INTERNET OF THINGS
FOR
HEALTH CARE

A thesis submitted in partial fulfilment of the requirements for the award of the degree of

Master of Science

in

Computer Science

By

Ashish Gupta
16419CMP008

Department of Computer Science
Institute of Science
Banaras Hindu University, Varanasi - 221005

2018

CANDIDATE’S DECLARATION

I hereby certify that the work, which is being presented in the report/thesis, entitled **Internet of Things for Health Care**, in partial fulfilment of the requirement for the award of the Degree of **Master of Science** and submitted to the institution is an authentic record of my/our own work carried out during the period *Feburary-2018* to *May-2018* under the supervision of Dr. S. Suresh. I also cited the reference about the text(s) /figure(s) /table(s) /equation(s) from where they have been taken.

The matter presented in this thesis as not been submitted elsewhere for the award of any other degree or diploma from any Institutions.

Date: Signature of the Candidate

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

Date: Signature of the Research Supervisor

The Viva-Voce examination of *Ashish Gupta*, M.Sc./M.C.A. Student has been held on __________________.

Signature of
External Examiner

Signature of
Head of the Department

ABSTRACT

Heart Disease is the major cause of deaths in the world. In last few years, several efficient tools and techniques have been proposed by researchers for building effective medical decision-making systems. In this project, the support vector machine, Multi-Layer Perceptron and logistic regression are used to build effective medical decision support systems for treatment of heart disease. In order to determine the applicability of the proposed decision-making system it is tried on Heart Disease datasets and arrhythmia dataset. The datasets were obtained from UCI machine learning repository. The rapid-growth in technology has increased the scope of remote health monitoring system. Internet of things (IOT) is a network of connected uniquely identifiable smart objects which are accessible through the internet. Among the applications enabled by IOT, real time health monitoring system is a trending application. In this project, a four-tier based remote health monitoring system framework is adopted to solve and process a huge volume of wearable sensor data. The machine learning based decision model is used at its core in the framework for the prediction of heart diseases. Tier-1 focuses on collection of health data through wearable sensor devices. Tier-2 focuses local data storage at patients' premises that interfaces between sensors and other centralized data repository and/or healthcare providers. Tier-3 uses scalable storage (centralized data repository) to store the huge volume of data sent from local storage at patients' premises. Tier-4 uses Apache Spark to develop machine learning based decision support system for heart diseases. In summary SVM based decision system performed better on heart disease datasets and Multi-Layer Perceptron based decision system performed better on arrhythmia dataset.

Keywords: Apache Spark, Internet of things, SVM, Logistic Regression, Multi-Layer Perceptron, Heart Disease, IOT for healthcare, Decision support system.

ACKNOWLEDGEMENTS

The satisfaction that accompanies the successful completion of this project would be incomplete without the mention of the people who made it possible. I would like to express my deepest appreciation to all those who provided us the possibility to complete this report. I convey thanks to my project guide Dr. S. Suresh of Department of Computer Science, Banaras Hindu University for providing encouragement, constant support and guidance which was of a great help to complete this project successfully. I would also like to thank our Head of the Department Prof. S. Karthikeyan, committee Coordinator Ms Ankita Vaish, Dr. Gaurav Baranwal for their valuable support throughout the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure No **Title** **Page No**.

CHAPTER 1

INTRODUCTION

1.1 HEART DISEASE

The Term heart disease is often called cardiovascular disease. Diseases under the heart disease umbrella include blood vessel diseases, such as coronary artery disease, arrhythmias, Myocardial infarction and congenital heart defects.

- Coronary artery disease: - When a substance called plaque builds up on the walls of arteries. This causes coronary arteries to narrow, limiting blood flow to the heart. Symptoms may include chest pain, heart attack
- Myocardial infarction: - This is also known as a heart attack, cardiac infarction, and coronary thrombosis. An interrupted blood flow damages or destroys part of the heart muscle. This is usually happens due to a blood clot that develops in one of the coronary arteries and can also occur if an artery suddenly narrows
- Arrhythmia: - Improper beating of the heart, whether irregular, too fast or too slow. Symptoms may include fluttering in the chest, chest pain, fainting or dizziness
- congenital heart defects: - Defect in the structure of the heart from birth of that person

According to the Centre for Disease Control (CDC), heart disease is the major cause of death in the, Australia, United States, Canada and United Kingdom. One in every four deaths in the U.S. occurs due to the heart disease [21].

There are many factors associated with heart disease. Some of the risk factors for coronary artery disease are age, chest pain, resting blood pressure, sugar level, cholesterol, heart rate, exercise induced chest pain etc. Having so many factors to analyse, physicians make decision based on the current medical test results of the patients. This procedure is a tedious one and requires a highly skilled physician and any mistake in the decision can put the life of the patient in danger.

In last few years, due to the advancement in the field of artificial intelligence have led to the development of machine learning based decision support model for effective treatment of heart diseases. In this project SVM [5,23], Multi-Layer Perceptron [5,22] and logistic regression are used to build effective medical decision support systems for treatment of heart diseases. Multi-Layer Perceptron, SVM and logistic regression [24] are popular machine learning algorithms.

Remote health monitoring (RHM) system is a technology by which one can monitor patient's health outside hospital or clinic premises (e.g. in the home). Incorporating RHM in treatment of heart diseases increase care of the patients, patient feels comfortable in a homely environment without reducing the quality of medical services, reduces healthcare delivery costs, reducing the number of emergency department visits and duration of hospital stays.

1.2 IOT

Internet of Things(IOT) can be thought of as the interconnection of uniquely identifiable smart objects and devices within todays internet infrastructure. Devices and objects with built in sensors are connected to an IOT platform, which merges data from the different devices and applies analytics to share the most valuable information with applications built to address specific needs. The internet of things provides wide range of applications such as smart home, wearables, connected cars, smart cities, waste management, logistics, retails, industrial control and health care. According to Gartner and Forbes, it is estimated that by 2020, the Internet of Things (IoT) will contribute $1.9 trillion to the global economy and $117 billion to the IoT-based healthcare industry. Based on the estimate, Medical care and health care represents one of the most attractive applications areas for the IOT. The IOT has the potential to give rise to many medical applications such as RHM, fitness programs, chronic diseases and elderly care.

In this project we also provide a design of remote health monitoring system which consists of a four-tier architecture to store and process a huge volume of wearable sensor data. and uses

- Tier-1 focuses on collection of health data through wearable sensor devices.
- Tier-2 focuses local data storage at patients' premises that interfaces between sensors and other centralized data repository and/or healthcare providers.
- Tier-3 uses scalable storage (centralized data repository) to store the huge volume of data sent from local storage at patients' premises. In general, relational database are used in IOT enabled RHM system to store clinical data. There has been increase in the variety and quantity of smart wearable devices in recent times and IOT devices continuously generate huge amount of data, so we cannot store and process this data using traditional data management techniques. Scalable NOSQL databases have to be used in the IoT-based health monitoring system. Researchers have started the use of big data and NOSQL technologies in various IoT

applications. For example, Hassanalieragh et al. [7] have used cloud computing with big data technologies to store the clinical data generated by various IOT devices.

- Tier-4 uses Apache Spark to develop the decision support system (discussed above) for treatment of heart diseases.

In this application, the proposed design of RHM continuously observes the individual's health condition. When, the health metrics such as heart rate, ECG, blood pressure, respiratory rate, sweating, skin temperature, and heart sound go beyond standard values, the IoT devices send an alert message with the observed health measures to the doctor and other care holders.

. Finally, the performance of the proposed design of IoT-based RHM system is comparatively analysed with the help of various performance evaluation metrics

1.3 OBJECTIVE

1. Literature review on existing machine learning based model for prediction of heart diseases
2. To implement a Machine learning based model for prediction of heart diseases.
3. Provide a IOT-based Health monitoring system design which consists of a four-tier architecture to store and process a huge volume of wearable sensor data and uses the machine learning model at its core for prediction of heart diseases.
4. Tier-1 focuses on collection of health data through wearable sensor devices.
5. Tier-2 focuses local data storage at patients' premises that interfaces between sensors and other centralized data repository and/or healthcare providers.
6. Tier-3 uses scalable storage (centralized data repository) to store the huge volume of data sent from local storage at patients' premises.
7. Tier-4 uses Apache Spark to develop machine learning based prediction model for heart diseases.
8. The performance of prediction model is comparatively analysed with the help of various performance evaluation metrices to assess the reliability of our expectations.
9. The discussion for improving the accuracy of our prediction model.

CHAPTER 2

LITERATURE REVIEW

2.1 HEART DISEASE AND IOT BASED RHM SYSTEM

Hybrid Neural Network model which is the combination of artificial neural network and fuzzy neural network was proposed. The proposed method achieved accuracy values 84.24% and 86.8% for Pima Indians diabetes dataset and Cleveland heart disease dataset, respectively [11].

SAS base software 9.1.3 is used for diagnosing of the heart disease. A neural networks ensemble method is used in the centre of the proposed system. The proposed method obtained 89.01% classification accuracy for Cleveland heart disease database [12].

A classification model is proposed for diagnosis of heart diseases using tangra tool. Algorithms used are Naive Bayes, KNN, Decision Tree and their respective accuracy of the algorithms are 52.33%, 52%, 45.67% for Cleveland heart disease database [15].

Researchers from Turkey [16] have argued that improving accuracies of machine learning algorithms is very important in improving the performance of computer-aided diagnosis (CADx) systems. In the experiments performed they have used three datasets Parkinson's, diabetes and Cleveland heart diseases to evaluate their classification performances. Their experimental details are as follows: -

1. Correlation-based feature selection (CFS) algorithm is used for reducing the dimensions of the datasets used in the experiment.
2. classification performances of 30 machine learning algorithms are calculated for three datasets.
3. 30 classifier ensembles are constructed based on Rotation Forest(RF) algorithm and evaluate the performances of respective classifiers with the same three datasets.
4. Performance evaluation metrices used are kappa error (KE), classification accuracy (ACC), and area under ROC curve.

The classification performance of 30 base classifiers and their respective ensemble classifiers on Cleveland heart disease dataset is shown in the table 1. In the table 1, 'e' means RF classifier ensemble corresponding to base classifier measures. 'Diff' means 'Difference' while 'AVG' stands for 'average'.

Algorithm	ACC (%)	eACC (%)	KE	eKE	AUC	eAUC
Bayesian Logistic Regression	64.97	64.97	0.39	0.39	0.602	0.612
Bayes Net	83.16	82.83	0.65	0.63	0.9	0.875
Naive Bayes	84.48	82.17	0.68	0.63	0.897	0.887
Logistic Regression	82.83	82.5	0.65	0.64	0.902	0.906
Multi-Layer Perceptron	82.5	82.5	0.64	0.64	0.866	0.895
RBF Network	83.49	84.48	0.66	0.68	0.887	0.895
Simple Logistic	82.83	82.83	0.65	0.65	0.899	0.908
SMO	83.49	83.82	0.66	0.67	0.83	0.879
KSTAR	78.54	78.87	0.56	0.57	0.881	0.865
Locally Weighed Learner	71.61	81.51	0.43	0.62	0.847	0.9
Fuzzy Lattice Reasoning	79.63	79.63	0.57	0.57	0.889	0.893
HiperPipes	55.77	76.56	0.32	0.51	0.516	0.796
Voting Feature Intervals	81.18	77.55	0.62	0.54	0.867	0.816
Conjunctive Rule	71.94	82.17	0.43	0.64	0.715	0.89
JRIP	80.19	82.83	0.59	0.65	0.798	0.886
Nnge	80.85	83.49	0.61	0.66	0.807	0.886
OneR	71.61	83.82	0.43	0.67	0.716	0.892
PART Decision Learner	81.18	82.17	0.62	0.64	0.845	0.898
Ripple-Down Rule learner	77.22	82.83	0.53	0.65	0.768	0.893
ZeroR	54.45	54.45	0.3	0.3	0.487	0.487
Decision Table	82.83	82.83	0.64	0.64	0.875	0.879
Best-first Decision Tree	78.54	82.5	0.56	0.64	0.81	0.897
Decision Stump	71.61	80.85	0.43	0.61	0.68	0.891
Functional Tree Learner	81.51	84.15	0.62	0.67	0.843	0.9
J48	77.22	82.17	0.53	0.64	0.795	0.895
Alternating Decision Tree	81.51	82.5	0.62	0.63	0.879	0.882
Logistic Model Trees	82.83	82.83	0.65	0.65	0.899	0.907
Random Tree	78.54	80.52	0.57	0.6	0.791	0.875
Fast Decision Tree Learner	79.2	82.17	0.57	0.64	0.837	0.895
Simple Chart	79.86	82.17	0.59	0.63	0.835	0.894
AVG	77.52	80.49	0.56	0.61	0.805	0.862

Table 1 – Cleveland heart disease classification results [16].

A scalable three-tier architecture to store and process such huge volume of wearable smart devices is proposed in [13]. Tier-1 focuses on collection of data from wearable smart devices. Tier-2 uses Apache HBase for storing the large volume of wearable smart devices. Tier-3 uses Apache Mahout for developing the logistic regression-based prediction model for heart diseases.

Researchers from Harvard University have developed a health care project to measure individual health parameters such as ECG, EKG, EMG, SpO2, pulse oximeter and Mica2 motes. Electronic devices such as laptops, personal computers and PDAs are used in the CodeBlue project for necessary action from doctors and care holders when the patients' health condition deteriorates [14].

A three-tier architecture has been proposed for remote health monitoring. The first-tier deals with data acquisition using wearable sensor devices, Second tier deals with Transmission of data to a cloud storage and third tier deals with Analytics. The paper also highlights opportunities and challenges of IOT for health care in future [18].

A comprehensive survey for the internet of things for health care is provided in [17]. The survey presents a detailed review of advances in IOT-based health care technologies, network architectures, IOT based health care applications and services, Industrial trends in IOT based health care solutions. In addition, the survey presents a deep observation into security and privacy issues surrounding IOT healthcare solutions and proposes an intelligent security model to minimize security risks.

The researchers from Selcuk University, have used principal component analysis to reduce the dimension of arrhythmia dataset from 279 features to 15 features and then the reduced dataset is used to train least square support vector machine (LS-SVM) classifier. They have used two classes as the presence or absence of arrhythmia in their experiments. The classifier consists of three stages: 50–50% of training-test dataset, 70–30% of training-test dataset and 80–20% of training-test dataset, the obtained classification accuracies; 96.86%, 100% and 100% respectively [19].

The existing works in the field of Remote Health Monitoring system [13, 18] propose only three-tier architecture to collect physiological parameters from patient's body and then the collected data is transferred to cloud based storage on which analytics is performed

The three-tier architecture has the following disadvantages: -

1. Cloud based datacentre are centralized, so all the data from different regions can cause congestion in the core network.
2. During patient’s critical condition, a crash or network jam can be catastrophic. Therefore, we need a RHM system which is reliable and has a very low time.
3. Every bit of data needs to be sent to the cloud and hence it consumes lot of bandwidth. Therefore, the efficiency is reduced
4. A costlier approach, since it requires a huge amount cloud storage and processing power.
5. There is a limited support for mobility of patients or IOT devices. Suppose if there is a network outage then the health data captured is lost because it could not be sent to the cloud.

Considering the disadvantages of the three-tier architecture we propose a four-tier architecture based on fog computing for RHM system.

The four-tier architecture has the following advantages: -

1. Need of bandwidth is reduced by not sending every bit of information to the cloud, and instead aggregating or pre-processing it at certain access points.
2. Lowers the cost as it requires less cloud storage and processing power.
3. If the trained prediction model is stored in the fog node or tier-2 of the proposed architecture, it enables real-time monitoring and improves quality of service.
4. This kind of distributed strategy, may help in lowering cost and improve efficiencies.
5. Mobility is supported.
6. Latency is reduced and reduces the congestion in the core network of the centralized datacentre.
7. Improved security of encrypted data as it stays closer to the end user

In [13] researchers have used Apache Mahout for developing the logistic regression-based prediction model for heart diseases. Apache Mahout is machine learning, data mining, and math library on top of MapReduce. Apache Spark provide an easier to use alternative to MapReduce and run programs up to 100 times faster than Hadoop MapReduce in memory or 10 times faster on disk. Details about Apache spark and Apache Hadoop has been discussed in Chapter 3 of this project.

CHAPTER 3

BACKGROUND DETAILS

In this chapter we are going to discuss about the details of the dataset, principal component analysis algorithm and tools used in our experiments.

3.1 DATASET DESCRIPTION

For the experiments taken place in this project, we have used arrhythmia and heart disease dataset, both the dataset used in our experiment was taken form UCI Machine Learning Repository.

Heart Disease Dataset

The UCI heart disease directory contains 4 databases concerning heart disease diagnosis. The data was collected from the following four locations.

1. Cleveland Clinic Foundation (Cleveland. Data)
2. Hungarian Institute of Cardiology, Budapest (Hungarian. Data)
3. V.A. Medical Centre, Long Beach, CA (long-beach-va.data)
4. University Hospital, Zurich, Switzerland (Switzerland. Data)

Each of the database contains 76 attributes. However, majority of the existing studies have used only a maximum of 14 attributes (UCI, 2009; UCI, 2010).
The description of each of the 14 attributes are listed as follows:

1. age: age in years
2. Sex: male, female
3. cp: chest pain type
    - -- Value 1: typical angina
    - -- Value 2: atypical angina
    - -- Value 3: non-anginal pain
    - -- Value 4: asymptomatic
4. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholesterol in mg/dl
6. fbs: fasting blood sugar > 120 mg/dl (1 = true; 0 = false)

7. restecg: resting electrocardiographic results

   -- Value 0: normal

   -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)

   -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria

8. thalach: maximum heart rate achieved
9. exang: exercise induced angina (1 = yes; 0 = no)
10. oldpeak: ST depression induced by exercise relative to rest
11. slope: The slope of the peak exercise ST segment

    -- Value 1: upsloping

    -- Value 2: flat

    -- Value 3: downsloping

12. ca: number of major vessels (0-3) coloured by fluoroscopy
13. thal: 3 = normal; 6 = fixed defect; 7 = reversible defect
14. num: The predicted attribute

    This field refers to the presence of heart disease in the patient

    -- 0: No presence

    -- 1: Presence

    -- 2: Presence

    -- 3: Presence

    -- 4: Presence

The Dataset contains five class attributes indicating class 0 as healthy and class 1,2,3,4 as one of the sick types.

Class Distribution in each of the dataset are as follows: -

Table 2- Class Distribution of Heart Disease Dataset

Database	0	1	2	3	4	Total
Cleveland	164	55	36	35	13	303
Hungarian	188	37	26	28	15	294
Switzerland	8	48	32	30	5	123
Long Beach VA	51	56	41	42	10	200

For this project work, we have used only Cleveland. Data and Switzerland. Data and multi-class classification problem is converted into a binary classification problem, i.e. class 0 as positive and class 1,2,3,4 as negative.

The number of Instances in each of the dataset are as follows: -

Database	No of instances:
Cleveland	303
Hungarian	294
Switzerland	123
Long Beach VA	200

Arrhythmia Dataset

The Arrhythmia Dataset provided by UCI repository aims to distinguish between the presence and absence of cardiac arrhythmia and to classify it in one of the 16 groups. The Arrhythmia dataset has 452 instances and 279 attributes. There are several missing attributes in the dataset denoted by '?'

Attribute Information:

1 Age: Age in years, linear

2 Sex: Sex (0 = male; 1 = female), nominal

3 Height: Height in centimetres, linear

4 Weight: Weight in kilograms, linear

5 QRS duration: Average of QRS duration in msec., linear

6 P-R interval: Average duration between onset of P and Q waves in msec., linear

7 Q-T interval: Average duration between onset of Q and offset of T waves in msec., linear

8 T interval: Average duration of T wave in msec., linear

9 P interval: Average duration of P wave in msec., linear

Vector angles in degrees on front plane of, linear

10 QRS

11 T

12 P

13 QRST

14 J

15 Heart rate: Number of heart beats per minute, linear

Of channel DI:

Average width, in msec., of: linear

16 Q wave

17 R wave

18 S wave

19 R' wave, small peak just after R

20 S' wave

21 Number of intrinsic deflections, linear

22 Existence of ragged R wave, nominal

23 Existence of diphasic derivation of R wave, nominal

24 Existence of ragged P wave, nominal

25 Existence of diphasic derivation of P wave, nominal

26 Existence of ragged T wave, nominal

27 Existence of diphasic derivation of T wave, nominal

Of channel DII:

28 .. 39 (similar to 16 .. 27 of channel DI)

Of channels DIII:

40 .. 51

Of channel AVR:

52 .. 63

Of channel AVL:

64 .. 75

Of channel AVF:

76 .. 87

Of channel V1:

88 .. 99

Of channel V2:

100 .. 111

Of channel V3:

112 .. 123

Of channel V4:

124 .. 135

Of channel V5:

136 .. 147

Of channel V6:

148 .. 159

Of channel DI:

Amplitude, * 0.1 milivolt, of

160 JJ wave, linear

161 Q wave, linear

162 R wave, linear

163 S wave, linear

164 R' wave, linear

165 S' wave, linear

166 P wave, linear

167 T wave, linear

168 QRSA, Sum of areas of all segments divided by 10, (Area= width * height / 2), linear

169 QRSTA = QRSA + 0.5 * width of T wave * 0.1 * height of T wave. (If T is diphasic then

the bigger segment is considered), linear

Of channel DII:

170 .. 179

Of channel DIII:

180 .. 189

Of channel AVR:

190 .. 199

Of channel AVL:

200 .. 209

Of channel AVF:

210 .. 219

Of channel V1:

220 .. 229

Of channel V2:

230 .. 239

Of channel V3:

240 .. 249

Of channel V4:

250 .. 259

Of channel V5:

260 .. 269

Of channel V6:

270 .. 279

Class Distribution:

Table 3- Class Distribution of Arrhythmia Dataset

Class code	Class	Number of Instances
1	Normal	245
2	Ischemic changes (Coronary Artery Disease)	44
3	Old Anterior Myocardial Infarction	15
4	Old Inferior Myocardial Infarction	15
5	Sinus tachycardia	13
6	Sinus bradycardia	25
7	Ventricular Premature Contraction (PVC)	3
8	Supraventricular Premature Contraction	2
9	Left bundle branch block	9
10	Right bundle branch block	50
11	1. degree AtrioVentricular block	0
12	2. degree AV block	0
13	3. degree AV block	0
14	Left ventricule hypertrophy	4
15	Atrial Fibrillation or Flutter	5
16	others	22

3.2 DIMENSIONALITY REDUCTION

Consider a dataset having thousands or tens of thousands of features, when these types of dataset are fed to clustering and classification algorithms (and or data analysis algorithms), these algorithms have trouble with high dimensional data resulting in reduced classification accuracy and poor-quality clusters. The dimensionality reduction is the way to handle this problem.

Benefits of dimensionality reduction: -

- Data mining algorithms work better if number of features in a dataset is lower because dimensionality reduction may remove noise and irrelevant features.
- Can lead to more understandable model because the model may involve fewer attributes.
- Data can be easily being visualized
- Amount of time and memory required by data mining algorithm is reduced with a reduction in dimensionality.

One of the most popular and commonly used method for dimensionality reduction is principal component analysis.

PRINCIPAL COMPONENT ANALYSIS (PCA)

It is a statistical method that uses an orthogonal conversion to convert a set of instances of possibly correlated features into a set of values of linearly uncorrelated features called principal components. If there are n instances with p features, then the number of distinct principal component is min (n-1, p). Therefore, the original data is projected to much smaller space, resulting in dimensionality reduction [4].

Attribute sub-set selection reduces the attribute set size by retaining a subset of the initial set of attributes, PCA “combines” the essence of attributes by creating an alternative, smaller set of variables. The principal components are sorted in decreasing order of “significance” or variance. The sorted axis is such that the first axis shows the most variance among the data, the second axis shows the next highest variance and so on. Because the components are sorted in decreasing order of “significance,” the data size can be reduced by eliminating the weaker components, that is, those with low variance. Using the strongest principal components, it should be possible to reconstruct a good approximation of the original data.

Problem formulation

Given two features, x1 and x2, we want to find a single line that effectively describes both features at once. We then map our old features onto this new line to get a new single feature. The same can be done for the three features, where we map them to a plane. The goal of PCA is to reduce the average of all the distances of every feature to the projection line. This is known as projection error

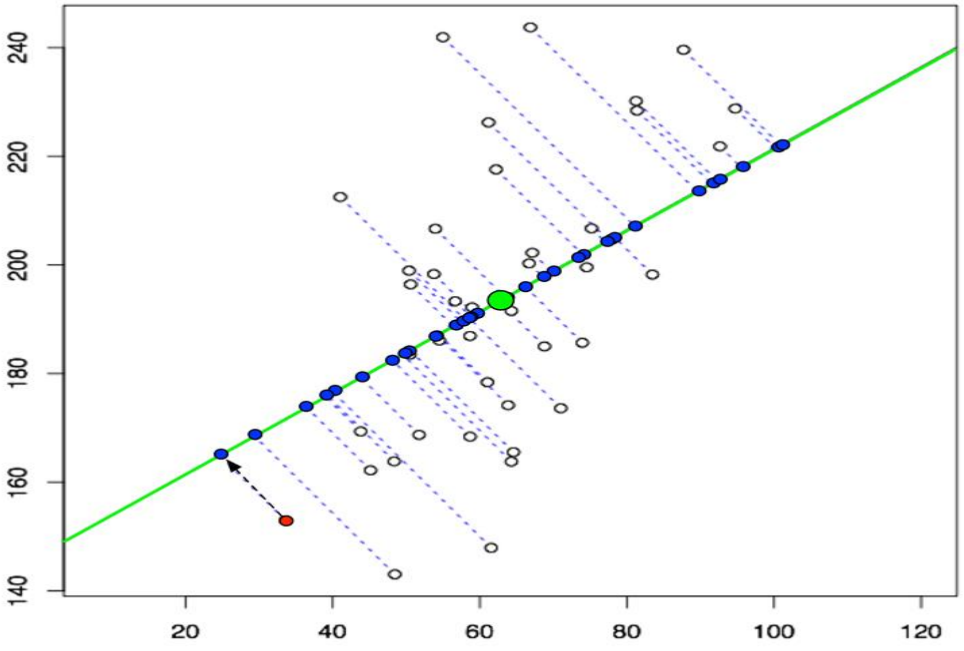

Figure 1- Reduce 2D to 1D

In the above figure 1 from [25], one can see that to reduce a 2D data to 1D we find a direction (a vector u $^{(1)}$ ∈ R^{n}) onto which to project the data so as to minimize the projection error. In a more general way, PCA is a method to reduce data from n-dimension to k-dimension: Find k vectors u $^{(1)}$, u $^{(2)}$, ..., u $^{(k)}$ onto which to project the data so as to minimize the projection error.

The figure 1 looks like a linear regression example but PCA is not a linear regression. Following are the reasons: -

- In linear regression, we are minimizing the squared error from every point to our predictor line. These are vertical distances.
- In PCA, we are minimizing the shortest distance, or shortest orthogonal distances, to our data points.

More generally, in linear regression we are taking all our examples in x (independent variable) and applying the parameters in Θ to predict y (dependent variable) but in PCA, we are taking a number of features x_1, x_2, ..., x_n, and finding a closest common dataset among them. We are not trying to predict any result and we are not applying theta weights to the features.

Principal Component Analysis Algorithm

To Reduce a Dataset X having m instances from n-dimensions to k-dimensions

Before applying PCA on a dataset, there is a data pre-processing step we must perform:

Step 1 DATA PRE-PROCESSING

- Given training set: x(1), x(2), ..., x(m)
- Do feature scaling or mean normalization

$\mu_j = \frac{1}{m}\sum_{i=1}^{m} x_j^{(i)}$

Replace each $x_j^{(i)}$ with $x_j^{(i)} - \mu_j$

If different features on different scales (e.g., x_1 = size of house, x_2 = number of bedrooms), scale features to have comparable range of values.

Above, we first subtract the mean of each feature from the original feature. Then we scale all the features $x_j^{(i)} = \frac{(x_j^{(i)} - \mu_j)}{s_j}$

Step 2 COMPUTE COVARIANCE MATRIX

$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)})(x_j^{(i)})^{T}$ [We denote the covariance matrix with a capital sigma]

$x^{(i)}$ is an $n \times 1$ vector, $(x^{(i)})^{T}$ is a 1×n vector and X is a $m * n$ matrix (row-wise stored examples). The product of those will be an $n * n$ matrix, which are the dimensions of Σ.

Step 3 COMPUTE "EIGENVECTORS" OF COVARIANCE MATRIX Σ

[U, S, V] = SVD(Sigma)

SVD stands for singular value decomposition in linear algebra

Step 4 TAKE THE FIRST K COLUMNS OF THE U MATRIX AND COMPUTE Z

Assign the first k columns of U to a variable called 'Ureduce'.

U will have dimensions $n * n$

Therefore, Ureduce will have $n * k$ matrix.

Ureduce^T will have dimensions $k * n$ while $x^{(i)}$ will have dimensions $n * 1$.

We compute z with:

$z^{(i)} = \quad \text{Ureduce}^T * x^{(i)}$

The product $\text{Ureduce}^T * x^{(i))}$ will have dimensions k×1.

Therefore, Z is the k dimensional vector, which is our reduced dataset

Choosing the optimal value of k (Number of Principal Components)

Given the average squared projection error $\frac{1}{m}\sum_{i=1}^{m}\left|x^{(i)} * xapprox^{(i)}\right|^2$

Given the total variation in the data $\frac{1}{m}\sum_{i=1}^{m}\left|x^{(i)}\right|^2$

PCA tries to minimize the average squared projection error.

Therefore, we should choose: k to be the smallest value so that

$$\frac{\frac{1}{m}\sum_{i=1}^{m}\left|x^{(i)} * xapprox^{(i)}\right|^2}{\frac{1}{m}\sum_{i=1}^{m}\left|x^{(i)}\right|^2} \quad <=0.01$$

In other words, the squared projection error divided by the total variation should be less than one percent, so that 99% of the variance is retained.

Algorithm for choosing k

1. Try PCA with k=1, 2, ...
2. Compute Ureduce, z, x
3. Check the formula given above that 99% of the variance is retained. If not, go to step one and increase k.

This procedure would be a time-consuming process

During PCA implementation we get S matrix by calculating SVD(Sigma)

Pick Smallest value of k for which

$$1 - \frac{\sum_{i=1}^{k} s_{ii}}{\sum_{i=1}^{n} s_{ii}} \quad <=0.01$$

3.3 BIG DATA

Big Data is data sets, which cannot be stored and processed by single system or we can say which cannot be processed using traditional system. The well-known definition of Big Data jointly given by Gartner and IBM is a four Vs concept: volume, velocity, variety and veracity. Therefore, we can say that the data which has large volume, comes with high velocity, from different sources and formats and has great uncertainty is referred as Big Data. Volume-represents the amount of data i.e. Big data has large volume. Velocity-represents the speed at which the data is generated i.e. rate of entering streaming data is very fast. Variety -refers different form of data i.e. unstructured or semi-structured data (click stream, text, audio, video, log file, sensor data, XML) generated from different sources. Veracity-refers uncertainty of data i.e. quality of data being captured. Big data challenges include data analysis, capturing data, sharing, data storage, search, transfer, querying, visualization, updating. The term "Big Data" is not limited to data perspective but it has emerged as a stream that includes associated technologies, tools and real-world applications.

3.4 MAPREDUCE FRAMEWORK

MapReduce is a programming paradigm or distributed computing based on java. MapReduce is a framework using which we can write applications to process large datasets in parallel, on large clusters of commodity hardware in a reliable manner.

A MapReduce program executes in three stages: -

1. Map: Each data node or worker node applies the map function to the local data and writes the output to a temporary storage. A master node ensures that only one copy of redundant input data is processed
2. Shuffle: The data node shuffles the data based on the output keys produced by the map function, such that the all the data having the same key is located on the same data node.
3. Reduce: Each data node process data belonging to the same group (i.e. same key), in parallel.

3.5 APACHE HADOOP

Apache Hadoop [2] software is open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. The library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

The base Apache Hadoop framework is composed of the following modules:

- Hadoop Common – contains libraries and utilities needed by other Hadoop modules.
- Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.
- Hadoop YARN – a platform responsible for managing computing resources in clusters and using them for scheduling users' applications.
- Hadoop MapReduce – an implementation of the MapReduce programming model for large-scale data processing [2].

MapReduce-based programs implemented on Hadoop do not fit well iterative processes because each iteration requires a new reading phase from disk. This feature is critical when dealing with huge datasets. This issue led to the introduction of Spark, which enables the nodes of the cluster to cache data and intermediate results in memory, instead of reloading them from the disk at each iteration.

The following figure 2 from [26] explains how MapReduce works on iterative operations

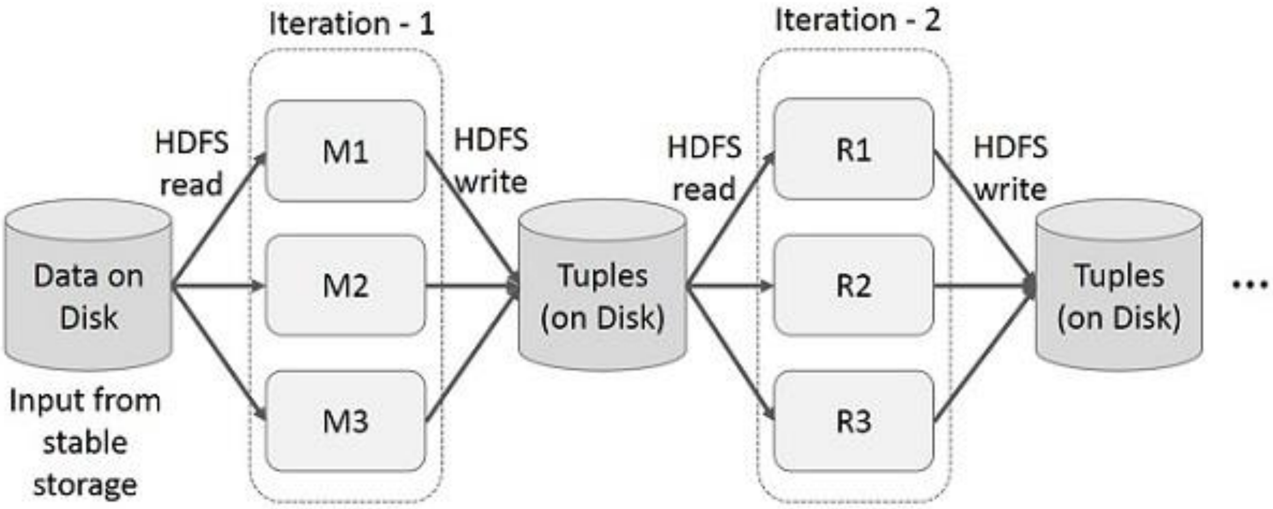

Figure 2- Iterative operations on MapReduce

The following figure 3 from [26] explains how Spark works on iterative operations

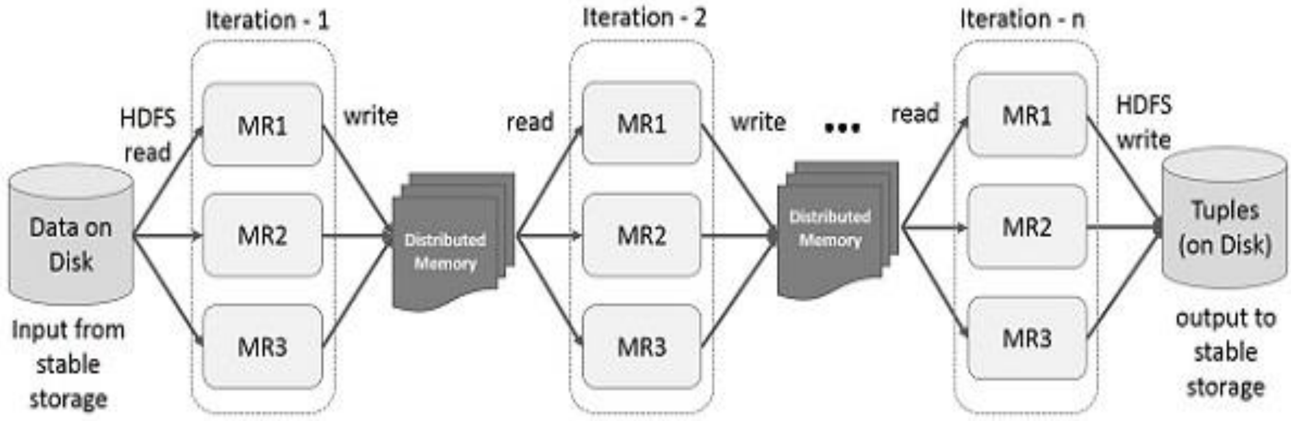

Figure 3- Iterative operations on Spark RDD

3.6 APACHE SPARK

Apache spark is an open source, cluster computing frame work, designed for fast computation. It is based on Hadoop MapReduce and it extends the MapReduce Model (or we can say it is an alternative to MapReduce) to efficiently use it for more type of computations which includes interactive queries and stream processing. Since its release, Apache Spark has seen rapid adoption by enterprises across a wide range of industries. Internet powerhouses such as Netflix, Yahoo, and eBay have deployed Spark at massive scale, collectively processing multiple petabytes of data on clusters of over 8,000 nodes. It has quickly become the largest open source community in big data, with over 1000 contributors from 250+ organizations [9].

Evolution of apache spark

Spark is one of Hadoop's sub project developed in 2009 in UC Berkeley's AMPLab by Matei Zaharia. It was Open Sourced in 2010 under a BSD license. It was donated to Apache software foundation in 2013 [8].

Features of apache spark

Apache spark has the following features: -

1. Speed: Spark runs an application in a Hadoop cluster, up to 100x faster in memory and 10x faster in disks by exploiting in memory computations and other optimizations.
2. Supports multiple languages: Spark provides built-in APIs in Java, Scala, R or Python. Therefore, you can write applications in different languages. Spark offers over 80 high-level operators that make it easy to build parallel apps.
3. Advanced Analytics: Spark not only supports ‘Map’ and ‘reduce’. It comes with a package of libraries including SQL and DataFrames, MLlib for machine learning, GraphX, and Spark Streaming. we can combine these libraries seamlessly in the same application.
4. Runs Everywhere: We can run Spark using its standalone cluster mode, on EC2, on Hadoop YARN (MapReduce 2.0), on Mesos, or on Kubernetes. It can access data from HDFS, Apache Cassandra, Apache HBase, Apache Hive, and hundreds of other data sources.

Components of apache spark

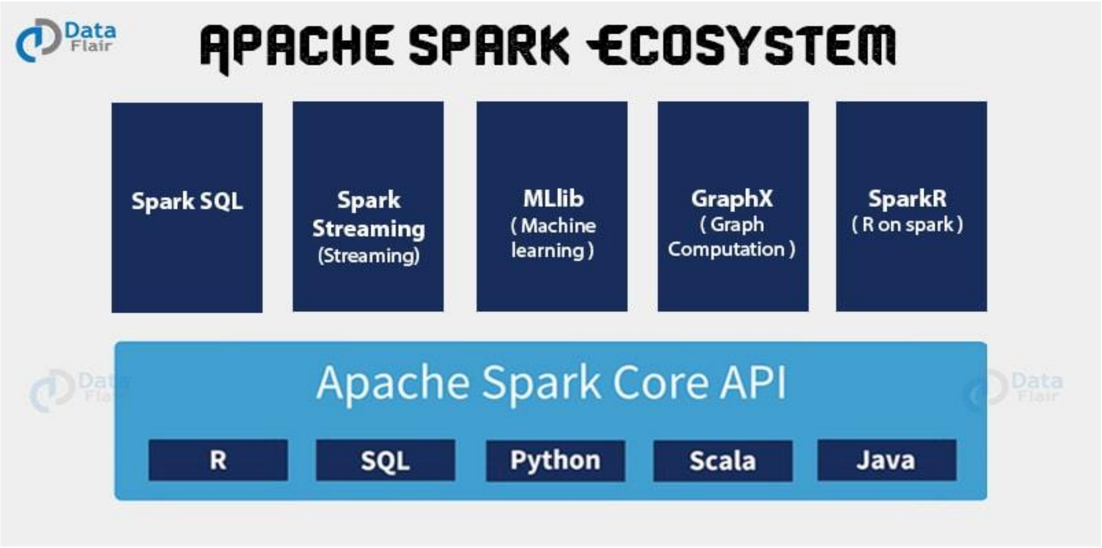

Figure 4: Depicts the various components of Apache Spark

Apache Spark Core

Spark Core is a general execution engine for spark platform and it is a vital element upon which all other functionality is built upon. It provides In-Memory computation and referencing datasets in external storage systems.

Spark SQL

Spark SQL lets us query structured data inside Spark programs, either using SQL or a familiar DataFrame API. Usable in Java, Scala, Python and R. We can use spark SQL to run SQL or HiveQL queries on existing warehouses. It also allows JDBC and ODBC connectivity to query big data

Spark Streaming

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics.

MLlib

MLIb is a Machine Learning Library which fits into Spark's APIs and interoperates with NumPy in Python and R libraries. Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration and can yield better results than the one-pass approximations sometimes used on MapReduce [1].

GraphX

GraphX is a distributed graph-processing framework on top of Spark. It provides an API for expressing graph computation that can model the user-defined graphs by using Pregel abstraction API. GraphX competes on performance with the fastest graph systems while retaining Spark's flexibility, fault tolerance, and ease of use [1].

Resilient distributed datasets

Resilient Distributed Datasets[RDD] is a fundamental data structure of spark. It is a permanent distributed collection of data objects.

The features of RDDs are: -

Resilient: - It means reliable i.e. fault-tolerant so able to recompute missing or damaged partitions due to failure of nodes

Distributed: - data residing on multiple nodes in a cluster.

Dataset: - is a collection of partitioned data

CHAPTER 4
PROPOSED APPROACH

4.1 INTRODUCTION

In this section we provide a design of remote health monitoring system for early detection of heart diseases. The IOT based RHM system consists of a four-tier architecture to store and process huge volume of IOT sensor data. Tier-1 focuses on collection of health data through wearable sensor devices. Tier-2 focuses local data storage at patients' premises that interfaces between sensors. Tier-3 uses scalable storage (centralized data repository) to store the huge volume of data sent from local storage at patients' premises. Tier-4 uses Apache Spark to develop the decision support system for treatment of heart diseases.

4.2 FOUR TIER ARCHITECTURE OF IOT BASED RHM SYSTEM

A detailed description of the four-tier architecture is provided below: -

Tier 1: Data collection

Data collection block is used for collecting patients' health data by attaching various wearable IOT sensor devices to the patients' body. The rapid growth in technology has helped to measure various physiological parameters of body easily. There are various types of smart devices available in online shopping sites, with the help of these wearable smart devices one can easily measure heart rate, blood pressure, body temperature, oxygen saturation. A smart phone camera can easily be used to collect data about open wounds, eye disorder, skin disorder etc. The data collected from various sensor devices are send to a local storage via Bluetooth, Infrared (IR) light or any other wireless network.

Tier 2: Local Storage

The data collected through smart devices are sent to local storage which is located at patient's premises. The term local storage can be used for private datacentre, smartphones, personal computers, virtual machines or any other computing device. The data stored here is pre-processed before sending it to the centralized repository. When the clinical measure of patient exceeds its normal value, the device sends an alert message with the clinical value to the doctor and care holder.

Tier 3: Centralized Data repository

The data sent form Local storage is sent to a Centralized Data repository for permanent storage. IOT devices have the objective of sending clinical measures continuously. It is very difficult to store such a huge amount of data using traditional storage mechanisms. Therefore, we use a cloud storage technology to store the data in distributed fashion. Apache HBase is a popular open-source, distributed, non-relational database. An account is created with Amazon, Google or Microsoft to get the virtual machines with Apache HBase database.

Tier 4: Data analytics

Data Analytics block is used for development of machine learning based decision support model. Apache Spark based is used to develop machine learning based decision support system.

Machine learning algorithms used are SVM, Multi-Layer Perceptron and logistic regression. The results show that applying SVM yields a better accuracy on heart disease dataset and Multi-Layer Perceptron yields a better accuracy on arrhythmia dataset.

4.3 MACHINE LEARNING BASED DECISION SUPPORT SYSTEM

The proposed method is illustrated in figure 5 given below, is implemented using Apache Spark framework. Details about Apache Spark is mentioned in Chapter 3.6 of this project report. Since we have considered a binary classification for Heart Disease Dataset [mentioned in chapter 3.1 of this project report], linear support vector machine, Multi-Layer Perceptron and binomial logistic regression is used. For Arrhythmia Dataset [mentioned in chapter 3.1 of this project report] we have considered a multilabel classification, Multinomial logistic regression, Multi-Layer Perceptron and combination of linear support vector machine and one-vs-rest classifier is used.

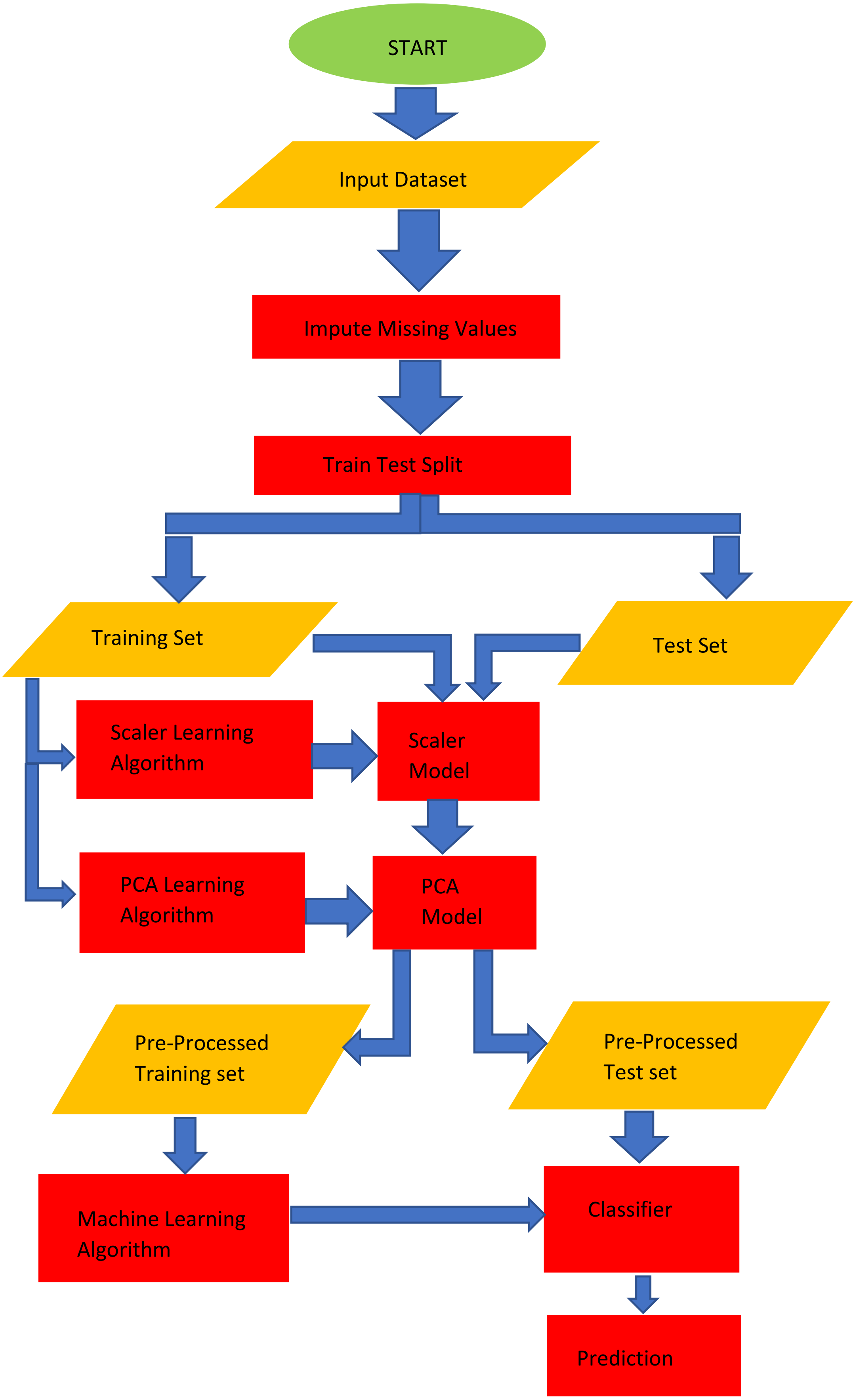

Figure 5- Work Flow Diagram of proposed decision support system

A detailed step-wise explanation to develop machine learning based decision support system is provided below: -

Input: A Heart Disease or Arrhythmia Dataset as DB
Output: An efficient classifier for treatment of heart disease

Step 1 Impute the missing values present in the Dataset DB, either using the mean or the median of the columns in which the missing values are located.

Step 2 Split the dataset DB randomly in two parts. The first part as the training set (80 % of DB) and second part as the test set (20 % of DB)

Step 3 Train a MaxAbsScaler (in Apache Spark MLlib) model using the training set (obtained in step 2). The scaler model rescales each feature to range [-1, 1] by dividing through the maximum absolute value in each feature.

Step 4 Train a Principal Component Analysis(PCA) model using the training set (obtained in Step 2). To choose the optimal value of k refer to dimensionality reduction [mentioned in chapter 3.2 of this project report]

Step 5 Pass the training set obtained in Step 2 first through the scaler model and then through the PCA model. A pre-processed training set is obtained as result.

Step 6 Pass the test set obtained in Step 2 first through the scaler model and then through the PCA model. A pre-processed test set is obtained as result.

Step 7 Train a machine Learning algorithm (SVM, logistic regression or Multi-Layer Perceptron) using the pre-processed training set. A classifier is obtained as a result.

Step 8 Evaluate the performance of the classifier using the pre-processed test set and various evaluation metrices.

Step 9 End

CHAPTER 5
IMPLEMENTATION

5.1 INTRODUCTION

We have evaluated our machine learning based decision support system on local Spark-2.2.0 cluster installed at personal laptop running Ubuntu 17.04 64bit, where the laptop has dual core Intel i3 processors running at 2.10 GHz and RAM 8 GB.

Machine Learning Algorithms used in our experiment are Support Vector Machine, Multi-Layer Perceptron and Logistic Regression. Datasets used are Cleveland heart disease, Cleveland and Switzerland mixed heart disease, arrhythmia. Tool used is Apache Spark.

We have used 80% of the dataset as training set and the rest 20% is used as test set. Therefore, no of training and test instances for the three datasets are as follows:

Table 4- Number of training and test instances for the three datasets

	Cleveland	Cleveland+switzerland	Arrhythmia
Total Instances	303	426	452
Train instances	244	338	361
Test instances	59	88	91

5.2 PERFORMANCE EVALUATION METRICES

Recall or True positive rate=TP/(TP+FN)

False positive rate=FP/(FP+TN)

Precision or positive predicted value=TP/(TP+FP)

Negative predicted value=TN/(TN+FN)

Accuracy=(TP+TN)/(TP+FP+FN+TN)

Receiver operating characteristics (ROC): -Receiver operating characteristics (ROC) or ROC curve, is a graphical plot that explains the performance of a binary classifier system and is created by plotting the true positive rate against the false positive rate.

Area under the curves (AUC): - It is the area below the ROC curve.

Precision Recall Curve: - A system with high recall but low precision returns many results, but most of its predicted labels are incorrect when compared to the training labels. A system with high precision but low recall returns very few results, but most of its predicted labels are correct when compared to the training labels. An ideal system with high precision and high recall will return many results, with all results labelled correctly. Thus, the precision recall curve shows this relationship.

Table 5- A Confusion Matrix for binary classification problem

	Predicted class=1	Predicted class=0
Actual class=1	TP	FN
Actual class=0	FP	TN

where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives respectively.

CHAPTER 6

RESULTS AND DISCUSSION

In this section we will see the results our machine learning based decision support system.

Note: - Here K represents no of principal components.

6.1 Experimental Results

Table 6- Classification Results of Cleveland heart disease dataset

Classifier	K	Reg Param	Max Iterations	Accuracy	AUC	TPR	FPR	PPV	NPV
SVM	8	0.0	25	0.88	0.87	0.79	0.06	0.92	0.93
Logistic	8	0.1	20	0.84	0.85	0.79	0.1	0.88	0.9
MLP	5	Layers (5, 2)	20	0.83	0.83	0.79	0.13	0.85	0.86

Table 7- Confusion Matrix of Cleveland heart disease dataset using Support Vector Machines

	Predicted class=1	Predicted class=0
Actual class=1	24	5
Actual class=0	2	28

Table 8- Confusion Matrix of Cleveland heart disease dataset using Logistic Regression

	Predicted class=1	Predicted class=0
Actual class=1	23	6
Actual class=0	3	27

Table 9- Confusion Matrix of Cleveland heart disease dataset using MLP

	Predicted class=1	Predicted class=0
Actual class=1	23	6
Actual class=0	4	26

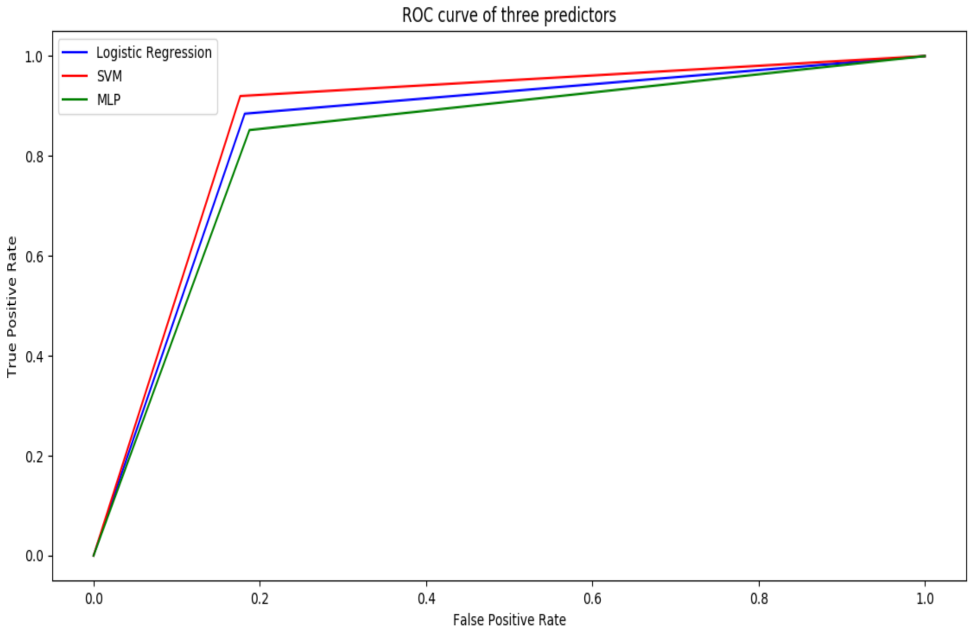

Figure 6-ROC curve of three predictors on cleveland heart disease dataset

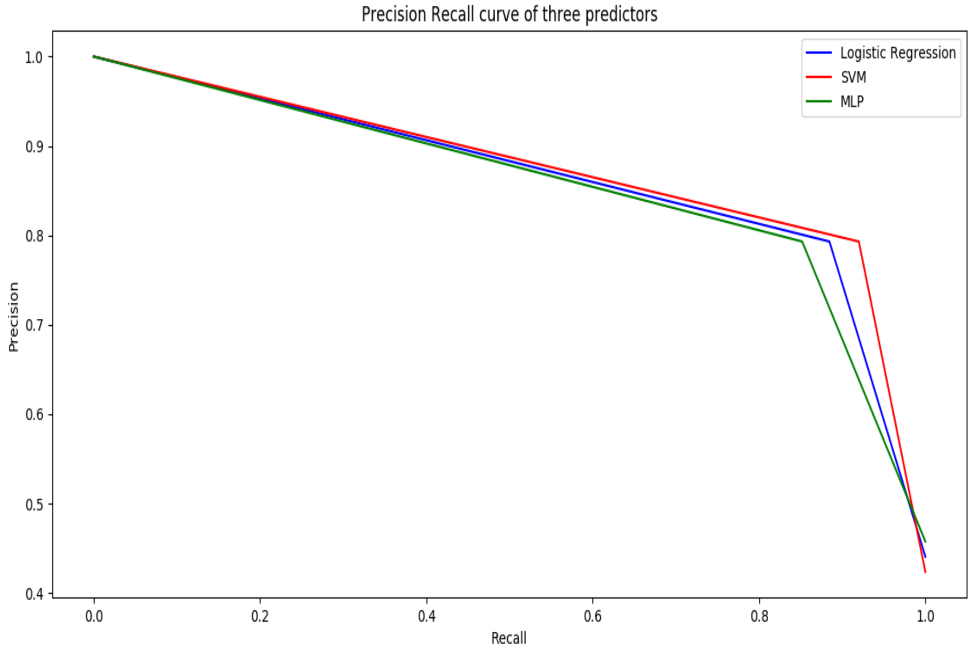

Figure 7-Precision Recall curve of three predictors on cleveland heart disease dataset

Table 10- Classification Results of Cleveland + Switzerland (Mixed) Heart Disease Dataset Results

Classifier	K	Reg Param	Max Iterations	Accuracy	AUC	TPR	FPR	PPV	NPV
SVM	5	0.6	25	96.59	0.95	0.96	0.03	0.98	0.96
Logistic	8	0.0	20	94.31	0.93	0.94	0.06	0.96	0.93
MLP	8	Layers (8, 2)	20	94.31	0.93	0.94	0.06	0.96	0.93

Table 11- Confusion Matrix of Cleveland + Switzerland (Mixed) Heart Disease using Support Vector Machines

	Predicted class=1	Predicted class=0
Actual class=1	57	2
Actual class=0	1	28

Table 12- Confusion Matrix of Cleveland + Switzerland (Mixed) Heart Disease using Logistic Regression

	Predicted class=1	Predicted class=0
Actual class=1	56	3
Actual class=0	2	27

Table 13-- Confusion Matrix of Cleveland + Switzerland (Mixed) Heart Disease using MLP

	Predicted class=1	Predicted class=0
Actual class=1	56	3
Actual class=0	2	27

Table 14- Classification Results of arrhythmia dataset

Classifier	k	Reg Param	Max Iterations	Accuracy
SVM	60	0.2	30	0.74
Logistic	65	0.0	25	0.70
MLP	60	Layers (60,48,16)	30	0.75

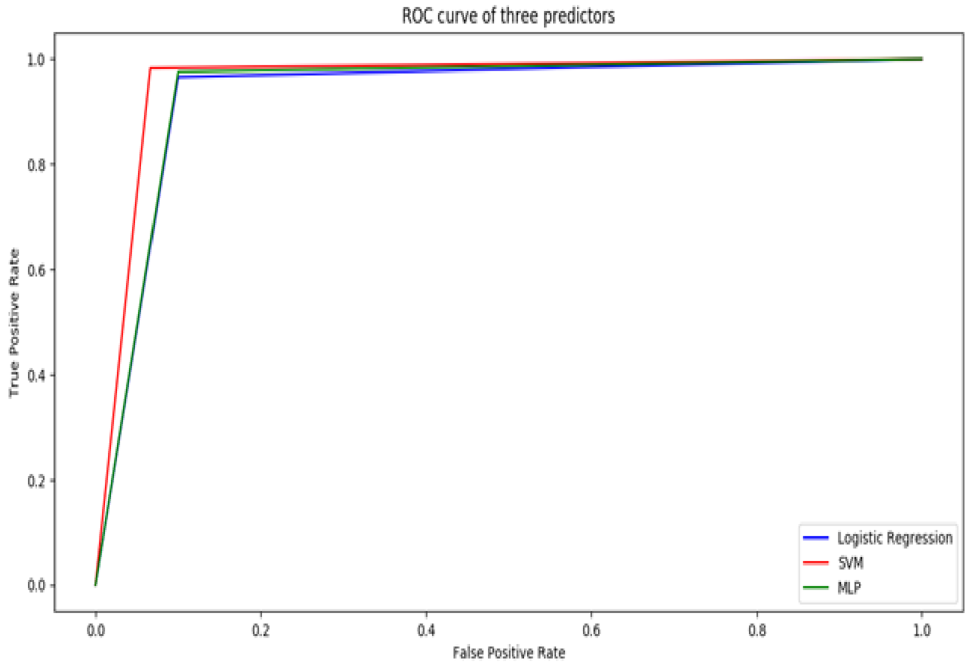

Figure 8- ROC curve of three predictors on cleveland+switzerland(Mixed) heart disease dataset

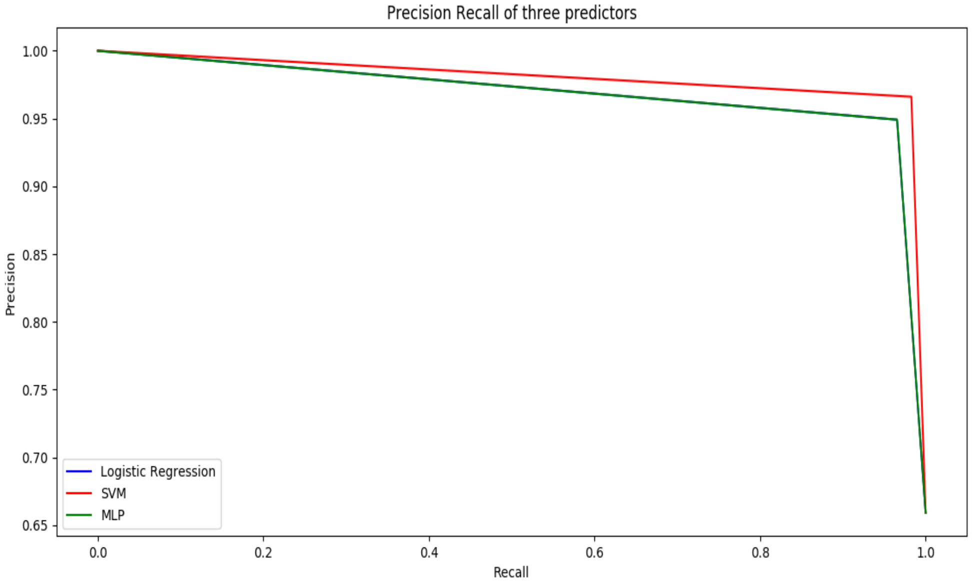

Figure 9- Precision Recall curve of three predictors on cleveland+switzerland (Mixed) heart disease dataset

6.2 Discussion

For the experiment taken place in the project, For the first experiment Cleveland heart disease database was chosen. The literature review of this project discusses about the accuracy obtained by different researchers and whose work is published by a reputed publisher. As can be seen in Table 6, 88% classification accuracy is obtained using svm classifier. Table 6,7,8,9 highlights the result obtained using different evaluation metrices on the Cleveland datasets. Figure 6 shows the ROC curve of the three-classification model on Cleveland dataset. As it can be seen the curve of SVM has the highest area under the cover, so we can say SVM performs better than the other two models. Figure 7 shows the precision-recall curve on the same dataset. It can be seen that precision of SVM drops at a higher stage of recall compared to other predictors.

To obtain a better accuracy we mixed the Cleveland and Switzerland heart disease dataset and performed our second experiment on the mixed data set. As can be seen in Table 10, 97% classification accuracy is obtained using svm classifier. Table 10,11,12,13 highlights the result obtained using different evaluation metrices on the Cleveland datasets. Figure 8 shows the ROC curve of the three-classification model on the said dataset. As it can be seen the curve of SVM has the highest area under the cover, so we can say SVM performs better than the other two models. Figure 9 shows the precision-recall curve on the same dataset. As it can be seen precision-recall curve for Logistic regression and MLP are same, so no separate curve for logistic regression is showed in the graph. Precision of SVM is higher than other two models at recall level 0.3 to 0.9 and drops at a higher stage of recall compared to other predictors.

We performed out third experiment on arrhythmia dataset. Researchers from Selcuk University [19] have used two classes as the presence or absence of arrhythmia in their experiments and classification accuracy obtained by them is 100%. In our experiment we have considered a multi-label classification. As can be seen in Table 14, 75% classification accuracy is obtained using multi-layer perceptron classifier. In this case MLP performs better than other two classifiers.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Machine Learning applications, particularly decision support systems, need classifiers which have higher accuracy. Such applications in general require a three-step process: (i) Pre-process the data (ii) Relevant feature selection (iii) a high accuracy classifier to obtain the highest classification performance.

In this project work we did not evaluate the effect of feature selection algorithm on our classifier performances. Instead, we used a scaler model to rescale each feature to range [-1, 1] by dividing through the maximum absolute value in each feature and used principal component analysis (PCA) algorithm to reduce the dimension of the dataset. The reduced dataset was used to train machine learning based classifiers. We found that SVM based classifier performed better on Cleveland and Cleveland+Switzerland (mixed) heart disease dataset having an accuracy of 88% and 97% respectively, MLP based classifier performed better on arrhythmia dataset having and accuracy of 75%. In addition, we also proposed a scalable IOT based four-tier architecture remote health monitoring system which uses the decision support system in its data analytics block.

7.2 FUTURE SCOPE

In future we plan to improve the accuracies of classifier by using classifier ensemble approaches

REFERENCES

[1] Apache Spark, https://spark.apache.org

[2] Apache Hadoop, http://hadoop.apache.org/

[3] Heart Disease Dataset, http://archive.ics.uci.edu/ml/datasets/heart+Disease

[4] Principal Component Analysis, https://en.wikipedia.org/wiki/Principal_component_analysis

[5] Tan, P., Steinbach, M. and Kumar, V. (2016). *Introduction to Data Mining*. 1st ed. Noida Uttar Pradesh (India): Pearson.

[6] Sudhakar Singh, Pankaj Singh, Rakhi Garg and P K Mishra, "Big Data: Technologies, Trends and Applications", In: International Journal of Computer Science and Information Technologies, Vol. 6(5), 2015

[7] Hassanalieragh M, Page A, Soyata T, Sharma G, Aktas M, Mateos G, Kantarci B, Andreescu S (2015) Health monitoring and management using internet-of-things (iot) sensing with cloud-based processing: Opportunities and challenges. In: 2015 IEEE international conference on services computing (SCC), IEEE

[8] Apache Spark, https://en.wikipedia.org/wiki/Apache_Spark

[9] Apache Spark, https://databricks.com/spark/about

[10] Apache Hadoop, https://en.wikipedia.org/wiki/Apache_Hadoop

[11] Design of a hybrid system for the diabetes and heart diseases Humar Kahramanli *, Novruz Allahverdi Department of Electronic and Computer Education, Selcuk University, Konya, Turkey.

[12] Das, Resul & Turkoglu, Ibrahim & Sengur, Abdulkadir. (2009). Effective diagnosis of heart disease through neural networks ensembles. Expert Syst. Appl.. 36. 7675-7680. 10.1016/j.eswa.2008.09.013.

[13] Priyan, M.K. & Gandhi, Usha. (2017). A novel three-tier Internet of Things architecture with machine learning algorithm for early detection of heart diseases. Computers & Electrical Engineering. 65. 10.1016/j.compeleceng.2017.09.001.

[14] Malan, David, Thaddeus Fulford-Jones, Matt Welsh, and Steve Moulton. 2004. CodeBlue: An ad hoc sensor network infrastructure for emergency medical care. Paper presented at the International Workshop on Wearable and Implantable Body Sensor Networks, April, London, UK.

[15] Asha Rajkumar, G.Sophia Reena, Diagnosis Of Heart Disease Using Datamining Algorithm, Global Journal of Computer Science and Technology 38 Vol. 10 Issue 10 Ver. 1.0 September2010.

[16] A. Ozcift and A. Gulten, “Classifier ensemble construction with rotation forest to improve medical diagnosis performance of machine learning algorithms,” Comput. Meth. Prog. Bio., vol. 104, no. 3, pp. 443–451, Dec. 2011.

[17] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," in IEEE Access, vol. 3, pp. 678-708, 2015. doi: 10.1109/ACCESS.2015.2437951

[18] M. Hassanalieragh et al., "Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-Based Processing: Opportunities and Challenges," 2015 IEEE International Conference on Services Computing, New York, NY, 2015, pp. 285-292. doi: 10.1109/SCC.2015.47

[19] K. Polat, S. Gunes, Detection of ECG Arrhythmia using a differential expert system approach based on principal component analysis and least square support vector machine, Appl. Math. Comput. 186 (2007) 898–906

[20] Arrhythmia Data, https://archive.ics.uci.edu/ml/datasets/arrhythmia

[21] Heart Disease, https://www.medicalnewstoday.com/articles/237191.php

[22] Multi-Layer Perceptron, http://neuralnetworksanddeeplearning.com/chap1.html

[23] Support Vector Machines, https://en.wikipedia.org/wiki/Support_vector_machine

[24] Logistic Regression, https://en.wikipedia.org/wiki/Logistic_regression

[25] Principal Component Analysis, https://liorpachter.wordpress.com/2014/05/26/what-is- principal-component-analysis/

[26] Apache Spark, https://www.tutorialspoint.com/apache_spark/apache_spark_rdd.htm