

PRIVACY PRESERVING DISEASE DETECTION USING NEURAL NETWORKS

A Project Report

*Submitted in partial fulfillment of
the requirements for the award of the degree of*

BACHELOR OF TECHNOLOGY

by

Batlanki Kanaka Pravallika

B170405EC

Dowluri Satya Ashok

B170962EC

J Sharan

B170823EC

Krishnam Veera Venkata Pramod Chaitanya

B170837EC

Under the guidance of
Dr. Deepthi P.P.



Department of Electronics and Communication Engineering

NATIONAL INSTITUTE OF TECHNOLOGY CALICUT

Calicut, Kerala, India – 673 601

2020-2021

Acknowledgments

We take this opportunity to express our deepest gratitude to everyone who supported us through this online semester, and helped and motivated us in completing this project.

We sincerely express our whole hearted gratitude to our project guide **Dr. Deepthi P.P.**, for helping and guiding us through this project Without whom, we would not have been able to complete the project successfully.

We are very thankful to the research scholars, **Ms. Sona Alex** and **Dr. Aneesh M. Koya** for their invaluable insight into the project and directing us through the project.

We would like to thank the Project Co-ordinator **Mr.Jaikumar M G** and the evaluation committee for all the valuable suggestions put forward during the initial evaluations.

We would like to acknowledge the Department of Electronics and Communication Engineering, National Institute of Technology, Calicut, for extending its full support during the entirety of the project and helping us through the online semester.

We would also like to thank our beloved parents who supported us and have created a working environment at our homes in these difficult times.

Declaration

We hereby declare that except where specific reference is made to the work of others, the contents of this project report are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This project report is our own work and does not contain any outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

B K Pravallika (B170405EC)

D Satya Ashok (B170962EC)

J Sharan (B170823EC)

K V V Pramod Chaitanya (B170837EC)

NIT Calicut

Date: 14/05/2021

DEPARTMENT OF ELECTRONICS AND COMMUNICATION
ENGINEERING



Certificate

This is to certify that the project report entitled **Privacy Preserving Disease Detection Using Neural Networks** submitted by **B K Pravallika (B170405EC)**, **D Satya Ashok (B170962EC)**, **J Sharan (B170823EC)**, **K V V Pramod Chaitanya (B170837EC)** to National Institute of Technology Calicut for the award of the degree of Bachelor of Technology in Electronics and Communication Engineering, is a bonafide record of the project work carried out by them under my supervision and guidance. The content of the project report, in full or parts have not been submitted to any other institute or university for the award of any degree or diploma.

Dr. Deepthi P.P.
(Project Guide)
Dept. of Electronics and
Communication Engineering
NIT Calicut

Dr. Deepthi P.P.
(Professor and Head of Department)
Dept. of Electronics and
Communication Engineering
NIT Calicut

Date: 14/05/2021

(Office seal)

Abstract

Electrocardiogram (ECG) is a simple and cost efficient method used to diagnose heart disorders by detecting the electrical impulses from the heart. Bundle Branch Block (BBB) is a heart disorder caused by the blockage in the human heart. This disorder causes fluctuations in the ECG waves, from which it can be detected. Nowadays, since most of the operations happen in the cloud due to requirement for greater computation power, there is always a concern for privacy of data. So, we propose a privacy preserving disease detection model based on Neural Networks for the automatic detection of Left and Right Bundle Blocks. It includes computation of 5-statistical features obtained from the extracted QRS complex of the ECG wave. It uses CKKS Homomorphic encryption scheme to encrypt the computed features and sent to the cloud for detection using Neural Network.

The proposed technique uses CPSC 2018 Dataset and obtained an accuracy of 84.3 %

List of Abbreviations

ECG	Electro Cardiogram
BBB	Bundle Branch Blocks
RBBB	Right Bundle Branch Blocks
LBBB	Left Bundle Branch Blocks
SG filter	Savitzky-Golay Filter
NN	Neural Netowork
ANN	Artificial Neural Networks
WNN	Wavelet Neural Networks
CKKS	Cheon-Kim-Kim-Song
BGV	Brakerski-Gentry-Vaikuntanathan
BFV	The Brakerski/Fan-Vercauteren
HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
SWHE	Somewhat Homomorphic Encryption
PHE	Partially Homomorphic Encryption
CPSC	China Physiological Signal Challenge
PYFHEL	Python For Homomorphic Encryption Library

Contents

1	Introduction	1
1.1	Motivation	2
1.2	Challenges	2
2	Literature Review	4
3	Theory	6
3.1	Introduction	6
3.1.1	The Electricity Of The Heart	6
3.1.2	Electrocardiogram (ECG)	7
3.1.3	Bundle Branch Blockages	9
3.2	Filters Used In Feature Extraction	10
3.2.1	Median Filter	10
3.2.2	SG Filter	10
3.3	Neural Networks	10
3.3.1	Multi Layer Perceptron (MLP)	10
3.3.2	Wavelet Neural Networks (WNN)	11
3.3.3	Activation Functions	12
3.3.4	Loss Functions	13
3.3.5	Optimization Algorithms	13
3.3.6	Tensorflow	14
3.4	Expansions Used For Approximations	14
3.4.1	e^x Expansion	14
3.4.2	$\cos(x)$ Expansion	14
3.4.3	$\frac{1}{1+e^{-x}}$ Expansion	14
3.5	Homomorphic Encryption	15
3.5.1	Introduction To Homomorphic Encryption	15
3.5.2	Types Of Homomorphic Encryption	15
3.5.3	Limitations Of Homomorphic Encryption:	16
3.6	CKKS Encryption Scheme	16
3.6.1	Encoding And Decoding	17

3.6.2	Encryption And Decryption	18
4	Database and Software	20
4.1	Dataset	20
4.2	Major Libraries Used	20
4.2.1	Scipy Library	20
4.2.2	Keras	21
4.3	Microsoft Seal	21
4.3.1	Pyfhel Library	21
4.4	Computation Platform	22
5	Block Diagram	23
6	Methodology	25
6.1	QRS - Complex Detection	25
6.2	Feature Extraction	28
6.3	Classification	28
6.3.1	Classification Using WNN	28
6.3.2	Privacy Preserving Classification	29
7	Results	34
7.1	Feature Extraction	35
7.2	Classification Using ANN And WNN	35
7.3	Classification With Approximated Activation Functions	36
7.4	Classification Using Privacy Preserving Classifier	38
8	Conclusion	40
9	Future Works	41
	Bibliography	42

List of Figures

3.1	ECG Nodes and Wiring diagram of the Heart	7
3.2	Ideal ECG graph	7
3.3	Parts of the QRS complex [13]	8
3.4	A Sample ECG plotted on an ECG paper [13]	8
3.5	CKKS scheme	17
5.1	Block Diagram of the proposed model	23
6.1	QRS interval extraction procedure	27
6.2	The implemented WNN	29
6.3	The implemented privacy preserving neural network	33
7.1	Sample ECG from CPSC 2018 dataset	34
7.2	Loss and Accuracy plot using WNN	36
7.3	Accuracy vs number of layers, nodes in each layer	37
7.4	Loss and Accuracy plot using Approximated Activation functions	38

List of Tables

3.1	Comparision of RBBB and LBBB	9
6.3	Original and Approximated functions	30
6.4	Plots of the original and approximated functions	31
7.1	Number of Patterns used for testing, validation and training of the model	34
7.2	Features Extracted from ECG of a normal, a LBBB and a RBBB patient	35
7.3	Accuracies of the model using different activation functions	36
7.4	Maximum possible Train, Validation, Test Accuracies of model with Normal Activation functions vs Approximated Activation functions	37
7.5	Test Accuracy of model on encrypted data and noise Budget Available with varying parameters.	39

Chapter 1

Introduction

According to the World Health Organisation, an estimated 17.9 million people die each year due to Heart Diseases which is 31% of all the deaths worldwide. India has even a higher rate of mortality compared to global average. The 1.21 billion people of this country are experiencing mortality due to Cardiovascular diseases at an increased rate. So in this work we attempt to detect some heart diseases from the ECG data using neural networks with great accuracy levels in such a way that the privacy and security of the data are preserved during cloud based analysis operations.

Electrocardiogram (ECG) records the electrical signal from the heart. Electrodes are placed on the chest to record the heart's electrical signals, which records heart beat. ECG signal consists of a P,T-waves and a QRS-complex. The P-wave represents atrial depolarization, the QRS-complex and T-wave the ventricular depolarization and repolarization. Duration and amplitude of QRS-complex, PR-interval, RR-interval, ST-segment, and QT-interval are the parameters to be taken care of for diagnosis of heart diseases. Bundle branch block is a state in which there's a discontinuation or clog along the path that electrical pulses travel to make a heart beat. At times it makes it tough for the heart to pump blood efficiently through the body. BBB is of two types Left Bundle Branch Block(LBBB) and Right Bundle Branch Block(RBBB). If LBBB is present the left ventricle of the heart is delayed, which causes the left ventricle to contract after the right ventricle. For RBBB the right ventricle is not directly activated by impulses travelling through the right bundle branch. When there is a LBBB QRS duration will be greater than 120 milliseconds, Q wave will be absent in leads I, V5 and V6, T wave displacement will be opposite in direction to the major deflection of the QRS complex and there will be a monomorphic R wave in I, V5 and V6 and ST. When there is a RBBB QRS duration will be greater than 120 milliseconds ,slurred S waves in leads I, aVL and frequently in V5 and V6 and there will be rsR' "bunny

ear” pattern in the anterior precordial leads (leadsV1-V3) [1]

The work we presented includes QRS detection using novel pre-processing techniques and kurtosis based enhanced efficiency. Since the neural network based data analysis operations are computationally intensive, the users will have to send the private medical data to the third party cloud for reducing the power consumption in the lightweight edge devices. The main concern of classification in untrusted 3rd party cloud is that there is a chance to leak patients’ private data. So, to overcome this privacy issue, It is required to encrypt the patient’s private data and then transmit it to the cloud for classification. Homomorphic encryption allows the user to perform computations on encrypted data without decrypting it. There are different Homomorphic encryption schemes which can support data processing in encrypted domain..

1.1 Motivation

For patients having previous record of heart problems, it is crucial to monitor the heart in real-time as the diseases related to heart are very fatal and any delay in treatment could lead to worst consequences. Also the detection system should be very accurate as the person using the system is absolutely relying on it for real-time monitoring.

Inorder to increase the resource allocations and system functioning, systems based on IoT should adapt to system’s configuration based on user’s requirements. Data analytic approaches such as AI and ML are needed to improve the decision-making approaches in the IoT systems. The choice of suitable encryption schemes and pre-processing techniques highly influence the data security, delay and accuracy in detection operations.

Therefore in this work, QRS detection using novel pre-processing techniques and kurtosis based enhanced efficiency are used for feature extraction. Wavelet neural networks are used to get the high accuracy levels while CKKS homomorphic encryption scheme is used for privacy preserved computations.[2]

1.2 Challenges

IoT systems are resource constrained. So, Computations on large amount of the ECG data that is being generated is very difficult in IoT based systems. There is always a trade - off between power available and accuracy in case of IoT systems.

Data security during computations in third party cloud is also a matter of concern because several organizations can misuse the patients data. Our work proposes a technique that performs operations and classifies encrypted data using Neural Network without decrypting it thereby enhancing the security and privacy of data. [3]

Chapter 2

Literature Review

An extensive survey of literature available in the relevant research areas is carried out. Major related works are described here.

Continuous Wavelet Transform (CWT) was found to be useful by Ilic¹⁶ to identify LBBB and RBBB.[4]

Wavelet neural network(WNN) having Morlet and Mexican hat wavelet functions as hidden layer activation functions was tested by Ceylan et al and obtained accuracies of 97.9% for LBBB and 99.2% for RBBB.[5]

Hybrid techniques such as Bacterial Forging-Particle Swarm Optimization (BF-PSO) for feature extraction with Levenberg-Marquardt neural network for classification was proposed by Kora and Kalva. They got accuracies of 98.3% for LBBB, 98.15% for RBBB and 98.1% for normal condition.They utilized 20 statistical features for classification, whereas our work uses only 5 statistical features and trained on huge data of 2000 patient records and nearly 52000 patterns were extracted from these 2000 records for the classification[6]

A neural network structure on fuzzy clustering was proposed by Yuksel and Bekir for BBB detection. But for mixed classification FCNN was found to give better recognition rates.[7]

A combination of minimum distance classifier ,weighted linear discriminant classifier and support vector machine (SVM) for heartbeat classification was proposed by Huang et al.¹⁹ . Obtained results have 91.4% sensitivity ,37.3% positive predictive value for LBBB and 92.8% sensitivity ,88.8% positive predictive value for RBBB.[8]

QRS detection using novel pre-processing techniques and kurtosis based enhanced efficiency was tested on FTD, MIT-AD databases. For FTD, accuracy is high with 99.90% Se, 99.91% +p, 0.19% DER and 99.81% Ac. For MIT-AD also the results were acceptable with 99.50% Se, 99.56% +p, 0.93% DER and 99.08% Ac.[9]

Z. Brakerski, C. Gentry and V. Vaikuntanathan has proposed FHE schemes such as BGV and BFV for homomorphic arithmetic on encrypted integers, and CKKS for the same on encrypted real numbers.[10]

Chen et al proposed bootstrapping with CKKS to perform more number of computations on ciphertext but CKKS with bootstrapping is said to be unstable in practical applications. CKKS without bootstrapping is more suited for machine learning models because of its support for floating numbers and stability.[11]

Owusu-Agyemang, Kwabena & Qin, Zhen & Zhuang, Tianming & Qin, Zhiguang. (2019) have shown that Homomorphic encryption is useful in Privacy preserving classification using NN, as we can perform arithmetic operations on the encrypted data without decrypting it. But the number of operations possible is often limited to few such as multiplication. This limits the activation functions that can be used in our neural network. Any activation function which is not a linear must be approximated to a polynomial to be used in our model. This in turn affects the accuracy of our model.[12]

Chapter 3

Theory

3.1 Introduction

World wide survey has clearly indicated that heart disease is an important cause for death.

So it is of utmost importance to find an accurate method for the early diagnosis of Heart diseases.

3.1.1 The Electricity Of The Heart

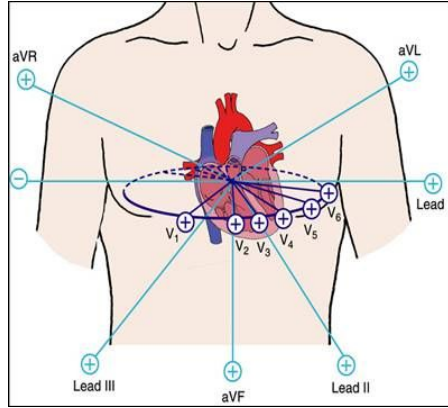
When electrodes are attached to the body's surface as shown in Figure 3.1(a), one can observe electrical changes(depolarization) which are associated with muscular contractions. Electrical changes related to the heart muscle contraction will only be visible if the patient is in fully relaxed state.

In the electrical point of view, heart as only two chambers since both the atria contract together which is then followed by both the ventricles contracting together.

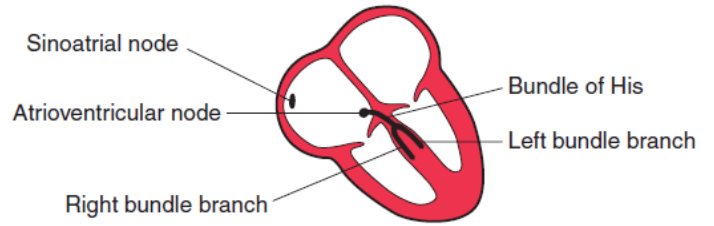
A labelled diagram of cross-section of heart is shown in Figure 3.1(b). At Sinoatrial(SA) node in the right atrium, cardiac cycle's electric discharge begins. Then through fibres of atrial muscle, the depolarization spreads. There is a special region in atrium where there would be a pause as depolarization spreads through atrioventricular(AV) node. And then it moves through the 'Bundle of His' which is divided into left and right bundle branches in the septum between the ventricles. Within the ventricular muscle, the left bundle branch splits into two sections and then the conduction spreads a bit more slowly through 'Purkinje Fibres'

Other than the SA node, electrical activation of the heart may start anywhere.

The part of the heart that controls the activation sequence is referred to as the 'rhythm'. Sinus rhythm refers to the natural heart rhythm that begins with electrical activation in the SA node.



(a) ECG Nodes



(b) Wiring Diagram of Heart [13]

Figure 3.1: ECG Nodes and Wiring diagram of the Heart

3.1.2 Electrocardiogram (ECG)

The ECG is crucial for diagnosing irregular heart rhythms and, as a result, managing them. It aids in the detection of the cause of chest pain, and it is essential for the proper use of early intervention in myocardial infarction. It may aid in the diagnosis of dizziness, syncope, and shortness of breath.

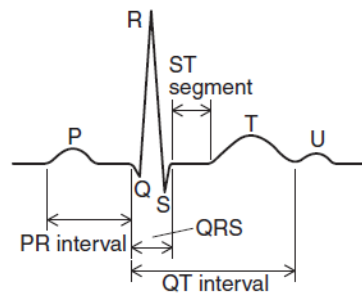


Figure 3.2: Ideal ECG graph

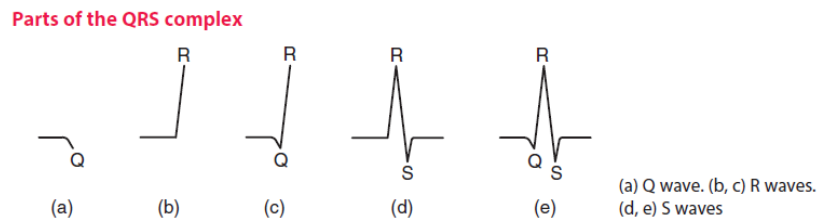


Figure 3.3: Parts of the QRS complex [13]

Parts of ECG

From the Figure 3.2 We can find a number of changes in the ECG wave from the axis. The contraction of atria causes P wave, depolarisation of ventricles causes QRS complex and the ventricles returning to resting state causes T wave. An extra U wave can sometimes be seen just after T wave.

By looking at the abnormality in distances between the different parts of the ECG wave, we can diagnose the specific problems in the heart.

The parts of the QRS complex i.e. Q, R, S are shown in detail in Figure 3.3

ECG is normally recorded by drawing a trace on a paper with medium sized squares, where each square of side 5mm represents 0.2 seconds.

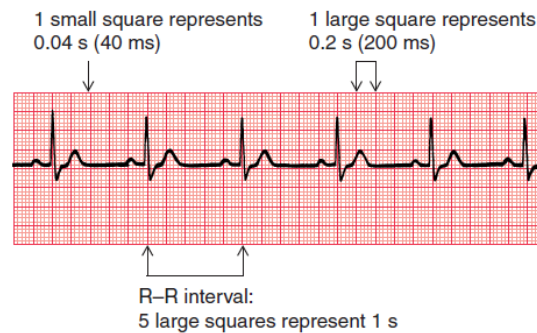


Figure 3.4: A Sample ECG plotted on an ECG paper [13]

3.1.3 Bundle Branch Blockages

Bundle Branch Blockage or Heart block is a heart condition caused due to a delay or a blockage in the path of electrical impulses that are required for the heart to beat.

There are 3 types of Heart blocks:

- **First degree heart block** : It is the least serious of the three. It occurs when there is a delay in the electrical impulses reaching the ventricles.
- **Second degree heart block** : It occurs when some of the electrical impulses are lost on the way, leading to skipped beats.
- **Third degree heart block** : No electrical impulses reach the ventricles. It is very serious and a pacemaker is absolutely necessary.

If the blockage occurs on the left side of the heart, it is called LBBB if it occurs on the right side of the heart, it is called RBBB.

Table 3.1: Comparision of RBBB and LBBB

Right Bundle Branch Block	Left Bundle Branch Block
Wide QRS complexes (160 ms)	Wide QRS complexes (160 ms)
Sinus rhythm with rate 60/min	Sinus rhyth with rate 100/min
Normal PR interval	Normal PR interval
Normal Cardiac axis	Normal Cardiac axis
RSR1 pattern in lead V1 and deep, wide S waves in lead V6	M pattern in the QRS complexes, best seen in leads I, VL, V5, V6
Normal ST segments and T waves	Inverted T waves in I, II, VL leads

3.2 Filters Used In Feature Extraction

3.2.1 Median Filter

Median filter is a nonlinear filter, that replaces each value in a signal with the median of the neighbouring values. The window of neighbouring values used to calculate the median can be changed, to adjust the noise removal and smoothening of the wave. It is mostly used to remove noise from the signal. It preserves the corners and steps in the signal.

3.2.2 SG Filter

Savitzky Golay filter fits successive subsets of equally spaced datapoints with a low order polynomial (varies from order 2 to 6) using the method of least squares. The resultant output is a smoothened version of the input. It is mostly used to smoothen the signal.

3.3 Neural Networks

Neural networks(NN) are analogous to the nerve cells called neurons present in our brain. Neural networks are implemented by mimicking the functions of neurons. Machine learning models doesn't have the capability to extract the underlying features in the data whereas neural networks have the ability to extract the underlying features in the data by adding non-linearity to it.

Now a days, Deep Neural Networks are being used everywhere. It has wide range of applications in Face recognition, object detection, Medical data classification etc. There are different types of neural networks like perceptron , Multi-layer perceptron, Recurrent Neural Network(RNN), Convolutional Neural Network(CNN), Long Short-Term Memory(LSTM) etc.

In our work, for classification of medical data we will be using Artificial Neural Network(ANN) and Wavelet Neural Network(WNN).

3.3.1 Multi Layer Perceptron (MLP)

Deep learning along with artificial intelligence is known as neural networks. General machine learning models cannot manage too complex or out of reach applications . Neural Networks have the ability to solve the complex problems very easily.

Artificial neural networks(ANNs) are based on neurons present in the human brain.

Just like the way in which the neurons in our brain are connected to each other, Artificial Neural Networks have neurons called nodes that are connected to each other to transmit the information or data .

MLP contains nodes arranged in layers. Each layer contains nodes that are connected to the nodes in another layer. Each node in a layer connects with every other node in the following layer with a certain value known as weight [14] and each node contains a bias associated with it. Bias is added to prevent the Neural Network from always giving a zero output. MLP involves 2 stages known as Forward propagation and backward Propagation.

In forward propagation, input is fed to the input layer of Neural Network. Input data is then multiplied with the weights of nodes and added with a bias. The output of the layer is given as input to special kind of functions known as activation functions. The role of activation function is to add non-linearity to the data and also to decide whether to activate the node or not. The output of the activation functions is then given as input to the next layer. This process continues till we achieve the final output in the output layer.

Once we obtain the final output from the output layer, we compare it with the expected output and then the loss function is calculated. Once the error (loss function) is calculated we will update the weights accordingly to minimize the loss. This process is known as back propagation and once again the forward propagation continues. This process continues till the local minima of loss function and the maximum accuracy is obtained. NNs use different kinds of activation functions like softmax, relu, sigmoid, tanh, logistic etc. In our work, we will be making use of relu, mexican, morlet, softmax and sigmoid activation functions.

3.3.2 Wavelet Neural Networks (WNN)

Wavelet Neural networks(WNNs) are special class of neural networks that uses wavelet functions like mexican hat, Morlet functions as activation functions for the layers in the network. Wavelet Neural Networks have been used in many applications. For our work, we will be using a wavelet neural network based classifier and it is proved to be more efficient than other NNs for plain domain data.

A wavelet is a mathematical function that divides a continuous-time signal or function into scale components. Each scale part may usually be assigned a frequency range. After that, each scale part can be studied at a resolution that corresponds to its scale. [15]

3.3.3 Activation Functions

Activation functions are used to calculate the output of a node given a set of inputs. Each layer in a neural network is set an activation function according to the requirement.

Generally the activation functions used in neural networks are nonlinear. There are many activation functions. The ones used in the project are mentioned below.

Mexican Hat

Mexican hat is a Wavelet function and is generally used when the input data is obtained from a wave like data. It is given by,

$$y = (1 - 0.1x^2)e^{-2x^2} \quad (3.1)$$

Morlet

Morlet is also a wavelet function and works well when used if the input data is obtained from a wave like data. It is given by,

$$y = \cos(1.75x)e^{\frac{-x^2}{2}} \quad (3.2)$$

Softmax

Also known as softargmax. It is mostly used in the last layer i.e. output layer of a neural network to normalise the output. The sum of values obtained in all nodes in output layer will be equal to 1.0 if the activation function used is softmax. It's equation is given by,

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (3.3)$$

Sigmoid

Sigmoid function is a 'S' shaped curve and is a non-linear function. It is given by,

$$y = \frac{1}{1 + e^{-x}} \quad (3.4)$$

Relu

Relu (Rectified Linear Unit) is an activation function that gives the same value as output if input is positive, and 0 as output when input is negative. It is given by,

$$y = \max(0, x) \quad (3.5)$$

3.3.4 Loss Functions

It is also known as Cost function. The loss function compares the outputs of our neural network with the original labelled values of the inputs in the dataset, and gives us the Prediction Error that tells us how good or how bad our model is. The gradient of the Loss function is then used to update the weights in our model, so as to minimise the loss i.e. error in prediction.

The Loss function used in our model is Categorical Cross Entropy.

Categorical Cross Entropy

It is also called Softmax Loss function. It is mainly used in Multi Class problems where the input data can belong to only one class, and to use it, the number of output nodes should be equal to the number of classes. While using Categorical Cross Entropy function we mostly use Softmax activation function at the output layer.

It is given by the formula,

$$Loss = - \sum_x p(x) * \log(q(x)) \quad (3.6)$$

where $p(x)$ is value of class x in the label, $q(x)$ is the prediction of the value of class x by our model.

3.3.5 Optimization Algorithms

The neural network is initially given some random weights and the loss is calculated using the loss function. The gradient of the Loss gives an idea of how to change the weights to reduce the error at the output and increase accuracy. We try to find the local minimum in the Loss function and find the corresponding optimal weights. This is done by changing the weights and observing the loss until we get to the minimum loss using an Optimization Algorithm. Here the algorithm defines by how much and which specific weights should be changed in each step. In our model, Adam optimizer is used.

Adam Optimiser

Adam (Adaptive Motion Estimation) combines the best of AdaGrad and RMSProp optimisers and is very easy to get it up and running. It is the most used Optimiser.

The configuration parameters of Adam optimiser are Learning Rate (α), exponential decay rate of first moment(β_1) and second moment(β_2) estimates, a small number to prevent division by 0 error(ϵ).

3.3.6 Tensorflow

Tensorflow is an popular open source python library by Google, that can be used to build optimised Machine learning models and neural networks.

The models built using Tensorflow can be run on a GPU(Graphics card) or a TPU(Tensor Processing Unit) for faster processing.

3.4 Expansions Used For Approximations

3.4.1 e^x Expansion

The series expansion of e^x is given by,

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \dots \quad (3.7)$$

3.4.2 $\cos(x)$ Expansion

The series expansion of $\cos(x)$ is given by,

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{n!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots \quad (3.8)$$

3.4.3 $\frac{1}{1+e^{-x}}$ Expansion

The series expansion of $\frac{1}{1+e^{-x}}$ is given by,

$$\frac{1}{1+e^{-x}} = \frac{1}{2} + \frac{x}{4} - \frac{x^3}{48} + \dots \quad (3.9)$$

3.5 Homomorphic Encryption

3.5.1 Introduction To Homomorphic Encryption

Generally, to operate on encrypted data we should have knowledge about the decryption function but in Homomorphic Encryption scheme we don't have to know about the decryption function to perform arithmetic operations on encrypted data. Assume $E_K()$ is an encryption function and its key is K and $D_K(.)$ is its decryption function. In this case $E_K(.)$ is said to be homomorphic with the operator $(.)$. If there is an efficient algorithm Alg which satisfies the below condition:

$$Alg(E_K(x), E_K(y)) = E_K(x.y) \quad (3.10)$$

Example:

In RSA, $E_K(x) = x^K \bmod n$ (all operations are done in Z_n) Given another ciphertext $E_K(y)$, we can compute $E_K(x.y)$ by simply multiplying the two ciphertexts:

$$E_K(x.y) = (x.y)^K \bmod n = x^K y^K \bmod n = E_K(x)E_K(y) \quad (3.11)$$

But addition is not supported by this scheme.[19]

3.5.2 Types Of Homomorphic Encryption

HE schemes are classified based on the operations they can perform on the encrypted data, difference lies in the operations they can perform and the depth of those operations(i.e, number of times the operation can be performed). There are three types of HE, namely, partially, somewhat and fully homomorphic encryption.

Partially homomorphic encryption scheme can be used when there is a need for only a single type of operation, addition or multiplication, but never both. It doesn't restrict the depth of the operation. This scheme is useful only when we need to perform either addition or multiplication on the encrypted data. The RSA encryption scheme is an example of a PHE that allows an unbounded number of modular multiplications.

Somewhat homomorphic encryption scheme can be used when both addition and multiplication operations are needed, but there is a restriction on the depth (e.g. a depth of at most 5). Leveled Homomorphic Encryption is a subset of SHE, it can perform operations with variable depth, but the depth must be set prior to encryption, however, sometimes we will have to perform operations of arbitrary depth.

Fully homomorphic encryption schemes can be used when both addition and multiplication are required, but in contrast to SHE, FHE has an unlimited depth, which makes it suitable for deep learning applications. Although many people have proposed FHE schemes in the last decade, it has been difficult to use them practically. Now FHE are built on top of SHE known as bootstrapping. FHE has schemes such as BGV and BFV for integer arithmetic, and CKKS for arithmetic on real numbers.

Though FHE seems to be the most powerful scheme, to put such scheme into practice, we need to consider other factors as well, like the ciphertexts size, evaluation cost, plaintexts domain(integers or real numbers), and the bootstrapping cost for FHE schemes.

3.5.3 Limitations Of Homomorphic Encryption:

Homomorphic encryption is very useful in Privacy preserving, as we can perform arithmetic operations on the encrypted data without decrypting it. But the number of operations possible is often limited to few like addition (/subtraction) and multiplication. This limits the activation functions that can be used in our neural network. Any activation function which is non linear must be approximated to a polynomial to be used in our model. This affects the accuracy of our model.

Operations in encrypted domain are extremely slow and consume larger memory when compared to plain domain and Each multiplication adds a significant amount of noise to the output whereas in case of addition, the noise added is not that significant. As the number of multiplicative operations on a ciphertext increases, noise increases rapidly making the ciphertext impossible to be decrypted correctly. There is always a tradeoff between number of computations possible on a ciphertext(performance) and also the noise level of the ciphertext.

3.6 CKKS Encryption Scheme

Cheon-Kim-Kim-Song (CKKS) scheme is a homomorphic encryption scheme that allows us to perform operations like addition, multiplication on encrypted data. CKKS performs operations on vector of complex values. So, it can be used for real numbers as well. As shown in Figure 3.5, the message vector m is encoded into a **plaintext** polynomial. This is because polynomials provide good trade-off between security and efficiency than vectors [20]. The encoded polynomial is then encrypted using public key and the encrypted message C contains 2 polynomials

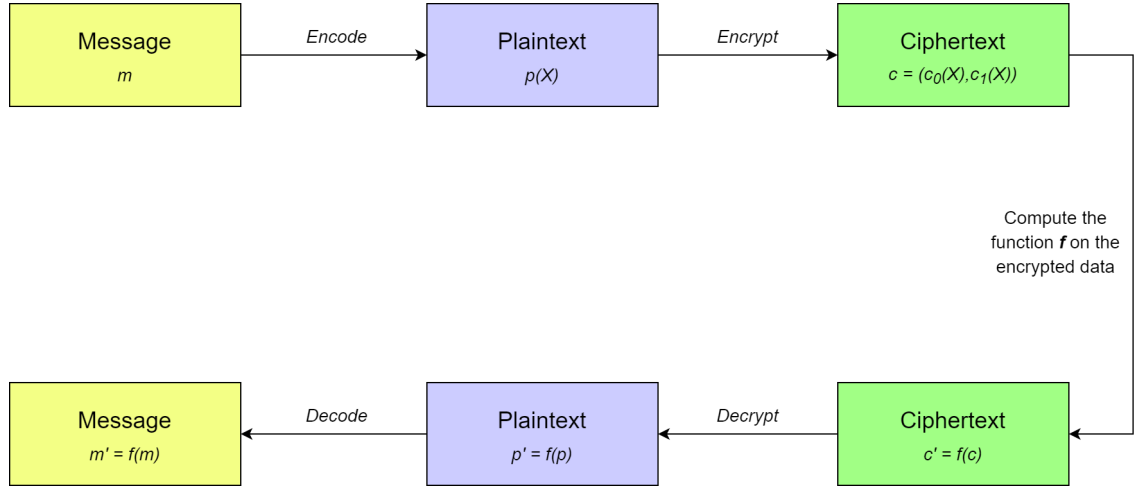


Figure 3.5: CKKS scheme

as shown in the Figure 3.5. The encrypted message is known as **ciphertext**. Then the operations will be performed on the encrypted message. The message is then decrypted using secret key. Decrypted message is then decoded to get final message.

3.6.1 Encoding And Decoding

To understand the encoding and decoding in CKKS scheme, one should have good understanding of linear algebra and ring algebra. The aim of encoder is to encode the vector $Z \in C^N$ to polynomial ring in $\frac{Z[X]}{X^{N+1}}$. To encode the vectors into polynomials, CKKS scheme uses canonical embedding given by $\sigma : \frac{C[X]}{X^{N+1}} \rightarrow C^N$. σ is used to convert the polynomial ring into a vector. Therefore, σ^{-1} is used for the encoding of vectors into polynomial ring.

Since $Z[x]$ has integral coefficients, the roots of the polynomial will be conjugate to each other. $\sigma(R) \subseteq H = z \in C^N$ contains conjugate pair of roots. Therefore we can say that any element of $\sigma(R)$ is in space of $N/2$ instead of N . Now projection of an element in H onto C^N is known as π operation in CKKS. It is also a type of isomorphism. Here π projects whereas as π^{-1} expands. So, $\pi^{-1}(Z) \in H$.

To project π^{-1} on $\sigma(R)$ CKKS uses a technique known as coordinate-wise random rounding [21]. Using this technique CKKS rounds a value x to its nearest integer.

R has orthogonal basis $(1, X, \dots, X^{N-1})$ and $\sigma(R)$ has orthogonal basis $\beta = (b_1, b_2, \dots, b_N) =$

$(\sigma(1), \sigma(X), \dots, \sigma(X^{N-1}))$. Now projection of z on β is given by $z = \sum z_i b_i$ where $z_i = \frac{\langle z, b_i \rangle}{\|b_i\|^2}$ and $\langle x, y \rangle$ is hermitian product. Once we have the coordinates of z_i we use coordinate-wise random rounding to random them. Once we have projected on $\sigma(R)$, we can apply σ^{-1} to get the encoded polynomial we desired. One important point here is that before projection on $\sigma(R)$ we will multiply by a term Δ for precision.

The decoding process is very simple, we can get our message back by computing $z = \pi \circ \sigma(\Delta^{-1}.m)$

3.6.2 Encryption And Decryption

Ring Learning with Error(RLWE) [22] is the foundation of CKKS encryption. RLWE problem is to distinguish noisy pairs of the form $(a_i, b_i) = (a_i, \langle a_i, s \rangle + e_i)$ from the random ones in $\frac{Z_q^n}{x^{N+1}} X \frac{Z_q^n}{x^{N+1}}$. Here $a_i, s \in \frac{Z_q^n}{x^{N+1}}$, a_i is uniformly sampled and s is secret key and $e_i \in \frac{Z_q^n}{x^{N+1}}$ is a small noise. RLWE problems are very hard to solve. So, encryption scheme based on RLWE will be secure against attacks.

Lets say we have taken a secret key $s \in \frac{Z_q^n}{x^{N+1}}$ and we publish n pairs $(a_i, \langle a_i, s \rangle + e_i)$. The above equation can be written in Matrix form as $(A, A.s + e)$ with $A \in \frac{Z_q^{n \times n}}{x^{N+1}}$ and $e \in \frac{Z_q^n}{x^{N+1}}$. So, according to the RLWE problem we have discussed above, it is very hard to recover the secret key from this couple.

We will use $p = (-A.s + e, A)$ as our public key, where s, A are secret key and matrix respectively as discussed above. Now, let's encrypt message m in $\frac{Z_q^n}{x^{N+1}}$ using the public key.

Encryption: $c = (m, 0) + p = (m - A.s + e, A) = (c_0, c_1)$

Decryption using s : $m' = c_0 + c_1.s = m - A.s + e - A.s = m + e \approx m$.

We can observe that the encryption of encoded polynomials gives a pair of polynomials in encrypted domain and decryption of this pair gives approximated message polynomial.

Addition

Suppose we have two messages, m, m' and let's say corresponding encrypted messages are $c = (c_0, c_1)$ and $c' = (c'_0, c'_1)$. Now, the addition of these two messages yields $c_{add} = c + c' = (c_0 + c'_0, c_1 + c'_1)$

Now, decryption of c_{add} yields

$$c_{add,0} + c_{add,1}.s = c_0 + c'_0 + (c_1 + c'_1).s = c_0 + c_1.s + c'_0 + c'_1.s = m + m' + 2e \approx m + m'$$

Clearly, we can observe that the decrypted value of addition of two encrypted messages is approximately equal to addition of their corresponding plaintext messages.

Multiplication of a cipher text with a plain text

Let's take a message m and its cipher text $c = (c_0, c_1)$ and another message m' . Multiplication of m' with ciphertext c is given by $c_{mult} = (m'.c_0, m'.c_1)$.

Now, Decrypting c_{mult} we get

$$m'.c_0 + m'.c_1.s = m'.(c_0 + c_1.s) = m'.(m + e) = m'.m + m'.e \approx m'.m$$

we can observe that the decrypted value of multiplication of a plaintext(m') and a ciphertext(c) is approximately equal to multiplication of the plaintext(m') and corresponding plaintext(m) of the ciphertext.

Multiplication of two cipher texts

Multiplication of two cipher texts is not straight forward. We have to find c_{mult} and Decrypt_mult such that $\text{Decrypt_mult}(c_{mult}(c, c'), s) = \text{Decrypt}(c, s) \cdot \text{Decrypt}(c', s)$

Let's compute $\text{Decrypt}(c, s)$. $\text{Decrypt}(c', s)$ first

$$\text{Decrypt}(c, s) \cdot \text{Decrypt}(c', s) = (c_0 + c_1.s).(c'_0 + c'_1.s) = c_0.c'_0 + (c_0.c'_1 + c'_0.c_1).s + c_1.c'_1.s^2 = d_0 + d_1.s + d_2.s^2.$$

Therefore decryption in case of multiplication of two cipher text can be seen as a polynomial of degree 2 as $d_0 + d_1.s + d_2.s^2$

$$\text{Therefore } C_{mult}(c, c') = c_{mult} = (d_0, d_1, d_2) = (c_0.c'_0, c_0.c'_1 + c'_0.c_1, c_1.c'_1) \\ \text{Decrypt_mult}(c_{mult}, s) = d_0 + d_1.s + d_2.s^2$$

We are successfully able to perform multiplication of cipher texts but the problem here is that the length of c_{mult} is 3, whereas in previous cases it is just 2. As, the number of multiplicative operations on a cipher text increases, the size of cipher text will grow further. To overcome this problem, relinearization operation is performed to resize the ciphertext to a size of 2 after every multiplication.

Chapter 4

Database and Software

4.1 Dataset

The China Physiological Signal Challenge (CPSC) 2018 Dataset is used to train and evaluate our model[25]. CPSC Dataset contains 12 lead ECG data collected from 11 different hospitals. All data from 11 hospitals were combined to constitute the dataset. The training set contains 6,877, 12 leads ECG recordings lasting from 6 s to 60 s and the test set contains 2,954 ECG recordings with the similar lengths. All ECG recordings were sampled at 500 Hz and provided in MATLAB format (.mat format). CPSC dataset contains data of patients suffering from 9 different diseases. For detection of Bundle Branch Block disease we only consider data which contains Normal, LBBB, RBBB as its labels. Some records have multiple labels because the patient may suffer from 2 or more diseases.

4.2 Major Libraries Used

4.2.1 Scipy Library

Scipy is an open source python library mainly used for Scientific computing. It is built on top of Numpy library and uses Numpy arrays data structure for all computations. It has many tools for functions like Integration, Linear Algebra, Signal processing, Image processing and many more. Scipy was used in our model for computing Median filter and SG filter.

4.2.2 Keras

Keras is an open source, user-friendly and easy to use library that provides python interface for implementation of different Neural Networks[26]. Keras uses Tensorflow as its backend. Keras contains basic building blocks like layers, activation functions, optimizers for very easy implementation of Neural Networks. It also allows training model on Graphics processing units (GPU) and tensor processing units (TPU)

4.3 Microsoft Seal

Microsoft seal is a open- source homomorphic encryption technology which allows the user to perform computations directly on encrypted Integers or real numbers[23]. Microsoft seal supports CKKS, BFV scheme but it doesnot support BGV scheme. For evaluation of machine learning models, CKKS is a very good choice. Microsoft seal provides implementation of CKKS so it is better for machine learning models. Microsoft seal is implemented in C++ and it requires Microsoft visual studio to work.

4.3.1 Pyfhel Library

Python for homomorphic encryption library (pyfhel) is an open - source library that can be easily installed and used in any python environment and doesnot require any extra dependencies[24] . Pyfhel uses microsoft seal as its backend and can perform many homomorphic encryption operations such as addition, subtraction, scalar multiplication in python. pyfhel is very well suited for machine learning models. The noise budget available can be controlled depending on various parameters such as:

- **Plaintext modulus(p)**: This is the modulus of the plain text.
- **Polynomial modulus(m)**: This is the degree of the cyclotomic polynomial used in encryption ($x^m + 1$).
- **security parameter(Sec)**: Using this parameter we can adjust the security of our encryption to 128,192, 256 bit etc.
- **intDigits, FracDigits** : This parameter helps us to adjust the number of digits in integral and fractional part of a ciphertext.

Depending on values of these parameters one can adjust the noise budget available but there is always a trade-off between noise level and the performance of the

model. Here performance refers to how accurately a cipher text can be decoded to its actual plain text value.

- Higher p value gives slightly less noise budget but the performance is better.
- Higher m value gives significantly more noise budget but performance is not so good.
- Higher sec value less noise budget and better performance
- Higher intDigits, FracDigits value gives slightly less noise budget but performance is better.

We will be using pyfhel library for homomorphic encryption because it supports CKKS scheme which can be operated on floating numbers and well suited for machine learning models.

4.4 Computation Platform

Google Colab is a open source python notebook that runs on google cloud. It allows us to write and use python code without any requirement of configuration and execute through the browser, well suited for implementation of machine learning algorithms. It makes computation easier and much faster. Colab supports both TPU, GPU version runtime which makes computations ever faster.

Chapter 5

Block Diagram

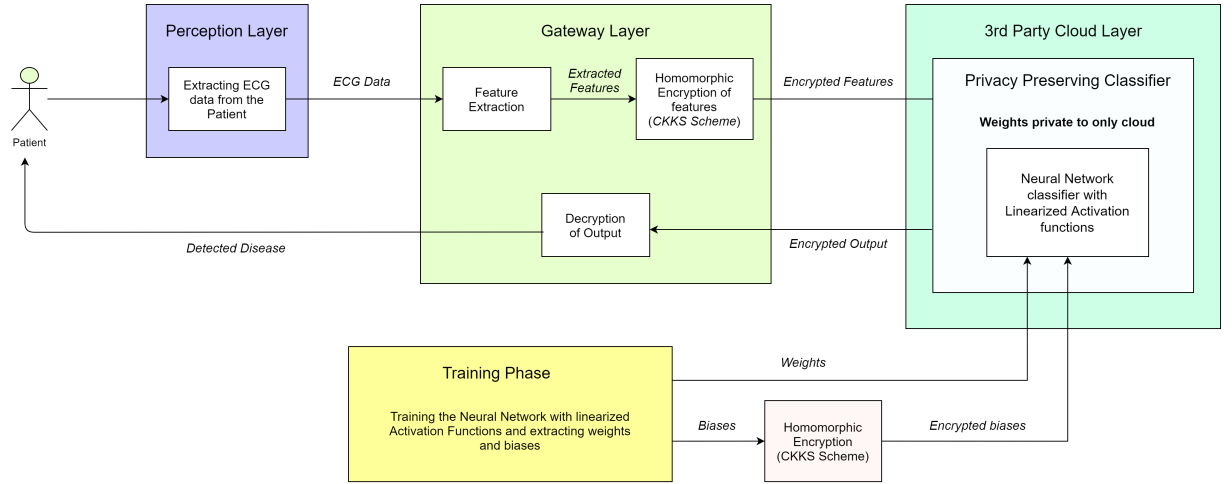


Figure 5.1: Block Diagram of the proposed model

ECG data is extracted from the patient using the leads attached to the body. The Extracted ECG data is then sent to the Gateway Node say Laptop, Mobile etc. The feature extraction is done in gateway node using methods discussed in Section 6.1, 6.2. The extracted features are then encrypted using CKKS homomorphic encryption scheme. The encrypted features are then sent to 3rd party cloud for classification. Only prediction will occur in cloud layer.

Model will be trained and validated in plain domain itself using dataset available, Optimized weights and biases after completion of training will be extracted from the model. Biases are encrypted using CKKS homomorphic encryption scheme and weights will remain unencrypted and then hosted in the cloud layer. Weights are made private only to cloud. So, only cloud can access the weights.

When the encrypted data from the gateway node is received by the cloud, using the weights and biases in the cloud layer, output is predicted and then sent back to gateway node. Gateway Node decrypts the data and sends the information back to the patient.

Chapter 6

Methodology

6.1 QRS - Complex Detection

QRS detection is the most vital part in detection of any heart disease using ECG. Detection of QRS complex is a well studied topic and detection of any other waves of ECG like P-wave, T-wave depends on extraction and detection of R- peak. In this work we have detected R peak using the technique proposed in paper [27]. This technique includes:

- Median filter is used to suppress the noise in the extracted ECG signal (say $x[n]$) from the patient. In stage 1, Window width for median filter is $fs/2$ samples (fs = sampling frequency of ECG data). For CPSC Dataset $fs=500Hz$. In stage 2, window width is taken as fs and output of stage 1 is given as input to stage 2. Output of stage 2 is subtracted from original $x[n]$ to get baseline free ECG data [27] as shown in Figure 6.1(c).
- Savitzky–Golay (SG) is used as smoothing filter to smooth out a noisy signal [28]. The SG filter coefficients, the order and the frame size is selected as described in [29]. The smoothed ECG signal is shown in Figure 6.1(d).
- Since R peak is the tallest peak among all waves in ECG data, Computation of third power of ECG makes it even more significant and easy to detect as shown in Figure 6.1(e).
- Baseline identification is one of the important feature in QRS complex detection. In our work baseline identification is done using Sauvola’s technique [30].

$$th = \mu[1 + k(\frac{\sigma}{\sigma_m - 1})] \quad (6.1)$$

where μ is mean, σ is standard deviation, k is a parameter in the range $[0.2, 0.5]$, σ_m is maximum standard deviation.

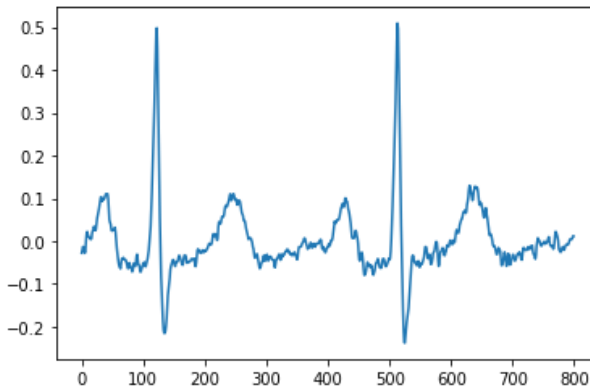
- To determine the QRS region, threshold is calculated as
Upward QRS region : $z[n] > th + rms$
Inverted QRS region : $z[n] < th - 3*rms$

For upward QRS region, the value $th+rms$ will be greater than amplitude of P wave, T wave and for Inverted QRS region, the value $th - 3*rms$ will be lesser than P wave and T wave . So, the region greater than this threshold for upward QRS region can be considered as a part of QRS complex that contains R - peak and vice versa for inverted QRS complex. The detected R-peaks are shown in Figure 6.1(f).

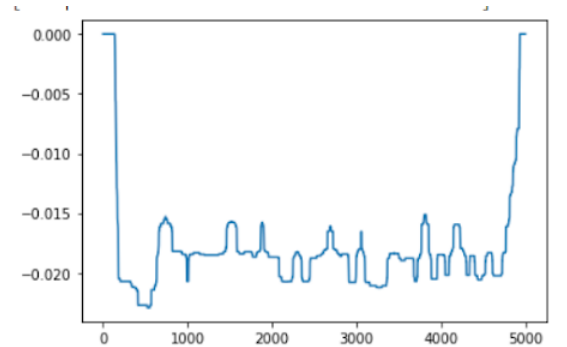
- Local maxima of the determined region using threshold is termed as R - Peak. In case of inverted QRS complex, local minima of the determined region is termed as R - peak.
- Sometimes T waves maybe greater than threshold value and can be detected as R - peak. A T- wave is expected in next 450ms from a R - Peak. Inorder to overcome this problem kurtosis is calculated for detected R- Peaks those having difference less than 450ms. Higher kurtosis value is expected for R-Peak compared to T- wave.

$$\text{Kurtosis } (\beta) = \frac{E[(X - \mu_x)^4]}{(E[(X - \mu_x)^2])^2} \quad (6.2)$$

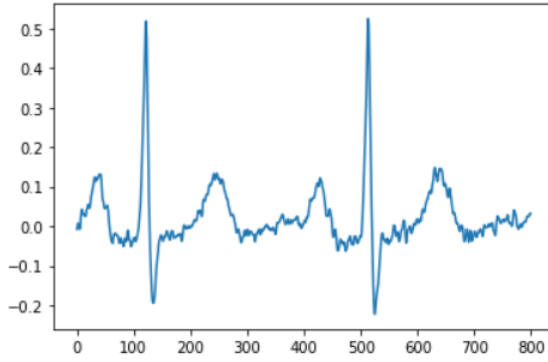
- With reference to the detected R peak local minima to the left of 40 samples(80 ms) is termed as Q wave and local minima to the right of 40 samples(80 ms) is termed as S wave. A detected QRS wave is shown in Figure 6.1(g).



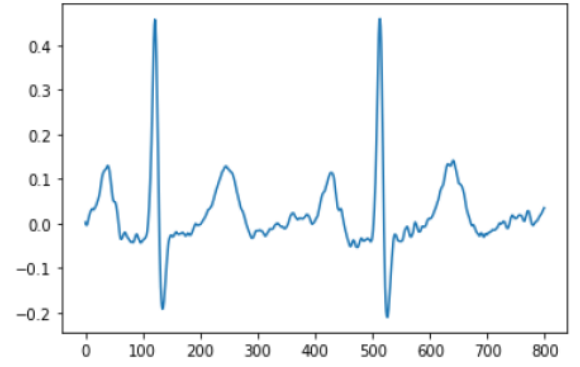
(a) ECG signal from lead I



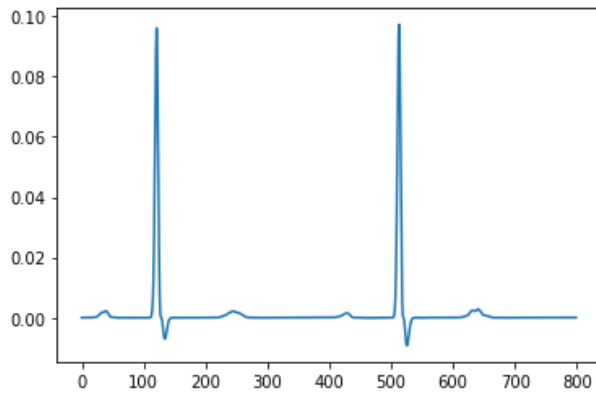
(b) Baseline drift in ECG signal



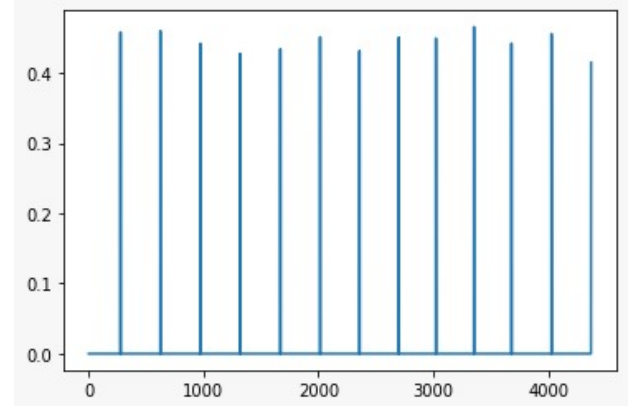
(c) Median Filtered Signal with noise removed



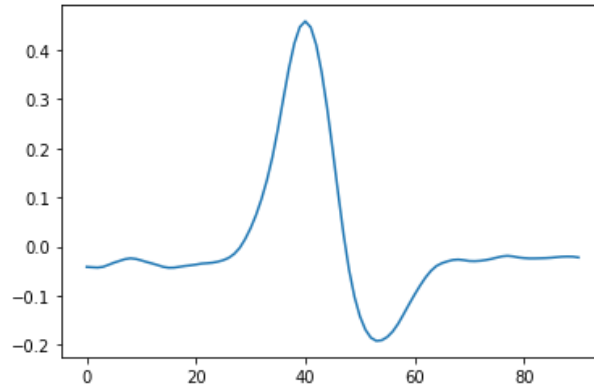
(d) Signal after Smoothing with SG filter



(e) Signal with boosted R peaks after cubing



(f) Detected R peaks



(g) Extrated QRS complex

Figure 6.1: QRS interval extraction procedure

6.2 Feature Extraction

Presence of LBBB or RBBB causes change in the shape of the QRS complex. Thus statistical features can be used for classification. The extracted QRS complex is termed as $x[n]$. Number of samples in $x[n]$ is termed as N . The following 5 statistical features i.e. Mean, Variance, Standard Deviation, Skewness and Kurtosis are computed for each QRS complex using the below equations as proposed in paper [31].

1.
$$\text{Mean } (\mu_x) = \frac{1}{N} \sum x[n] \quad (6.3)$$

2.
$$\text{Variance } (\sigma_x^2) = E[(X - \mu_x)^2] \quad (6.4)$$

3.
$$\text{Standard Deviation } (\sigma_x) = \sqrt{E[(X - \mu_x)^2]} \quad (6.5)$$

4.
$$\text{Skewness} = E\left[\left(\frac{X - \mu_x}{\sigma_x}\right)^3\right] \quad (6.6)$$

5.
$$\text{Kurtosis} = \frac{E[(X - \mu_x)^4]}{(E[(X - \mu_x)^2])^2} \quad (6.7)$$

Extracted counts of QRS complexes are shown in Table 7.1. The Extracted 5 features are fed as input to the Neural Network for classification.

6.3 Classification

6.3.1 Classification Using WNN

After computation of 5 statistical features from the QRS complex, these features are fed as input to the Neural Network for classification. In our work 28162 patterns are used to train the model, 7041 for validation and 17340 patterns were used for testing our model.

In this work, we used Wavelet Neural Network for feature classification instead of normal Neural Network especially for the classification of Bundle Branch Blocks as proposed in paper [32]. Special activation functions known as wavelet functions

were used in the hidden layers for the classification of Bundle Branch Blocks. Mexican Hat and Morlet wavelet functions were used for the classification. Equations of Mexican Hat and Morlet wavelet functions are shown in Fig..5. Block diagram for WNN is shown in Figure 6.2

In our work, the model was trained using training set and 1 input layer, 2 hidden layers and 1 output layer were used for classification. Wavelet functions are used as activation functions for hidden layers and softmax max is used as activation functions for output layer. Input layer contains 5 nodes and the hidden layers contain 5,4 nodes respectively. Output nodes contain 3 nodes to classify as either LBBB or RBBB or Normal. Output labels are encoded using OneHotEncoding. Categorical Crossentropy is used as Loss function and Adam optimizer is used for optimization. WNN is implemented using the keras library in python. The implemented NN is shown in Figure 6.2.

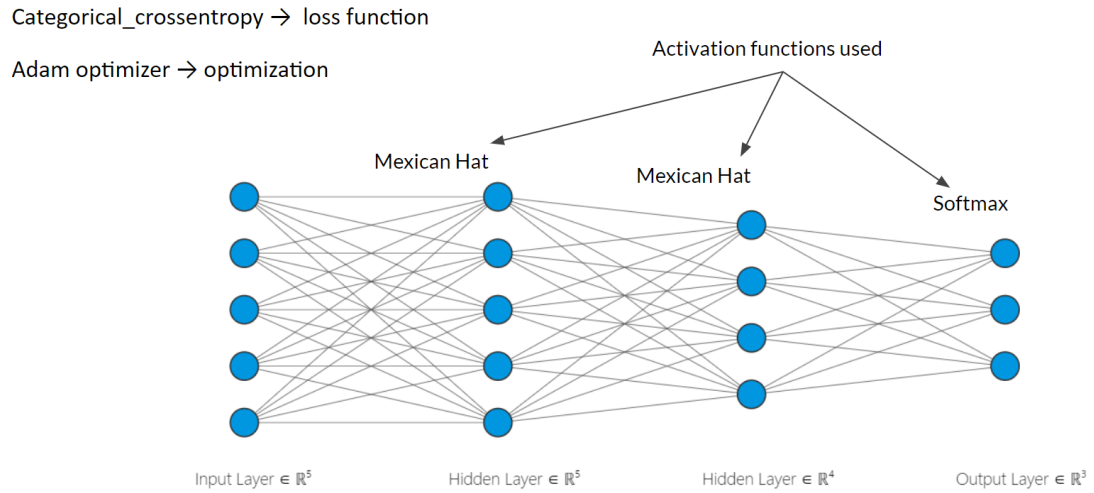


Figure 6.2: The implemented WNN

6.3.2 Privacy Preserving Classification

In previous Section 6.3, classification of extracted features is accomplished using Wavelet Neural Network(WNN) in **plain domain**. As discussed in block diagram, the classification of features will be done in 3rd Party cloud service. Concern over loss of privacy and business value of private data is an overwhelming barrier to the adoption of cloud services. The best way to overcome this problem is to encrypt

the extracted features and then perform operations on them without decrypting them.

Encryption of Extracted features

After Extraction of ECG data from the patient, the data will be sent to the gateway node. QRS - complex detection, Feature Extraction will be done in gateway node as discussed in Section 6.1, 6.2 respectively. Extracted Features are then encrypted using CKKS homomorphic encryption scheme. Then the encrypted features will be sent to the cloud for classification.

Approximation of Activation functions

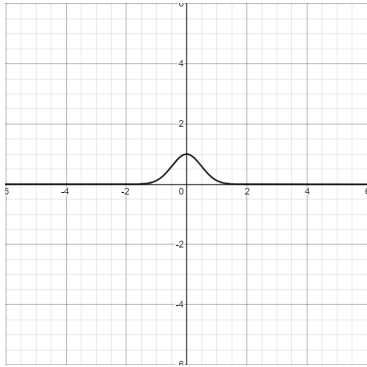
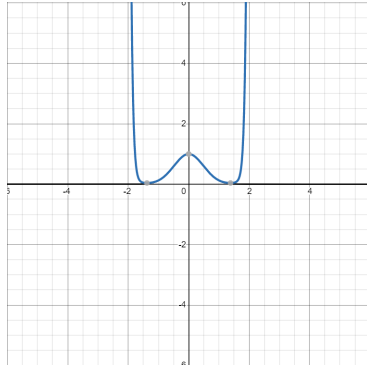
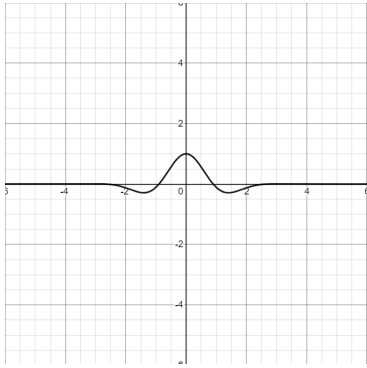
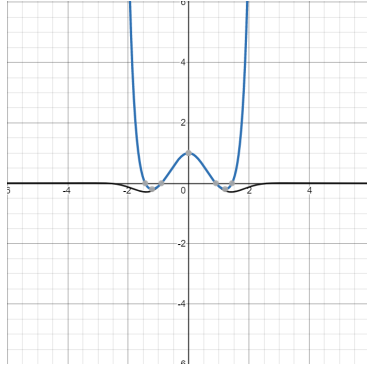
One of the major limitation of homomorphic encryption is that it can only perform multiplication, addition operations. Operations like division, exponential are not possible in case of homomorphic encryption. All activations functions in machine learning such as Relu, Mexican, Morlet, Sigmoid, Softmax etc are non-linear in nature. i.e, they involve both division and exponential operations as well. So, direct usage of these activation functions in encrypted domain is not feasible. We have to approximate these functions into polynomials which involves only multiplication and addition operations. Polynomials are good approximations because they still have capability to add non-linearity to the data. The approximation of the activation functions used in our model are shown below:

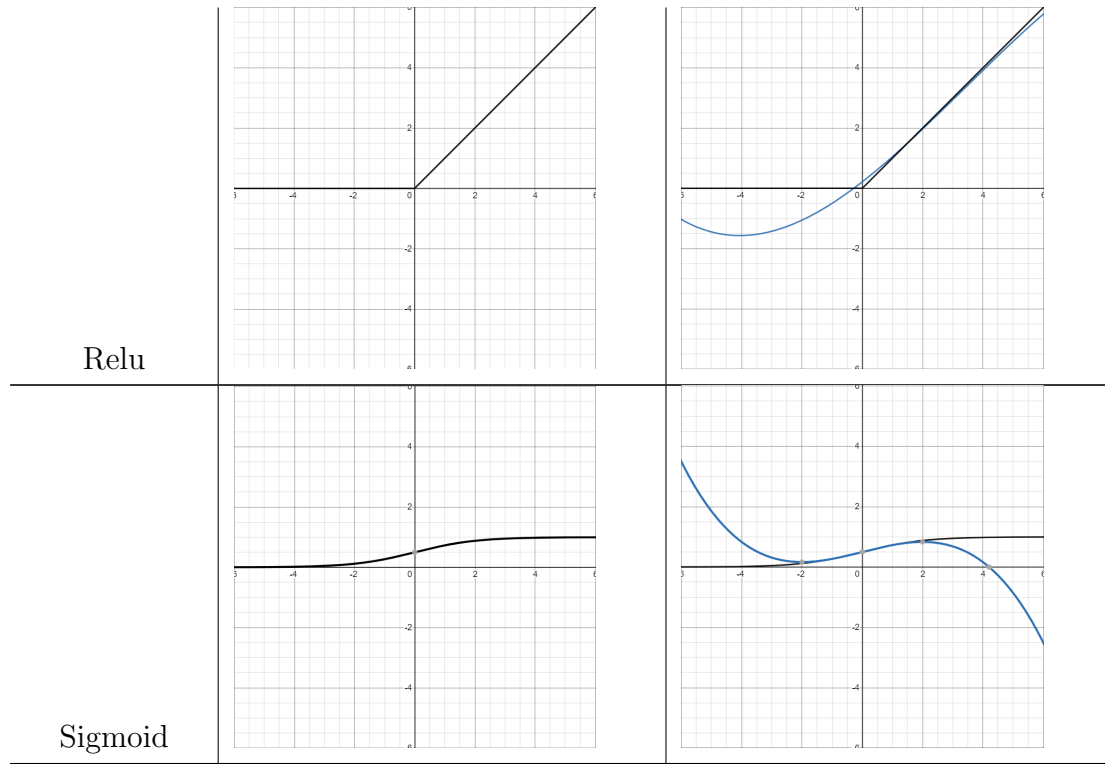
Table 6.3: Original and Approximated functions

Name	Original	Approximated
Mexican Hat	$y = (1 - 0.1x^2)e^{-2x^2}$	$1 - 2.1x^2 + 2.2x^4 - 1.53x^6 + 0.8x^8 - 0.33x^{10} + 0.11x^{12} - 0.034x^{14} + 8.88 \times 10^{-3}x^{16} - 2.04 \times 10^{-3}x^{18} + 4.23 \times 10^{-4}x^{20} - 7.95 \times 10^{-5}x^{22} + 13.68 \times 10^{-6}x^{24} - 21.70 \times 10^{-7}x^{26} + 3.19 \times 10^{-7}x^{28}$
Morlet	$y = \cos(1.75x) * e^{\frac{-x^2}{2}}$	$y = 1 - 2.03x^2 + 1.28x^4 - 0.44x^6 + 0.10x^8 - 0.01x^{10} + 2.58 \times 10^{-3}x^{12} - 2.96 \times 10^{-4}x^{14} + 2.87 \times 10^{-5}x^{16} - 2.42 \times 10^{-6}x^{18} + 1.79 \times 10^{-7}x^{20} - 1.19 \times 10^{-8}x^{22} + 7.10 \times 10^{-10}x^{24} - 3.61 \times 10^{-11}x^{26} + 6.23 \times 10^{-12}x^{28}$

Relu	$y = \max(0, x)$	$y = 0.215 + 0.78x^1 + 0.06x^2 - 0.0059x^3$
Sigmoid	$y = \frac{1}{1+e^{-x}}$	$y = 0.5 + \frac{x}{4} - \frac{x^3}{48}$

Table 6.4: Plots of the original and approximated functions

Name	Original	Approximated
Mexican Hat		
Morlet		



Privacy Preserving Classifier

Extracted features after encryption will be sent to cloud for classification. As discussed earlier in block diagram, the training and validation of model will be done in plain domain. After completion of training, optimized weights and biases will be extracted from the trained model. Biases were encrypted using CKKS scheme but weights remains unencrypted. Weights and encrypted biases will be sent to the cloud and weights are made private only to cloud. That is, only cloud can access the weights for computations.

In case of classification in plain domain (Section 6.3.1), we have observed that the Mexican function as activation function in the hidden layers have yielded very good accuracy in plain domain when compared to relu. Approximated wavelet functions require a degree of atleast 28 for good classification accuracy and acceptable input range. In case of encrypted domain, usage of approximated Mexican, Morlet functions are highly impractical because computation of x^{28} is not feasible due to limitation in number of multiplicative operations possible. We have observed that for our model, approximated relu, sigmoid activation functions with degree 3 is sufficient for good classification accuracy. As number of layers increases, number of computations on ciphertext also increases. Due to the limitation in number of

operations that can be performed on a ciphertext, we cannot afford to have more than 2 hidden layers in our model for activation functions with degree 3. So, we can only have atmost 2 hidden layers in our model in case of encrypted data classification

Our work Proposes a technique that operates on encrypted data with 2 hidden layers. 5 nodes in 1st hidden layer with approximated Relu activation function and 4 nodes in 2nd hidden layer with approximated Relu activation function and 3 nodes in output layer with approximated Sigmoid activation function to classify as LBBB, RBBB and Normal. Since, Softmax is very difficult to approximate we have used approximated Sigmoid as activation function for output layer.

Weights extracted from the trained model remains unencrypted and made private only to cloud because multiplication of two cipher texts will further increase the noise level in the circuit. So, because of this reason weights remain unencrypted and multiplication of plain text and cipher text will be performed here. Biases will be encrypted because addition of 2 cipher texts won't add much noise to the cipher text. Categorical Crossentropy is used as Loss function and Adam optimizer is used for optimization. When the encrypted data is received by the cloud, using the encrypted biases and private weights, the model predicts the output and sent back to gateway node.

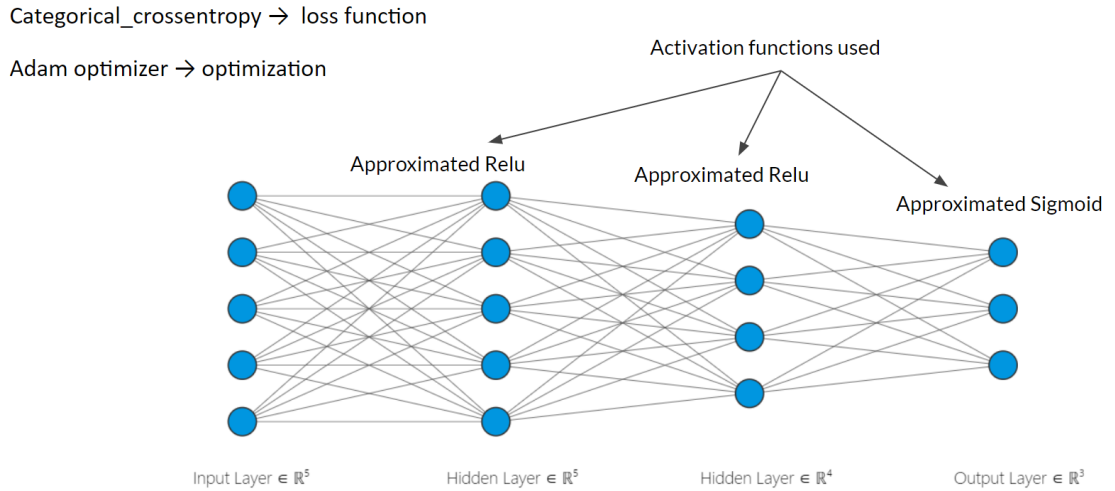


Figure 6.3: The implemented privacy preserving neural network

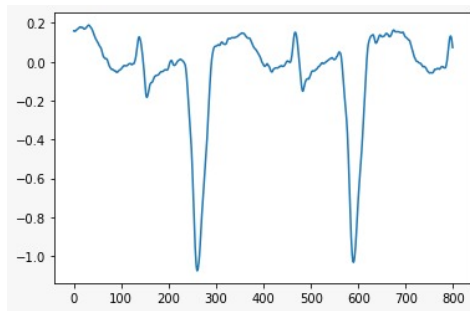
Chapter 7

Results

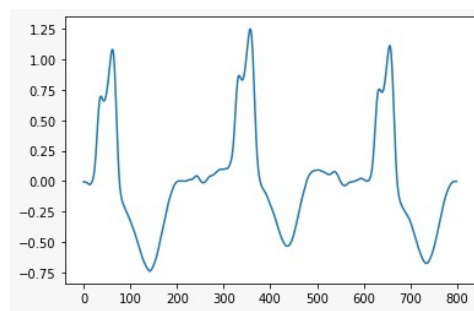
All the models has been implemented using python in Google Colab with 12GB RAM and GPU supported version. CPSC 2018 dataset was used for training and classification. The division into training, testing and validation sets is tabulated below.

Table 7.1: Number of Patterns used for testing, validation and training of the model

S No.	Data Category	No. of patterns
1.	Training	28162
2.	Validation	7041
3.	Testing	17340
	Total	52543



(a) ECG of LBBB patient



(b) ECG of RBBB patient

Figure 7.1: Sample ECG from CPSC 2018 dataset

7.1 Feature Extraction

The 5 features, i.e. Mean, Standard deviation, Variance, Skewness and kurtosis, are extracted from each QRS complex using the method stated in the Methodology.

The sample values of features extracted from a normal, RBBB and LBBB patient using this method are tabulated in Table 7.2 for reference.

Table 7.2: Features Extracted from ECG of a normal, a LBBB and a RBBB patient

Type	Mean	Std. Deviation	Variance	Skewness	Kurtosis
Normal	-0.05325	0.22818	0.05206	-0.39035	3.50845
LBBB	-0.36599	0.38930	0.15155	-0.50152	1.79253
RBBB	0.22846	0.36022	0.12976	0.10278	1.83997

7.2 Classification Using ANN And WNN

The extracted features were fed into a neural network and the respective accuracies were noted by changing the number of hidden layers and nodes in each layer.

Performance of our model is evaluated using training, validation and test accuracies. For our work, 2 hidden layers are concluded as the optimal number of hidden layers and hidden layers have 5,4 nodes respectively. Performance of the model is evaluated for different activation functions like Relu and wavelet functions (Mexican Hat and Morlet). Softmax was used at the output layer for best classification.

Mexican hat activation function in the hidden layers gave an accuracy of 92.6% and Relu activation function in the hidden layers gave an accuracy of 91.3% and for Morlet function accuracy of 91.7% is obtained. Detailed stats are provided in Table 7.3 and plots of loss function vs number of epochs, accuracy vs number of epochs for training and validation are shown in Figure 7.1

Table 7.3: Accuracies of the model using different activation functions

S.No	Neural Network used	Train Accuracy	Validation Accuracy	Test Accuracy
1.	WNN with Mexican Hat	92.7%	91.9%	92.6%
2.	WNN with Morlet	92.2%	91.5%	91.7%
3.	ANN with Relu	92.2%	91%	91.3%

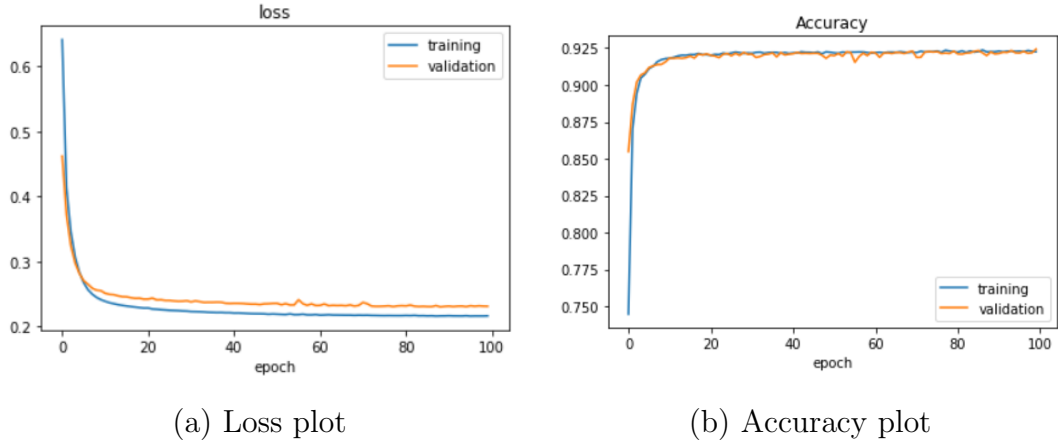


Figure 7.2: Loss and Accuracy plot using WNN

7.3 Classification With Approximated Activation Functions

To model the privacy preserving classifier, we use Homomorphic Encryption to encrypt data and perform operations on them. But HE only supports addition and multiplication. So only polynomial activation functions can be used. So we need to approximate the existing activation functions into polynomials to use them in the privacy preserving classifier.

Using different approximated activation functions in the neural network classifier, a maximum accuracy of 88.7% was obtained when approximated relu was used in the hidden layers and approximated sigmoid was used at the output layer. A detailed comparison with the classifier with normal activation functions is given in Table 7.4.

The number of layers and nodes in each layer were varied to find the maximum

accuracy and the results are plotted in the Figure 7.3. The maximum accuracy is obtained when 7 hidden layers were used with 11 nodes in each layer.

The accuracy was appreciably high using 4 hidden layers and 5 nodes in each layer, Plots of loss function vs number of epochs, accuracy vs number of epochs for traning and validation are shown in Figure 7.4

Table 7.4: Maximum possible Train, Validation, Test Accuracies of model with Normal Activation functions vs Approximated Activation functions

S.No	Accuracy	Using Normal Activa- tion functions	Using Approximated activation functions
1.	Training	92.7%	89%
2.	Validation	91.6%	89.1%
3.	Testing	92.6%	88.7%

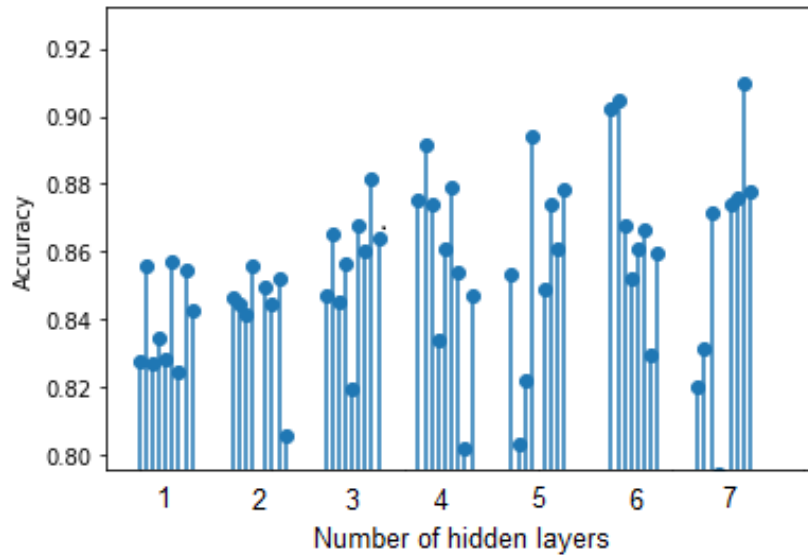


Figure 7.3: Accuracy vs number of layers, nodes in each layer

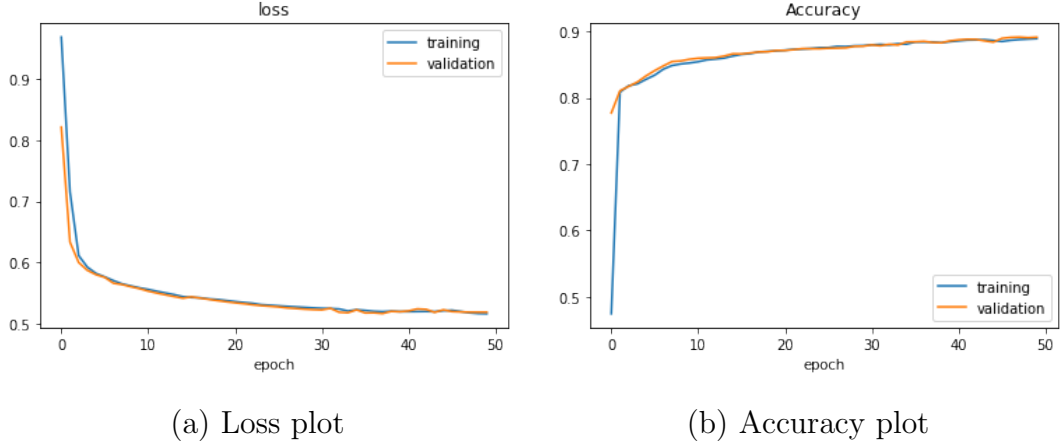


Figure 7.4: Loss and Accuracy plot using Approximated Activation functions

7.4 Classification Using Privacy Preserving Classifier

The weights and biases obtained from training using approximated activation functions in Section 7.3 are used as the weights and biases in our Neural Network. So, no separate training step is needed. The approximated activation functions are encrypted using CKKS scheme using Pyfhel library and are used as activation functions in our Neural Network.

The extracted 5 features are encrypted using CKKS scheme and are given as inputs to the Neural Network and the encrypted output is obtained which can be decrypted using CKKS scheme and resultant accuracy is noted.

For proper classification, the noise at output should be as low as possible. As in CKKS scheme, each multiplication adds in a significant amount of noise, the number of multiplications that can be done is very limited. So using Mexican Hat and Morlet as activation functions is not practical as proper approximation requires polynomials of atleast order 28. A satisfactory approximation of Softmax function was also not possible.

Approximated Relu in the hidden layers and approximated Sigmoid at the output layer were used as activation functions as they can easily be approximated to an order of 3 and we obtained an accuracy of 88.7% using these functions in the previous Section 7.3.

The parameters of level of encryption in CKKS i.e. Plaintext Modulus(p) and Polynomial coefficient modulus (m) were changed and the resulting accuracies and noise margin available were noted and tabulated in Table 7.5.

A maximum accuracy of 84.3% was obtained when $p = 2047$ and $m = 8192$ using Approximated Relu at the two hidden layers with 5 nodes in the first hidden layer and 4 nodes in the second hidden layer and Approximated Sigmoid at the output layer. The output layer has 3 nodes for classification into 3 classes i.e. Normal, LBBB, RBBB.

Table 7.5: Test Accuracy of model on encrypted data and noise Budget Available with varying parameters.

S.No	Parameter Values	Test Accuracy	Noise Budget
1.	$p = 63, m = 8192$	81.6%	205
2.	$p = 1023, m = 8192$	82.9%	198
3.	$p = 2047, m = 8192$	84.3%	192

Chapter 8

Conclusion

The proposed work presents a technique for detection of Bundle Branch Blocks without loss of privacy. BBB detection includes statistical features extraction from QRS complex and disease detection with a privacy preserving neural network. The privacy preserving neural network performs operations on the CKKS encrypted statistical features and outputs encrypted disease status without decrypting CKKS encrypted data. Since Homomorphic encryption schemes support only multiplication and addition operations, many Non-linear activation functions have been approximated to polynomials and accuracies were measured.

The model performed best while using Relu and Sigmoid as the activation functions, giving a test accuracy of 84.3%.

Approximation of activation functions limits the effective implementation of the privacy preserving neural network classifier. So, machine learning algorithms that involve only linear operations like Decision tree, Random forest etc. could be tried for better performance.

Chapter 9

Future Works

As the main reason for accuracy dropping from plain domain to encrypted domain is due to the approximation of activation functions. So future models can utilise other machine learning models like decision tree, random forest which have only linear operations so that it may help to improve the accuracy in encrypted domain disease detection.

Attempts to improve the accuracy need to be done without causing a very high increase in computational complexity which can further cause increased power dissipation and delay in processing.

Bibliography

- [1] L. Dev Sharma, R. K. Sunkaria and A. Kumar, "Bundle branch block detection using statistical features of QRS-complex and k-nearest neighbors," 2017 Conference on Information and Communication Technology (CICT), Gwalior, 2017, pp. 1-4, doi: 10.1109/INFOCOMTECH.2017.8340585.
- [2] Gupta, Rajeev & Mohan, Indu & Narula, Jagat. (2016). Trends in Coronary Heart Disease Epidemiology in India. *Annals of Global Health*. 82. 307-315. 10.1016/j.aogh.2016.04.002.
- [3] Chen, Jialu et al. "Lightweight Privacy-Preserving Training And Evaluation For Discretized Neural Networks". *IEEE Internet Of Things Journal*, vol 7, no. 4, 2020, pp. 2663-2678. Institute Of Electrical And Electronics Engineers (IEEE), doi:10.1109/jiot.2019.2942165.
- [4] Ilic, Sinisa. (2007). Detection of the Left Bundle Branch Block in Continuous Wavelet Transform of ECG Signal. *Elektronika ir Elektrotechnika*. 33-36.
- [5] Ceylan, Rahime & Özbay, Yüksel. (2011). Wavelet Neural Network for Classification of Bundle Branch Blocks. *Proceedings of the World Congress on Engineering 2011, WCE 2011*. 2. 1003-1007.
- [6] Kora, P., Kalva, S.R. Hybrid Bacterial Foraging and Particle Swarm Optimization for detecting Bundle Branch Block. *SpringerPlus* 4, 481 (2015). <https://doi.org/10.1186/s40064-015-1240-z>.
- [7] Ozbay, Yüksel & Ceylan, Rahime Karlik, Bekir. (2006). A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Computers in biology and medicine*. 36. 376-88. 10.1016/j.compbimed.2005.01.006.
- [8] Huang H, Liu J, Zhu Q, Wang R, Hu G. Detection of inter-patient left and right bundle branch block heartbeats in ECG using ensemble classifiers. *Biomed Eng Online*. 2014 Jun 5;13:72. doi: 10.1186/1475-925X-13-72. PMID: 24903422; PMCID: PMC4086987.

- [9] Sharma, Lakhan & Sunkaria, Ramesh. (2016). A Robust QRS Detection using Novel Pre-Processing Techniques and Kurtosis based Enhanced Efficiency. Measurement. 87. 10.1016/j.measurement.2016.03.015.
- [10] Cheon, Jung Hee, KyooHyung Han, Andrey Kim, Miran Kim, and Yongsoo Song. 2019. "A Full RNS Variant of Approximate Homomorphic Encryption." Selected Areas in Cryptography – SAC 2018, 347–368. doi:10.1007/978 – 3 – 030 – 10970 – 7₁₆.
- [11] Chen, Hao, Ilaria Chillotti, and Yongsoo Song. 2019. "Improved Bootstrapping for Approximate Homomorphic Encryption." Advances in Cryptology – EUROCRYPT 2019, 34–54. doi:10.1007/978 – 3 – 030 – 17656 – 3₂.
- [12] Owusu-Agyemang, Kwabena & Qin, Zhen & Zhuang, Tianming Qin, Zhiguang. (2019). MSCryptoNet: Multi-Scheme Privacy-Preserving Deep Learning in Cloud Computing. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2901219.
- [13] Hampton, J. R., amp; Hampton, J. (2019). The ECG made easy. Edinburgh: Elsevier.
- [14] "-, G., By, -, Team, G., amp; Here, P. (2021, April 26). Types of neural networks and definition of neural network. Retrieved May 04, 2021, from <https://www.mygreatlearning.com/blog/types-of-neural-networks/>"
- [15] "https://www.sciencedirect.com/science/article/pii/S0893608013000129"
- [16] " Brownlee, J. (2021, January 12). Gentle introduction to the adam optimization algorithm for deep learning. Retrieved May 04, 2021, from <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>: :text=Adam"
- [17] " Brownlee, J. (2020, December 22). A gentle introduction to cross-entropy for machine learning. Retrieved May 04, 2021, from <https://machinelearningmastery.com/cross-entropy-for-machine-learning/> "
- [18] M. K. Sandhya. "Secure Data Aggregation in Wireless Sensor Networks Using Privacy Homomorphism", Communications in Computer and Information Science, 2011
- [19] "Symmetric-Key Homomorphic Encryption For Encrypted Data Processing — Proceedings Of The 2009 IEEE International Conference On Communications". Dl.Acm.Org, 2021 gives few methods of homomorphic encryption.

- [20] Cheon, Jung & Kim, Andrey & Kim, Miran Song, Yongsoo. (2017). Homomorphic Encryption for Arithmetic of Approximate Numbers. 409-437. 10.1007/978-3-319-70694-815.
- [21] Lyubashevsky V., Peikert C., Regev O. (2013) A Toolkit for Ring-LWE Cryptography. In: Johansson T., Nguyen P.Q. (eds) Advances in Cryptology – EUROCRYPT 2013. EUROCRYPT 2013. Lecture Notes in Computer Science, vol 7881. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-38348-93>
- [22] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2013. On Ideal Lattices and Learning with Errors over Rings. *J. ACM* 60, 6, Article 43 (November 2013), 35 pages. DOI:<https://doi.org/10.1145/2535925>
- [23] "https://www.microsoft.com/en-us/research/wp-content/uploads/2017/06/sealmanual_v2.2.pdf"
- [24] "https://readthedocs.org/projects/pyfhel/"
- [25] Liu, Feifei Liu, Chengyu Zhao, Lina Zhang, Xiangyu Wu, Xiaoling Xu, Xiaoyan Liu, Yulin Ma, Caiyun Wei, Shoushui He, Zhiqiang Li, Jianqing Ng, Eddie. (2018). An Open Access Database for Evaluating the Algorithms of Electrocardiogram Rhythm and Morphology Abnormality Detection. *Journal of Medical Imaging and Health Informatics*. 8. 1368-1373. 10.1166/jmih.2018.2442.
- [26] Gulli, A., Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.
- [27] Sharma, Lakhan Sunkaria, Ramesh. (2016). A Robust QRS Detection using Novel Pre-Processing Techniques and Kurtosis based Enhanced Efficiency Measurement. 87. 10.1016/j.measurement.2016.03.015.
- [28] R.W. Schafer, What is a Savitzky–Golay filter? [lecture notes], *IEEE Signal Process. Mag.* 28 (4) (2011) 111–117.
- [29] C. J. Deepu and Y. Lian, "A Joint QRS Detection and Data Compression Scheme for Wearable Sensors," in *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 1, pp. 165-175, Jan. 2015, doi: 10.1109/TBME.2014.2342879
- [30] F. Shafait, D. Keysers, T.M. Breuel, Efficient implementation of local adaptive thresholding techniques using integral images, *Proc. SPIE Int. Soc. Opt. Eng.* 6815 (2008) 6815.

- [31] L. Dev Sharma, R. K. Sunkaria and A. Kumar, "Bundle branch block detection using statistical features of QRS-complex and k-nearest neighbors," 2017 Conference on Information and Communication Technology (CICT), Gwalior, 2017, pp. 1-4, doi: 10.1109/INFOCOMTECH.2017.8340585.
- [32] Ceylan, Rahime Özbay, Yüksel. (2011). Wavelet Neural Network for Classification of Bundle Branch Blocks. Proceedings of the World Congress on Engineering 2011, WCE 2011. 2. 1003-1007.