

Prediction of Genetic Mutation from Clinical Data of Sickle Cell  
Disease using Few-Shot Siamese Bidirectional LSTM and  
Federated Learning

by

Salman Alam

19301037

Atquiya Labiba Oni

19301039

Jubair Samir

22241149

Asif Mosharrof Hossain

19201006

A thesis submitted to the Department of Computer Science and Engineering  
in partial fulfillment of the requirements for the degree of  
B.Sc. in Computer Science

Department of Computer Science and Engineering  
School of Data and Sciences  
Brac University  
May 2023

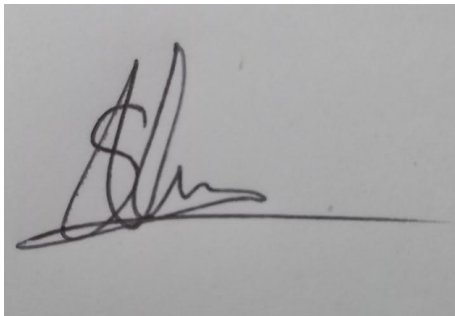
© 2023. Brac University  
All rights reserved.

# Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

## Student's Full Name & Signature:



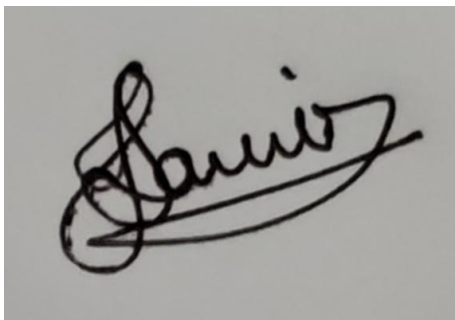
---

Salman Alam  
19301037



---

Atquiya Labiba Oni  
19301039



---

Jubair Samir  
22241149



---

Asif Mosharrof Hossain  
19201006

# Approval

The thesis/project titled “Prediction of Genetic Mutation from Clinical Data of Sickle Cell Disease using Few-Shot Siamese Bidirectional LSTM and Federated Learning” submitted by

1. Salman Alam (19301037)
2. Atquiya Labiba Oni (19301039)
3. Jubair Samir (22241149)
4. Asif Mosharrof Hossain (19201006)

Of Spring, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on May 22, 2023.

## Examining Committee:

Supervisor:  
(Member)



---

Dr.Md.Golam Rabiul Alam  
Professor  
Department of Computer Science and Engineering  
Brac University

Co-Supervisor:  
(Member)



---

Mr.Md.Tanzim Reza  
Lecturer  
Department of Computer Science and Engineering  
Brac University

Program Coordinator:  
(Member)

---

Dr.Md.Golam Rabiul Alam  
Professor  
Department of Computer Science and Engineering  
Brac University

Head of Department:  
(Chair)

---

Dr.Sadia Hamid Kazi  
Associate Professor, Chairperson  
Department of Computer Science and Engineering  
Brac University

# Abstract

Sickle Cell Disease is a monogenic genetic disorder which often leads to various repercussions affecting multiple vital organs simultaneously. However, the treatment for Sickle Cell is diverse and often varies from patient to patient, but several background studies revealed the progression and symptoms of Sickle Cell can be predicted to a great extent based on a patient's genetic mutation type in the HBB gene. Moreover, such research regarding genetic mutation prediction can be seen in other fields of medicine such as cancer, but in the case of Sickle Cell it is scarce. Furthermore, other limitations include complexity and unavailability of genetic testing, limited clinical data available and privacy concerns regarding medical information of patients. Hence, our study aimed to build a Federated Siamese Bidirectional LSTM to predict the Sickle Cell genotype from clinical data, in case of sparse and decentralized data. Consequently, a Sickle Cell clinical dataset with 216 instances and 4 different genotype class labels was pre-processed accordingly to train and evaluate the model performance. The dataset was then used to create pairs with corresponding similarity scores and the Siamese Bi-LSTM was trained for several epochs to compute similarity between two instances. The data was divided among client devices in case of federated, while the Siamese Bi-LSTM trained locally to update the global model and the test data was then used to assess their performance. Thus, based on the performance analysis the Siamese Bi-LSTM achieved accuracy of 90.45% with f1 score of 90.66% and the Federated Siamese Bi-LSTM model (FFSB-LSTM) achieved accuracy of 88.25% and f1 score of 88.57% showing significant improvement compared to the baseline KNN and Logistic Regression models.

**Keywords:** Sickle Cell, Clinical Data, Genotype, Federated Learning, Few-Shot Siamese, Federated Siamese Bidirectional LSTM

## **Acknowledgement**

Firstly, all praise to Allah for whom our thesis have been completed without any major difficulties. Secondly, to our respected supervisor Dr.Md.Golam Rabiul Alam sir and co-advisor Md.Tanzim Reza sir for their kind support and advice in our work. And finally to our parents without their throughout support it may not be possible.

# Table of Contents

<b>Declaration</b>	<b>i</b>
<b>Approval</b>	<b>ii</b>
<b>Abstract</b>	<b>iv</b>
<b>Acknowledgment</b>	<b>v</b>
<b>Table of Contents</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>List of Tables</b>	<b>ix</b>
<b>Nomenclature</b>	<b>x</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem Statement . . . . .	2
1.2 Research Contributions . . . . .	4
<b>2 Literature Review</b>	<b>5</b>
2.1 Sickle Cell Disease . . . . .	5
2.2 Genetic Mutation . . . . .	6
2.3 Related Works . . . . .	6
<b>3 Dataset</b>	<b>10</b>
3.1 Data Collection . . . . .	10
3.2 Dataset Features . . . . .	11
3.3 Exploratory Data Analysis . . . . .	12
3.4 Dataset Pre-Processing . . . . .	15
3.5 Dataset Correlation . . . . .	17
<b>4 Methodology</b>	<b>18</b>
4.1 Models . . . . .	20
4.1.1 K-Nearest Neighbor . . . . .	20
4.1.2 Logistic Regression . . . . .	21
4.2 Few Shot Siamese Learning . . . . .	22
4.3 LSTM Architecture . . . . .	23
4.3.1 Long Short Term memory (LSTM) . . . . .	23
4.3.2 Bidirectional LSTM . . . . .	25

4.3.3	Lambda Layer (L1-Distance Computation) . . . . .	25
4.3.4	Dense Layer . . . . .	26
4.3.5	Sigmoid Activation Function . . . . .	26
4.3.6	Adam optimizer . . . . .	26
4.3.7	Loss Function . . . . .	27
4.4	Federated Learning . . . . .	27
4.5	FFSB-LSTM Working Principle . . . . .	29
4.5.1	Training . . . . .	29
4.5.2	Testing . . . . .	31
<b>5</b>	<b>Implementation and Result Analysis</b>	<b>32</b>
5.1	Proposed Model Specifications . . . . .	32
5.1.1	Siamese Bi-directional LSTM . . . . .	32
5.1.2	Federated Few-Shot Siamese Bi-directional LSTM . . . . .	33
5.2	Performance Metrics . . . . .	34
5.2.1	Confusion Matrix . . . . .	34
5.2.2	Accuracy . . . . .	35
5.2.3	Precision . . . . .	35
5.2.4	Recall . . . . .	35
5.2.5	F1 Score . . . . .	35
5.3	Result Analysis . . . . .	36
5.3.1	K-Nearest Neighbour . . . . .	36
5.3.2	Logistic Regression . . . . .	37
5.3.3	Siamese Bi-Directional LSTM . . . . .	37
5.3.4	Federated Siamese Bi-Directional LSTM . . . . .	38
5.4	Overall Performance Comparison . . . . .	39
<b>6</b>	<b>Conclusion</b>	<b>41</b>
	<b>Bibliography</b>	<b>44</b>



# List of Figures

3.1	Dataset with clinical data and HBB genotype . . . . .	11
3.2	Dist Plot of Numerical Features . . . . .	13
3.3	Box-plot of the columns with most outliers . . . . .	14
3.4	Bar-Graph for imbalance in the label column . . . . .	14
3.5	Genotype class label after resampling . . . . .	16
3.6	Heatmap showing correlation between features of the dataset . . . . .	17
4.1	Top level overview of proposed FFSB-LSTM . . . . .	18
4.2	KNN visualization in binary classification and imputation [18] . . . . .	20
4.3	Graphical representation of the Sigmoid function [19] . . . . .	21
4.4	Basic Siamese Network [20] . . . . .	22
4.5	Siamese Network Similarity Computation [22] . . . . .	23
4.6	Basic LSTM Architecture [24] . . . . .	24
4.7	Federated Learning Architecture [30] . . . . .	28
4.8	FFSB-LSTM Model Architecture . . . . .	30
5.1	Siamese Bi-directional LSTM model summary . . . . .	33
5.2	Confusion Matrix for Multi-Class Classification [32] . . . . .	34
5.3	KNN Confusion matrix . . . . .	36
5.4	Logistic Regression Confusion matrix . . . . .	37
5.5	Siamese Bi-LSTM Confusion matrix . . . . .	37
5.6	FFSB-LSTM Confusion matrix . . . . .	38
5.7	Performance comparison Bar-Chart . . . . .	39

# List of Tables

3.1	Dataset Features . . . . .	11
3.2	Number of instances in each genotype . . . . .	13
3.3	Sickle Cell Genotype Column Mapping . . . . .	15
4.1	Pairs Dataset to Train Bi-LSTM . . . . .	29
5.1	Performance comparison between models . . . . .	39

# Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

*Bi – LSTM* Bi-directional Long Short Term Memory

*FFSB – LSTM* Federated Few-Shot Siamese Bi-directional Long Short Term Memory

*HBB* Haemoglobin Subunit Beta

*KNN* K-Nearest Neighbors

*LSTM* Long Short Term Memory

# Chapter 1

## Introduction

Sickle Cell Disease is among the lethal monogenic genetic diseases which is responsible for the deaths of numerous children and adults around the world, especially in the African continent. Moreover, the varying symptoms of Sickle Cell Disease and its damage to various organs simultaneously such as brain, lungs, liver or kidney makes the genetic disease even more deadly. The most common causes of death include extreme pain, kidney failure or stroke [1]. Even though those with Sickle Cell trait tend to survive longer, the rise of complications are always varying, sudden and could pose serious threat to a person's life if not anticipated early. Moreover, Genetic mutation is the change found in the genome sequence which might cause abnormality in a person, in this case it is the mutation in the Haemoglobin Subunit Beta (HBB) gene that is the fundamental cause of inheriting the Sickle Cell disorder.

The numerous research studies conducted on Sickle Cell Disease and its consequences, resulted in the development of better testing, detection and prevention to a great extent. Additionally, tests like HPLC exist for detection of Sickle Cell, but genetic testing is still required to confirm Sickle Cell in a person or child. Furthermore, the Sickle Cell complications and symptoms are dependent on the type of Sickle Cell genotype to a great extent [2]. However, it is observed that physical tests available for the identification of the type of genetic mutation have their own limitations and constraints, which includes, unavailability in many regions of Africa and other countries, false positives, invasive tests and many more [2]. Hence, in a lot of these cases vital information regarding a Sickle Cell patient and their future complications remains unattainable. Additionally, several research studies regarding cancer and other genetic disorders have shown the implementation of deep learning models to predict genetic mutation from some sort of clinical data. However, such studies are hardly observed in the field of Sickle Cell even though the prediction of genetic mutation can provide valuable information.

The other major issue in case of genetic disorders, including Sickle Cell, remains to be a shortage of clinical data due to several reasons, such as not recording data centrally, lack of enough patients willing to submit their clinical information etc. Moreover, the issue of privacy concern, decentralized data, and medical data inaccessible to the public are also prevalent in the field of Sickle Cell Disease. Hence, our study proposes a model to address the issue of predicting genetic mutation using deep learning along with limited and decentralized data by combining the concepts of Few-shot

Siamese and Federated Learning, using a Bidirectional LSTM model, which will be called Federated Few-Shot Siamese Bi-directional LSTM (FFSB-LSTM). Furthermore, LSTM is often used for sequential data but our research hopes to discover its capability of capturing dependencies and complex patterns within structured or tabular data. Therefore, the aim of this particular research study will be the prediction of genetic mutation from clinical data of Sickle Cell patients using the Federated Siamese Bidirectional LSTM network.

## 1.1 Problem Statement

Sickle Cell is a severe genetic disease which is polypharmacy in nature, that means the genetic disorder gives rise to multiple diverse symptoms affecting various vital organs simultaneously including brain, lungs and kidney. Hence, a single patient requires several different treatments throughout their lifetime, but such situations can be avoided or the risk can be minimised if the future symptoms can be predicted at an early stage and treated accordingly. Consequently, knowledge regarding the genetic mutation of a Sickle Cell patient in the HBB gene can provide vital information about the complications that might arise in the future, since a lot of these symptoms are highly correlated with the type of mutation [3]. Even though the most common method of identifying the genetic mutation is conducting a genetic test, such tests often pose certain limitations which restrict its widespread use such as in the African continent where Sickle Cell is more prevalent. The limitations include its unavailability in all medical centres and false positives leading to retesting making the procedure expensive and time consuming. Moreover, emotions play a role when genetic testing is required, since it is observed that parents often do not approve when the concerned patient is their child [4]. Therefore, knowing the genetic mutation in the HBB gene responsible for the Sickle Cell genetic disease can be significant for a patient in terms of early detection and treatment, but the limitations regarding genetic testing often constricts the possibility of gaining knowledge about the genotype in patients.

Research studies regarding the prediction of genetic mutation from clinical data is prevalent in case of cancer studies where genotype is a significant factor like Sickle Cell. The papers [5] and [6] discuss techniques and deep learning models achieving great accuracy while predicting the genotype from H&E images in case of breast cancer and liver cancer respectively. However, such studies and implementation of automated deep learning models to predict genetic mutation is rarely observed in case of Sickle Cell, even though these factors are highly significant in comprehending a particular Sickle Cell affected patient's situation and developing treatment for the future, known as precision medicine. Moreover, in the domain of genetic diseases, like Sickle Cell in our case, there lies the issue of data shortage and dataset inaccessibility due to several reasons. There is a shortage of data for reasons which include genetic diseases are not as common as other prevalent ones or the spread of Sickle Cell disease is most often noticed in the African continent where data recording and collection is scarce to some extent. Hence, to tackle this issue of data shortage we will be implementing Few-Shot Siamese techniques to our deep learning model, where we aim to build an effective genotype prediction model using a small amount

of data. Furthermore, the data collected by different organizations and hospitals regarding Sickle Cell patients is often not accessible by the public, since there are security issues and privacy concerns. Thus, the concept of federated learning which utilizes decentralized data to train a global model by training local models at client and receives only the weight and parameters, could solve the issue of data inaccessibility due to privacy concerns to a great extent. Consequently, the advent of deep learning in such a sector will progress further studies regarding gene therapy and personalized treatment since medical professionals can predict certain aspects of a Sickle Cell patient at an early stage.

Additionally, our research aims to build a Few-Shot Siamese Bidirectional LSTM deep learning model called FFSB-LSTM for the prediction of genotype from tabular or structured data of Sickle Cell patients, even though LSTM models are meant for sequential or time-series data and not for non-sequential tabular data as in our case. However, we will be using the LSTM to compute the similarity between two rows of data with each consisting of multiple features. Hence, any conventional technique to compute similarity between two rows or vectors such as distance metrics (Euclidean/Manhattan), K-means clustering or embedding methods would only determine the similarity based on the feature values and not consider the complex relationships or dependencies between the features. However, LSTM is capable of computing similarity between two instances based on their values and also by capturing the temporal dependencies or complex patterns within the features. Hence, our research also aims to deduce the credibility and efficiency of Bidirectional LSTM while computing similarity between two instances from a structured dataset.

Therefore, the problem statement can be established as:

**How effective will a Federated Siamese Bidirectional LSTM model be when predicting genetic mutation from clinical data of Sickle Cell patients?**

Thus, the above discussion reveals a research gap in the field of Sickle Cell studies where it is possible to predict the genotype using deep learning models with limited data, which is the fundamental target of this study. Moreover, the introduction of deep learning in the field of medical science will only make the process more precise and efficient by helping the medical professionals instead of substituting them. Consequently, the prediction of genotype will lead to the development of more effective personalised treatments, early detection and act as an aid to medical professionals before opting for other physical testing methods.

## 1.2 Research Contributions

The research aims to predict the genetic mutation of Sickle Cell patients from clinical data using our proposed Federated Siamese Bidirectional LSTM (FFSB-LSTM), in order to develop effective personalized treatments in future based on the outcome, early detection and aid clinicians in deciding whether further physical testing is required. Thus, the study consists of multiple aspects and domains each with its own objectives, while also fulfilling the ultimate goal of predicting genotype. Therefore, the research contributions can be established as:

1. We implemented our proposed model to predict the genotype of Sickle Cell from clinical data, in order to aid medical professionals for determining future complications and appropriate treatments, where genetic testing is not viable.
2. We introduce our proposed model combining the concepts of Few-Shot Siamese and Federated Learning, the Federated Few-Shot Siamese Bidirectional LSTM (FFSB-LSTM), to handle and train using both limited and decentralized data protecting patient privacy.
3. Our research study explored the capability of Bi-LSTM when learning temporal dependencies and complex relationships between features from non-sequential data of a structured dataset.
4. Training and testing our proposed FFSB-LSTM model on a small dataset with 216 instances and 4 unique genotype classes regarding Sickle Cell clinical data showed substantial improvement in performance from baseline KNN and Logistic Regression models, with FFSB-LSTM achieving 88.25% accuracy and 88.57% f1-score.

# Chapter 2

## Literature Review

The background study for our research revealed that work regarding the prediction of genotype in the case of Sickle Cell is rare, even though it has significance. Thus, this section focused on reviewing papers based on different aspects of our study, including Sickle Cell Disease, significance of genotype in Sickle Cell and related works where studies concerning genotype prediction done for other diseases such as Cancer is discussed.

### 2.1 Sickle Cell Disease

Sickle cell disease (SCD) is a collection of inherited disorders regarding blood, characterized by abnormal hemoglobin, which is caused by mutation in Hemoglobin Subunit Beta (HBB), a gene that codes for proteins [1]. The mutation in HBB leads to abnormal hemoglobin Hbs. Usually, red blood cells (RBC) are disc-shaped, however in case of SCD, the presence of Hbs causes RBCs to become crescent/sickle shape and become rigid. This in turn causes blockages in blood vessels by sticking on its walls resulting in problems in blood flow, and can even cause tissue or organ damage. Between 300,000 and 400,000 neonates are estimated to be affected globally each year, where the majority of the cases are in the sub-Saharan African Region [1]. There are many types of SCDs ranging from Severe, Moderate, Mild to Very Mild [7]. The different characteristics depending on gene mutation and occur in different geographical regions, (for example Severe SCD - eastern Mediterranean region & India). One of the most common clinical complications is acute pain. Individuals with SCD can also experience neurocognitive dysfunction, retinopathy, pulmonary hypertension, Anaemia Leukocytosis, chronic pain and other complications in pregnancy [1]. Early diagnosis is very important to improve survivability rate, and steps such as newborn screening, post neonatal testing have been taken in Europe, USA and India [1]. However, these programmes have been a challenge to implement in low-income countries in Africa. Till this date, reports have suggested that no African country has implemented a national screening programme for SCD, where this is a region to have reported 75% of the births with SCD worldwide [1]. Although some evidence supports the use of blood transfusions and hydroxycarbamide in some situations, SCD clinical care is still at a very basic level, and there are medications that have yet to be produced which would particularly target the pathophysiology of this disease [7]. Gene therapies are seeming to become a viable option however it is very expensive and would be very difficult for aiding low-income regions [7].



## 2.2 Genetic Mutation

Genotype is the genetic composition of an individual while the phenotype is the visible physical characteristics which result from an individual's genotype. Most frequently occurring genotypes that causes SCD are, Hb SS, Hb SC, Hb S $\beta$ + thalassemia and Hb S $\beta$ 0- thalassemia. The severity and frequency of clinical complications is dependent on genotype as it differs between different genotypes of sickle cell disease. Symptoms tend to be more severe in SS and S $\beta$ 0- thalassemia and milder in SC disease with the exception of proliferative sickle retinopathy (PSR) [3]. If clinical complications are compared between people having Hb SC or Hb S $\beta$ + thalassemia mutation and people having Hb SS mutation, the Hb SS mutation results in lower hemoglobin values and more hemolysis indicators [8]. As a result, people with Hb SS are more prone to diseases like ulcers, stroke, vaso-occlusive episodes, and early death. Moreover, depending on the molecular  $\beta$ -thalassemia gene mutation and the level of HbA generated, sickle cell S $\beta$ + thalassemia possesses a very wide clinical spectrum. Avascular necrosis that affects individuals with all kinds of sickle cell genotypes tends to manifest earlier in individuals with Hb SS disease, potentially leading to greater morbidity and reduced quality of life [8]. Along with genotype, geographic areas more susceptible to malaria also increase sickle cell disease severity [8]. Lastly, the author concludes with the remarks that genetic mutation which can cause variation in phenotype needs to be known for better prediction of medical complications severity in Sickle Cell disease [8].

The most prevalent method of identifying a certain genetic mutation is through genetic testing, but the papers [9] and [4] discuss the limitations of genetic testing in various situations. According to the paper [9], there are multiple disadvantages of genetic testing such as the tests still have a high rate of false positives and the results are also sometimes unclear which often results in retesting and makes the process more expensive. Moreover, there are ethical limitations to genetic testing and sometimes individuals are uncomfortable with such testing methods [9]. Further, the limitations are also discussed in [4] which states a person's emotions often influence their decision of accepting a genetic test. Furthermore, the paper raises the issue of availability of such genetic testing in various regions and hopes these techniques will become more easier to access in the future.

## 2.3 Related Works

The study [5] focuses on the creating a deep learning model in order to predict the genetic mutation in the BRCA gene from Histopathology Images to prevent breast cancer in patients. According to the paper [5], predicting the genetic mutation in BRCA1/2 can provide valuable information regarding the future risk of developing breast cancer and hence patients can opt for gene therapy for early prevention. Their dataset included H&E images from two medical centers in China and the total of 222 H&E images were collected from the two datasets. Moreover, the study developed a deep CNN model of ResNet on whole-slide images to predict the genetic mutation in breast cancer. Thus, their model showed a 95 percent confidence interval and the paper stated it was successful in predicting the genetic mutation in gBRCA from images alone using the deep learning model developed.

The paper [6] also develops a deep learning model in order to classify and determine genetic mutation from Histopathology Images in Liver cancer. According to the paper [6], Hepatocellular carcinoma (HCC) is one the severe types of liver cancer, which is often detected at terminal stages when recovery becomes difficult and painful. However, identifying the genetic mutations which cause HCC can become one of the ways to prevent severe cases through early detection and targeted treatment [6]. Hence, the paper trains a deep learning neural network, known as inception V3 developed by Google, on 491 histopathological images collected from the Genomic Data Databases. Additionally, the model was trained to predict ten most common genetic mutations that are considered responsible for HCC, which includes CTNNB1, FMN2 or ZFX4. Moreover, the paper states the performance of the classifier as high with a 95 percent confidence interval at distinguishing tumor from healthy liver. Furthermore, the performance of genetic mutation prediction, based upon Matthew's correlation coefficient, showed a 96 percent accuracy, which was equivalent to the expertise of a 5-year experienced pathologist. Therefore, the paper [6] concluded with the remarks that such deep learning models could aid medical professionals in predicting various diseases or genetic mutations for early prevention and targeted treatment.

The research article [10] discusses the importance of examination of histopathology images for evaluating lung tumors extensively and predicting gene mutations. Moreover, Adenocarcinoma (LUAD) and squamous cell carcinoma (LUSC) are most common variant of lung tumor so they require skillful pathologist to differentiate them [10]. Additionally, predicting gene alterations from histopathology images with low cost would help cancer patients to get better treatment. Consequently, they built and trained a deep CNN network known as Inception V3 to classify LUAD, LUSC and also predict gene mutation in which the image is taken as the input. They collected data from the Genomic Data Commons database containing 1,634 whole-slide images, 1,176 tumor tissues and 459 normal tissues [10]. According to the paper [10], their model was able to predict six most frequently found gene mutations in LUAD which are STK11, EGFR, FAT1, SETBP1, KRAS and TP53 from images and their AUC ranged from 0.733 to 0.856. Moreover, their model performed better than existing works and also their results were similar to pathologist's evaluations. Hence, the paper [10] concluded with observations that deep learning models can be used to aid pathologists in classification from images and also predict gene mutation which would help patients to receive better personalized treatment according to their individual's genotype.

In the paper [11], have developed a deep learning model which can predict EGFR mutation from CT images. According to them EGFR prediction is important because it can assist doctors in providing treatments for Lung Adenocarcinoma (LA) and their method of prediction using CT image is inexpensive and easily accessible. They have used 14926 CT images of tumors to train their model and 1.28 million images from ImageNet dataset for transfer learning [11]. When their deep learning model is trained by a CT image it predicts the chance of the tumor caused by EGFR mutation and also the characteristics due to the mutation. Furthermore, they observed the response by filter after giving image as input which helped them to derive the

attributes filtered by the filter. Their model performed with 95% of confidence interval with AUC of 0.81 and the results outperformed other existing works. Therefore, they believe that as CT image is easily accessible and frequently done by LA patients with the help of deep learning models EGFR mutation can be predicted easily.

The paper [12] also used deep learning models in their research to predict EGFR along with KRAS mutation from CT images to bring advancement in personalized treatment for lung cancer patients. According to the paper [12], lung cancer has the highest death rate compared to other cancers. Hence, personalized treatment is crucial for lung cancer treatment as it improves recovery rate but in it requires classifying gene mutation. To train their model they used training dataset containing 363 patients gathered from their partner hospital and the validation dataset consisting 162 patients from The Cancer Imaging Archive (TCIA). In this study [12], they developed a multi-channel and multi-task deep learning (MMDL) model in which multi-channel helped the multi-task deep learning (MMDL) model to distinguish the lumps extensively. They compared their results with existing studies performed using commonly used models to predict the EGFR and KRAS mutation. In the training dataset, their proposed model had AUC and accuracy of 86.56%, 79.43%, for EGFR mutation, which was higher than other traditional models. For KRAS prediction their model had higher scores of AUC and accuracy with 78.97%, 72.25%. Moreover, in the validation dataset their model outperformed other models in both cases of mutation prediction with AUC, accuracy of 81.29%, 75.06% for EGFR and 74.23%, 69.64% for KRAS mutation. According to the research [12], their method of predicting gene mutation with a deep learning model from CT image is cost effective, fast, convenient and also gives better results than existing works which would benefit lung cancer patients.

Pancreatic cancer is one of the fatal cancers and the death rate is high [13]. They have used 107 images of pancreatic cancer patients to predict the p53 mutation and programmed death ligand 1 (PD-L1) status. The features were chosen by Mann-Whitney U test and random forest function and they have extracted image features which helped to develop their model for mutation prediction [13]. Their model had AUC of 0.795 for p53 and AUC of 0.683 for programmed death ligand 1 (PD-L1) status. Moreover, the concept of radiogenomics was used for other cancers except pancreatic cancers and this prediction can contribute in developing personalized medicine.

Glioblastoma (GBM) is a rapidly growing brain tumor which affects nearby brain tissues by invading them [14]. The aim of this research was to effectively predict Isocitrate Dehydrogenase 1 (IDH1) mutation in GBM using different machine learning techniques with the aid of Quantitative Radiomic Data, which could aid with new data from multiparametric magnetic resonance imaging (MRI). Mutation of the IDH1 gene is said to arise in about 12% of the GBM cases. Noninvasive techniques for the precise prediction of IDH1 mutation status have received a lot of attention since they can be used without incurring the expense of testing or the risk of surgery. So, the purpose of this work was to predict IDH1 mutation status using machine learning-based classification models based on preoperative MRI characteristics. For data collection, a group of 88 cases dated between May 2010 and June

2015 were chosen from the Department of Neurosurgery’s data registry, following some specific set of predetermined criterias, and all patients were treated following the same treatment regimen to keep consistency. After that MRI images were obtained, and apparent diffusion coefficient (ADC) images were deduced. Then, volumetric segmentation and ROI analysis were done based on the MRI image outcome. Furthermore, statistical analysis was done based on the results where Chi-squared, Fisher exact tests were used. Wilcoxon rank sum test was done for deducing dissimilarities in continuous variables between IDHmut and IDHwt groups. Lastly, Label (IDHmut/wt) and features information extracted from inputs images were passed through a classification algorithm consisting of 8 machine learning classifiers: Support Vector Machine (SVM), K- Nearest Neighbour (KNN), Decision Tree, Adaboost, Random Forest, Naive Bayes, Gradient Boost and Linear Discriminant Analysis. For training the classifier sample size was 88 with 5 K-fold, whereas for testing sample size was 35, which was independently created from another registry. Among all the machine learning classifiers KNN performed the best both during training (87.3% and 81.3% accuracy respectively) while Naive Bayes performing the worst in training (70.3% accuracy) and Linear Discriminant Analysis testing (66.3%) [14]. The current study’s findings demonstrate that quantitative radiomic data may accurately predict the molecular status of GBM, and that machine learning technologies can be applied to increase prediction accuracy.

An crucial biomarker for the detection and analysis of glioma is isocitrate dehydrogenase (IDH) gene mutation [15]. Convolutional neural networks (CNN) provides decent performance in IDH mutation prediction, but it is unable to learn from network and geometric data (non- Euclidean data). The aim of this research was to develop a multi-modal learning framework to extract attributes from the focal tumor picture, tumor geometrics, and global brain networks using three different encoders. Anatomical MRI data was collected from 407 glioma patients , where 20 patients were used for training a self-supervised learning model [15]. The remaining 387 were split into testing and training with a 7:3 ratio. The training models were then reevaluated with 117 patients data independent of the previous ones. At first , a self-supervised learning method was used which aided in the creation of brain networks from anatomical multi-sequence MRI. After that, creation of hierarchical graph attention for the brain network encoders was accomplished which helped in retrieving tumor-related properties of the brain network. Furthermore, bi-level multi-modal contrastive loss was designed to align with tumor-related network attributes with focal tumor attributes over the domain gap. At last, a population graph including the multi-modal data and forecast the genotype of the patients was created. The learning framework showed promising results when compared with the modern up to date models. All their study showed promising results; it was later discussed that it has some limitations such as the scarcity of training data due to glioma being rare, and their dataset being imbalanced of the IDH mutant. However, it was concluded that their findings outperformed traditional CNNs, but further development is still needed to improve accuracy by involving larger dataset.

# Chapter 3

## Dataset

### 3.1 Data Collection

The dataset selected for this particular research has been collected from the study [16], which physically obtains the clinical data and HBB genotype of 217 children infected with *Plasmodium falciparum* from Jaramogi Oginga Odinga Teaching and Referral Hospital in Western Kenya between 2018 and 2019. Moreover, the study cohort included patients who were from 1 years old to 16 years old and about 50% of the data being males and the rest being females. Even though the study focused on Kenyan children with Sickle Cell, it can be observed that the study population is diverse with varying characteristics. Additionally, for each patient 16 different medical aspects were tested and identified, which means the dataset has 217 instances with 19 features or columns including their gender and age. Furthermore, the reasons for selecting this particular dataset can be established as follows:

1. Several medical aspects for each patient have been tested and recorded, which will aid for an in-depth analysis of our prediction model
2. Four different types of genotype in HBB gene combining all Sickle Cell patients is available in this dataset. Hence, diverse targeted labels are present in one dataset
3. The dataset has no missing values, which means significant imputation or assumption will not be required
4. The study populations' age is very diverse, the gender distribution being close to 50% and all the patients belong to the same country. Thus, the impact of external environmental factors and demographics will be minimum

Moreover, according to [17], the data recorded for the patients has been collected by conducting physical medical tests. For instance, the genetic mutation was identified through genetic testing using Taqman SNP Genotyping Assay and rest of the medical features were mostly collected from blood samples of the Sickle Cell patients. For instance, Haematological values such as red blood cell count, white blood cell count or mean cell volume were determined from the blood samples. Consequently, the hemoglobin level or platelet count were deduced from the blood smears of the study cohort. Hence, the medical features are relatively simple to identify since only blood samples were required except for the HBB genotyping. Therefore, the dataset used

in our research is a tabular or structured dataset arranged with various medical features and the subsequent HBB genotype of the study cohort.

Months	SEX	WBC	LYM%	MON%	GRAN%	RBC	MCV(FL)	HCT	MCH(Pg)	MCHC	RDW	HB	THR	MPV	PCT	PDW	Pf. Infect	Sickle Cell Genotype	
0	1	0	12.43	52.5	19.5	28.0	4.46	70.9	31.6	23.5	33.2	10.8	10.5	348	4.5	0.16	8.3	Negative	Homozygous HbAA/HbAA
1	1	0	10.76	59.0	12.5	28.0	5.21	94.3	49.1	27.4	29.1	12.4	14.3	312	5.6	0.17	9.6	Negative	Homozygous HbAA/HbAA
2	1	0	11.15	49.7	16.3	34.0	4.98	96.0	47.8	25.1	26.1	12.6	12.5	280	6.5	0.18	10.9	Negative	Homozygous HbAA/HbAA
3	1	0	9.37	69.9	5.6	24.5	5.39	75.5	40.6	22.6	30.0	11.6	12.2	239	6.0	0.14	11.9	Positive	Heterozygous HbAA/HbSS
4	1	0	7.81	16.5	7.4	46.1	5.14	84.8	43.5	22.7	26.8	13.4	11.7	344	5.5	0.19	8.8	Negative	Heterozygous HbAA/HbSS
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
212	3	1	7.57	41.0	11.5	47.5	4.47	76.8	34.3	24.6	32.0	11.2	11.0	237	5.2	0.12	8.4	Negative	Homozygous HbAA/HbAA
213	3	1	4.81	36.1	10.3	53.6	4.74	78.8	37.3	24.6	31.3	10.5	11.7	196	5.8	0.11	10.5	Negative	Homozygous HbAA/HbAA
214	3	1	5.51	45.9	11.3	42.8	4.55	83.8	38.1	24.1	28.8	11.7	11.0	209	5.8	0.12	10.4	Negative	Homozygous HbAA/HbAA
215	3	1	4.42	19.3	8.4	72.3	5.75	82.8	47.6	22.9	27.7	9.5	13.2	111	5.6	0.06	12.6	Positive	Homozygous HbAA/HbAA
216	3	1	7.60	20.9	9.2	69.9	4.60	80.5	37.0	22.8	28.3	15.8	10.5	357	5.7	0.20	10.6	Negative	Homozygous HbAA/HbAA

Figure 3.1: Dataset with clinical data and HBB genotype

## 3.2 Dataset Features

There are 19 clinical and demographics features in the dataset. The features with their full form are represented in the table below:

Table 3.1: Dataset Features

Feature Number	Feature	Full Form of features
1	Months	Months
2	SEX	SEX
3	WBC	White Blood Cell Count
4	LYM%	Lymphocytes
5	MON%	Monocyte
6	GRAN%	Granulocyte
7	RBC	Red Blood Cell Count
8	MCV(FL)	Mean Cell Volume
9	HCT	Hematocrit Values
10	MCH(Pg)	Mean cell Haemoglobin
11	MCHC	Mean cell Haemoglobin Concentration
12	RDW	Red Cell Distribution levels
13	HB	Haemoglobin level
14	THR	Thrombocytopenia
15	MPV	Mean Platelet Volume
16	PCT	Procalcitonin levels
17	PDW	Platelet Distribution Width
18	Pf. Infect	Plasmodium falciparum
19	Sickle Cell Genotype	Sickle Cell Genotype

The dataset contains demographic features like age/months and gender/sex along with clinical features identified from blood samples of the Sickle Cell patients. Consequently, the feature Sickle Cell Genotype will be the target label from the dataset for our research.

The paper [16], [17] and various other studies assert that the type of genetic mutation has a significant impact on most of the symptoms or medical complications in case of sickle Cell and could provide valuable information regarding the future of these patients, especially the children. Moreover, according to [16], the differences in several Haematological aspects is comparable with the type of HBB genotype in a patient as revealed in their study. For instance, anaemia is more prevalent in children with the genotype HBSS along with malaria and red blood cell count and haemoglobin levels are inversely proportional to the red blood cell width in case of the patients possessing HBSS genotype for Sickle Cell. Similarly, the study [17] reveals significant relation between the type of genotype and other clinical aspects including white blood cell count, monocytes or red blood cell count. Furthermore, the medical features besides genotype in the dataset are also correlated amongst themselves since most of the values are derived from blood smears of the patients.

Thus, the above mentioned papers assert the significance of genotype in case of Sickle Cell disease along with the correlation between haematological features and genotype. Hence, the dataset used for this study provides adequate clinical and demographic features to build a model for the prediction of genotype in Sickle Cell patients, while also studying the underlying complex relationships between the features and type of genetic mutation.

### 3.3 Exploratory Data Analysis

Initially, the dataset was analysed where it was observed that there are 217 instances and 19 columns, with no null values in any of the columns. Moreover, the two columns ‘Pf.infect’ and ‘Sickle Cell Genotype’ are categorical columns, which will require encoding when pre-processing the dataset. Furthermore, a detailed multi-class analysis of the Sickle Cell genotype column revealed that there are 5 unique genetic mutation classes in the column such as Homozygous HbAA/HbAA, Heterozygous HbAA/HbSS, Homozygous HbSS/HbSS, Homozygous HbAA/HbSS and Heterozygous HbSS/HbSS. The number of instances for each genotype is shown in table below:

Table 3.2: Number of instances in each genotype

Genotype	Number of instances
Homozygous HbAA/HbAA	148
Heterozygous HbAA/HbSS	43
Homozygous HbSS/HbSS	23
Homozygous HbAA/HbSS	2
Heterozygous HbSS/HbSS	1

Additionally, the genetic mutations are provided in the form where the first and second mutation are separated by '/', since each genetic mutation is from one copy in the HBB gene. Analysing the number of instances in each genotype reveals that the genotype Heterozygous HbSS/HbSS has only one instance and should be dropped since there is no significant data available for this genotype. Moreover, the genotype Homozygous HbAA/HbAA is the normal HBB genotype and not Sickle Cell, which was added in the dataset as reference, while the rest of the genotypes represent the genetic mutation in HBB gene in case of Sickle Cell.

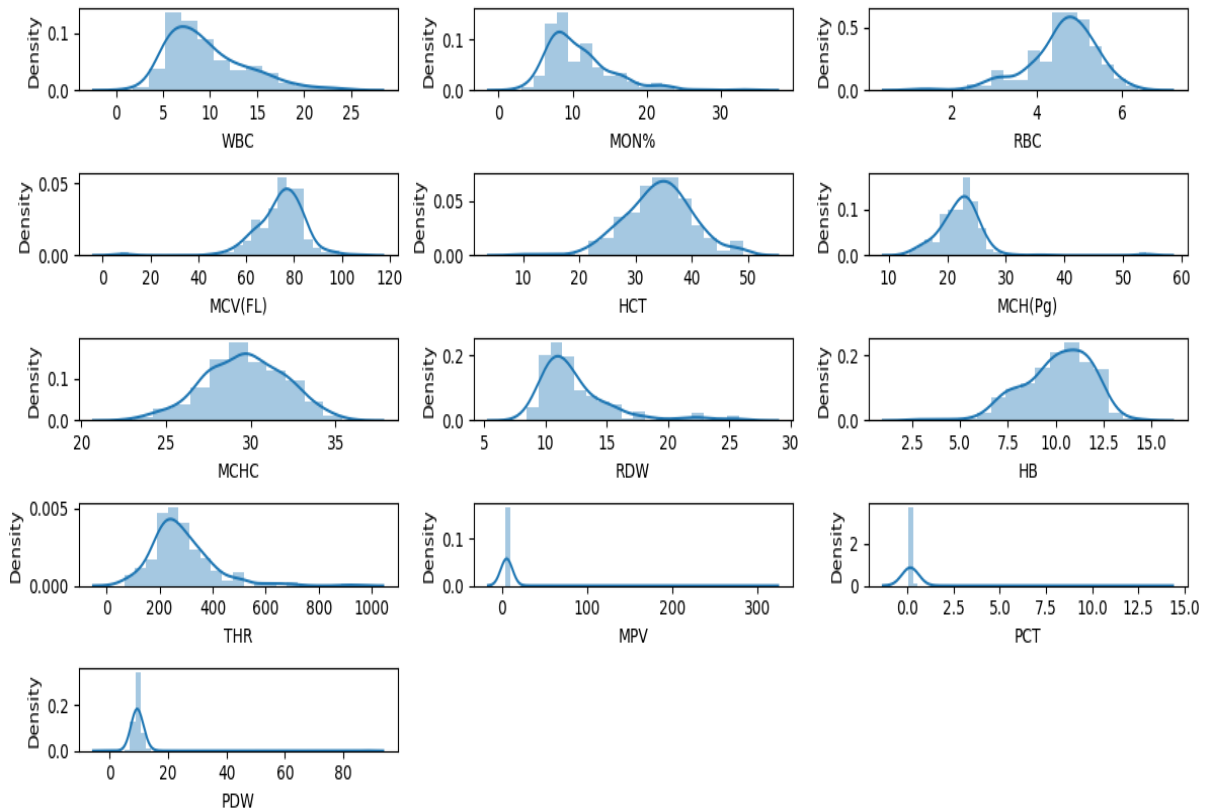


Figure 3.2: Dist Plot of Numerical Features



Moreover, the above dist-plots depict density with respect to the data distribution for each of the numerical features. Hence, observing the distplots it can be deduced that data distribution for most of the numerical columns have a gaussian distribution, which means the distribution is somewhat symmetric about the mean of the data. Thus, it would be better to scale the dataset using a standard scaler since most of the columns have normal distribution of data.

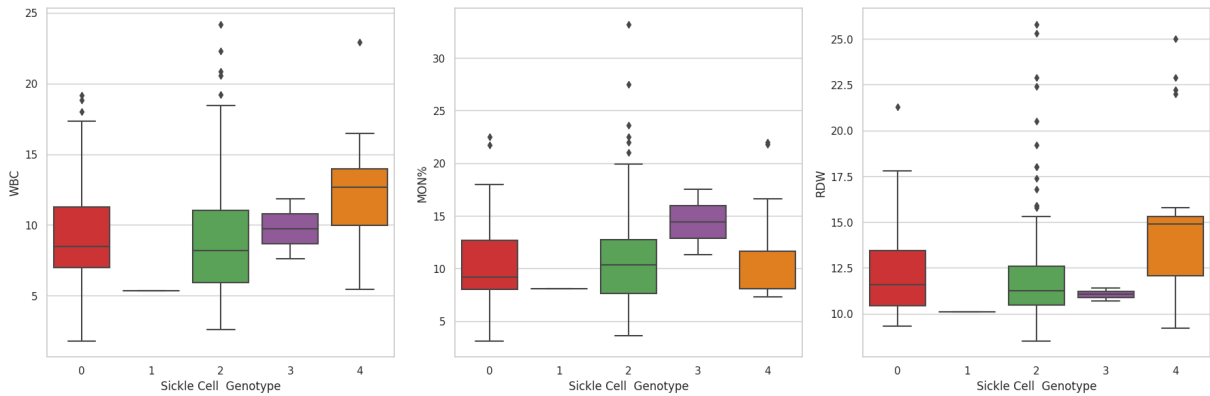


Figure 3.3: Box-plot of the columns with most outliers

Therefore, analysing the box-plots of the features with the label column Sickle Cell genotype reveals that a few of the columns have outliers within them, which might cause biases when training the models. Furthermore, the columns ‘WBC’, ‘MON%’ and ‘RDW’ contain the most outliers as depicted in the figure above. As a result, the outliers will be required to be removed during dataset pre-processing in order to avoid any sort of biases while training the models.

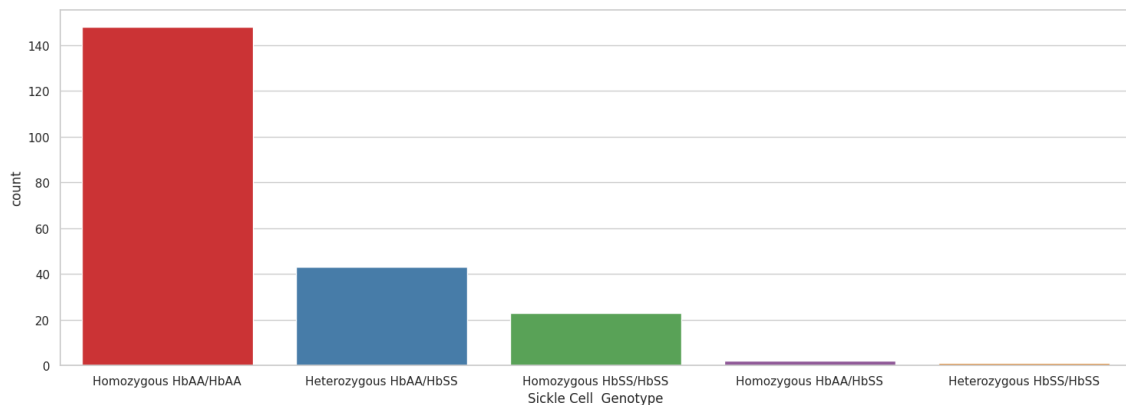


Figure 3.4: Bar-Graph for imbalance in the label column

Additionally, the above bar chart reveals that there is imbalance within the genotypes present in the label column Sickle Cell Genotype and could hamper the training of the models if the imbalance is not removed. Observing the bar graph, it can be deduced that Homozygous HbAA/HbAA has majority of the instances and Homozygous HbAA/HbSS along with Heterozygous HbSS/HbSS make up the minority class of the dataset. Thus, the identified imbalance will be removed during pre-processing of the dataset.

### 3.4 Dataset Pre-Processing

Several pre-processing techniques are applied on the dataset, in order to clean the data and make it usable for implementing the machine learning and deep learning models. Moreover, Python was used to conduct the dataset analysis, data pre-processing and the model analysis.

The initial dataset analysis revealed that the genotype Heterozygous HbSS/HbSS has only one instance and would not provide adequate data for training and testing of the models. Hence, the instance with genotype Heterozygous HbSS/HbSS is dropped from the dataset and the updated shape becomes 216 instances with 4 unique genotype classes. Consequently, the categorical features such as ‘Pf.infect’ and ‘Sickle Cell Genotype’ can not be processed by the deep learning models unless converted to numerical values. Hence, encoding techniques like label encoder from the sklearn library were used to map certain categorical features. As a result, the binary values in ‘Pf.infect’, which are negative and positive were mapped to 0 and 1 respectively. On the other hand, the label classes in the ‘Sickle Cell Genotype’ column were mapped as follows:

Table 3.3: Sickle Cell Genotype Column Mapping

Label Classes	Mapping
Heterozygous HbAA/HbSS	0
Homozygous HbAA/HbAA	1
Homozygous HbAA/HbSS	2
Homozygous HbSS/HbSS	3

In the next step, the columns ‘months’ representing the age of the children were dropped from the dataset since it has no correlation with the output label genotype. Additionally, during data analysis it was noted that few of the columns had outliers among them and so the outliers were identified for the numerical columns using the z-score formula, which is:

$$z - score = \frac{x - mean}{standard\ deviation} \quad (3.1)$$

The instances with a z-score value greater than 2 or less than -2 were established as outliers, since these values fall outside 95% of the data in that particular column and these outliers were substituted using the KNN imputer from the sklearn library was used with a k value of 2 and the distance function being Euclidean function, which will compute distance from the nearest 2 values without null and estimate for the numerical imputation. Furthermore, the dataset analysis showed imbalance with the genotype label classes and to remove such imbalance the SMOTE technique from the imbalanced learn library is used, which will create synthetic data for the

minority classes based on the nearest neighbours and create equality among all the label classes for proper training of the models. Hence, the after affect of smote is visualized in the following bar graph:

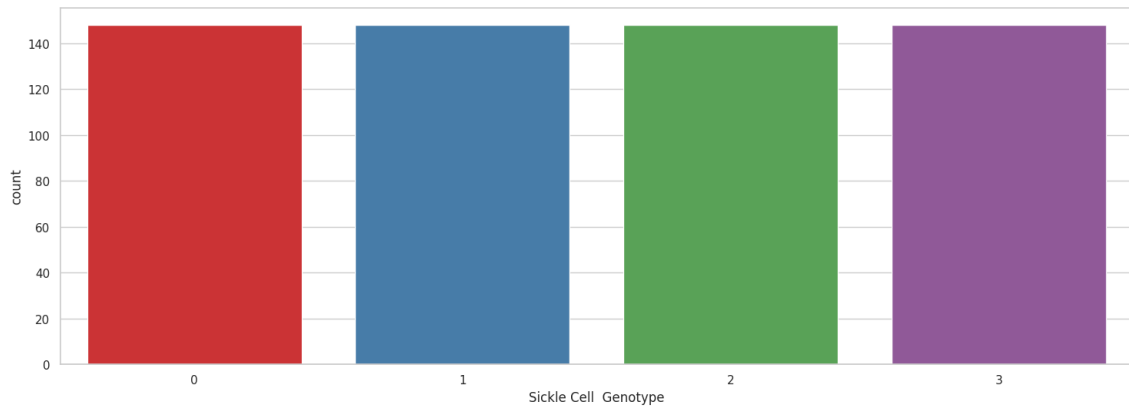


Figure 3.5: Genotype class label after resampling

In the end, the preprocessed dataset had 17 medical features for training and testing of the models in order to predict the genotype class. Furthermore, the dataset will be split into train and test with the ration 70:30 respectively along with label class being set as stratified. Moreover, to prevent biases towards a particular feature, the dataset was scaled using the standard scaler from sklearn library, which will fit on the train data and transform both the train and test data with the equation:

$$x - scale = \frac{x - mean}{standard deviation} \quad (3.2)$$

Therefore, the above mentioned techniques were applied to pre-process the dataset and make it viable for model implementation.

### 3.5 Dataset Correlation

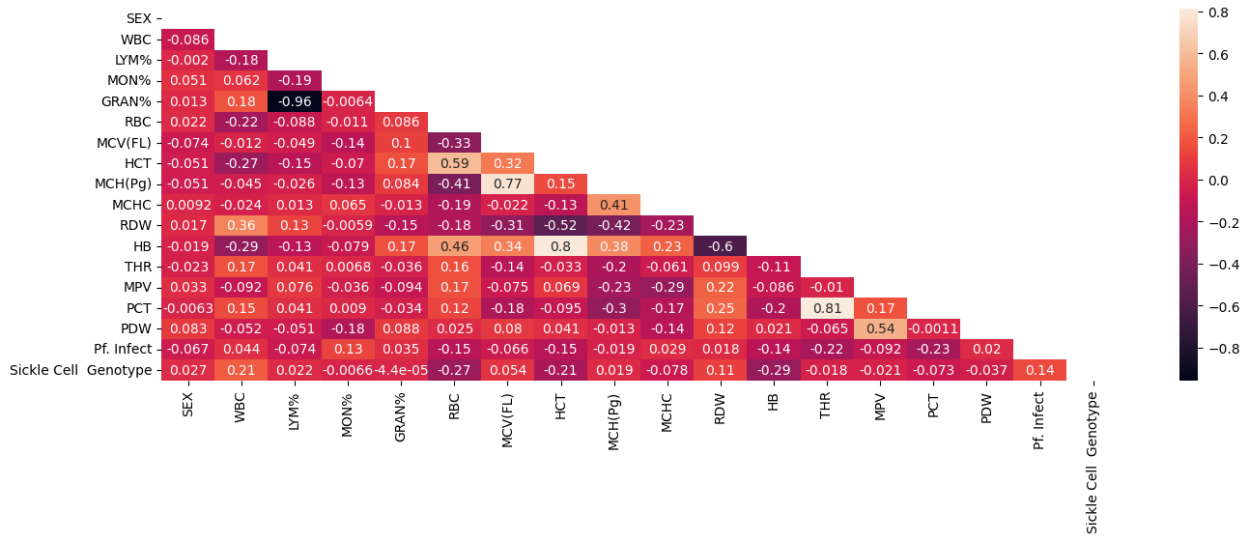


Figure 3.6: Heatmap showing correlation between features of the dataset

The dataset features had medical correlation as discussed above, but to visualize the correlation of the features with the label in the dataset, a heatmap was produced along with the correlation values. Consequently, observing the label Sickle Cell genotype, it has strong correlation with the rest of the features mostly being positively correlated. Furthermore, it can also be observed that the features also have correlation amongst themselves, which will be beneficial for the LSTM model.

# Chapter 4

## Methodology

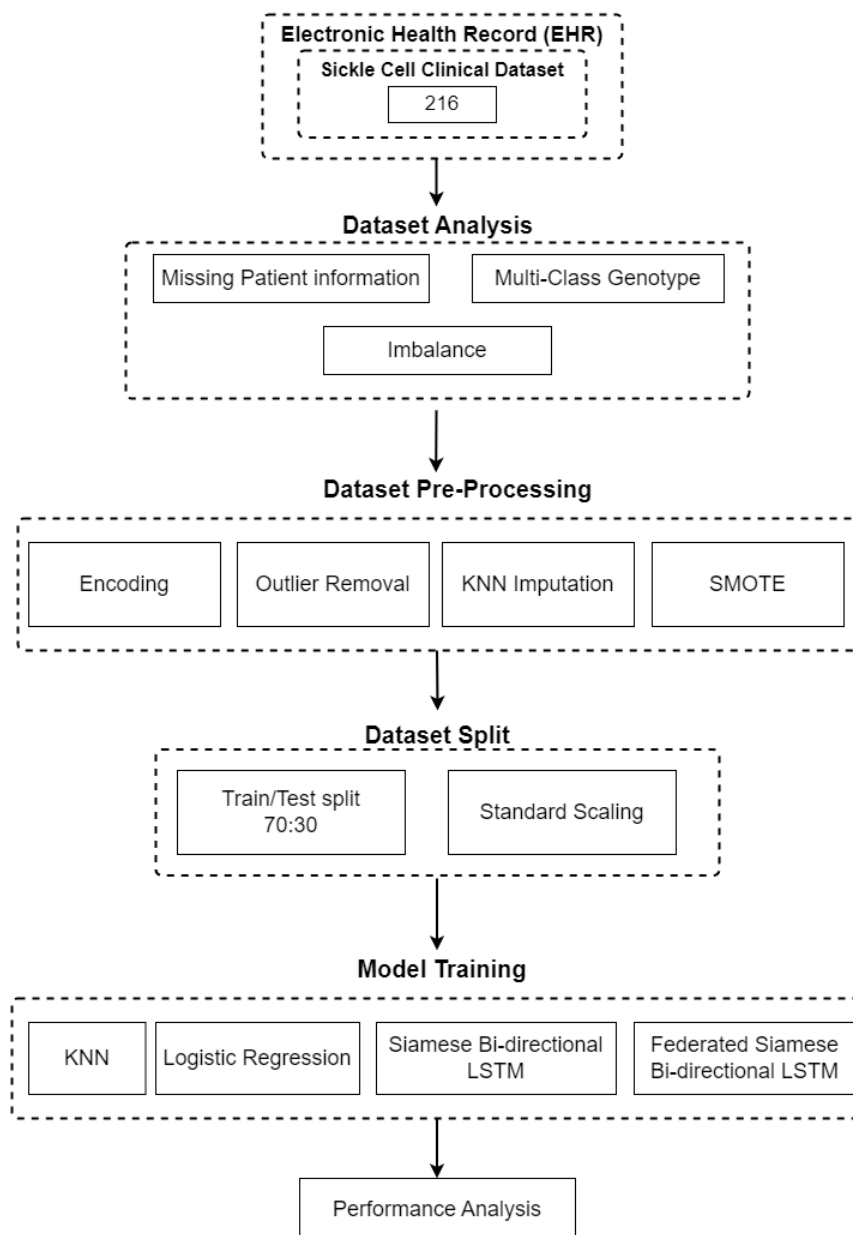


Figure 4.1: Top level overview of proposed FFBS-LSTM

Initially, the research study is established by identifying a problem statement, stating research objectives and reviewing different published papers related to this particular study. In the next step, the dataset is collected from a published paper which contains the clinical data of 216 Sickle Cell patients along with information regarding their genetic mutation type and other clinical features.

Consequently, the dataset required in-depth analysis where correlation between the features is established along with analyzing the null values, gender distribution and the multi-class analysis of genotype label column. Moreover, further analysis regarding outliers and imbalance in genotype class is identified using python. Moreover, the dataset is pre-processed using various techniques, which includes encoding categorical features, outlier removal and KNN imputation using z-score values and using SMOTE to create synthetic data to tackle the issue of imbalance in genotype label classes.

Additionally, the study aims to predict the Sickle Cell genotype from the clinical features using Siamese Bi-LSTM and Federated learning along with two machine learning models KNN and Logistic Regression as the Baseline models to compare. Hence the dataset was split into train and test according to ratio 70:30 and scaled using the standard scaler.

In the next step, the train data will be used to train the two machine learning baseline models KNN and Logistic Regression along with our custom deep learning models Siamese Bi-LSTM and Federated Siamese Bi-LSTM to compare their performance in predicting the Sickle Cell genotype from the clinical features and their ability to adapt in case of limited and decentralized data.

As part of our result analysis, the models' performance will be analyzed based on 4 performance criteria, namely accuracy, precision, recall and f1. Moreover, the confusion matrix is also produced for each case. Therefore, the discussed methodology is followed for this particular study regarding the prediction of genetic mutation using clinical data of Sickle Cell patients.

## 4.1 Models

### 4.1.1 K-Nearest Neighbor

K Nearest Neighbor (KNN) is an algorithm based on supervised learning that can be both used for classification and regression analysis. What KNN tries to do is to correctly classify the data points in the test set by computing the distance between the test data point and train points. Cosine Similarity, Euclidean distance and Manhattan distance are some of the most implemented distance functions. Now in case of classification problems, the KNN algorithm computes the probability of the test data point being in the classes of K training data where K represents all the points which are closest to the test data. Then based on the maximum probability the prediction class will be selected. On the other hand, in case of regression problems the test data will be the mean of the 'K' selected training data points [18].

For instance, the Euclidean distance function can be stated as:

$$D(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2} \quad (4.1)$$

a, b = Euclidean points

n = the dimension n-space

The KNN classifier imported from sklearn for this study had a k value of 5 and the Euclidean distances computed were considered uniform weights for all the points. Moreover, the KNN model has been used due to its simplicity, ability to handle multi class labels and yielding high accuracy in classification problems. Additionally, the KNN algorithm is also used for imputation in this particular study where it calculates the distance from k nearest points to estimate the missing value.

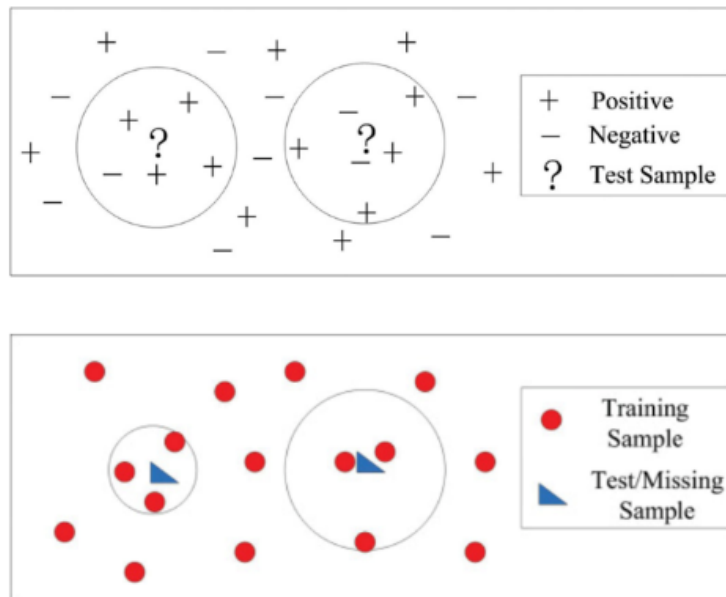


Figure 4.2: KNN visualization in binary classification and imputation [18]

## 4.1.2 Logistic Regression

Logistic Regression is a machine learning technique or algorithm used for both regression and classification tasks, but most of the time it is used for classification problems. The binary response variable belongs to any of the classes. With the aid of dependent variables, it is used to predict categorical variables. Consider that there are two classes, and it is necessary to determine which class a new data point would go under. Then, using algorithms, probability values between 0 and 1 are calculated. For instance, whether or not it will rain today. In logistic regression, the sigmoid curve is the result of passing the weighted sum of the input through the sigmoid activation function. The logistic function, a sigmoid function, has the shape of a "S" and converts any real value to a number between 0 and 1. A sigmoid function's output is classed as either 1 or 0, depending on whether it is greater than or less than 0.5. Y will be expected to be 0 if the graph ends negatively and vice versa.

The Sigmoid function can be established as follows:

$$f(a) = \frac{1}{1 + e^{-a}} \quad (4.2)$$

f(a) = Sigmoid function

a = Real value

Moreover, Logistic Regression is one of the best classification algorithms since our dataset is comparatively small with multiple classes, the algorithm usually prevents overfitting and better accuracy even for small datasets.

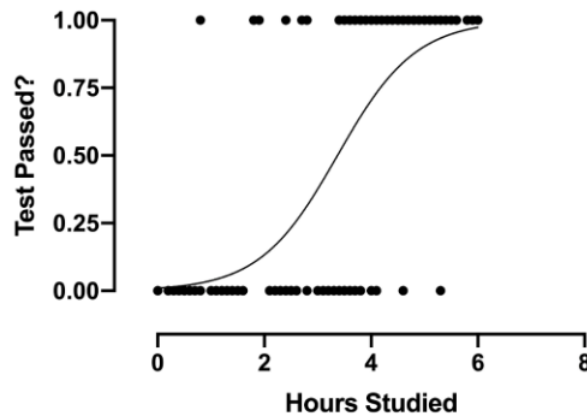


Figure 4.3: Graphical representation of the Sigmoid function [19]



## 4.2 Few Shot Siamese Learning

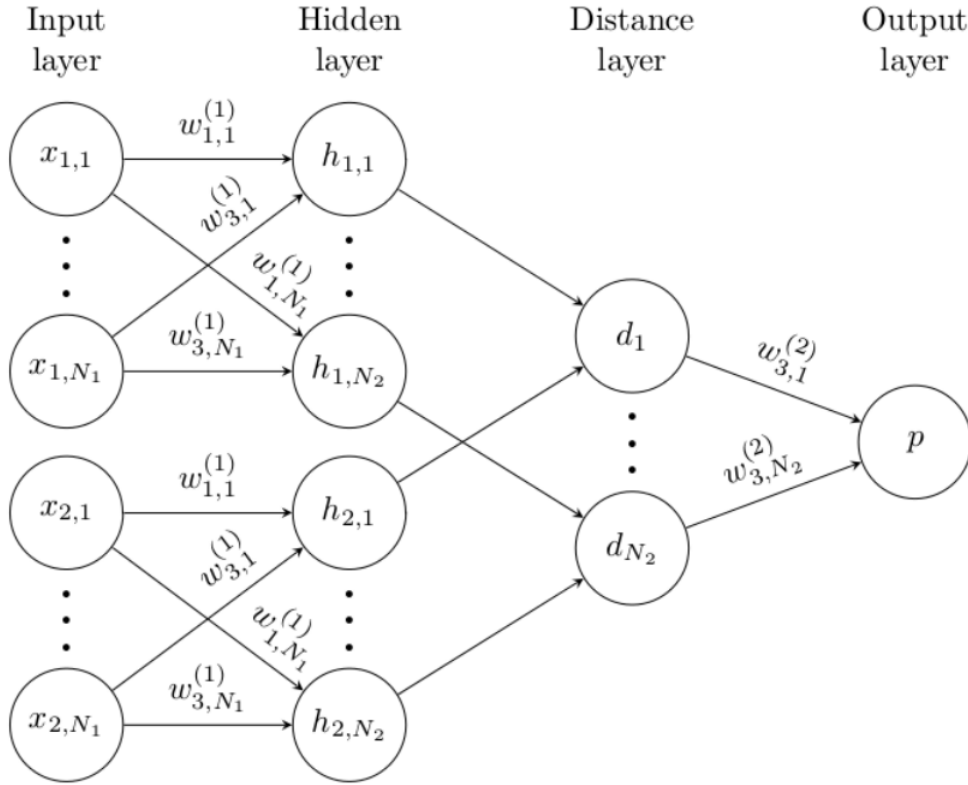


Figure 4.4: Basic Siamese Network [20]

Few Shot Siamese learning is a type of Few shot Learning Technique where models can be trained from a small amount of information in a dataset, typically less than the amount required to train a machine learning or deep learning model properly [21]. For better understanding, in case of traditional supervised learning approach test samples are generally from known classes and are never seen before. However in case of few-shot learning the query samples are from unknown classes. Generally this is used in image classification where the aim is not to let the model recognize images in the training set but to follow a “learn to learn” approach. In simple terms, while training the model for image classification, the model is presented with one or a few images of an object, let’s say different animals as support sets, and tasked to classify any query image (of a particular animal) correctly, which it had been revealed to the model before. The idea here is to find the similarity and difference between the query image and support set images to be able to correctly classify the image. Support sets can be defined as a small set of labeled images. If the support set has  $k$  number of classes where each class has  $n$  samples, it is called  $k$ -way  $n$ -shot support set. While doing few shots learning the, the prediction accuracy depends on the number of ways and shots, i.e  $k$  &  $n$  value. For example if compared between 4-way 1-shot learning and 8-way 1-shot learning, 4-way would have higher accuracy. Even though the concept of few shot learning is more prevalent in the domain of image classification, it can also be implemented for other forms of data such as structured or sentences in NLP.

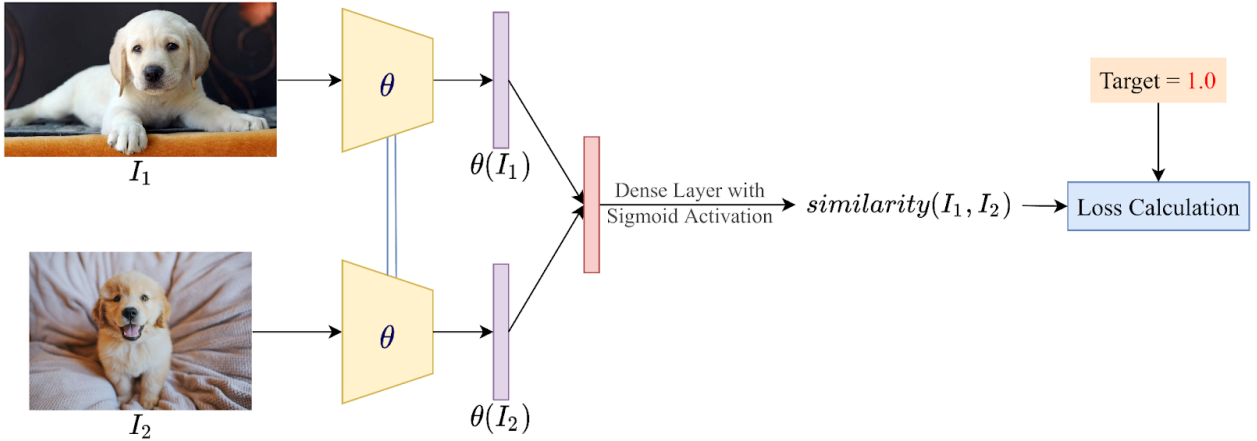


Figure 4.5: Siamese Network Similarity Computation [22]

Siamese Network is a type of neural network which is designed in such a way that it accommodates in learning similarity or dissimilarity between paired inputs. Consisting of two or more identical subnetworks (same architecture and weight), these are connected at the output layer where the similarities are computed from the given pairs for inputs. There are few methods on how the Siamese Network can be trained, one of them is the very basic approach of pairwise similarity. Here feature vectors of the input pairs are passed through a siamese network where similarity scores are calculated and parameters are readjusted to minimize the loss. Similarly score closer to 1 resembles that the input images or instances are similar whereas similarity score closer to 0 resembles that the inputs are dissimilar. Another method is the Triplet loss method, where we randomly choose an image randomly from the support set and name it as “anchor”. Next, we choose another image from the same class we choose the anchor image from and call it a positive sample. After that another image is selected from another class which is identified as a negative sample. The 3 image feature vectors are passed through the Siamese Network, where L2 normalized distance is calculated between the 2 pair (anchor and positive sample, anchor and negative sample, which is in turn helps us to find the loss function and aid in minimization of the loss function, as the model keeps training.

## 4.3 LSTM Architecture

### 4.3.1 Long Short Term memory (LSTM)

Long Short Term Memory (LSTM) Network can be stated as an extension to the architecture of the RNN network to solve the issue of vanishing gradient, where the network can not backpropagate the gradient updates or optimization information to the initial or input layers since it keeps decrementing with each passing layer [23]. Hence, RNNs’ are not capable of capturing long term dependency or complex patterns when the data sequence is long, input has high dimensionality or the input sequence has too many features and as a result, it impacts the performance of the network as whole. On the other hand, the architecture of LSTM makes it capable to learn complex hidden patterns and dependencies within high dimensional data using the concept of cell state, which acts like a long-term memory for the LSTM network. The basic LSTM network architecture can be visualised as follows:

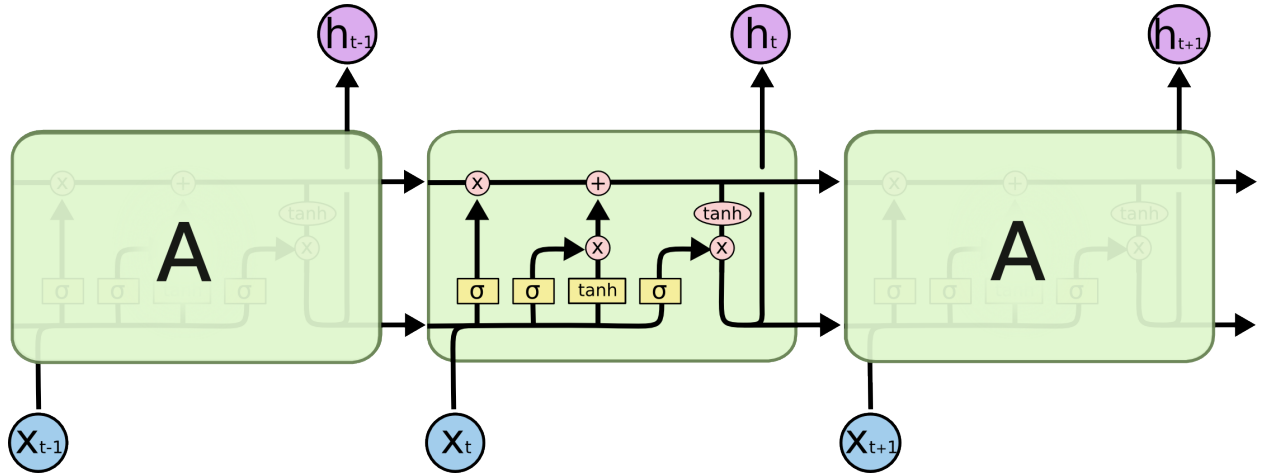


Figure 4.6: Basic LSTM Architecture [24]

In the above figure, the LSTM is spread out in time, since the processing of LSTM is done based on timesteps and the above figure shows 3 timesteps. Moreover, in each timestep, the state of 3 fundamental gates are computed within the LSTM unit, namely Forget gate, Input gate and Output gate. Hence,  $X_t$  represents the input in that particular time step and  $h_t$  is the output, which is this timestep's hidden state that is passed on as a parameter to the next timestep.

Therefore, in order to update the forget gate, which is the decision whether to keep the current cell memory or erase it, the current input multiplied with the relevant weights is summed with the previous timestep hidden state and passed through a sigmoid function, which will convert the value within the range from 0 to 1. Consequently, if the value is 0 the cell memory is erased and it is kept if the value is 1 from the sigmoid function.

In the next step, the state of the input gate is computed, which decides whether the current input is relevant enough to be stored in the cell memory. The computation is done by multiplying the hidden state and current input with their corresponding weights, then passed through a sigmoid function. Subsequently, the weight multiplied value is also passed through a tanh function and then the output of the sigmoid and tanh gates is multiplied to determine the new state of the input gate. Furthermore, the tanh function compresses a value between -1 and 1 with the following formula where  $x$  is the input value:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4.3)$$

Finally, the state of the output gate is determined by passing the current cell memory through tanh function and the current input through sigmoid function and the outputs from tanh and sigmoid gates are multiplied to produce the new state of the output gate, which is also the hidden state for the next time step. Hence, updating the three fundamental gates completes one timestep for the LSTM network and with each timestep the network keeps learning new patterns and dependencies with the input sequence.

Furthermore, the above description explains the architecture of a single unit in LSTM, but in reality several such interconnected units are processed simultaneously to produce an output with the values of all the hidden states from the units. Therefore, LSTM is capable of learning long term dependencies along with complex hidden patterns within the data features or sequences and output the encoded vector with the values of the hidden states and parameters.

### 4.3.2 Bidirectional LSTM

The Bidirectional LSTM is the extension of the LSTM network architect where the same is input sequence is processed by the network from the forward direction and backward direction simultaneously to produce two sets of hidden state representation, which is then concatenated to produce the final output vector with the hidden states values [25]. Furthermore, one of the LSTM networks in the Bidirectional LSTM will process the input starting from the initial time step and move forward, but the backward LSTM network will process the input starting from the last time step and move towards the first time step updating its parameters learning from both past and future timesteps simultaneously. As a result, the output hidden state vector dimension will be doubled from that of the LSTM network, which is dependent on the number of units present in the network.

Hence, the fundamental advantage of implementing the bidirectional LSTM involves the network learning and updating its parameters with a deeper understanding of the complex relationships between the features in the input sequence which the unidirectional LSTM might have missed. However, such networks can lead to cases of overfitting and might require proper hyperparameter tuning to produce optimal results.

### 4.3.3 Lambda Layer (L1-Distance Computation)

The lambda layer from the keras library provides the opportunity to build layers for our model with self-coded expressions and formulas, which can be added to the built model architecture for further ease and compatibility. Hence, in this case the lambda layer is used to compute the element-wise L1-distance or the Manhattan distance between the two different encoded hidden state representation output from the Bidirectional LSTM network. Therefore, the L1-distance formula can be established as:

$$L1 - Distance = |x_1 - x_2| \tag{4.4}$$

Here,

L1-Distance = the element-wise absolute difference between  $x_1$  and  $x_2$

$x_1$  = hidden states represented as vector

$x_2$  = hidden states represented as vector

### 4.3.4 Dense Layer

Dense layer in context of LSTM or any neural network denotes that the units in this dense layer will be connected with all the output neurons from the previous layer. Hence, it is called the dense layer due to its deep connection with the previous layer and the purpose of this particular layer is to provide the final output in the dimensions required by multiplying the input with learned weights and adding a bias term before passing it to the activation function. Similar to other layers the dense layer has units and an activation function. For instance, if the output is binary then the dense layer will have 1 unit with sigmoid as the activation function since the output will be within 0 and 1.

### 4.3.5 Sigmoid Activation Function

The sigmoid activation function translates any input value within the range 0 and 1. Hence, the sigmoid function is beneficial in cases where the output will be similarity score between 0 and 1 or the output will be the probability of a certain outcome or event. Moreover, the sigmoid is used within the LSTM unit to compute and update the states of the forget, input and output gates. The formula used to perform the sigmoid operation can be established as:

$$\sigma = \frac{1}{(1 + e^{-x})} \quad (4.5)$$

Here,

$\sigma$  = Sigmoid output within [0,1]

x = input to the function

### 4.3.6 Adam optimizer

In deep learning models, optimizers play a vital role in enhancing model's performance by reducing loss. Adam optimizer is acquired from adaptive moment estimation because the algorithm is designed in such a way that adam optimizer utilizes approximation of first and second moments of gradient to adjust the learning rate for different variables of the network [26]. Moreover, adam optimizer function is developed by incorporating the concepts of AdaGrad and RMS prop algorithm which are extensions of Stochastic Gradient Descent(SGD). The formula for bias correction and update in Adam optimizer are [26] :

Bias correction formula for 1st moment:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (4.6)$$

Here,

$\hat{m}_t$  = 1st moment

$\beta_1^t$  =hyperparameter

Bias correction formula for 2nd moment:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (4.7)$$

Here,

$\hat{v}_t$  = 2nd moment

$\beta_1^t$  =hyperparameter

Updating weight:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (4.8)$$

Here,

$\theta_t$  = weight

$\alpha$ =step size

$\epsilon$ = mitigate zero division

### 4.3.7 Loss Function

Loss function can be used to understand the performance of a model. If the model can predict outcome properly or approximately to actual results the loss value will be smaller. On the other hand, if the model performs badly in predicting then the loss value will be higher. As our Siamese model will determine the similarity and dissimilarity between two rows Binary Cross Entropy loss function is used in which 0 denotes dissimilar and 1 denotes similar. Hence, the loss is determined by calculating the deviation between the actual value and predicted outcome.

## 4.4 Federated Learning

Federated learning is a machine learning method or technique that allows several clients to collectively train a model based on its own dataset while maintaining the privacy of their respective user data [27]. It uses the approach of dividing the data among clients to handle privacy issues while maintaining the ability to build precise models. According to Martineau, federated learning provides a technique that allows us to develop new applications by training different models without the need of centralizing the data in one place [28].

The fundamental objective of Federated learning is that data can be stored on individual users' devices and yet be used to train machine learning or deep learning models, instead of using the traditional method of using centralized dataset [27]. For instance, Given that protected health information cannot be exchanged as freely as other businesses due to HIPAA and other laws, the healthcare sector is one of those that can gain the most from federated learning. Thus, this strategy is especially significant in situations when extremely sensitive data, like patient records cannot be shared or held centrally. Hence, there is no issue regarding data privacy. In this manner, the development of AI models can benefit from vast amounts of heterogeneous data from various healthcare databases and devices while being compliant with laws [29].

The basic concept of Federated Learning is, the model training process or the dataset must be divided into a few smaller sub-tasks, each of which is given to a different user device. Next, the outcomes of these sub-tasks are then sent back to the central server. Additionally, these sub-tasks are then forwarded to the devices, where they are executed on the local data. The server then aggregates or averages the weights

and parameters from each device in order to update the global model. Moreover, there are three variations of this decentralized training approach, where the primary model in horizontal federated learning is trained on comparable datasets, second the data are complementary in vertical federated learning; movie and book evaluations, for instance, are integrated to predict a person’s musical tastes. Finally, federated transfer learning involves training a model that has already been trained to accomplish one task, such as detecting automobiles, on a different dataset to perform another task, such as, recognizing cats [28].

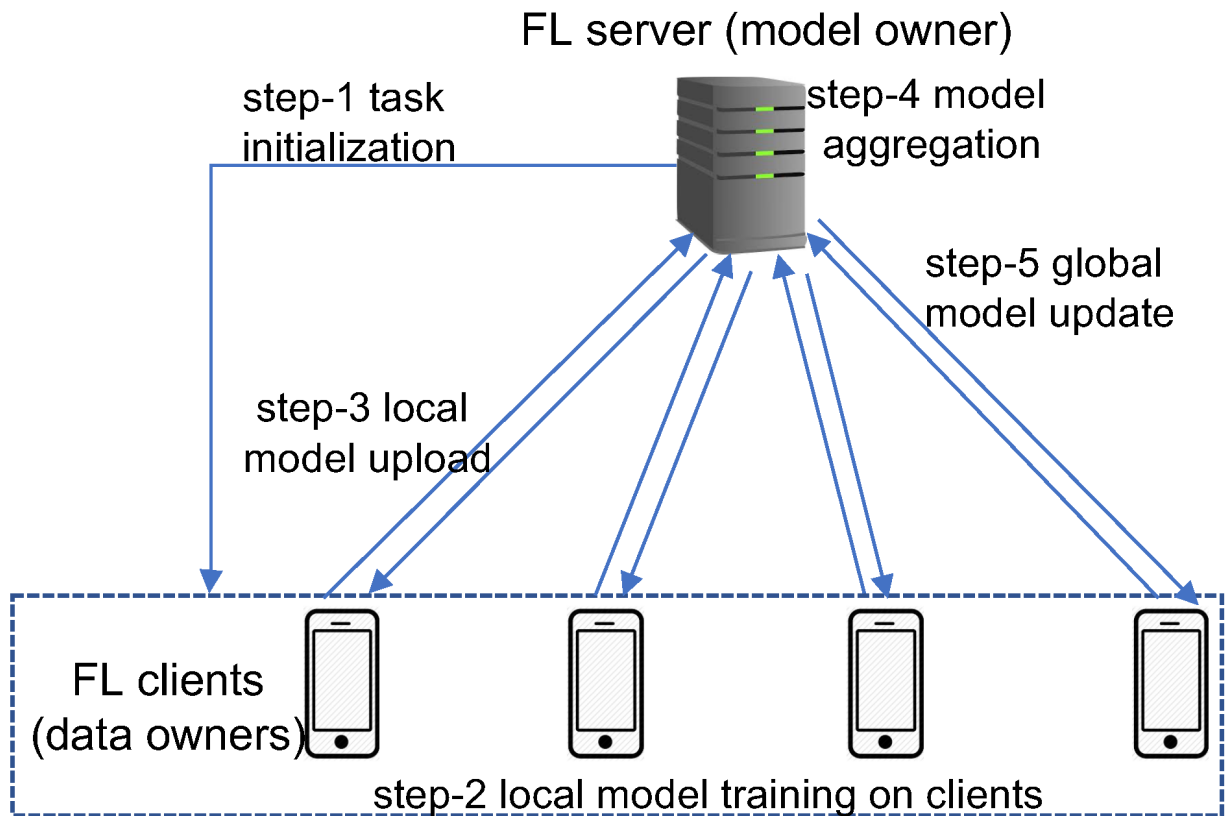


Figure 4.7: Federated Learning Architecture [30]

The advantages of Federated Learning can be established as:

1. **Increased Accuracy:** The model’s accuracy is increased by using training data from several devices to create a more diverse and representative dataset
2. **Decreased Computational Power:** The heavy load on the central server is reduced by dividing the data and training to client devices, which in order makes the overall training effective by spreading the computation to local devices.
3. **Maintain Privacy:** Data is still stored locally, decreasing the possibility of data breaches and hampering user privacy. Users can opt to participate in the model training process and have control over their data.
4. **Data Security:** This federated learning technique uses secure communication methods and proper encryption to guarantee the privacy and integrity of the data by keeping the data in local devices only.

5. **Relevance in Real-World:** The decentralized data used in federated learning more accurately depicts actual environments, improving the model’s effectiveness in real-world applications.
6. **Collaborative Training:** Federated learning effectively uses collaborative model training efforts between various clients without hampering the privacy of their data.
7. **Scalability:** Federated learning can be used in situations where there are a large number of user devices since it allows for the wide-scale training of models without the need for centralized data storage.

These benefits make federated learning an efficient approach for machine learning in situations where maintaining data privacy is significant and contribute to its growing popularity in various industries research domains. Federated learning is a crucial method for furthering machine learning in a privacy-conscious world despite these difficulties. Natural language processing, picture identification, and health care are just a few of the uses it has already seen. Federated learning will probably gain popularity as a machine learning approach as more organizations begin to understand the value of data privacy.

## 4.5 FFSB-LSTM Working Principle

### 4.5.1 Training

Initially, the train data after split is used to create pairs of similar and dissimilar instances based on the genotype class label to train the Siamese Bi-directional LSTM network to compute the similarity between two instances from the dataset. For each of the genotype classes, an equal number of similar pairs with target 1.0 and dissimilar pairs with target 0.0 are created. Since, for each class  $nC2$  Similar pairs can be created, where  $n$  is the number of instances in one class label, the pairs dataset is comparatively larger than the original dataset and adequate to train an LSTM model. Therefore, the pairs dataset would have the following format, where  $f$  represents a feature value and target is the similarity score:

Table 4.1: Pairs Dataset to Train Bi-LSTM

patient_input1	patient_input2	target
[f1,f2,f3,f4,.....,fk]	[f1,f2,f3,f4,.....,fk]	1.0
[f1,f2,f3,f4,.....,fk]	[f1,f2,f3,f4,.....,fk]	0.0
[f1,f2,f3,f4,.....,fk]	[f1,f2,f3,f4,.....,fk]	1.0
...	...	...
...	...	...
[f1,f2,f3,f4,.....,fk]	[f1,f2,f3,f4,.....,fk]	0.0



Subsequently, our proposed FFSB-LSTM model would have the following fundamental architecture or work flow:

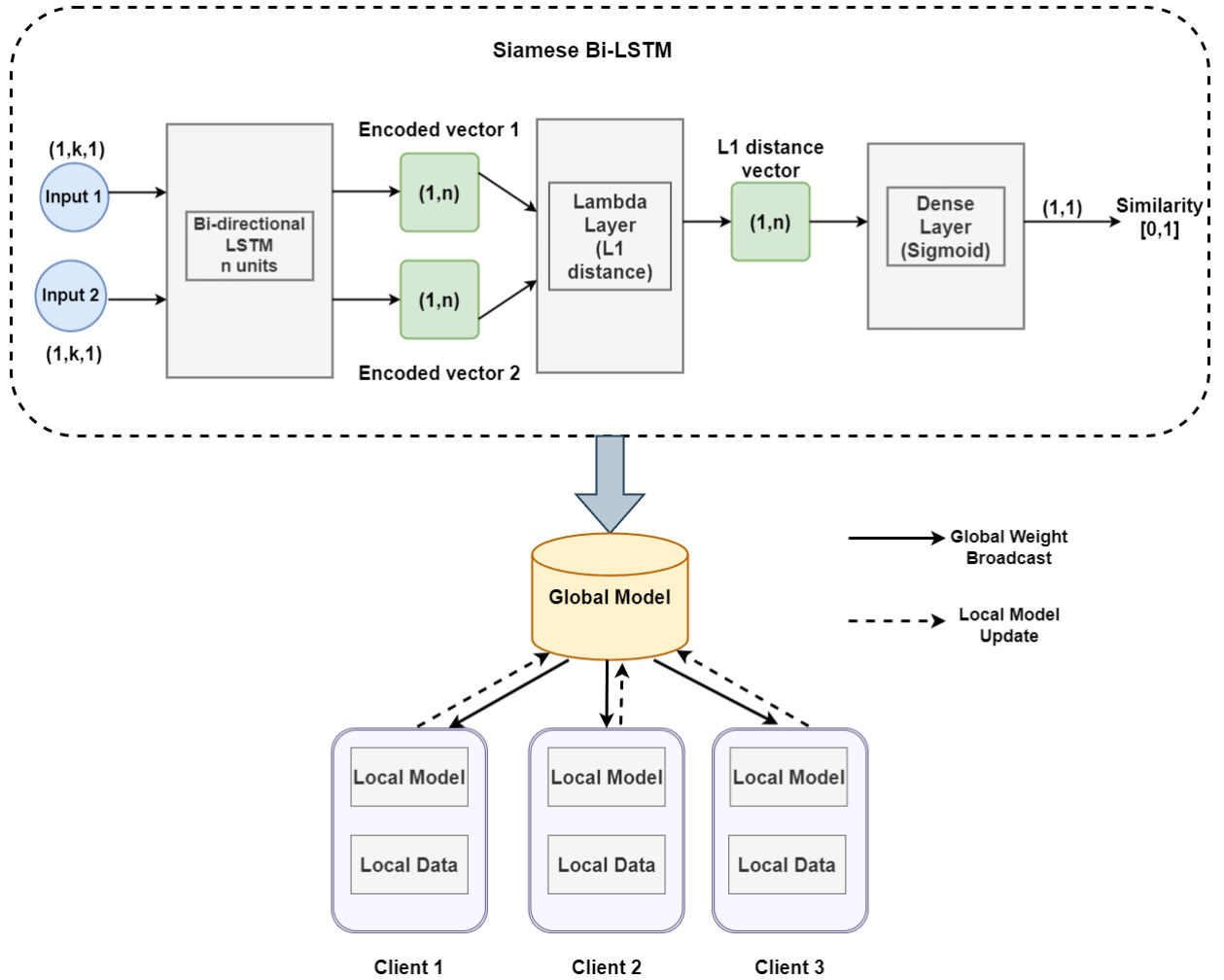


Figure 4.8: FFSB-LSTM Model Architecture

Initially, the input sequence with  $k$  clinical features is reshaped to the format of a 3D-tensor with shape (batch-size, time-steps, sequence-length). Hence, the shape of each input sequence becomes  $(1,k,1)$ , with each feature representing one time-step and as a result the Bi-LSTM will process each feature in a single time-step learning complex patterns between the features. Each pair from `patient_input1` and `patient_input2` is fed in the Bi-LSTM separately as two inputs. Thus, for each input the Bi-LSTM will run  $k$  timesteps and produce an output of shape  $(1,n)$ , with  $n$  hidden state representations based on its learning of the input sequence, named Encoded Vector since the model will have  $n$ -units. The two encoded vectors from the two inputs will then be passed to a lambda layer, which will compute the element-wise difference between the vectors and output the L1-Distance vector with shape  $(1,n)$ . Consequently, the distance vector is passed to the last Dense layer, which will first multiply the vector with its learnable weights of shape  $(n,1)$  and add a bias term to the scalar value of shape  $(1,1)$ . This value is then passed through the sigmoid function to translate it within the range  $[0, 1]$ , which represents the similarity score between the two inputs.

Hence, after the first iteration the model will compute the Binary Crossentropy loss by comparing with the true similarity values given during training and based on the loss value it will optimize the internal parameters and weights during backpropagation using the Adam optimizer which uses stochastic gradient descent. Thus, the model will then run for several epochs on the whole training pairs dataset, updating its parameters after each batch of train data.

Additionally, in the case of Federated Siamese BI-LSTM, the same model is wrapped as the global model with little changes to hyperparameters such as units or batch-size and sent to the client devices. Moreover, the train data is randomly and equally divided among the clients, with each receiving about equal instances of data from the original dataset. Next, the global FFSB-LSTM model is sent to the client devices and each client will then create its own pairs dataset and train the custom Siamese BI-LSTM model for several epochs with computing loss function to optimize using Adam optimizer and update the global model by averaging the parameters from the clients after each round. Additionally, the global model will send back the updated global model weights to the client models before each round. As a result, the global model will only receive updates on internal parameters and weights but the training on data will occur at client level.

## 4.5.2 Testing

The Siamese Bi-LSTM model built can compute similarity between two instances of data, but the prediction will be classification of the genotype label with unique classes. In order to predict the genotype, first a support set is created, which in this case will be the train data. The support set is the data with the genotype column, with which the model will compare the test data to to predict the genotype class.

Hence, a test row with  $k$  features and unknown genotype is sent to the trained Siamese Bi-LSTM model, which will compute the similarity score between the test row and all the other rows in the support set based on its learning of the complex hidden relationships within the features. The top 10 rows or instances in the support set with most similarity with the test instance are selected and a voting method is applied to these top 10 rows, where the genotype class with most votes is selected from the support set as the prediction for the given test instance.

Furthermore, Sickle Cell genotype prediction is done for all the instances in the test data and then the true genotype labels for the corresponding test data along with the predictions from the model is used to compute the performance metrics accuracy, precision, recall and f1-score.

# Chapter 5

## Implementation and Result Analysis

### 5.1 Proposed Model Specifications

#### 5.1.1 Siamese Bi-directional LSTM

The Siamese Bi-directional LSTM network is built using python and the tensorflow keras library according to the specifications required for our model. The Siamese Bi-directional LSTM built for our study will be trained to compute the similarity between two instances of the Sickle Cell dataset for prediction of the genotype. Thus, the model had multiple layers and the hyperparameters were tuned and set as per our requirements as follows:

1. A Bi-directional LSTM layer with 150 units along with dropout and recurrent dropout set to 0.2 to prevent overfitting
2. A lambda layer to compute element-wise difference between the two outputs of LSTM layer
3. Final Dense layer with 1 unit and sigmoid activation
4. Adam optimizer to update the internal parameters and weights during back-propagation
5. Binary Crossentropy loss function to compute the loss of the model during training
6. Batch size will be 32
7. Training the model for 30 epochs

The dataset initially had 216 instances, which was then pre-processed and SMOTE was applied to remove genotype class imbalance from the dataset. Hence, after SMOTE the dataset had equally divided instances among the 4 classes and was further split into train and test based on the ratio 70:30. Consequently, the train data with 414 instances is then used to create the pairs dataset with 16000 similar and dissimilar pairs and the dataset will be used to train the Siamese Bi-LSTM with each input shape being (1,17,1) since there are 17 features. Moreover, output of the

Bi-LSTM layer will be of shape (1,300) vector of hidden representations since it has 150 units in both forward and backward direction. Hence, the model will be trained for 30 epochs with batches of 32, while being optimized using the Adam optimizer based on the Binary Crossentropy loss function value, according to the architecture discussed in the model working principle section.

Hence, the model summary can be established as:

```

Model: "model"
-----
Layer (type)                Output Shape          Param #          Connected to
-----
input_1 (InputLayer)        [(None, 17, 1)]      0                []
input_2 (InputLayer)        [(None, 17, 1)]      0                []
bidirectional (Bidirectional) (None, 300)          182400           ['input_1[0][0]',
                    'input_2[0][0]']
lambda (Lambda)             (None, 300)          0                ['bidirectional[0][0]',
                    'bidirectional[1][0]']
dense (Dense)               (None, 1)            301              ['lambda[0][0]']
-----
Total params: 182,701
Trainable params: 182,701
Non-trainable params: 0

```

Figure 5.1: Siamese Bi-directional LSTM model summary

### 5.1.2 Federated Few-Shot Siamese Bi-directional LSTM

The Siamese Bi-directional LSTM is then also implemented along with the concept of federated learning to tackle the issue of both limited and decentralised data. Furthermore, the tensorflow federated framework was used to implement federated learning in our study. Therefore, the Bi-directional LSTM network had the following specifications in case of Federated Siamese Bi-directional LSTM:

1. A Bi-directional LSTM layer with 200 units
2. A lambda layer to compute element-wise difference between the two outputs of LSTM layer
3. Final Dense layer with 1 unit and sigmoid activation
4. Adam optimizer with learning rate 0.01 to update the internal parameters and weights during training client devices
5. Binary Crossentropy loss function to compute the loss of the model during training

Moreover, the specifications set for the federated learning aspect are as follows:

1. Train data will be split between 3 clients randomly
2. Batch size will be 64

3. Training each client model for 30 epochs
4. Train the model for 40 rounds with each round as one step for global model

Hence, the train data with 414 instances is equally divided among the 3 clients, with each receiving 138 instances randomly. The global model Siamese Bi-LSTM is initialized and the initial model weights are broadcasted to the clients. The clients will create pairs dataset from their own respective data in order to train their local models for 30 epochs with batch size of 64. Furthermore, the input shape will be (1,17,1) representing a single time-step for each feature, but the output of Bi-LSTM layer will be (1,400) since the layer has 200 units in this case. Moreover, the data flow to the next lambda and dense layer is discussed in the model working principle section. Additionally, Adam optimizer with a slightly larger learning rate of 0.01 is used to optimize the local models based on the loss function since each client has a smaller number of instances in this case.

Thus, after training the local model for 30 epochs in one round, the clients will send back the updated model weights to the global model, which will average the weights and broadcast again the updated global weights for the next round. Therefore, the above specifications were set for the training of the FFSB-LSTM model and then analyze the performance based on the test data evaluation.

## 5.2 Performance Metrics

### 5.2.1 Confusion Matrix

Confusion Matrix is used to determine the performance of the classifier [31]. The confusion matrix is a NxN matrix and gives the summary in terms of predicted values and actual values. In binary classification, the matrix is of 2x2 and for multi class classification, the row and column depends on the number of class labels. As a result, the performance metrics can be calculated from the confusion matrix. The column represents the predictions of the model and the rows represent the actual value.

		PREDICTED classification			
		Classes	a	b	c
ACTUAL classification	a	TN	FP	TN	TN
	b	FN	TP	FN	FN
	c	TN	FP	TN	TN
	d	TN	FP	TN	TN

Figure 5.2: Confusion Matrix for Multi-Class Classification [32]

Here,  
TP: True Positive  
TN: True Negative  
FP: False Positive  
FN: False Negative

### 5.2.2 Accuracy

Accuracy can be established as the performance metrics to compute the performance of the algorithm. It represents the ratio of true predictions made out of the total predictions done and visualized in percentage.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (5.1)$$

### 5.2.3 Precision

Precision is another performance indicator of a model and shows the percentage of true predictions predicted by the model. It is the proportion of true positives and total number of positive outcomes predicted. In case of multi class classification, the term 'weighted' is assigned to average. Thus, precision is calculated for each label separately and then the weighted average is computed based on the number of true rows for each label

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (5.2)$$

### 5.2.4 Recall

Recall is the ability of a model to predict positive outcomes. It is the ratio between positive predicted outcomes and all actual positive classes. In case of multi class classification, the term 'weighted' is assigned to average. Thus, recall is calculated for each label separately and then the weighted average is computed based on the number of true rows for each label

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5.3)$$

### 5.2.5 F1 Score

F1 score is the harmonic mean of precision and recall. As it is calculated by the harmonic mean, the value depends on the precision and recall score of the model and also helps to compare models.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (5.4)$$

## 5.3 Result Analysis

The test data is used to analyze the Siamese Bi-directional LSTM and Federated Siamese Bi-directional LSTM based on the bagging method discussed above, along with the two baseline models KNN and Logistic Regression, using the performance metrics accuracy, precision, recall and f1 score. Since, the test data was from the main Sickle Cell dataset after resampling, it will consist of 178 instances.

In case of Sickle Cell Genotype, the classes are: {'Homozygous HbAA/HbAA':1, 'Heterozygous HbAA/HbSS':0, 'Homozygous HbSS/HbSS':3, 'Homozygous HbAA/HbSS ':2}

### 5.3.1 K-Nearest Neighbour

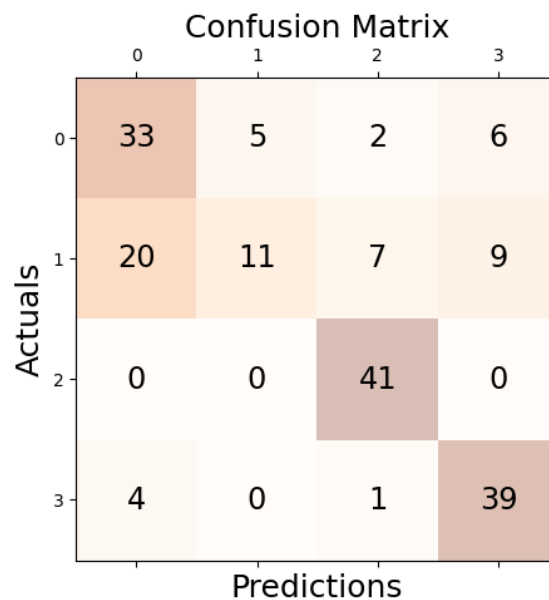


Figure 5.3: KNN Confusion matrix

The above figure depicts the number of instances successfully and unsuccessfully classified by KNN, which shows a total of 124 out of 178 instances being classified properly. Furthermore, the model could not predict the class labels 0 and 1 properly. Moreover, the accuracy along with precision, recall and f1 score for the model while predicting genetic mutation was 69.66%, 69.48%, 69.66% and 65.98% respectively.

### 5.3.2 Logistic Regression

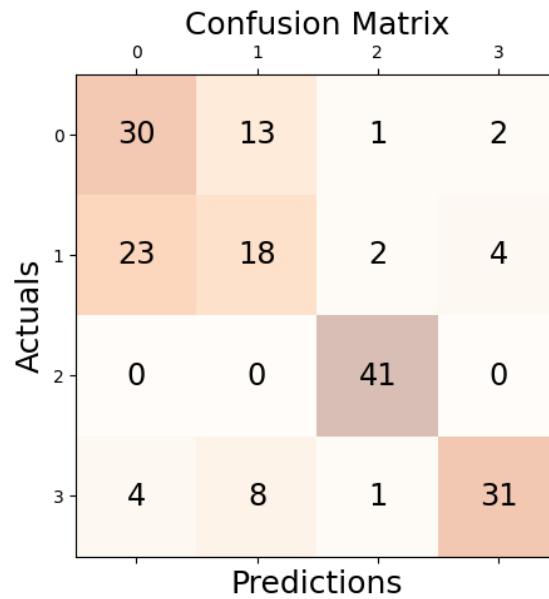


Figure 5.4: Logistic Regression Confusion matrix

The accuracy along with precision, recall and f1 score for the model Logistic Regression while predicting genetic mutation was 67.42%, 67.48%, 67.42% and 66.99% respectively. Consequently, the confusion matrix reveals LR successfully predicted 120 out of 178 instances. Additionally, the LR model had difficulty predicting 0, 1 and 3 genotype class labels.

### 5.3.3 Siamese Bi-Directional LSTM

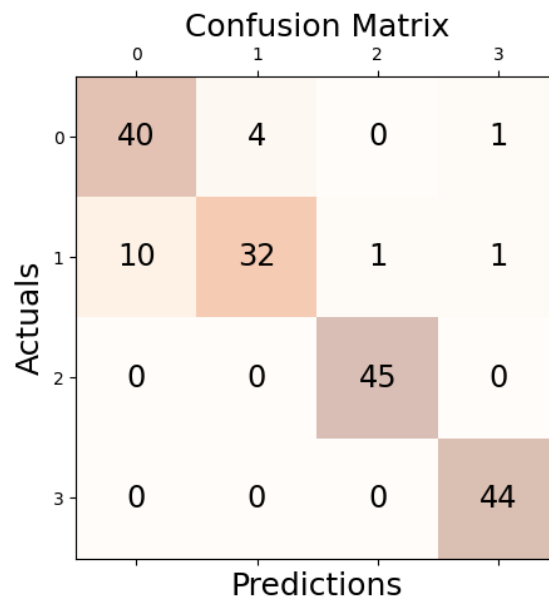


Figure 5.5: Siamese Bi-LSTM Confusion matrix



The accuracy along with precision, recall and f1 score for the model Siamese Bi-LSTM while predicting Sickle Cell genetic mutation was 90.45%, 91.36%, 90.45% and 90.66% respectively. Additionally, the Siamese Bi-LSTM model had great success in predicting all 4 genotype class labels to achieve great scores. Consequently, the confusion matrix reveals Siamese Bi-LSTM successfully predicted 161 out of 178 instances.

### 5.3.4 Federated Siamese Bi-Directional LSTM

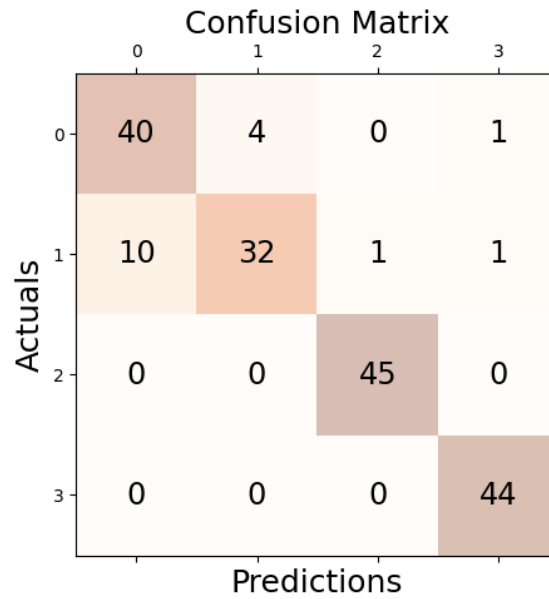


Figure 5.6: FFSB-LSTM Confusion matrix

The above figure reveals the number of instances successfully and unsuccessfully classified by FFSB-LSTM, which shows a total of 157 out of 178 instances being classified properly. Furthermore, the model could predict all the class labels to a great extent compared to the machine learning models. Moreover, the accuracy along with precision, recall and f1 score for the Federated Siamese Bi-LSTM while predicting genetic mutation was 88.20%, 89.82%, 88.20% and 88.57% respectively.

## 5.4 Overall Performance Comparison

Table 5.1: Performance comparison between models

Models	Accuracy	Precision	Recall	F1
KNN	69.66%	69.48%	69.66%	65.98%
Logistic regression	67.42%	67.48%	67.42%	66.99%
Siamese Bi-LSTM	90.45%	91.36%	90.45%	90.66%
Federated Siamese Bi-LSTM	88.2%	89.82%	88.2%	88.57%

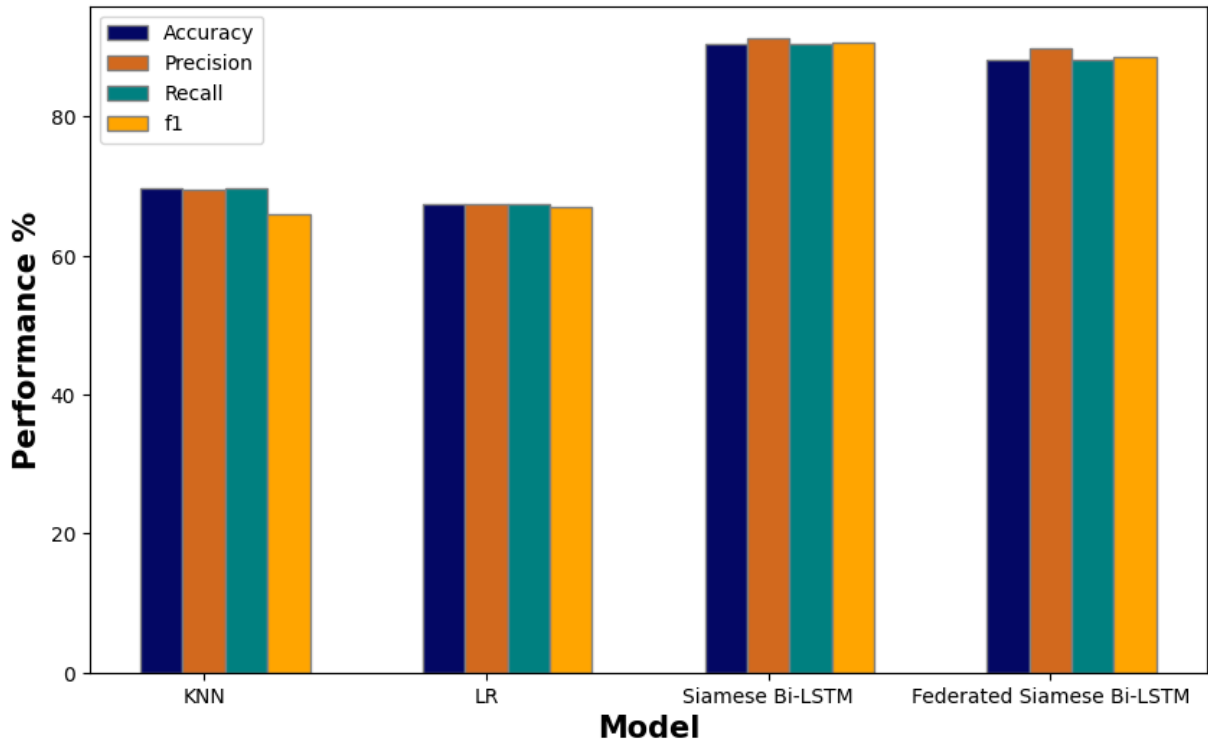


Figure 5.7: Performance comparison Bar-Chart

Based on table 5.1. and fig 5.7.,the performance of the baseline machine learning models KNN and Logistic Regression can be stated as moderate since they have accuracy and f1 score in the range 65% to 70%. Consequently, it proves that limited data, even after SMOTE resampling, was not adequate to train the machine learning models even though the algorithm of these models usually make them efficient in classification tasks.

On the contrary, the proposed FFSB-LSTM along with Siamese Bi-LSTM model in our study performed substantially better compared to the baseline models when predicting the Sickle Genotype even from limited and decentralised data. The Siamese Bi-LSTM achieved an accuracy of 90.45% and f1 score of 90.66%, which is slightly better than the Federated Siamese Bi-LSTM with accuracy of 88.20% and f1 score of 88.57%. Moreover, the performance scores reveal that the Siamese Bi-LSTM implementation is effective at learning underlying complex patterns and dependencies even from structured non-sequential dataset while computing the similarity scores. Therefore, the performance analysis asserts that our proposed Federated Siamese Bi-LSTM model was successful to a great extent in predicting Sickle Cell genotype from a limited and decentralised dataset.

# Chapter 6

## Conclusion

Sickle Cell is a chronic genetic disorder which progresses with time and becomes life-threatening as it gradually affects other organs such as liver, lung, kidneys etc. Furthermore, the progression of Sickle Cell can be predicted to a great extent based on the type of genetic mutation of a patient in the HBB gene, but the medical test available for these factors often pose multiple restrictions and limitations. Moreover, the issue of limited data and patient clinical data privacy-security concerns motivated us to implement a Federated Siamese Bi-directional LSTM. The Siamese Bi-LSTM was trained on data pairs to compute the similarity between instances, while the data was divided among clients in the case of federated learning, to develop a supervised model for the prediction of Sickle Cell genotype. Hence, the study used a Sickle Cell clinical dataset of 216 children in Africa with 4 different genotype classes to evaluate the performance of the developed Federated Siamese Bi-LSTM (FFSB-LSTM) model. The performance analysis reveals that baseline models like KNN had a max accuracy and f1 score of 69.66% and 65.98%. However, our Federated Siamese Bi-LSTM model showed substantial improvement at predicting the genotype by achieving an accuracy of 88.20% and f1 score of 88.57%. Additionally, the Siamese Bi-LSTM model without federated learning could achieve 90.45% test accuracy by training on the Sickle Cell dataset. Furthermore, the results assert that the Bi-LSTM model was successful in capturing the hidden patterns and relationships within the clinical features for non-sequential tabular data while computing similarity. Therefore, our research built a Federated Siamese Bi-LSTM to predict the genetic mutation of Sickle Cell patients from clinical data.

Therefore, our research study is limited to the implementation of only one kind of Few-Shot technique and so the future work based on our current model Federated Siamese Bi-directional LSTM to predict the Sickle cell genotype from limited and decentralised clinical data, could be the implementation of a different Few-shot algorithm such as Model Agnostic Meta Learning(MAML) or Relational Network, instead of a Siamese network. Moreover, the concept of Heterogeneous Federated Learning can also be introduced, where the data among the clients have different structure and features or might have model heterogeneity. Furthermore, custom optimizers, loss functions or layers can be built for the LSTM model to improve the accuracy and f1 scores further.

# Bibliography

- [1] G. J. Kato, F. B. Piel, C. D. Reid, M. H. Gaston, K. Ohene-Frempong, L. Krishnamurti, W. R. Smith, J. A. Panepinto, D. J. Weatherall, F. F. Costa, and et al., “Sickle cell disease,” *Nature Reviews Disease Primers*, vol. 4, no. 1, 2018. DOI: 10.1038/nrdp.2018.10.
- [2] F. B. Piel, M. H. Steinberg, and D. C. Rees, “Sickle cell disease,” *New England Journal of Medicine*, vol. 376, no. 16, pp. 1561–1573, 2017, PMID: 28423290. DOI: 10.1056/NEJMra1510865. eprint: <https://doi.org/10.1056/NEJMra1510865>. [Online]. Available: <https://doi.org/10.1056/NEJMra1510865>.
- [3] G. R. Serjeant, “The natural history of sickle cell disease,” *Cold Spring Harbor Perspectives in Medicine*, vol. 3, no. 10, 2013. DOI: 10.1101/cshperspect.a011783.
- [4] M. Bertolotti, “Opportunities, risks, and limitations of genetic testing: Looking to the future from patients’ point of view,” *Mayo Clinic Proceedings*, vol. 90, no. 10, pp. 1311–1313, 2015. DOI: 10.1016/j.mayocp.2015.08.015.
- [5] X. Wang, C. Zou, Y. Zhang, X. Li, C. Wang, F. Ke, J. Chen, W. Wang, D. Wang, X. Xu, L. Xie, and Y. Zhang, “Prediction of brca gene mutation in breast cancer based on deep learning and histopathology images,” *Frontiers in Genetics*, vol. 12, 2021, ISSN: 1664-8021. DOI: 10.3389/fgene.2021.661109. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fgene.2021.661109>.
- [6] M. Chen, B. Zhang, W. Topatana, J. Cao, H. Zhu, S. Juengpanich, Q. Mao, H. Yu, and X. Cai, “Classification and mutation prediction based on histopathology h&e images in liver cancer using deep learning,” *npj Precision Oncology*, vol. 4, no. 1, 2020. DOI: 10.1038/s41698-020-0120-3.
- [7] D. C. Rees, T. N. Williams, and M. T. Gladwin, “Sickle-cell disease,” *The Lancet*, vol. 376, no. 9757, pp. 2018–2031, 2010, ISSN: 0140-6736. DOI: [https://doi.org/10.1016/S0140-6736\(10\)61029-X](https://doi.org/10.1016/S0140-6736(10)61029-X). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S014067361061029X>.
- [8] S. L. Saraf, R. E. Molokie, M. Nouraie, C. A. Sable, L. Luchtman-Jones, G. J. Ensing, A. D. Campbell, S. R. Rana, X. M. Niu, R. F. Machado, and et al., “Differences in the clinical and genotypic presentation of sickle cell disease around the world,” *Paediatric Respiratory Reviews*, vol. 15, no. 1, pp. 4–12, 2014. DOI: 10.1016/j.prrv.2013.11.003.
- [9] R. Collier, “The downside of genetic screening,” *Canadian Medical Association Journal*, vol. 184, no. 8, pp. 862–864, 2012. DOI: 10.1503/cmaj.109-4169.

- [10] N. Coudray, P. S. Ocampo, T. Sakellaropoulos, N. Narula, M. Snuderl, D. Fenyö, A. L. Moreira, N. Razavian, and A. Tsirigos, “Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning,” *Nature Medicine*, vol. 24, no. 10, pp. 1559–1567, Sep. 2018. DOI: 10.1038/s41591-018-0177-5.
- [11] S. Wang, J. Shi, Z. Ye, D. Dong, D. Yu, M. Zhou, Y. Liu, O. Gevaert, K. Wang, Y. Zhu, and et al., “Predicting egfr mutation status in lung adenocarcinoma on computed tomography image using deep learning,” *European Respiratory Journal*, vol. 53, no. 3, p. 1800986, 2019. DOI: 10.1183/13993003.00986-2018.
- [12] Y. Dong, L. Hou, W. Yang, J. Han, J. Wang, Y. Qiang, J. Zhao, J. Hou, K. Song, Y. Ma, and et al., “Multi-channel multi-task deep learning for predicting egfr and kras mutations of non-small cell lung cancer on ct images,” *Quantitative Imaging in Medicine and Surgery*, vol. 11, no. 6, pp. 2354–2375, 2021. DOI: 10.21037/qims-20-600.
- [13] Y. Iwatate, I. Hoshino, H. Yokota, F. Ishige, M. Itami, Y. Mori, S. Chiba, H. Arimitsu, H. Yanagibashi, H. Nagase, and et al., “Radiogenomics for predicting p53 status, pd-11 expression, and prognosis with machine learning in pancreatic cancer,” *British Journal of Cancer*, vol. 123, no. 8, pp. 1253–1261, 2020. DOI: 10.1038/s41416-020-0997-1.
- [14] M. H. Lee, J. Kim, S.-T. Kim, H.-M. Shin, H.-J. You, J. W. Choi, H. J. Seol, D.-H. Nam, J.-I. Lee, and D.-S. Kong, “Prediction of idh1 mutation status in glioblastoma using machine learning technique based on quantitative radiomic data,” *World Neurosurgery*, vol. 125, e688–e696, 2019, ISSN: 1878-8750. DOI: <https://doi.org/10.1016/j.wneu.2019.01.157>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1878875019302578>.
- [15] Y. Wei, X. Chen, L. Zhu, L. Zhang, C.-B. Schönlieb, S. Price, and C. Li, “Multi-modal learning for predicting the genotype of glioma,” *IEEE Transactions on Medical Imaging*, pp. 1–1, 2023. DOI: 10.1109/TMI.2023.3244038.
- [16] P. Kosiyo, W. Otieno, J. Gitaka, E. O. Munde, and C. Ouma, “Haematological abnormalities in children with sickle cell disease and non-severe malaria infection in western kenya,” *BMC Infectious Diseases*, vol. 21, no. 1, 2021. DOI: 10.1186/s12879-021-06025-7.
- [17] —, “Association between haematological parameters and sickle cell genotypes in children with plasmodium falciparum malaria resident in kisumu county in western kenya,” *BMC Infectious Diseases*, vol. 20, no. 1, 2020. DOI: 10.1186/s12879-020-05625-z.
- [18] S. Zhang, X. Li, M. Zong, X. Zhu, and D. Cheng, “Learning k for knn classification,” *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 3, Jan. 2017, ISSN: 2157-6904. DOI: 10.1145/2990508. [Online]. Available: <https://doi.org/10.1145/2990508>.
- [19] S. Agrawal, *Logistic regression algorithm: Introduction to logistic regression*, May 2021. [Online]. Available: <https://www.analyticsvidhya.com/blog/2021/05/logistic-regression-supervised-learning-algorithm-for-classification/>.

- [20] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, “Siamese neural networks for one-shot image recognition,” in *ICML deep learning workshop*, Lille, vol. 2, 2015.
- [21] D. Chicco, “Siamese neural networks: An overview,” in *Artificial Neural Networks*, H. Cartwright, Ed. New York, NY: Springer US, 2021, pp. 73–94, ISBN: 978-1-0716-0826-5. DOI: 10.1007/978-1-0716-0826-5\_3. [Online]. Available: [https://doi.org/10.1007/978-1-0716-0826-5\\_3](https://doi.org/10.1007/978-1-0716-0826-5_3).
- [22] R. Kundu, *Everything you need to know about few-shot learning*, Jun. 2022. [Online]. Available: <https://blog.paperspace.com/few-shot-learning/>.
- [23] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: <https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf>. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [24] [Online]. Available: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [25] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM - a tutorial into long short-term memory recurrent neural networks,” *CoRR*, vol. abs/1909.09586, 2019. arXiv: 1909.09586. [Online]. Available: <http://arxiv.org/abs/1909.09586>.
- [26] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].
- [27] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, “A review of applications in federated learning,” *Computers Industrial Engineering*, vol. 149, p. 106 854, 2020, ISSN: 0360-8352. DOI: <https://doi.org/10.1016/j.cie.2020.106854>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835220305532>.
- [28] K. Martineau, *What is federated learning?* Oct. 2022. [Online]. Available: <https://research.ibm.com/blog/what-is-federated-learning>.
- [29] Editor, *Federated learning: The shift from centralized to distributed on-device model training*, Feb. 2022. [Online]. Available: <https://www.altexsoft.com/blog/federated-learning/>.
- [30] Q. Duan, S. Hu, R. Deng, and Z. Lu, “Combined federated and split learning in edge computing for ubiquitous intelligence in internet of things: State-of-the-art and future directions,” *Sensors*, vol. 22, no. 16, p. 5983, 2022. DOI: 10.3390/s22165983.
- [31] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, “Evaluation measures for models assessment over imbalanced data sets,” *Journal of Information Engineering and Applications*, vol. 3, pp. 27–38, 2013.
- [32] M. Grandini, E. Bagli, and G. Visani, *Metrics for multi-class classification: An overview*, 2020. arXiv: 2008.05756.