyMatrix – nxn matrix for the graph with n vertices, element (i,j) is equal to 1 if vertices i and j are connected in the graph
Description:	Determines the connectivity components of the graph
Output:	vertices – cell array, each cell contains all the vertices inside single graph component

10. perComponentDistance

Input:	components – cell array, each cell contains indices of the graph vertices corresponding to a single connectivity component inputX – training data, each point represents a single graph vertex
Description:	Determines the minimum distance between points of all components
Output:	cDist – array of pairwise per component distances

11. mIFeatureSelection

Input:	trainX – data entries from the training set trainY – labels corresponding to the data entries
Description:	Computes the mutual information (scores) for the values of the features with the labels of the class. The features with maximum scores are selected to be used in the next layers.
Output:	selection – indices corresponding to the selected features from the training set

12. normalizeInput

Input:	input – training data, each row is a single entry
Description:	Performs data centering and normalization in such way that mean of the data is equal to 0 and standard deviation to 1.
Output:	input – transformed data m0 – shift parameter used for centering s0 – normalization coefficient

13. normalizeInputWithParameters

Input:	input – training data, each row is a single entry

	m0 – shift parameter used for centering during training s0 – normalization coefficient obtained during training
Description:	Performs data centering and normalization based on the shift and normalization parameters obtained during training phase. It allows to have the same transformation for training and test data
Output:	inputN – transformed data

14. predictWithMultilayerMachinePCA

Input:	classifier – object containing all the parameters used to predict the labels: normalization parameters, selected features and projection vectors on each layer, trained knn prediction model testX – input dataset for which we need to make predictions
Description:	Performs the nonlinear transformation on each layer: normalization, feature selection, projection on the kernel principal components. The final transformation from the last layer is used to determine the model for knn classification
Output:	result – predicted class labels for the input data

15. trainWithMultilayerMachinePCA

Input:	inp – training data outp – training labels n – number of layers numberOfNeighbors – number of neighbors for knn
Description:	Determines the optimal transformation of the input data using kernel principal component analysis. And uses the final projection to train knn classifier

Output:	classifier – object containing all the parameters used to predict the labels: normalization parameters, selected features and projection vectors on each layer, trained knn prediction model

16. predictWithMultilayerMachineSupervised

Input:	classifier – object containing all the parameters used to predict the labels: normalization parameters, selected features and projection vectors on each layer, trained knn prediction model testX – input dataset for which we need to make predictions
Description:	Performs the nonlinear transformation on each layer: normalization, feature selection, projection on the components selected with the supervised kernel regression. The final transformation from the last layer is used to determine the model for KNN classification
Output:	result – predicted class labels for the input data

17. trainWithMultilayerMachinesupervised

Input:	inp – training data outp – training labels n – number of layers numberOfNeighbors – number of neighbors for knn
Description:	Determines the optimal transformation of the input data using components obtained with supervised kernel regression. And uses the final projection to train knn classifier
Output:	classifier – object containing all the parameters used to predict the labels: normalization parameters, selected features and projection vectors on each

	layer, trained knn prediction model

Table 1: Results Summary of kPCA algorithm at CAD dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	57.41	55.23	70.06	71.37	71.51
MSE,%	42.59	44.77	29.94	28.63	28.49
Sensitivity	57.26	53.37	94.99	100	100
Specificity	57.79	59.79	4.02	1	1.5
Cohen's kappa	0.31	0.30	0.04	0	0
Training time, s	578.61	1122.4	2230.9	3625.6	4757.7
Validation time,s	1444.2	13127	21478	22430	22502

Table 2: Results Summary of Supervised Regression algorithm at CAD dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	57.85	62.35	67.73	62.79	63.95
MSE,%	42.15	37.65	32.27	37.21	36.05
Sensitivity	57.91	72.34	89.97	76.55	77.15
Specificity	54.77	34.17	8.54	25.12	27.63
Cohen's kappa	0.30	0.23	0.07	0.17	0.20
Training time, s	1263.9	2892.1	5837.6	6062.4	8999
Validation time, s	17.56	23.62	29.25	31.72	33.46

Table 3: Results Summary of Unsupervised Latent Regression algorithm at CAD dataset

	1	2	3	4	5
Accuracy,%	57.85	62.35	67.73	62.79	63.95
MSE,%	42.15	37.65	32.27	37.21	36.05
Sensitivity	57.91	72.34	89.97	76.55	77.15
Specificity	54.77	34.17	8.54	25.12	27.64
Cohen's kappa	0.26	0.31	0.26	0.19	0.18
Training time, s	1122.4	2230.9	4671	5311	6621.2
Validation time, s	32	47.1	58.02	61.76	68.17

Table 4: Results Summary of Unsupervised Latent Regression with projection algorithm at CAD dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	61.30	61.30	61.30	61.30	61.30
MSE,%	38.69	38.69	38.69	38.69	38.69
Sensitivity	71.83	71.83	71.83	71.83	71.83
Specificity	35.5	35.5	35.5	35.5	35.5
Cohen's kappa	0.39	0.39	0.39	0.39	0.39
Training time, s	192.40	486.44	1318.25	3719.59	9958.89
Validation time, s	11.64	11.39	11.03	11.15	11.13

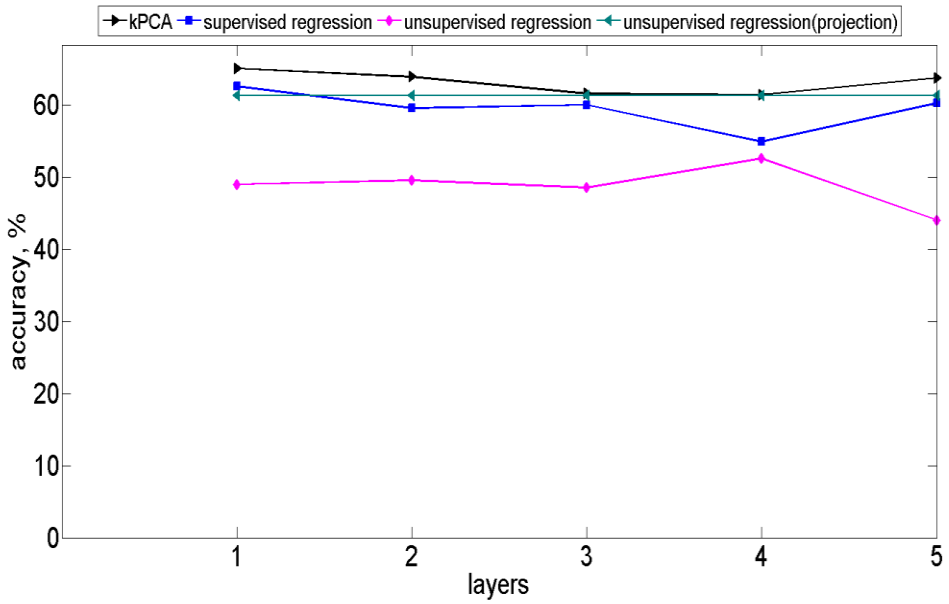


Figure 1: Accuracy vs. number of layers when applying the four algorithms for CAD dataset

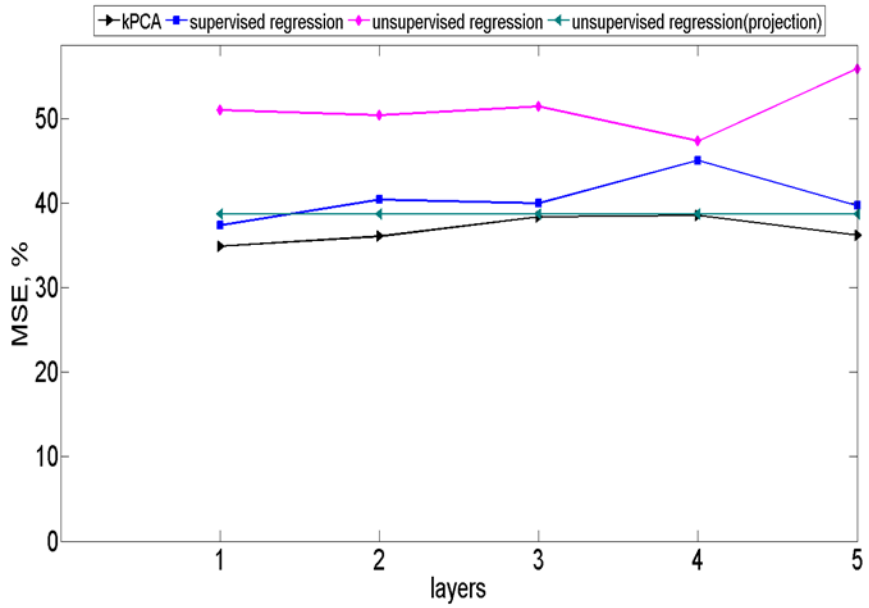


Figure 2: MSE vs. number of layers when applying the four algorithms for CAD dataset

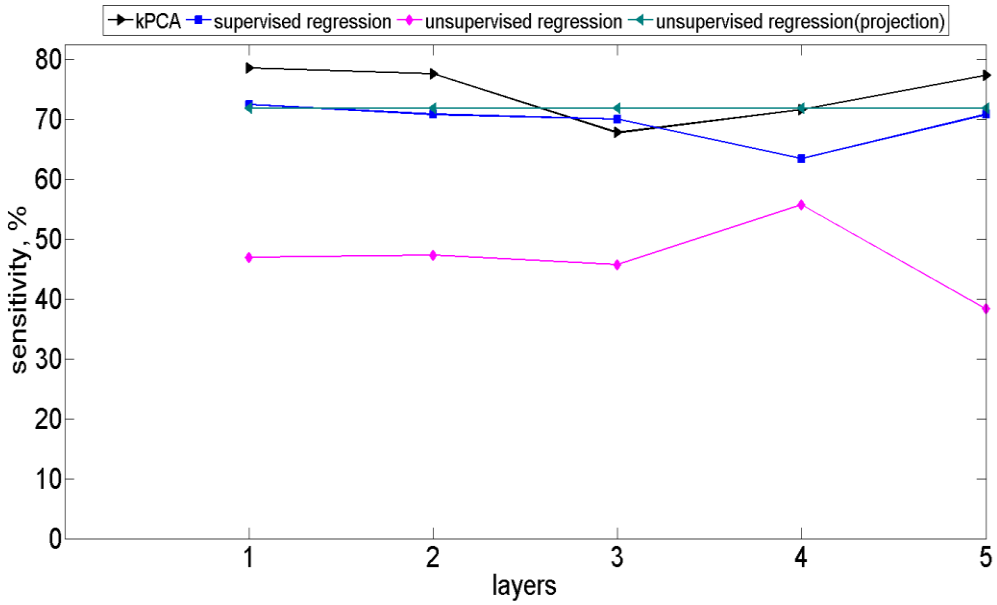


Figure 3: Sensitivity vs. number of layers when applying the four algorithms for CAD dataset

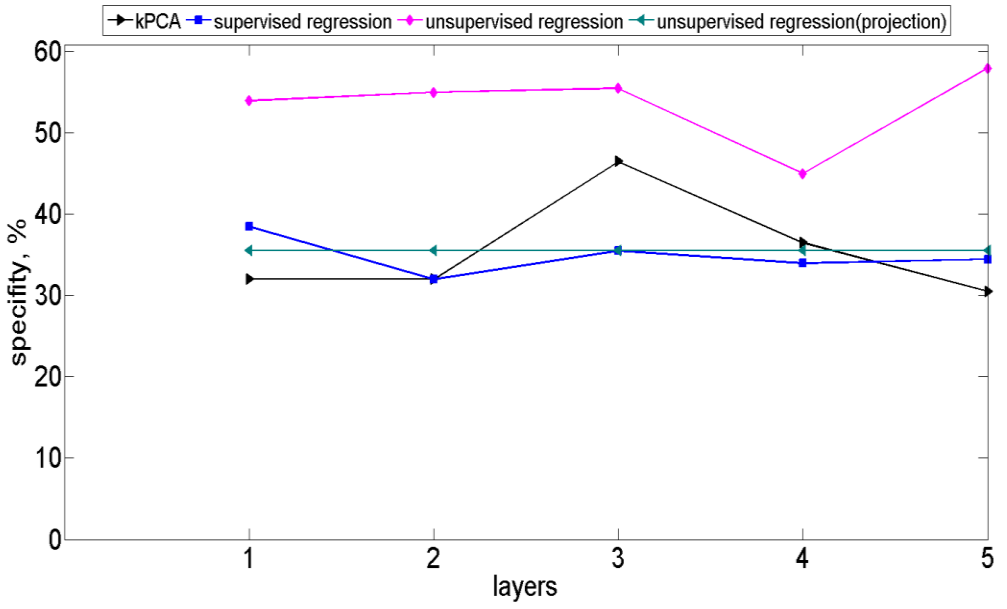


Figure 4: Specificity vs. number of layers when applying the four algorithms for CAD dataset

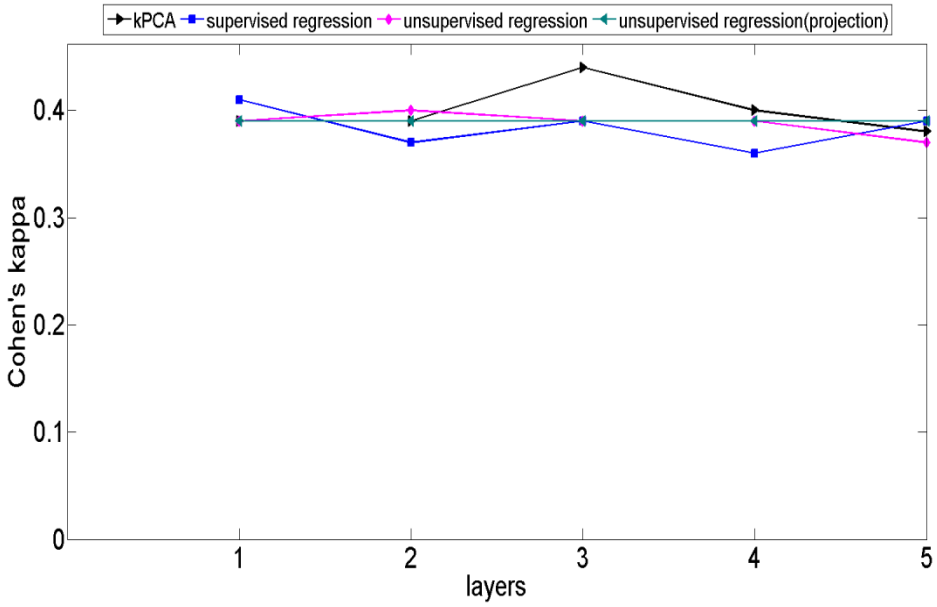


Figure 5: Cohen's Kappa vs. number of layers when applying the four algorithms for CAD dataset

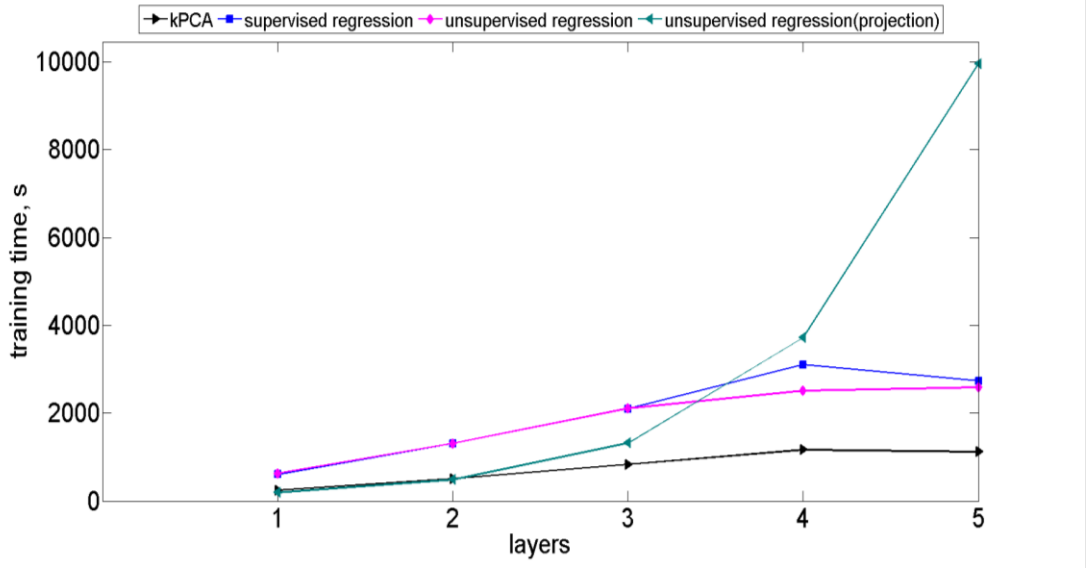


Figure 6: Training time (sec) vs. number of layers when applying the four algorithms for CAD dataset

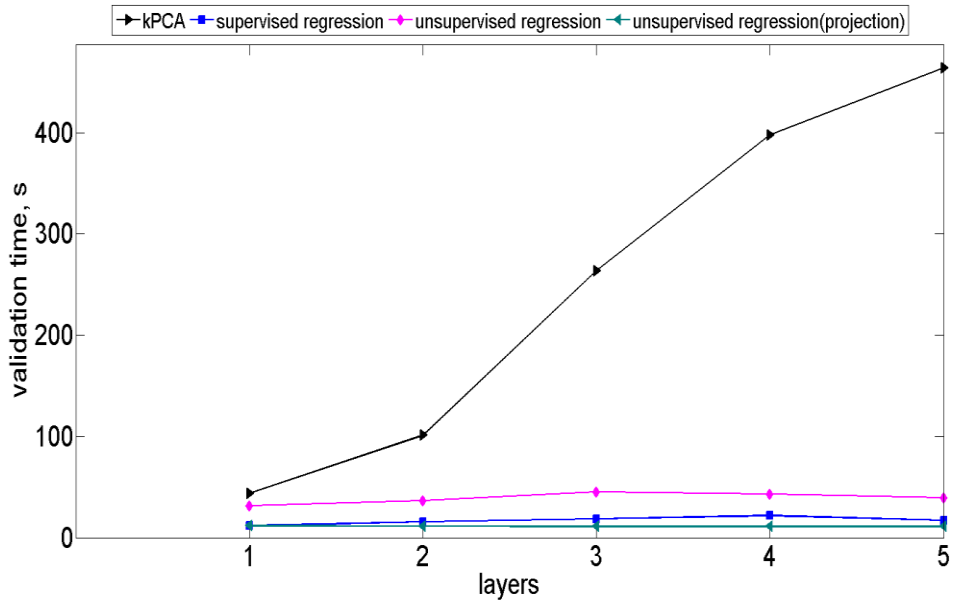


Figure 7: Validation time (sec) vs. number of layers when applying the four algorithms for CAD dataset

Table 5: Summary of Confusion Matrix for kPCA algorithm for hospital dataset

Number of Layers	Confusion Matrix			
1		**C**	**N**	**∑**
	C	280	84	**364**
	N	209	115	**324**
	∑	**489**	**199**	**688**
2		**C**	**N**	**∑**
	C	261	80	**341**
	N	228	119	**347**
	∑	**489**	**199**	**688**
3		**C**	**N**	**∑**
	C	474	191	**665**
	N	15	8	**23**
	∑	**489**	**199**	**688**
4		**C**	**N**	**∑**
	C	489	197	**686**
	N	0	2	**2**
	∑	**489**	**199**	**688**
5		**C**	**N**	**∑**
	C	489	196	**685**
	N	0	3	**3**
	∑	**489**	**199**	**688**

Table 6: Summary of Confusion Matrix for Supervised Regression algorithm for CAD dataset

Number of Layers	Confusion Matrix			
1		**C**	**N**	**∑**
	C	289	90	**379**
	N	200	109	**309**
	∑	**489**	**199**	**688**
2		**C**	**N**	**∑**
	C	361	131	**379**
	N	128	68	**309**
	∑	**489**	**199**	**688**
3		**C**	**N**	**∑**
	C	449	182	**631**
	N	40	17	**57**
	∑	**489**	**199**	**688**
4		**C**	**N**	**∑**
	C	382	149	**531**
	N	107	50	**157**
	∑	**489**	**199**	**688**
5		**C**	**N**	**∑**
	C	385	144	**529**
	N	104	55	**159**
	∑	**489**	**199**	**688**

Table 7: Summary of Confusion Matrix for Unsupervised Latent Regression algorithm for CAD dataset

Number of Layers	Confusion Matrix			
1		**C**	**N**	**∑**
	C	122	50	**172**
	N	367	149	**516**
	∑	**489**	**199**	**688**
2		**C**	**N**	**∑**
	C	241	70	**311**
	N	348	129	**477**
	∑	**489**	**199**	**688**
3		**C**	**N**	**∑**
	C	205	75	**280**
	N	284	124	**408**
	∑	**489**	**199**	**688**
4		**C**	**N**	**∑**
	C	289	119	**408**
	N	200	80	**280**
	∑	**489**	**199**	**688**

5		**C**	**N**	**∑**
	C	346	138	**484**
	N	143	61	**204**
	∑	**489**	**199**	**688**

Table 8: Summary of Confusion Matrix for Unsupervised Latent Regression with projection algorithm for CAD dataset

Number of Layers	**Confusion Matrix**			
1		**C**	**N**	**∑**
	C	351	128	**479**
	N	138	71	**209**
	∑	**489**	**199**	**688**
2		**C**	**N**	**∑**
	C	351	128	**479**
	N	138	71	**209**
	∑	**489**	**199**	**688**
3		**C**	**N**	**∑**
	C	351	128	**479**
	N	138	71	**209**
	∑	**489**	**199**	**688**
4		**C**	**N**	**∑**
	C	351	128	**479**
	N	138	71	**209**
	∑	**489**	**199**	**688**
5		**C**	**N**	**∑**
	C	351	128	**479**
	N	138	71	**209**
	∑	**489**	**199**	**688**

Table 9: Results Summary of kPCA algorithm at Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	65.34	64.41	66.04	64.18	65.81
MSE,%	34.65	35.58	33.95	35.81	34.18
Sensitivity	78.77	78.77	78.36	81.22	82.44
Specificity	47.56	45.40	49.72	41.62	43.78
Cohen’s kappa	0.5	0.48	0.51	0.46	0.48

Training time, s	161.85	290.41	439.78	617.78	716.48
Validation time,s	647.15	659.92	744.39	969.24	1014.18

Table 10: Results Summary of Supervised Regression algorithm at Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	69.53	65.81	61.62	63.48	59.06
MSE,%	30.46	34.18	38.37	36.51	40.93
Sensitivity	73.87	70.20	64.08	66.12	63.26
Specificity	63.78	60	58.37	60	53.51
Cohen’s kappa	0.59	0.55	0.52	0.53	0.48
Training time, s	447.63	700.93	1040.29	1348.98	1605.86
Validation time, s	6.58	7.27	9.33	9.43	10.18

Table 11: Results Summary of Unsupervised Latent Regression algorithm at Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	57.67	56.27	56.27	53.95	54.65
MSE,%	42.32	43.72	43.72	46.04	45.34
Sensitivity	100	90.61	90.61	71.02	80.40
Specificity	1.62	10.81	10.81	31.35	20.54
Cohen’s kappa	0.02	0.18	0.18	0.35	0.28
Training time, s	329.044	635.73	968.47	1189.06	1487.87
Validation time, s	15.96	18.15	20.52	20.75	22.81

Table 12: Results Summary of Unsupervised Latent Regression with projection algorithm at Arrhythmia dataset

Layers No. / Metric	1	2	3	4	5

Accuracy,%	68.0	68.0	68.0	68.0	68.0
MSE,%	32.0	32.0	32.0	32.0	32.0
Sensitivity	72.8	72.8	72.8	72.8	72.8
Specificity	61.6	61.6	61.6	61.6	61.6
Cohen's kappa	0.57	0.57	0.57	0.57	0.57
Training time, s	182	431	744	1200	1520
Validation time, s	15.4	11.1	74.9	54.7	47.4

Figure 8 below presents the accuracy values for the four algorithms according to the variation in the number of layers.

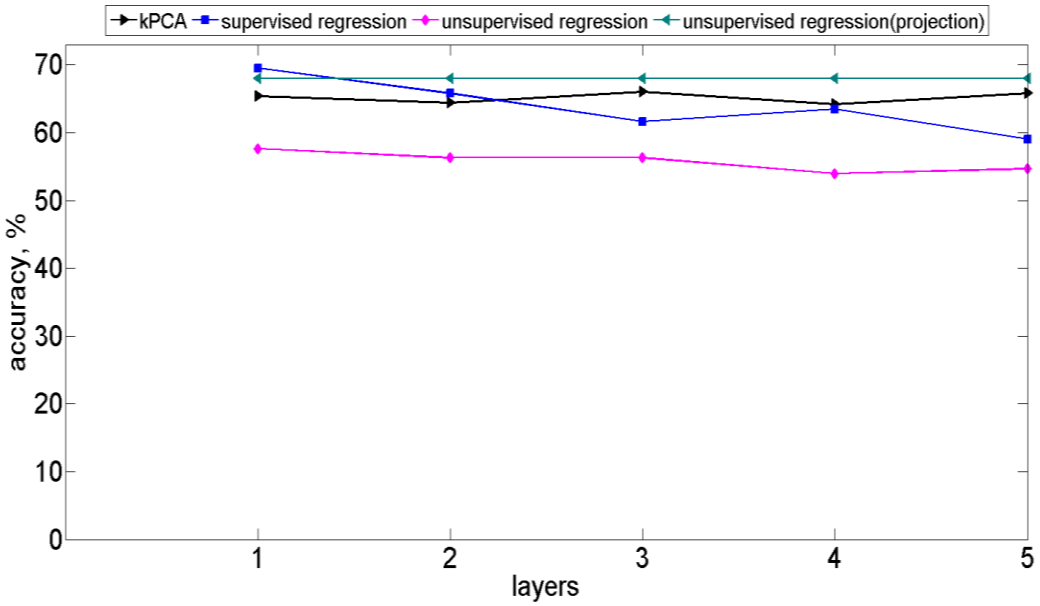


Figure 8: Accuracy vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 9 below presents the MSE values for the four algorithms according to the variation in the number of layers.

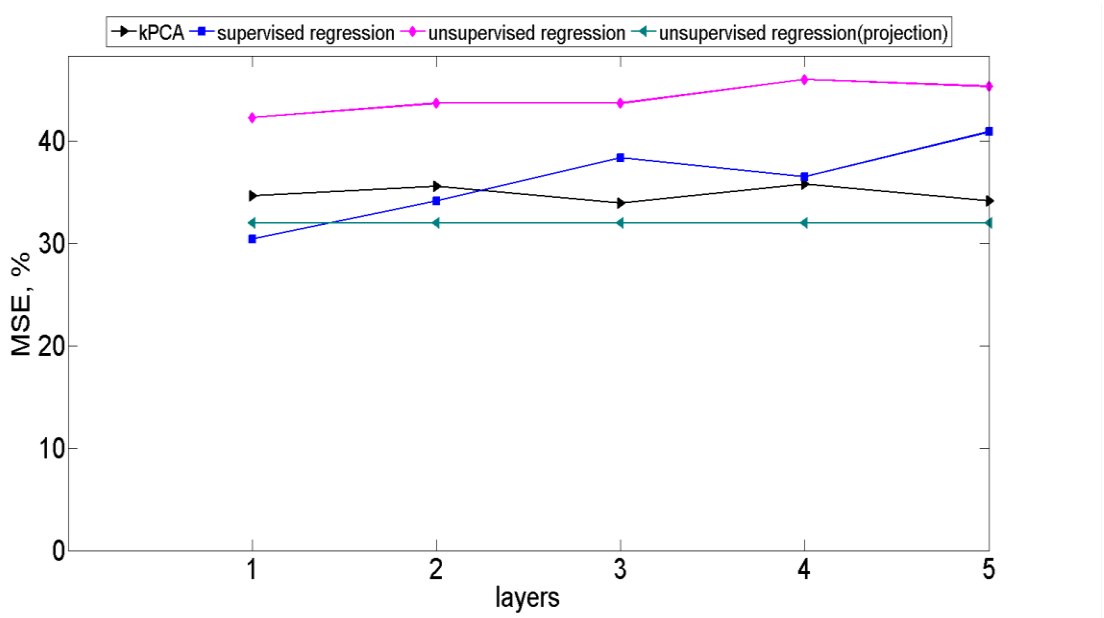


Figure 9: MSE vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 10 below presents the sensitivity values for the four algorithms according to the variation in the number of layers.

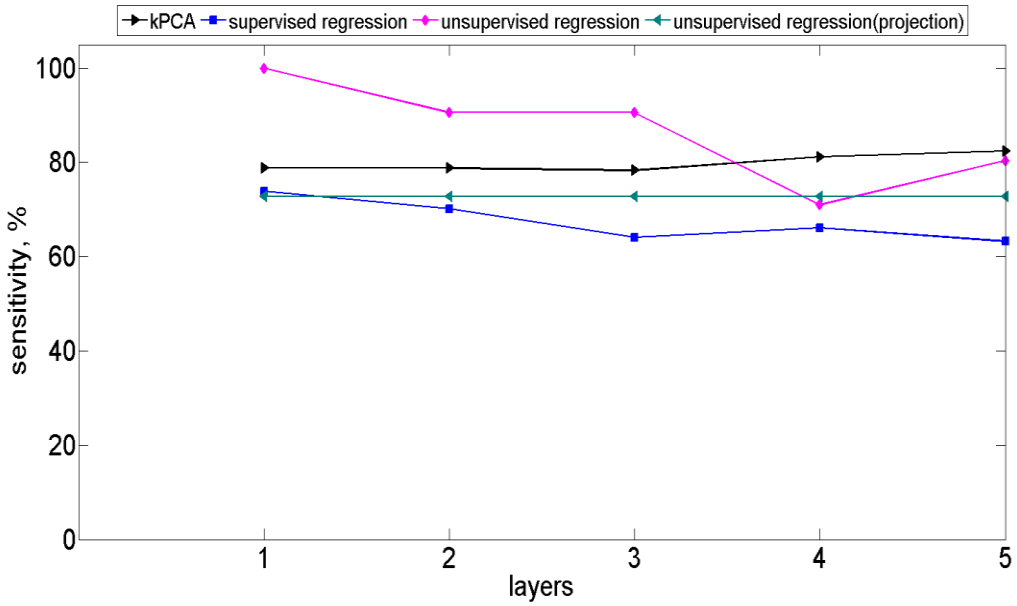


Figure 10: Sensitivity vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 11 below presents the specificity values for the four algorithms according to the variation in the number of layers.

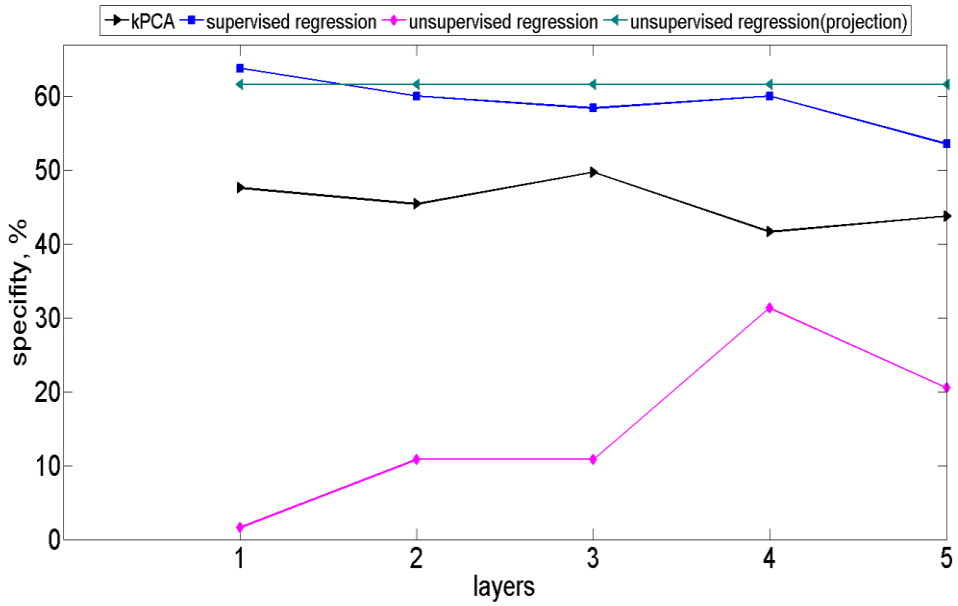


Figure 11: Specificity vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 12 below presents the Cohen's Kappa values for the four algorithms according to the variation in the number of layers.

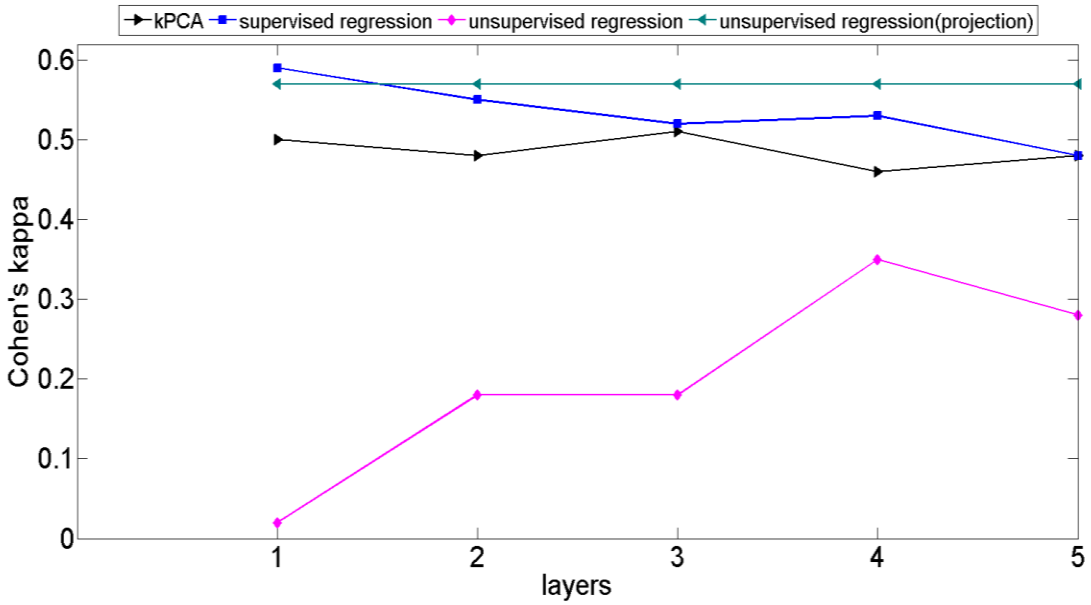


Figure 12: Cohen's Kappa vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 13 below presents the Training time values for the four algorithms according to the variation in the number of layers.

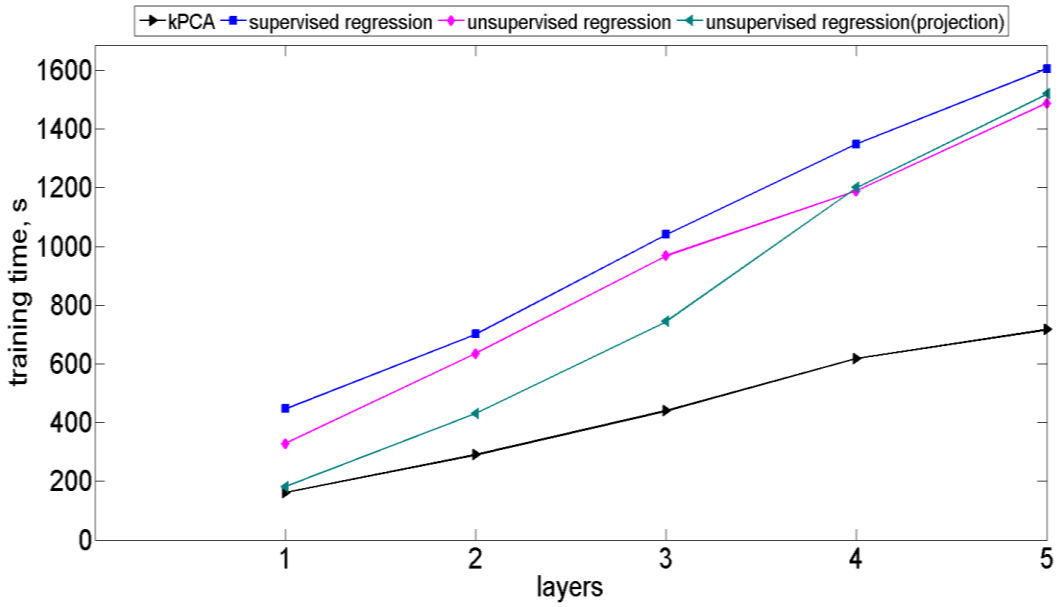


Figure 13: Training time (sec) vs. number of layers when applying the four algorithms for Arrhythmia dataset

Figure 14 below presents the Validation time values for the four algorithms according to the variation in the number of layers.

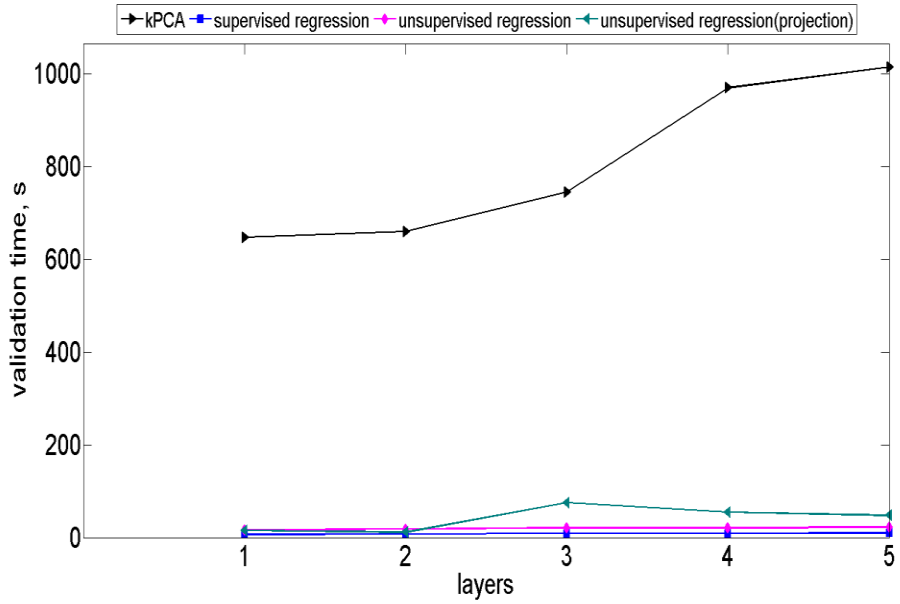


Figure 14: Validation time (sec) vs. number of layers when applying the four algorithms for Arrhythmia dataset

Table 13: Summary of Confusion Matrix for kPCA algorithm for Arrhythmia dataset

Number of Layers	Confusion Matrix			
1		**A**	**N**	∑
	A	193	97	**290**
	N	52	88	**140**
	∑	**245**	**185**	**430**
2		**A**	**N**	∑
	A	193	101	**294**
	N	52	84	**136**
	∑	**245**	**185**	**430**
3		**A**	**N**	∑
	A	192	93	**285**
	N	53	92	**145**
	∑	**245**	**185**	**430**
4		**A**	**N**	∑
	A	199	108	**307**
	N	46	77	**123**
	∑	**245**	**185**	**430**
5		**A**	**N**	∑
	A	202	104	**306**
	N	43	81	**124**
	∑	**245**	**185**	**430**

Table 14: Summary of Confusion Matrix for Supervised Regression algorithm for Arrhythmia dataset

Number of Layers	Confusion Matrix			
1		**A**	**N**	∑
	A	181	67	**248**
	N	64	118	**182**
	∑	**245**	**185**	**430**
2		**A**	**N**	∑
	A	172	74	**246**
	N	73	111	**184**
	∑	**245**	**185**	**430**
3		**A**	**N**	∑
	A	157	77	**234**
	N	88	108	**196**
	∑	**245**	**185**	**430**
4		**A**	**N**	∑
	A	162	74	**236**

	N	83	11	**94**
	∑	**245**	**185**	**430**
5		**A**	**N**	**∑**
	A	155	86	**241**
	N	90	99	**189**
	∑	**245**	**185**	**430**

Table 15: Summary of Confusion Matrix for Unsupervised Latent Regression algorithm for Arrhythmia dataset

Number of Layers	**Confusion Matrix**			
1		**A**	**N**	**∑**
	A	245	182	**427**
	N	0	3	**3**
	∑	**245**	**185**	**430**
2		**A**	**N**	**∑**
	A	222	165	**387**
	N	23	20	**43**
	∑	**245**	**185**	**430**
3		**A**	**N**	**∑**
	A	222	165	**387**
	N	23	20	**43**
	∑	**245**	**185**	**430**
4		**A**	**N**	**∑**
	A	174	127	**301**
	N	71	58	**129**
	∑	**245**	**185**	**430**
5		**A**	**N**	**∑**
	A	197	147	**344**
	N	48	38	**86**
	∑	**245**	**185**	**430**

Table 16: Summary of Confusion Matrix for Unsupervised Latent Regression with projection algorithm for Arrhythmia dataset

Number of Layers	**Confusion Matrix**			
1		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
2		**A**	**N**	**∑**
	A	179	71	**250**
	N	66	114	**180**

	∑	**245**	**185**	**430**
3		**A**	**N**	∑
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
4		**A**	**N**	∑
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**
5		**A**	**N**	∑
	A	179	71	**250**
	N	66	114	**180**
	∑	**245**	**185**	**430**

Table 17: Results Summary of kPCA algorithm at Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	96.7	98.7	98.5	96.25	96.8
MSE,%	3.3	1.3	1.5	3.75	3.2
Sensitivity	100	99.8	99.9	100	99.8
Specificity	95.25	95.25	94.9	94.6	95.4
Cohen's kappa	0.95	0.95	0.94	0.94	0.95
Training time, s	70	234.5	368.2	422.8	702
Validation time,s	69.2	103.6	226.9	535.2	688.9

Table 18: Results Summary of Supervised Regression algorithm at Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	93.4	93.6	94.5	92.7	93.5
MSE,%	6.6	6.4	5.5	7.3	6.5
Sensitivity	99.5	100	99.8	99.9	99.3
Specificity	90.8	90.8	92.0	89.5	91
Cohen's kappa	0.89	0.89	0.91	0.88	0.90

Training time, s	125.4	448.3	1076.7	893.6	1568.7
Validation time, s	36.8	30.3	47.9	41.45	61.2

Table 19: Results Summary of Unsupervised Latent Regression algorithm at Apnea dataset

Layers No. / Metric	1	2	3	4	5
Accuracy,%	99.65	75.1	69.3	88.5	58.4
MSE,%	0.35	24.9	30.7	11.5	41.6
Sensitivity	99.3	50	86.5	86.5	99.6
Specificity	99.7	86.2	53.1	86.2	36.23
Cohen's kappa	0.99	0.84	0.12	0.84	0.76
Training time, s	299	413.7	858.3	784.1	1604
Validation time, s	108	92.2	125.8	96.7	146.06

Table 20: Results Summary of Unsupervised Latent Regression with Projection algorithm at Apnea dataset

	1	2	3	4	5
Accuracy,%	95.3	95.3	95.4	95.4	95.3
MSE,%	4.7	4.7	4.6	4.6	4.7
Sensitivity	100	100	99.9	99.8	100
Specificity	93.2	93.2	93.4	93.2	93.2
Cohen's kappa	0.92	0.92	0.92	0.92	0.92
Training time, s	31.8	83.9	202.9	860.6	2262.3
Validation time, s	13.3	11.9	10.8	13.36	11.05

Figure 15 below presents the accuracy values for the four algorithms according to the variation in the number of layers.

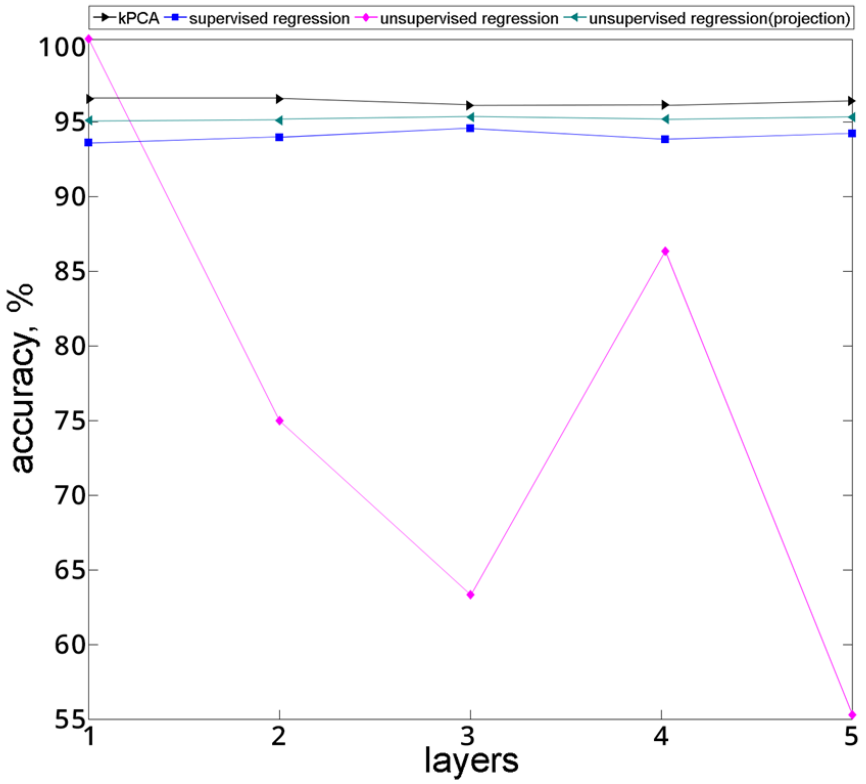


Figure 15: Accuracy vs. number of layers when applying the four algorithms for Apnea dataset

Figure 16 below presents the MSE values for the four algorithms according to the variation in the number of layers.

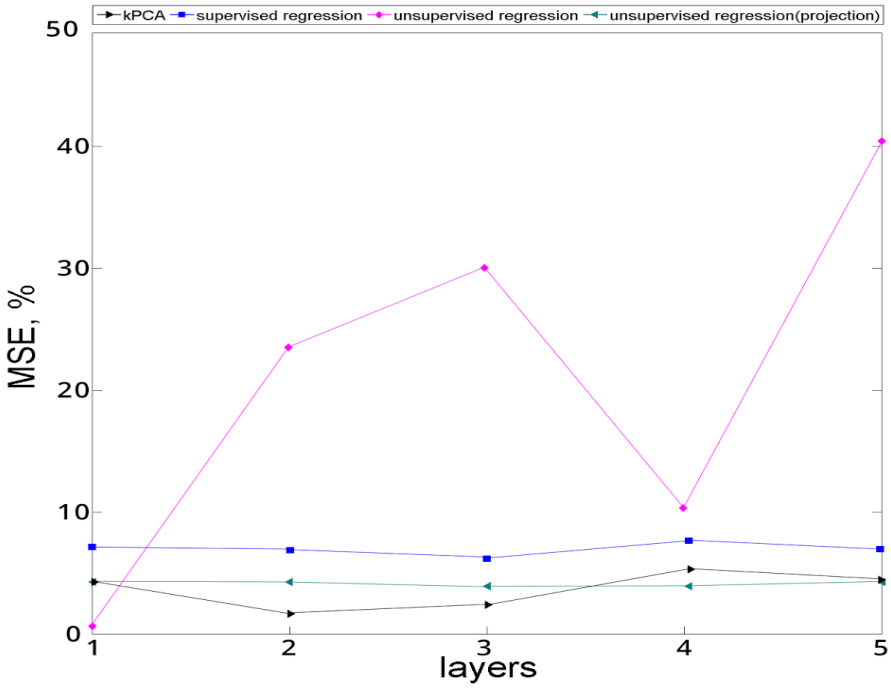


Figure 16: MSE vs. number of layers when applying the four algorithms for Apnea dataset

Figure 17 below presents the sensitivity values for the four algorithms according to the variation in the number of layers.

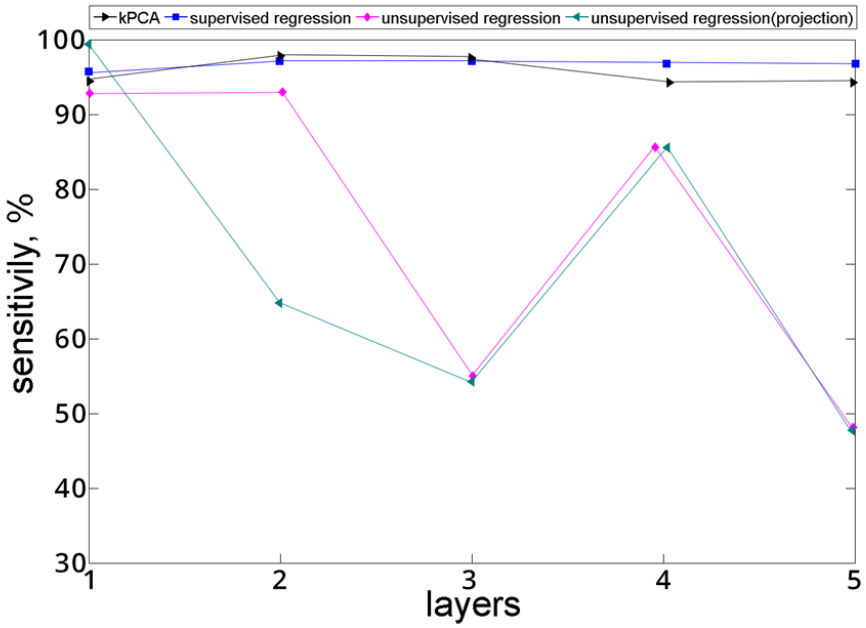


Figure 17: Sensitivity vs. number of layers when applying the four algorithms for Apnea dataset

Figure 18 below presents the specificity values for the four algorithms according to the variation in the number of layers.

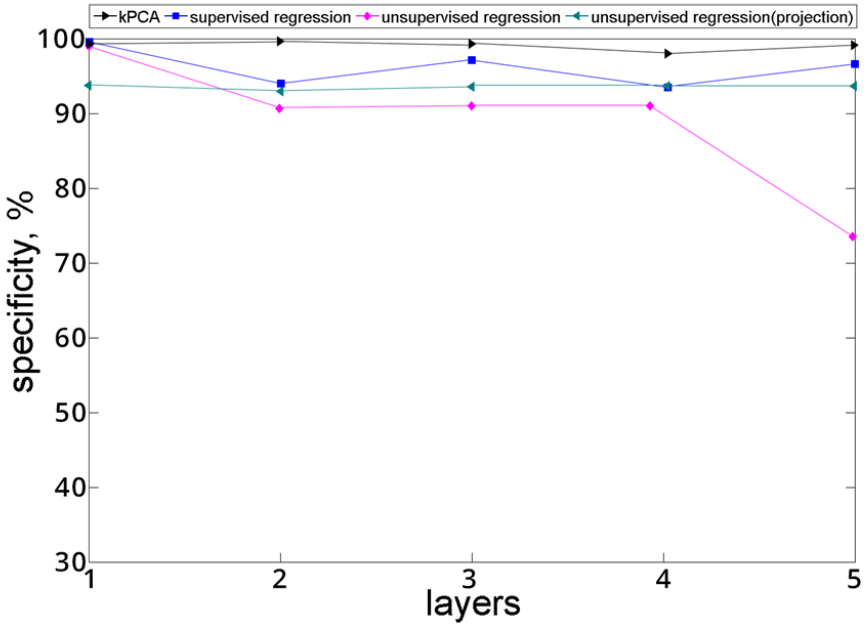


Figure 18: Specificity vs. number of layers when applying the four algorithms for Apnea dataset

Figure 19 below presents the Cohen's Kappa values for the four algorithms according to the variation in the number of layers.

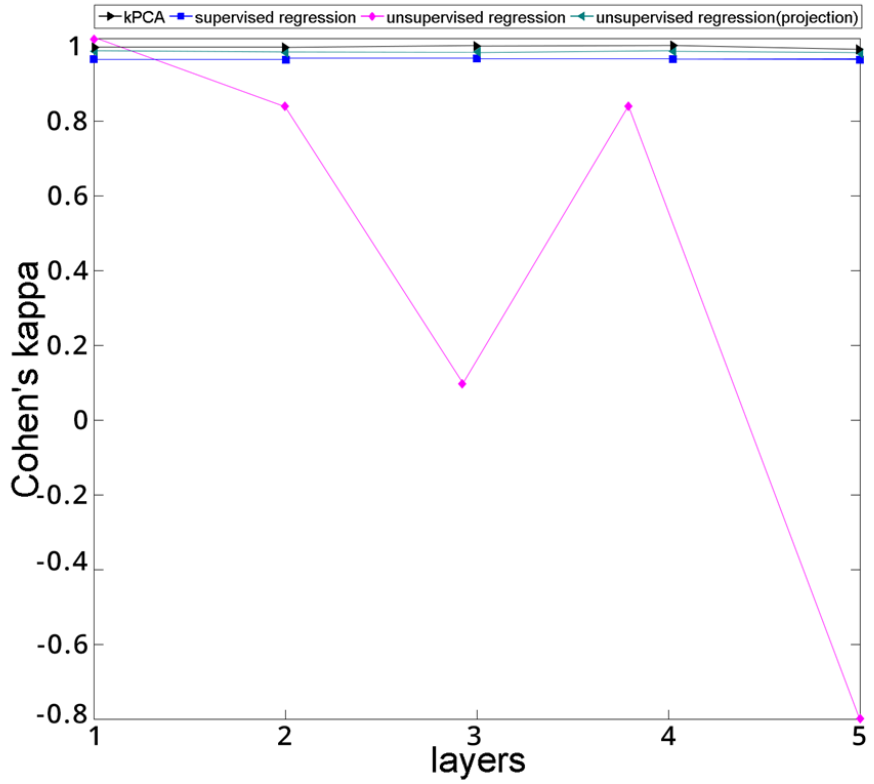


Figure 19: Cohen's Kappa vs. number of layers when applying the four algorithms for Apnea dataset

Figure 20 below presents the Training time values for the four algorithms according to the variation in the number of layers.

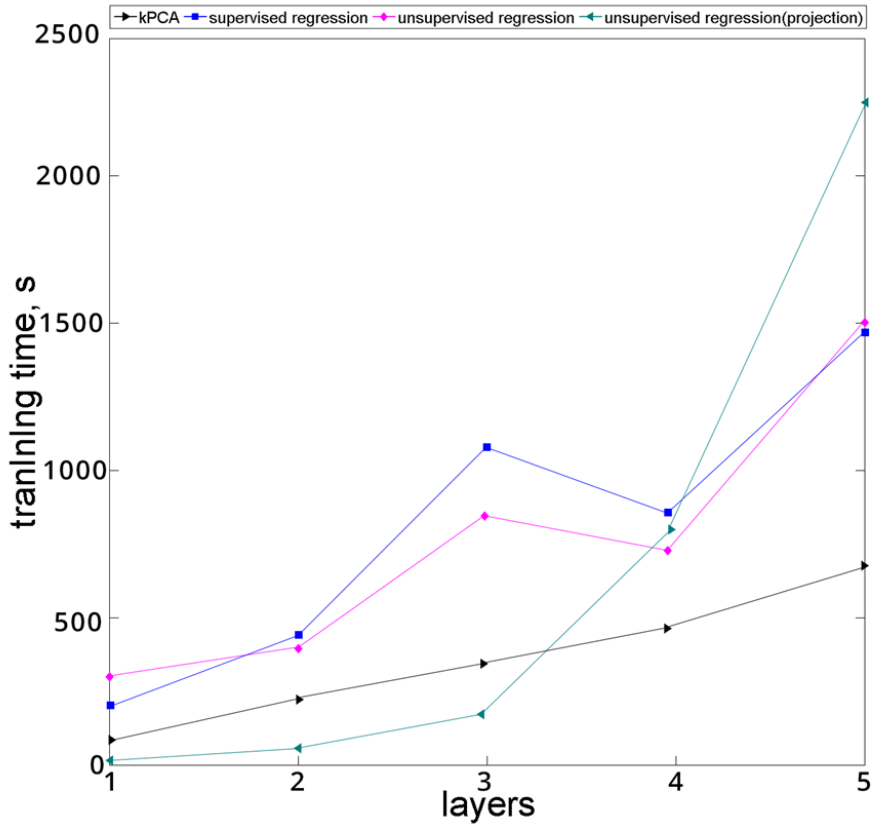


Figure 20: Training time (sec) vs. number of layers when applying the four algorithms for Apnea dataset

Figure 21 below presents the Validation time values for the four algorithms according to the variation in the number of layers.

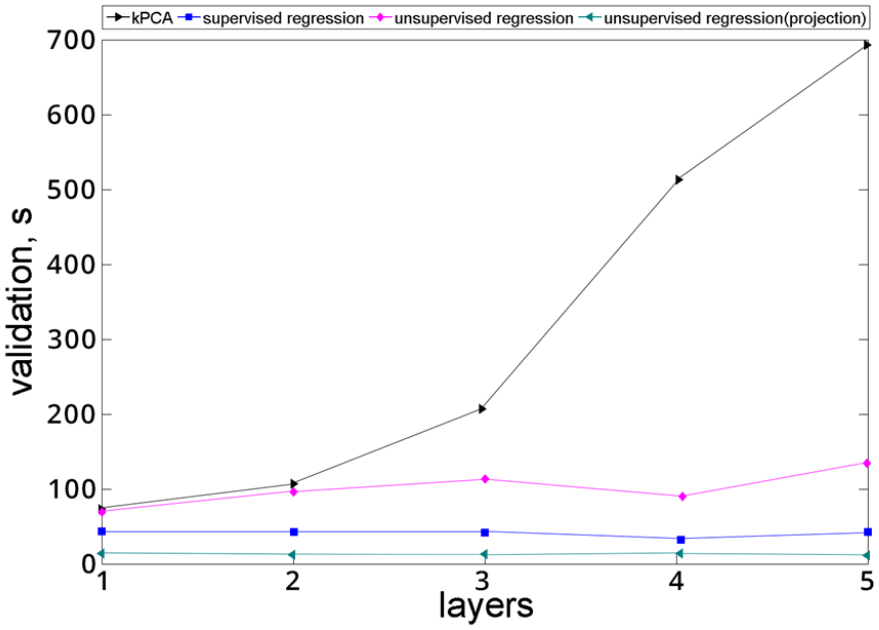


Figure 21: Validation time (sec) vs. number of layers when applying the four algorithms for Apnea dataset

The results for Confusion Matrix for each algorithm is shown in the following tables.

Table 21: Summary of Confusion Matrix for kPCA algorithm for Apnea dataset

Number of Layers	Confusion Matrix			
1		**A**	**N**	∑
	A	6447	500	**6947**
	N	67	9989	**10056**
	∑	**6514**	**10489**	**17003**
2		**A**	**N**	∑
	A	6497	200	**6697**
	N	17	10289	**10306**
	∑	**6514**	**10489**	**17003**
3		**A**	**N**	∑
	A	6477	224	**6701**
	N	37	10265	**10302**
	∑	**6514**	**10489**	**17003**
4		**A**	**N**	∑
	A	6379	502	**6881**
	N	135	9987	**10122**
	∑	**6514**	**10489**	**17003**
5		**A**	**N**	∑
	A	6461	498	**6959**

	N	53	9991	**10044**
	∑	**6514**	**10489**	**17003**

Table 22: Summary of Confusion Matrix for Supervised Regression algorithm for Apnea dataset

Number of Layers	**Confusion Matrix**			
1		**A**	**N**	**∑**
	A	6120	729	**6849**
	N	394	9760	**10154**
	∑	**6514**	**10489**	**17003**
2		**A**	**N**	**∑**
	A	6130	709	**6839**
	N	384	9780	**10164**
	∑	**6514**	**10489**	**17003**
3		**A**	**N**	**∑**
	A	6329	749	**7078**
	N	185	9740	**9925**
	∑	**6514**	**10489**	**17003**
4		**A**	**N**	**∑**
	A	6099	828	**6927**
	N	415	9661	**1076**
	∑	**6514**	**10489**	**17003**
5		**A**	**N**	**∑**
	A	6300	888	**7188**
	N	214	9601	**9815**
	∑	**6514**	**10489**	**17003**

Table 23: Summary of Confusion Matrix for Unsupervised Latent Regression algorithm for Apnea dataset

Number of Layers	**Confusion Matrix**			
1		**A**	**N**	**∑**
	A	6497	40	**6537**
	N	17	10449	**10466**
	∑	**6514**	**10489**	**17003**
2		**A**	**N**	**∑**
	A	5870	3585	**9455**
	N	644	6904	**7548**
	∑	**6514**	**10489**	**17003**
3		**A**	**N**	**∑**
	A	5921	4635	**10556**
	N	593	5854	**6447**
	∑	**6514**	**10489**	**17003**

4		**A**	**N**	**∑**
	A	6009	1458	**7467**
	N	505	9031	**9536**
	∑	**6514**	**10489**	**17003**
5		**A**	**N**	**∑**
	A	4871	5435	**10306**
	N	1643	5054	**6697**
	∑	**6514**	**1 489**	**17003**

Table 24: Summary of Confusion Matrix for Unsupervised Latent Regression with projection algorithm for Apnea dataset

Number of Layers	**Confusion Matrix**			
1		**A**	**N**	**∑**
	A	6117	400	**6517**
	N	397	10089	**10486**
	∑	**6514**	**10489**	**17003**
2		**A**	**N**	**∑**
	A	6105	391	**6496**
	N	409	10098	**10507**
	∑	**6514**	**10489**	**17003**
3		**A**	**N**	**∑**
	A	6120	394	**6701**
	N	394	10095	**10302**
	∑	**6514**	**10489**	**17003**
4		**A**	**N**	**∑**
	A	6125	399	**6524**
	N	389	10090	**10497**
	∑	**6514**	**10489**	**17003**
5		**A**	**N**	**∑**
	A	6100	389	**6489**
	N	414	10100	**10514**
	∑	**514**	**10489**	**17003**