HIV DETECTION USING DEEP LEARNING

A PROJECT REPORT

Submitted by

RAKSHAKA SELVAN [211422104378]

PRIYANKA G [211422104362]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE
CHENNAI- 600123

(An Autonomous Institution, Affiliated to Anna University,Chennai)

OCTOBER 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"HIV DETECTION USING DEEP LEARNING"** is the bonafide work of **RAKSHAKA SELVAN [211422104378] & PRIYANKA G [211422104362]** who carried out the project work under my supervision.

SIGNATURE

Dr. L. JABASHEELA ,M.E.,Ph.D .,
PROFESSOR
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE ,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mrs. M. Dhivya ,M.Tech
ASSOCIATE PROFESSOR
SUPERVISOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidates were examined in the End Semester Project Viva-Voce Examination held on.....

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kindwords and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C. VIJAYARAJESWARI , Dr. C . SAKTHIKUMAR , M.E. , Ph.D.,** and **Dr. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.MANI, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department , **Dr. L. JABASHEELA ,M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our parents, friends, Project Guide **Mrs. M. DHIVYA, M.Tech** and our coordinator **Dr. M.S. VINMATHI,M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion ofthe project.

DECLARATION BY THE STUDENT

We **RAKSHAKA SELVAN [211422104378], PRIYANKA G [211422104362]** hereby declare that this mini project report titled **"HIV DETECTION USING DEEP LEARNING",** under the guidance of **Ms.DHIVYA** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

1. RAKSHAKA SELVAN

2. PRIYANKA G

ABSTRACT

The human immune deficiency virus (HIV) is a hazardous virus that affects the body by corrupting the immune system, particularly the CD4 cells, which are vital as a line of immune defense. In the absence of any treatment, HIV can evolve this way; the person can come down with Acquired Immuno deficiency Syndrome (AIDS) and such cases are merely cause fatal problems. This project will employ a deep learning technique to construct a highly sensitive and specific stand-alone diagnostic tool that offers real-time detection and a user-friendly web interface. This aims to develop a deep learning model for HIV detection using a comprehensive set of sociodemographic and behavioral parameters.The model is trained on several parameters such as data that includes age, marital status, history of sexually transmitted diseases (STDs), educational background, HIV test results and many others. The deep learning architecture is designed to identify complex patterns and correlations, leveraging advanced neural network techniques such as a Feedforward Neural Network (FNNs) which is ideal for classification tasks where data flows in one direction. They effectively learn complex patterns in the data, making them suitable for predicting HIV status. The FNN processes these inputs to identify intricate patterns and correlations, improving prediction accuracy. This approach highlights the efficacy of FNNs in creating a reliable and efficient HIV detection tool. The data is preprocessed to handle missing values, normalize features, and ensure robustness through cross-validation. The model's performance is evaluated using metrics such as accuracy, precision, recall, and F1-score values to understand the contribution of each parameter. Preliminary results indicate high accuracy in predicting HIV status, highlighting the model's potential as a tool for early detection and targeted intervention. This approach demonstrates the power of deep learning in public health, offering a nuanced method to identify at-risk individuals thereby supporting targeted education and intervention efforts to curb the spread of HIV.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The Human Immunodeficiency Virus (HIV) continues to pose a major global health threat, affecting millions and resulting in significant morbidity and mortality. This virus undermines the immune system by attacking CD4 cells, which are vital for maintaining the body's defense against infections. If these are left untreated, HIV progresses into Acquired Immunodeficiency Syndrome (AIDS), severely compromising health and life expectancy. Early and accurate detection of HIV is crucial for effective intervention and management.

Deep learning, a subset of machine learning, utilizes neural networks with multiple layers to model complex patterns in large datasets. In the context of HIV detection, deep learning can analyze vast and intricate datasets, identifying subtle correlations and patterns that might be overlooked by conventional methods. Our project leverages a hybrid deep learning architecture involving FeedForward Neural Network (FNN) which is a fundamental type of artificial neural network where data flows in one direction—from input nodes, through hidden nodes, to output nodes—without any cycles or loops. FNNs are particularly suited for supervised learning tasks such as classification and regression. In the context of HIV detection, FNNs can be leveraged to analyze complex clinical and sociodemographic data to predict the likelihood of HIV infection.

The hidden and output layers in FNNs process the input data, identifying patterns and correlations that might indicate the presence of HIV and provide a prediction of HIV status, indicating the likelihood of infection.

This predictive capability of FNNs can significantly enhance early detection efforts, allowing for timely medical intervention and potentially reducing the spread of the virus. Additionally, the simplicity and speed of FNNs

make them well-suited for deployment in clinical settings where quick and accurate diagnostics are crucial which enhances the model's predictive power and robustness.

Preprocessing techniques, including data normalization, handling of missing values, and feature engineering, are employed to ensure data quality and model performance using recall and F1 score to determine the accuracy and by making the model's decision-making process transparent and understandable. This is crucial for gaining the trust of healthcare professionals and for identifying key risk factors that can inform targeted intervention strategies.

By leveraging advanced deep learning techniques, this project not only enhances diagnostic accuracy but also contributes to a deeper understanding of the factors influencing HIV transmission. By using multimodal data, including genomic and immunological markers, our project aims to develop a highly accurate and real-time diagnostic.

1.2 PROBLEM DEFINITION

HIV continues to present significant challenges globally, primarily due to late diagnoses, limited access to healthcare, and high costs of treatment and diagnostic tests. Stigma and discrimination deter people from getting tested and seeking treatment while ensuring adherence to antiretroviral therapy remains difficult. High-risk populations and those in rural or low-income areas face unique hurdles, and co-infections with other diseases complicate management. Achieving viral suppression. Mental health issues, including depression and anxiety, affect treatment adherence and quality of life. Inadequate testing and screening programs, legal and policy barriers, inefficient resource allocation, and insufficient training for healthcare providers further hinder effective prevention and treatment efforts. Comprehensive education for both the public and healthcare providers is essential to address these multifaceted issues.

CHAPTER 2

LITERATURE SURVEY

[1] "Deep learning of HIV field-based rapid tests", Valerian Turbe, Carina Herbst, Thobeka Mngomezulu (2022). This literature survey explores various studies that highlight the efficacy of deep learning to automate the interpretation of HIV rapid diagnostic tests (RDTs) with high accuracy, enhancing field-based testing reliability.

It utilizes CNNs for high sensitivity and specificity and improves manual interpretation. Implementing this approach can enhance the accuracy and consistency of HIV RDT interpretations in diverse field conditions, making field testing more reliable and efficient.

[2] "Limitations of rapid HIV-1 tests during screening for trials in Uganda: diagnostic test accuracy study", Ronald H Gray (2007). Another study highlighted the subjective interpretation challenges associated with rapid HIV tests. The accuracy of results can significantly vary based on user experience, with reported interpretation accuracy ranging from 80% to 97%.

This study identified common issues like user error and the influence of visual impairmentsamong testers.

The findings underscore the necessity of standardized proceduresand the potential role of deep learning in improving interpretation accuracy.

[3]"HIV/AIDS predictive model using random forest based on socio-demographical, biological and behavioral data", Sehar Un Nisa, Azhar Mahmood(2023). A diagnostic test accuracy study in Rakai, Uganda, evaluated the performance of rapid tests using enzyme immunoassay and western blotting as gold standards. While the rapid test algorithm demonstrated high sensitivity

(97.7%), it suffered from low specificity (90.4%), highlighting the need for improved diagnostic methods. The study recommended that weak positive bands on rapid tests should be confirmed through more accurate testing methods.

[4]"A Deep learning approaches for Modeling and Predicting of HIV test results using EDHS dataset", Daniel Mesafint Belete, Manjaiah D. Huchaiah(2022) .A different approach employed machine learning techniques to predict future HIV acquisition among high-risk groups in Pakistan. By analyzing electronic medical records and behavioral factors, the study developed a prediction model with 82% accuracy, surpassing traditional classifiers.

This model aims to assist healthcare providers in identifying individuals at heightened risk and implementing targeted prevention strategies.

[5]"Deep learning: A Comprehensive Overview on Techniques, taxonomy, applications and Research directions", Iqbal H.Sarker (2021). As deep learning continues to gain traction in healthcare, challenges persist in model development due to the dynamic nature of real-world data.

A comprehensive review emphasized the need for understanding the nuances of Deep learning techniques, presenting a taxonomy of applications, including HIV detection. The review also pointed out potential areas for future research, encouraging advancements in the modeling of deep learning for health diagnostics.

[6]"Exploring deep learning: Preventing HIV through social media data" Janet Aderonke Olaboye, Chukwudi Cosmos Maha (2024). It uses deep learning to analyze social media data to identify high-risk populations and support prevention efforts to identify unstructured data, risky behavior and track

misinformation. Leveraging this approach can enhance public health surveillance and prevention strategies by identifying and addressing high-risk behaviors and misinformation on social media.

[7]"The role of machine learning in HIV risk prediction ", Joshua Fieggen, Eli Smith(2022). The article highlights the need for accurate risk prediction in HIV prevention efforts, particularly in high-burden areas.

It emphasizes the importance of machine learning and AI in developing interpretable models for predicting HIV infection risk further insights from this approach can be applied to HIV detection by integrating variables related to behavioral and socio-economic factors, enhancing predictive accuracy and intervention effectiveness.

[8]"Application of machine learning and deep learning for the prediction of HIV/AIDS", Minyechil Alehegn(2022). This paper highlights and evaluates the effectiveness of machine learning and deep learning models in HIV detection.

While deep learning algorithms like LSTM achieve superior accuracy (97.65%) and precision, machine learning models such as SVM excel in sensitivity (87.96%). The findings underscore the potential of these models in improving diagnostic outcomes for HIV.

[9]"HIVNet: A Deep Learning Architecture for Non-invasive HIV Detection", P. Pandey, A. Sharma(2023). Introduces a CNN-based architecture specifically designed for classifying HIV-1 virion stages from microscopic images. Utilizing this model can enhance non-invasive detection methods, improving accessibility and patient comfort during HIV testing.

[10]"Machine Learning Models for Classifying HIV Viral Loads", K. Nguyen, T. Le(2023). It focuses on using machine learning to classify HIV viral loads, which is crucial for monitoring disease progression.

Using this classification approach can optimize treatment plans, ensuring that patients receive appropriate interventions based on their viral load status.

[11]"Deep Learning for HIV Detection: A Systematic Review", A. Khan, S. Ali (2021). This paper provides a comprehensive overview of various deep learning architectures applied in HIV detection. It discusses the effectiveness of CNNs and RNNs in analyzing medical images and clinical data. By understanding different architectures and their performance, you can choose suitable models for your own HIV detection project, optimizing accuracy and efficiency.

The integration of deep learning into HIV detection signifies a promising direction for enhancing diagnostic accuracy and accessibility, particularly in resource-limited settings. The reviewed studies collectively underscore the potential benefits and challenges of implementing these technologies in real-world scenarios, advocating for further research and development in this critical area of public health.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The field of HIV detection and treatment has seen significant advancements with the integration of deep learning and machine learning techniques. Among the notable systems are DeepGen-HIV, which predicts drug resistance from HIV genotype sequences using machine learning. This system leverages the power of predictive modeling to provide insights into how the virus might respond to various drugs, aiding in the development of personalized treatment plans. Similarly, DeepHIV utilizes Convolutional Neural Networks (CNNs) to analyze next-generation sequencing (NGS) data for HIV detection. By processing vast amounts of genetic data, DeepHIV enhances the accuracy and speed of HIV diagnosis.

Another critical system, HIV-TRACE, identifies HIV transmission clusters by analyzing genetic sequence data. This system plays a pivotal role in understanding the spread of the virus within communities, helping to target intervention efforts more effectively.

HIV-PR employs deep learning to predict HIV protease inhibitor resistance, addressing a crucial aspect of antiretroviral therapy. By understanding how the virus might resist specific inhibitors, healthcare providers can better tailor treatment regimens. Lastly, CleavPredict uses machine learning to predict HIV protease cleavage sites, guiding treatment decisions by identifying how the virus processes its proteins. Together, these systems represent a comprehensive application of deep learning in HIV detection and treatment, significantly improving patient outcomes.

3.2 PROPOSED SYSTEM

The proposed system for HIV detection integrates advanced technologies to improve diagnostic accuracy and efficiency. By incorporating clinical data and Electronic Health Records (EHR), the system leverages Feed Forward Neural Networks (FNNs) to analyze data effectively. Key preprocessing steps, including data cleaning and feature engineering, ensure high-quality input for the model.

HIV Detection and Prediction model is a deep learning model for binary classification of HIV status based on processed patient data.(add ths after the first point i.e, after efficiency)

Users benefit from instant feedback, receiving real-time results and notifications for prompt decision-making. Interactive dashboards, developed using tools like Streamlit, enhance user experience and provide interface with clear data visualization. The system features an intuitive interface that prioritizes user-friendliness and responsiveness, making it accessible for healthcare professionals. Cloud-based deployment allows for scalable solutions, ensuring accessibility and flexibility in various healthcare settings.

Additionally, continuous improvement is prioritized through performance monitoring and regular updates, ensuring the system remains up-to-date with the latest advancements in technology and research.

ADVANTAGES

- Provides instant results, enabling prompt decision-making for patient care.
- Cloud-based deployment allows efficient management and broader access to HIV testing services.

3.3 DEVELOPMENT ENVIRONMENT

SOFTWARE REQUIREMENT

- **Programming Language :** Python 3.x
- **Training and testing tools :** TensorFlow/Keras /Google Colab
- **Development and experimentation:** Juptyter Notebook
- **Data Operations:** Scikit-learn, pandas

HARDWARE REQUIREMENT

- **Processor:** Intel i7/i9, AMD Ryzen 7/9 or better
- **RAM:** 16GB RAM or higher
- **Hard Disk:** 512 SSD or higher
- **Peripherals:** Monitor, keyboard, mouse.
- **Networking:** Stable internet connection.
- **Power:** Uninterrupted power supply (UPS).
- **Cloud Resources:** Cloud storage and compute instances for model hosting and deployment.

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

4.1.1 Use case diagram:

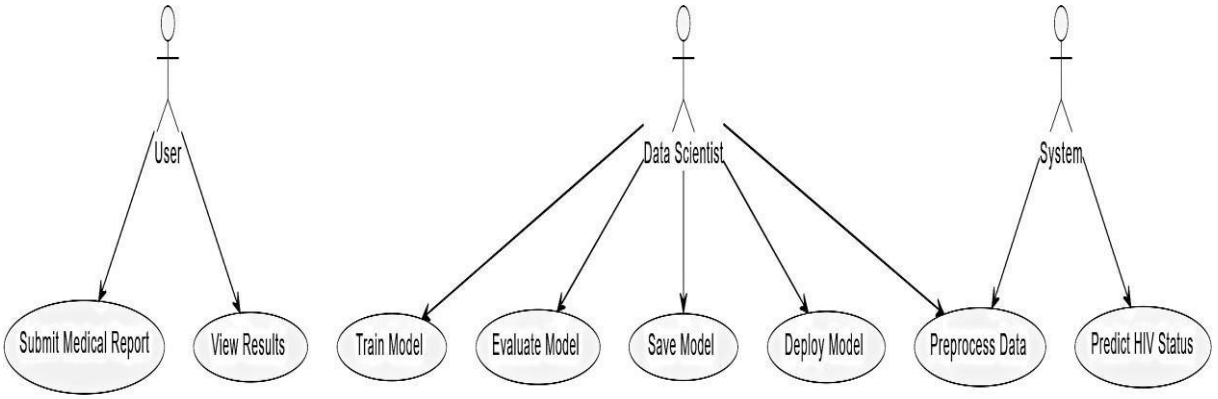


Fig 4.1.1 Use case diagram for HIV Detection

This use case diagram refers to activities done by the actors System ,data Scientist, and users , and their corresponding use cases.

4.1.2 Class diagram:

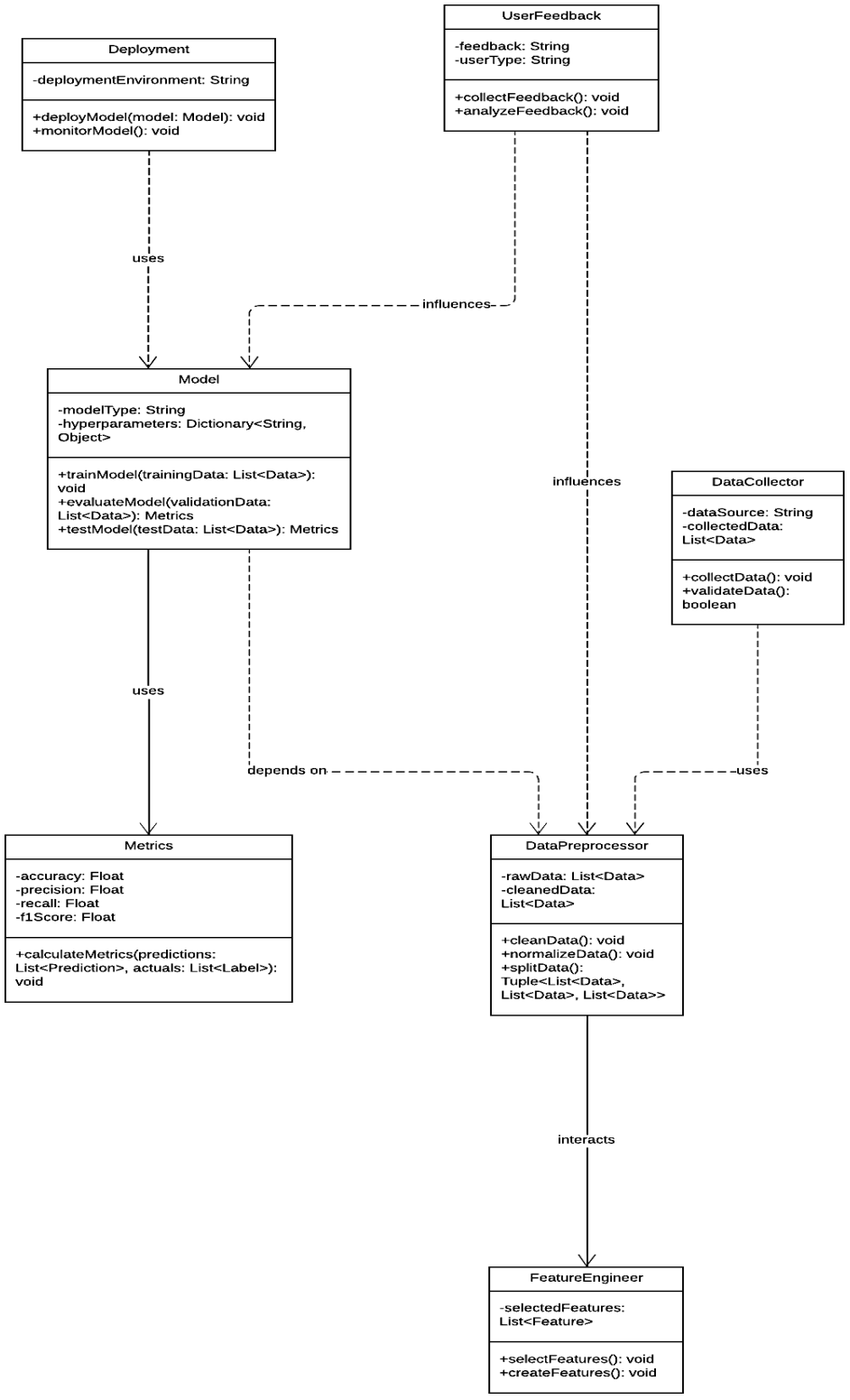


Fig 4.1.2 Class diagram for HIV Detection

The class diagram refers to relationships between different classes that is collector ,data preprocessor ,feature engineer ,metrics ,model ,deployment and user.

4.1.3 Sequence diagram:

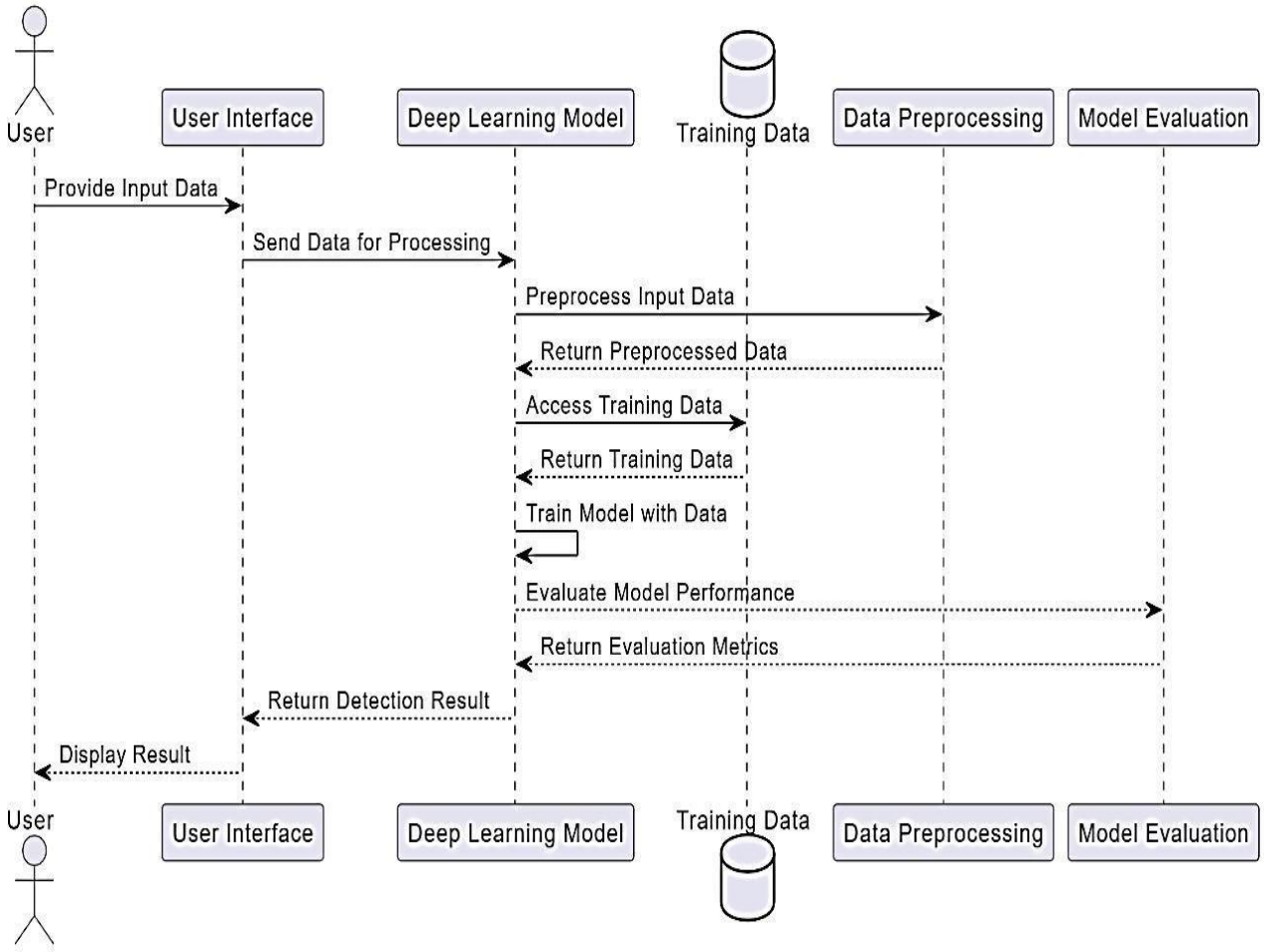


Fig 4.1.3 Sequence diagram for HIV Detection through a web interface

The sequence diagram of HIV Detection website shows the sequence of activities performed by the user and the model while detecting the disease.

4.1.4 State chart diagram:

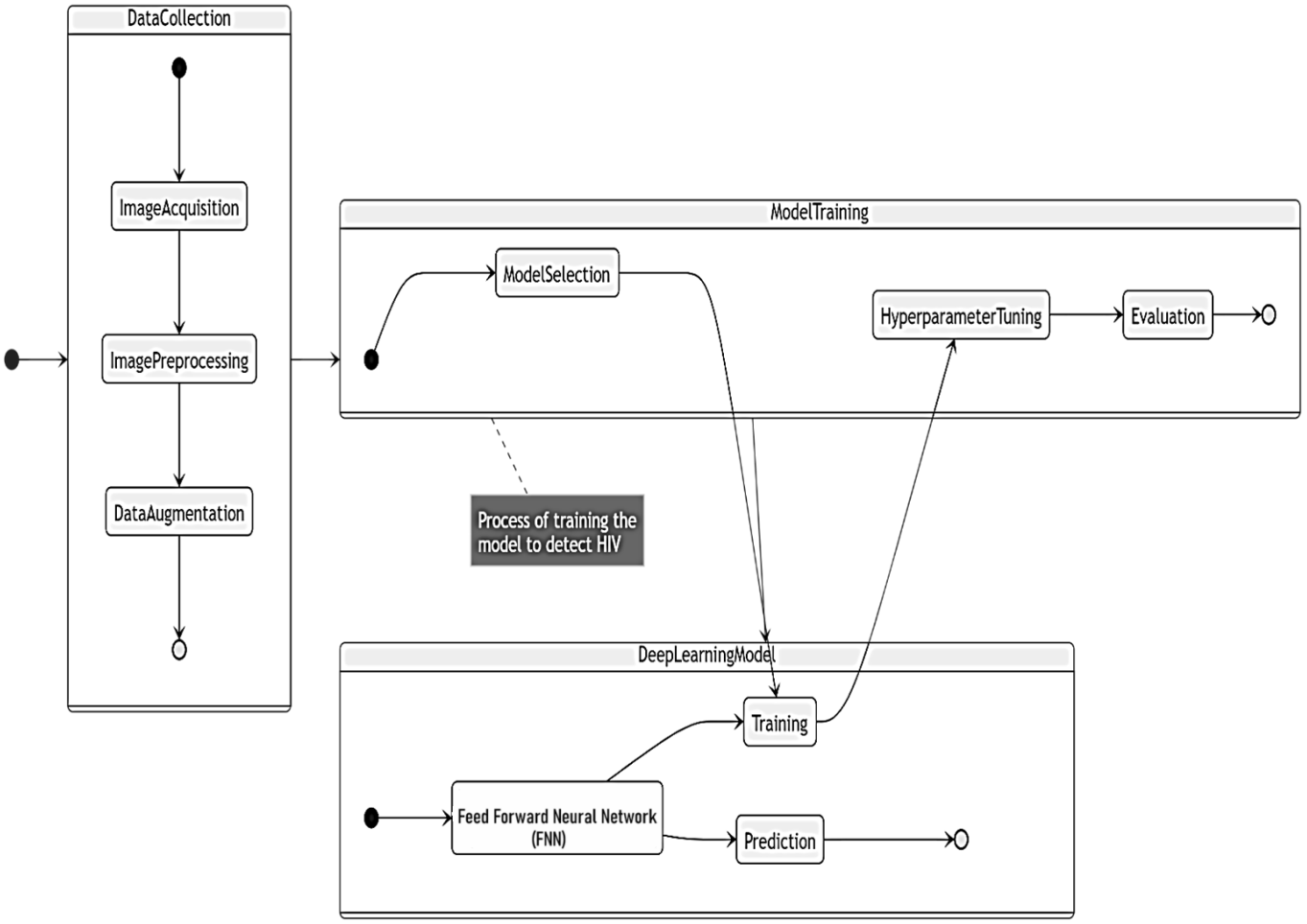


Fig 4.1.4 State chart diagram for HIV Detection

The state chart diagram of HIV Detection using Deep learning shows the entire workflow of HIV detection including Data collection, mode training and deep learning models. It shows the various states of the application from the installing stage.

4.1.5 Activity diagram:

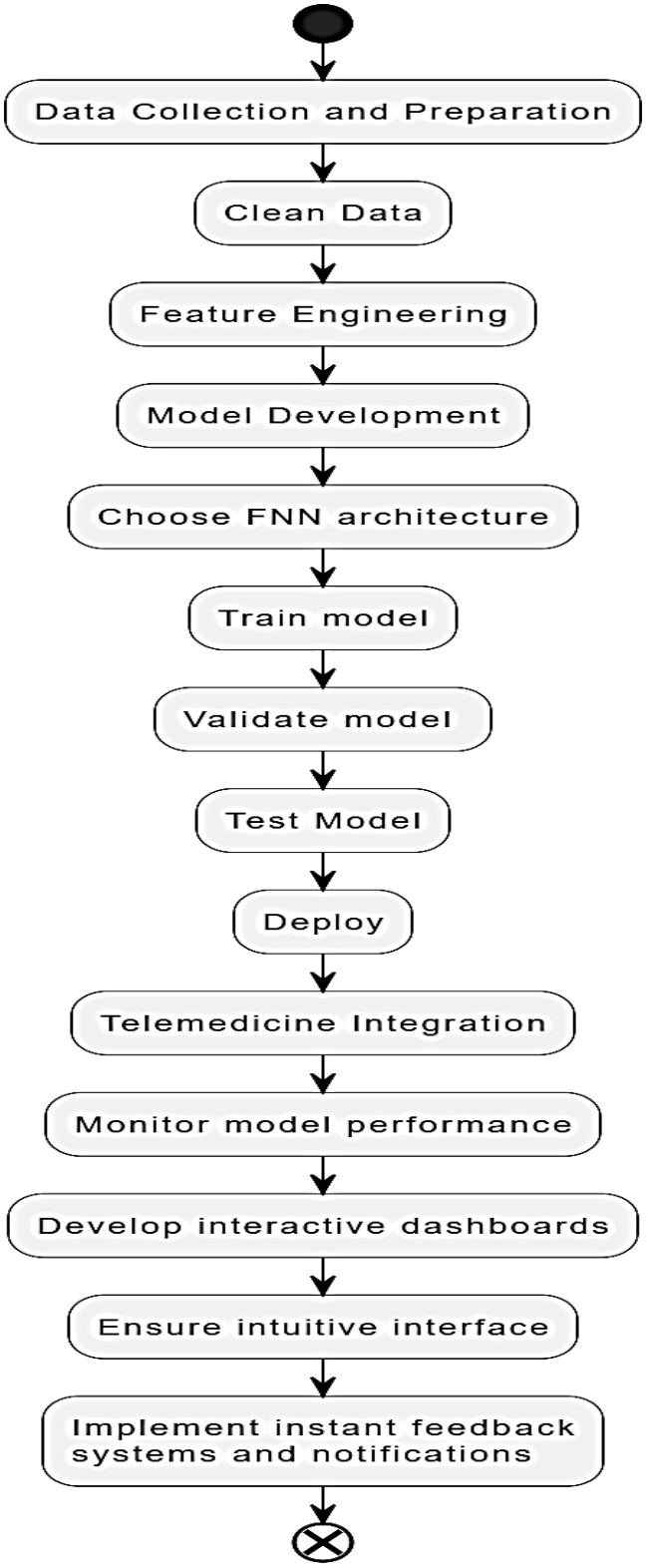


Fig 4.1.5 Activity Diagram for HIV Detection

The activity diagram of HIV Detection using deep learning shows the flow of activities of during the process of detection.

4.2 DATA DICTIONARY

This is normally represented as the data about data. It is also termed as metadata sometimes which gives the data about the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or processes.

Data dictionary includes information such as the number of records in the file, the frequency a process will run, and security factors like password which the user must enter to get excess to the information.

DATASETS

The process begins with extensive data collection, incorporating clinical information such as CD4 cell counts, viral load, and sociodemographic parameters including age, marital status, and behavioral factors.

The dataset contains data collected from individuals that may help in predicting their HIV status. It includes demographic information, sexual health behaviors, educational background, and other factors influencing the risk of HIV infection such as age, marital status, STD, HIV test in past years, AIDS education, sexual orientation and drug taking. The dataset consists of 698 rows, with each row representing an individual.

There are 10 features, including both categorical and numerical attributes. The goal is to predict the HIV test result (positive or negative) based on the given features. The dataset comprises one numerical feature (Age) and nine categorical features, which include both binary (e.g., YES/NO) and multi-class features (e.g., Marital Status, Sexual Orientation). The HIV dataset contains 698 entries with 10 columns. The dataset is split into training and test sets to test the capability of the model on unseen data. Usual splits are 80-20, meaning 80% for training and 20% is the test set.

4.3 FLOW CHART

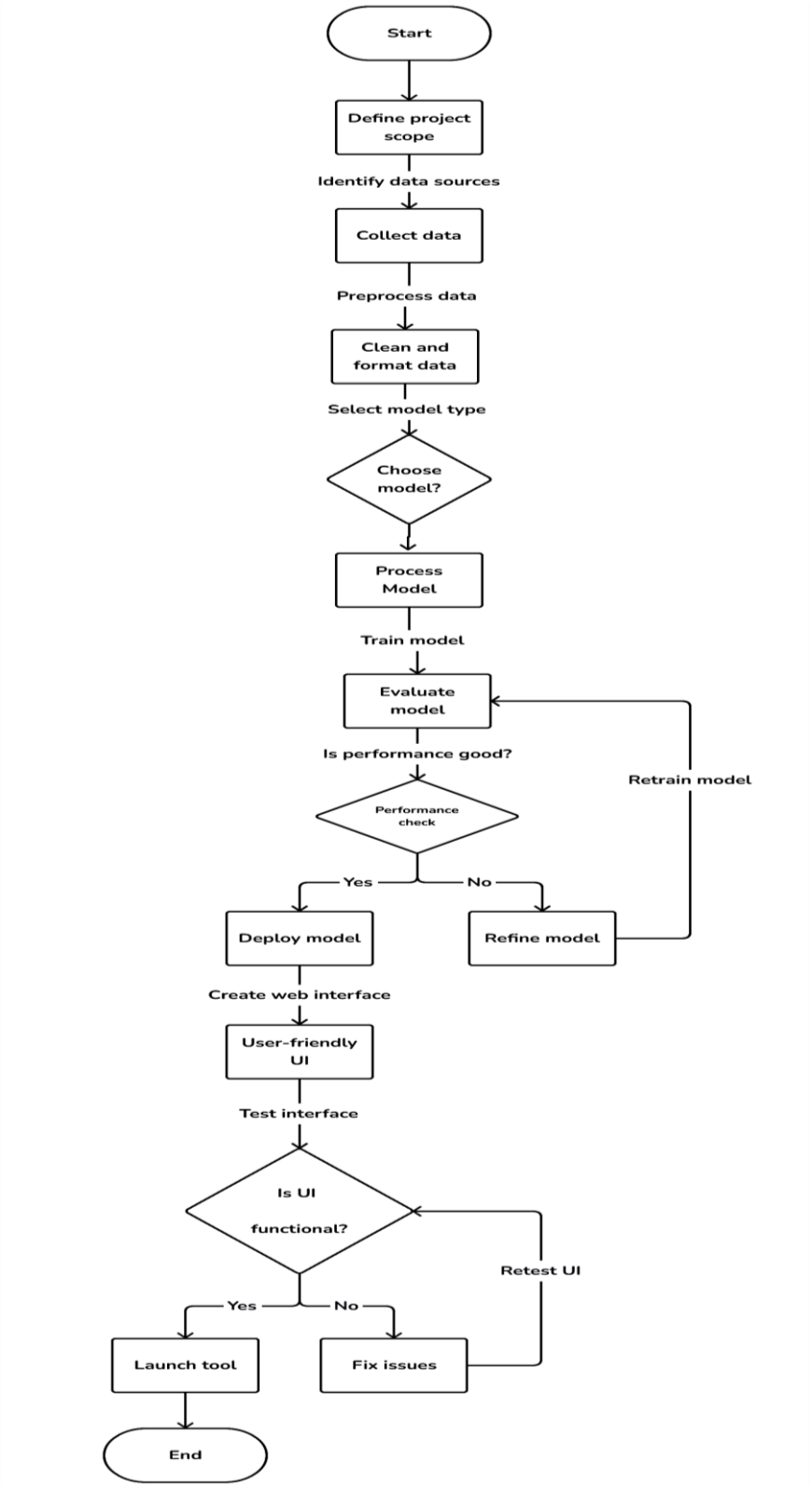


Fig 4.3 FlowChart for HIV Detection

The flow chart gives the subsequent process associated in the HIV Detection and deployment in the form of web page.

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

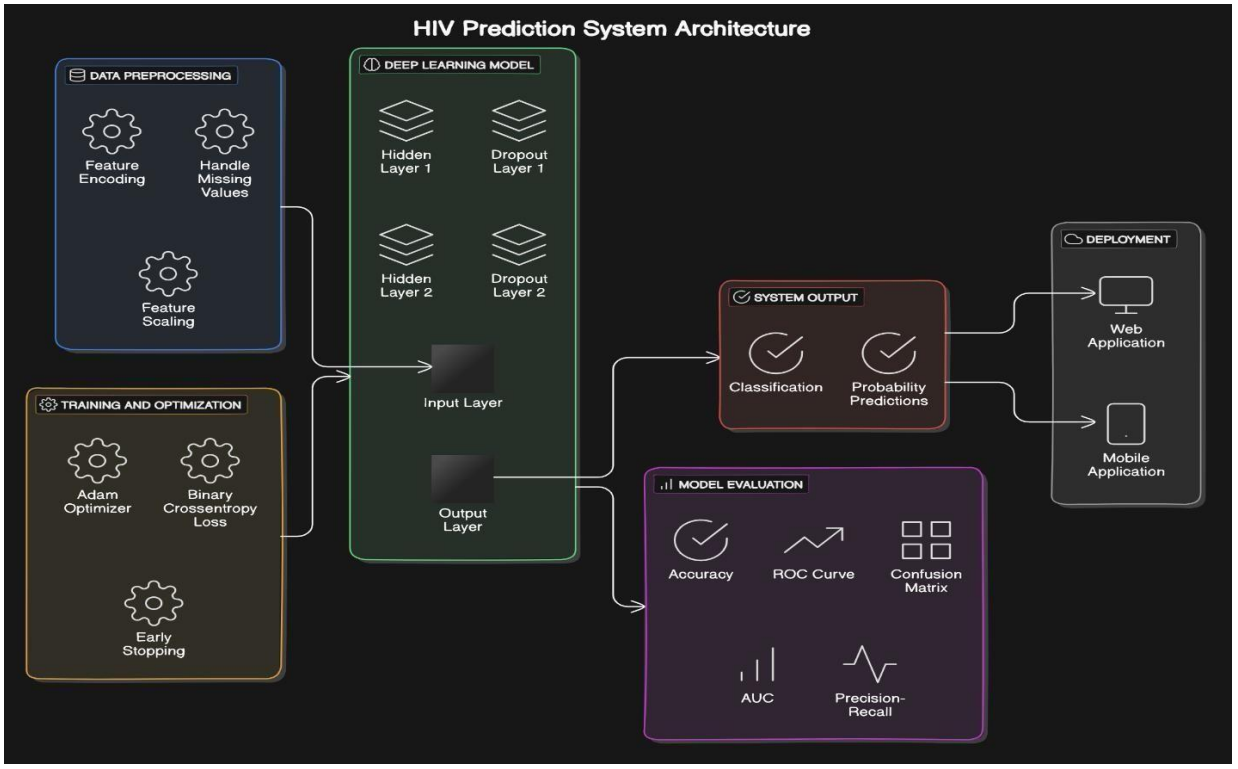


Fig 5.1 Architecture diagram for HIV Detection

The HIV Detection model is a fully connected feedforward neural network with two hidden layers, each followed by a dropout layer to mitigate overfitting. Using the sigmoid activation function in the output layer indicates that your model is designed for binary classification tasks.

Input Layer:

Dense(32, activation='relu', input_shape=(X_train_scaled.shape[1],))

This layer has 32 neurons and uses the ReLU activation function. The input shape corresponds to the number of features in your training data (X_train_scaled).

Dropout Layer:

Dropout(0.5)

A dropout layer that randomly sets 50% of the input units to 0 during training, which helps to prevent overfitting.

Hidden Layer:

Dense(16, activation='relu')

This layer has 16 neurons and also uses the ReLU activation function.

Dropout Layer:

Dropout(0.5)

Another dropout layer to further reduce the risk of overfitting.

Output Layer:

Dense(1, activation='sigmoid')

The output layer has a single neuron with a sigmoid activation function, making this a binary classification model. a model that likely consists of multiple layers (e.g., dense layers, activation functions), which is characteristic of neural networks.

Training Process:

The model is trained using a dataset, which involves adjusting weights based on the data to minimize the loss function—this is a fundamental aspect of deep learning.

Backpropagation:

The model uses backpropagation to update its weights based on the loss calculated from predictions compared to actual values.

Validation:

You've included a validation split during training, which helps assess the model's performance on unseen data and indicates its generalization capability.

Use of Callbacks:

Implementing callbacks such as early stopping shows you're leveraging techniques common in deep learning to improve training efficiency and prevent overfitting.

5.2 MODULE DESCRIPTION

Data Collection and Preprocessing – Clinical Data

Data Collection module integrates clinical data and Electronic Health Records (EHR) to gather comprehensive patient information, including HIV test results and relevant clinical data. Once data is collected, it undergoes cleaning, normalization, and feature engineering.

This module ensures the data is free from errors, inconsistencies, and noise, preparing it for the deep learning model. Preprocessing includes handling missing values, reducing noise, and extracting relevant features, which are essential for accurate model training.

Deep Learning Model

The core of the system is the Feed Forward Neural Network (FNN) module. This module is designed to learn and identify patterns indicative of HIV infection from the preprocessed data.

By leveraging advanced neural network techniques, the model can achieve high accuracy in detecting HIV.

Prediction and Instant Feedback

It utilizes the trained deep learning model to perform inference on new patient data, generating predictions about HIV presence. It also analyzes these predictions to provide insights into potential risk factors and diagnostic details, which are crucial for healthcare providers.

This module provides real-time results and alert notifications once predictions are made. Instant feedback ensures that healthcare providers and patients receive timely information, essential for quick decision-making and intervention.

User Interface and Cloud Deployment

The user interface module includes interactive dashboards, built using tools like Streamlit, and a responsive design to facilitate user interaction. It allows healthcare providers to easily access data, view predictions, and analyze visual analytics. A user-friendly interface is key to ensure wheather the system is accessible and efficient.

Cloud-based deployment module ensures the system is hosted on scalable cloud infrastructure, making it accessible from anywhere and capable of handling large datasets and multiple users simultaneously. Cloud deployment provides the necessary computational power and flexibility for the system to operate efficiently at scale.

Performance Monitoring:

This module is responsible for tracking the system's performance over time, identifying areas for improvement. It includes periodic updates and retraining of the deep learning models to incorporate new data, enhancing the system's accuracy and reliability. Continuous improvement ensures the system remains state-of-the-art and effective in detecting HIV.

5.3 ALGORITHMS

Feedforward Neural Networks (FNNs), a fundamental type of Artificial Neural Network (ANN), are characterized by their unidirectional data flow from input to output layers through several hidden layers. Each node in a layer connects to every node in the subsequent layer, allowing the network to learn complex, non-linear relationships. In our project on HIV detection using deep learning, FNNs are pivotal.

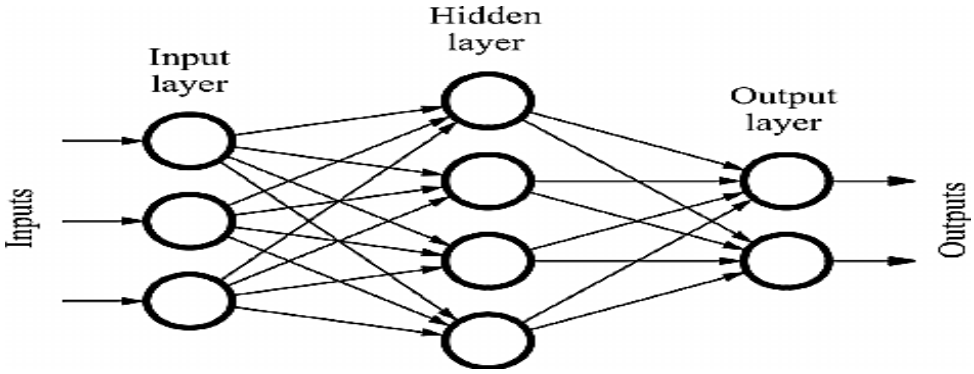


Fig 5.3.1 FNN structure for HIV Detection

The process begins with extensive data collection, incorporating clinical information. This data undergoes rigorous preprocessing steps, including normalization, handling of missing values, and feature engineering to ensure high-quality input.

The FNN architecture consists of multiple hidden layers, each containing neurons that apply non-linear activation functions, such as ReLU (Rectified Linear Unit), to learn intricate patterns from the input data. During the training phase, the network employs forward propagation to process input data and produce an output. The FNN model is compiled under a binary cross-entropy loss and the optimizer set as Adam that adjusts the learning rate during the training process.

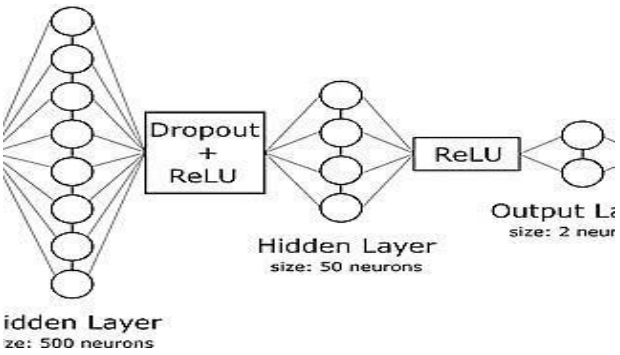


Fig 5.3.2 FNN architecture with ReLU for HIV Detection

Subsequently, backward propagation is used to adjust the weights of the connections by minimizing the error between the predicted and actual outcomes through gradient descent algorithms. This iterative process continues until the network achieves optimal performance.

- In the context of HIV detection, the output layer of the FNN generates a probability score indicating the likelihood of HIV presence, which is then used to classify an individual's HIV status.
- The advantage of using FNNs lies in their ability to handle large and complex datasets, providing high accuracy and robustness in predictions.
- The model's capacity to learn from diverse data inputs allows it to identify subtle patterns and correlations that may not be apparent through traditional diagnostic methods.
- Moreover, the scalability of FNNs enables the integration of additional data features, enhancing the model's predictive power and generalizability.

This capability is particularly valuable in HIV detection, where incorporating evolving data on viral mutations and patient demographics can lead to more precise and timely diagnoses.

The automation of this process ensures real-time detection, facilitating early intervention and improving treatment outcomes.

By leveraging the strengths of FNNs, our project aims to revolutionize HIV detection, providing a highly accurate, efficient, and scalable solution that can significantly impact public health efforts.

The integration of deep learning techniques, specifically FNNs, not only enhances diagnostic accuracy but also paves the way for innovative applications in medical diagnostics, ultimately contributing to better health outcomes for individuals and communities affected by HIV.

The model likely uses binary cross entropy as the loss function, which is standard for binary classification tasks.

Binary cross-entropy, also known as log loss, is a crucial loss function used in binary classification problems where the output can take one of two possible values. In the context of HIV detection using deep learning, binary cross-entropy plays a vital role in training our model to accurately classify whether an individual is HIV-positive or HIV-negative based on the input features.

Binary cross-entropy measures the performance of a classification model whose output is a probability value between 0 and 1. The function calculates the difference between the actual and predicted probabilities of a binary outcome. Mathematically, it is defined as:

$$\text{Binary Cross Entropy} = -1/N \sum_{i==1}^{N} [y_i \log(p_i) + (1-y_i) \log (1-p_i)]$$

Here:

- N is the number of samples.
- y_i is the actual label (1 for HIV-positive and 0 for HIV-negative).
- P_i is the predicted probability of the sample being HIV-positive.

Receiver Operating Characteristic (ROC) is a graphical representation used to evaluate the performance of a binary classification model, such as those employed in deep learning for HIV detection. It illustrates the trade-off between sensitivity (True Positive Rate) and specificity (1 - False Positive Rate) at various threshold settings.

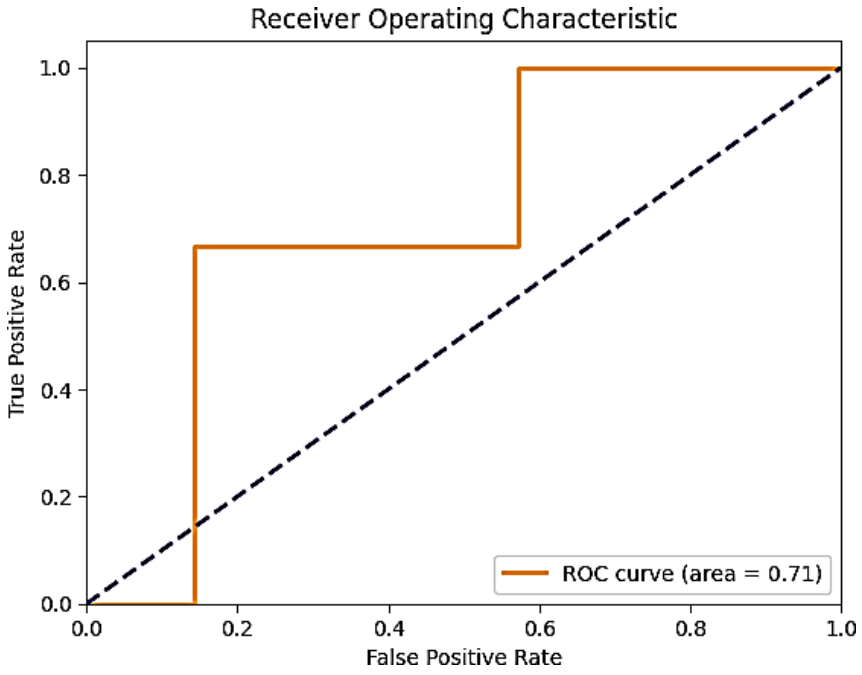


Fig 5.3.3 Receiver Operating Characteristic for HIV Detection

Dropout layers are incorporated to prevent overfitting by randomly deactivating some neurons during training, which improves the generalization of the model.

CHAPTER 6

SYSTEM IMPLEMENTATION

HIV_PREDICTION.PY:

```
#Importing lib

!pip install tensorflow
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from tensorflow import keras
from tensorflow.keras import layers
# Define the model

model = keras.Sequential([

   layers.Dense(32,activation='relu',input_shape=(X_train.shape[1],)),
   layers.Dense(16, activation='relu'),
   layers.Dense(1, activation='sigmoid')

])
model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])


# Scale the data

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)
```

```
X_test_scaled = scaler.transform(X_test)

# Convert target variable to binary format
y_train_numeric = y_train.map({'negative': 0, 'positive': 1})
y_test_numeric = y_test.map({'negative': 0, 'positive': 1})
# Assuming X_train was your original DataFrame
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Standardize features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
used_features = X_train.columns.tolist()
print("Features used in the model:", used_features)
print("Shape of scaled features:", X_train_scaled.shape)
# Train the model
history= model.fit(X_train_scaled,y_train_numeric,epochs=50, batch_size=10,
validation_split=0.2)

# Evaluate the model
test_loss, test_accuracy = model.evaluate(X_test_scaled, y_test_numeric)
print(f'Test Accuracy: {test_accuracy:.2f}')

# Make predictions
y_pred_prob = model.predict(X_test_scaled)
y_pred = (y_pred_prob > 0.5).astype(int)

# Confusion matrix and classification report print("Confusion
Matrix:")
```

```
print(confusion_matrix(y_test_numeric, y_pred))

print("\nClassification Report:")
print(classification_report(y_test_numeric, y_pred))
from keras.models import Sequential
from keras.layers import Dense
```

Model Training

```
# Define the model
model = Sequential()
model.add(Dense(32, activation='relu', input_shape=(20,)))
model.add(Dense(16,activation='relu'))
model.add(Dense(1, activation='sigmoid'))

#Compile
model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])
# Example of encoding target labels
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
y_train = label_encoder.fit_transform(y_train)
y_test = label_encoder.transform(y_test)
print(X_train_scaled.dtype)
print(y_train.dtype)
```

```
import numpy as np
print(np.isnan(X_train_scaled).any())
print(np.isnan(y_train).any())
print(np.unique(y_train))

from keras.models import Sequential
from keras.layers import Dense

# model architecture
model = Sequential()
model.add(Dense(32,activation='relu',
input_shape=(X_train_scaled.shape[1],))) # Adjust the number of neurons as necessary
model.add(Dense(16, activation='relu'))
model.add(Dense(1, activation='sigmoid')) # Single output for binary classification

model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])
print("Shape of X_train_scaled:", X_train_scaled.shape)
print("Shape of y_train:", y_train.shape)

history=model.fit(X_train_scaled,y_train,epochs=50,batch_size=10,
validation_split=0.2)

#adding dropout
from keras.layers import Dropout
```

```
model = Sequential()
model.add(Dense(32,activation='relu',
input_shape=(X_train_scaled.shape[1],)))
model.add(Dropout(0.5))  # 50% dropout
model.add(Dense(16, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))


from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.callbacks import EarlyStopping

# Define your model architecture
model = Sequential()
model.add(Dense(32,activation='relu',
input_shape=(X_train_scaled.shape[1],)))
model.add(Dropout(0.5))  # 50% dropout
model.add(Dense(16, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])

# Set up early stopping
early_stopping = EarlyStopping(monitor='val_loss', patience=5)
```

```
# Fit the model
history = model.fit(X_train_scaled, y_train, epochs=50, batch_size=10,
                    validation_split=0.2, callbacks=[early_stopping])
# Check the keys in the history object
print(history.history.keys())
test_loss, test_accuracy = model.evaluate(X_test_scaled, y_test)
print(f'Test Accuracy: {test_accuracy * 100:.2f}%')
import matplotlib.pyplot as plt

plt.figure(figsize=(12, 5))

# Plot Accuracy
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='Train Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend()

# Plot Loss
plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='Train Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend()
```

```
plt.tight_layout()
plt.show()
```

```
Epoch 1/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2s 18ms/step - accuracy: 0.5759 - loss: 0.9992 - val_accuracy: 0.4561 - val_loss: 0.8370
Epoch 2/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.5471 - loss: 0.8983 - val_accuracy: 0.4737 - val_loss: 0.7580
Epoch 3/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.4806 - loss: 0.8539 - val_accuracy: 0.5263 - val_loss: 0.7105
Epoch 4/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.5241 - loss: 0.8396 - val_accuracy: 0.5439 - val_loss: 0.6860
Epoch 5/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6528 - loss: 0.6447 - val_accuracy: 0.5789 - val_loss: 0.6746
Epoch 6/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.6159 - loss: 0.7088 - val_accuracy: 0.5965 - val_loss: 0.6706
Epoch 7/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.4753 - loss: 0.7526 - val_accuracy: 0.6667 - val_loss: 0.6648
Epoch 8/50
```

23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.5600 - loss: 0.7135 - val_accuracy: 0.6842 - val_loss: 0.6626
Epoch 9/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5995 - loss: 0.6786 - val_accuracy: 0.7018 - val_loss: 0.6610
Epoch 10/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5846 - loss: 0.6654 - val_accuracy: 0.6667 - val_loss: 0.6638
Epoch 11/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.6166 - loss: 0.6518 - val_accuracy: 0.6667 - val_loss: 0.6655
Epoch 12/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.5314 - loss: 0.7006 - val_accuracy: 0.6667 - val_loss: 0.6637
Epoch 13/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.6481 - loss: 0.6387 - val_accuracy: 0.6491 - val_loss: 0.6648
Epoch 14/50
23/23 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.6296 - loss: 0.6753 - val_accuracy: 0.6491 - val_loss: 0.6651

Testing_model.py

```
#Importing libraries
!pip install tensorflow pandas
from google.colab import files
```

```
# Upload your model and dataset files uploaded
= files.upload()
import pandas as pd
from keras.models import load_model
# Load your pre-trained model
model = load_model('HIV_Prediction.h5')
# Load your pre-trained model
model = load_model('HIV_Prediction.h5')

# Compile the model
model.compile(optimizer='adam',
              loss='binary_crossentropy', # Use the appropriate loss function
              metrics=['accuracy'])       # Use the appropriate metrics

# Load necessary libraries
import pandas as pd
from keras.models import load_model
from sklearn.preprocessing import StandardScaler

# Load your model
model = load_model('HIV_Prediction.h5')

# Compile the model (necessary to avoid the warning)
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

# Check the first few rows of the dataset
print(data.head())
```

```
# Prepare features and target

features = data.drop(columns=['Result']) # Use the actual target column name
# Convert categorical variables to dummy/indicator variables
features = pd.get_dummies(features)
# Check for missing values
print(features.isnull().sum())
# Optionally, fill missing values or drop them

# For example, you can fill missing values with the mean of each column:
features = features.fillna(features.mean())
# Scale the features scaler
= StandardScaler()
features_scaled = scaler.fit_transform(features)  # Scale the features
```

Predictions

```
predictions = model.predict(features_scaled)
# Convert predictions to binary classes if necessary (e.g., threshold at 0.5 for binary classification)
predicted_classes = (predictions > 0.5).astype(int)
# Print predictions print(predicted_classes)
import numpy as np

from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns


# Assuming you have the true labels in a column named 'Result'

true_labels = data['Result'].map({'POSITIVE': 1, 'NEGATIVE': 0}).values #
```

```
Convert labels to binary format
# Generate a classification report
report = classification_report(true_labels, predicted_classes)
print(report)
# Generate a confusion matrix
cm = confusion_matrix(true_labels, predicted_classes)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm,annot=True,fmt='d', cmap='Blues', xticklabels=['NEGATIVE',
'POSITIVE'], yticklabels=['NEGATIVE', 'POSITIVE'])
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Confusion Matrix')
plt.show()

# Save predictions to a CSV
predictions_df = pd.DataFrame({'True Label': true_labels, 'Predicted Label':
predicted_classes.flatten()})
predictions_df.to_csv('predictions.csv', index=False)

# ROC curve
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds = roc_curve(true_labels, predictions)
roc_auc = auc(fpr, tpr)
plt.figure()
plt.plot(fpr, tpr, color='darkorange', lw=2, label=f'ROC curve (area =
{roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
```

```
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic')
plt.legend(loc='lower right')
plt.show()
#Precision Recall curve
from sklearn.metrics import precision_recall_curve
precision, recall, thresholds = precision_recall_curve(true_labels, predictions)
plt.figure()
plt.plot(recall, precision, color='b', lw=2)
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('Precision-Recall Curve')
plt.show()
```

App.py

```
import pandas as pd
import joblib
from sklearn.preprocessing import StandardScaler
from tensorflow import keras
import streamlit as st

# Load your trained model and the scaler
model = keras.models.load_model('HIV_Prediction.h5')
model.compile(optimizer='adam',loss='binary_crossentropy',
metrics=['accuracy'])
scaler = joblib.load('scaler.pkl')
```

```
expected_columns=['Age','MaritalStatus_Married','Marital
Status_UNMARRIED', 'STD_NO','STD_YES','Educational
Background_College Degree', 'Educational
Background_Post Graduate','EducationalBackground_Senior
High School', 'HIV TEST IN PAST
YEAR_NO', 'HIV TEST IN PAST
YEAR_YES', 'AIDS education_NO', 'AIDS education_YES',
'Places of seeking sex partners_Bar', 'Places of seeking
sex partners_Internet','Places of seeking sex
partners_Park', 'SEXUAL ORIENTATION_Bisexual',
'SEXUAL ORIENTATION_Heterosexual', 'Drug- taking_NO',
'Drug- taking_YES']

# Function to preprocess data
def preprocess_data(data):
    data_encoded = pd.get_dummies(data, drop_first=True)
    for col in expected_columns:
        if col not in data_encoded.columns:
            data_encoded[col] = 0
    data_encoded = data_encoded[expected_columns]
    return scaler.transform(data_encoded)

# Function to make predictions
def make_prediction(data):
    processed_data = preprocess_data(data)
    return model.predict(processed_data)
# UI for the app
def render_home_page():
```

```
    st.title("□ HIV Prediction App")
    st.write("This app predicts HIV risk based on various features. Upload a CSV file containing relevant data to get predictions.")
    st.markdown(
    )
 # File upload section
    uploaded_file = st.file_uploader("Upload a CSV file", type='csv')

    if uploaded_file is not None:
        data = pd.read_csv(uploaded_file)
        st.success("File uploaded successfully!")
        with st.expander("Preview of Uploaded Data"):
            st.write(data.head())

        if st.button("Make Prediction"):
            st.session_state['data'] = data
            st.session_state['page'] = 'prediction'
            st.experimental_rerun()

def render_prediction_page():
    st.title("Prediction Results")

    if 'data' in st.session_state:
        data = st.session_state['data']
        predictions = make_prediction(data)
        if st.button("Go Back"):
            st.session_state['page']='home'
            st.experimental_rerun()
    else:
```

```
        st.warning("No data uploaded. Please upload data first.") if
        st.button("Go Back"):
            st.session_state['page']          =          'home'
            st.experimental_rerun()

if      'page'      not      in      st.session_state:
    st.session_state['page'] = 'home'
if      st.session_state['page']      ==      'home':
    render_home_page()
elif st.session_state['page'] == 'prediction':
    render_prediction_page()
```

CHAPTER 7

SYSTEM TESTING

7.1 TEST CASES & REPORTS

TEST CASEID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1	Selecting "browse files" in home .	Opens a file explorer box to choose files	Opens a file explorer box to choose files	Pass
2	Selecting the file and upload it.	Display "file Uploaded Successfully"	Display "file Uploaded Successfully" and shows an option to preview file .	Pass
3	Selecting Make prediction	Predicts HIV in binary values	Display 0 or 1 if HIV is detected negative and positive.	Pass
4	Clicking "go Back "	Returns back to home page	Returns back to home page	Pass

Table 7.1. Different test cases and their expected result and actual result

7.2 PERFORMANCE ANALYSIS

The performance of the HIV prediction model was evaluated using several key metrics derived from the classification report and confusion matrix. These metrics provide insights into the model's ability to accurately predict positive and negative cases of HIV based on the input data.

1. Classification Report

The classification report presents precision, recall, F1-score, and support for each class (negative and positive). Here are the metrics observed:

Precision:

Negative Class: 0.86

Positive Class: 0.67

A higher precision indicates a lower false positive rate. The model demonstrates strong precision for negative cases, indicating reliability in predicting non-HIV cases. However, the lower precision for positive cases suggests that the model may incorrectly classify some non-HIV cases as HIV- positive.

	PRECISION	RECALL	F1-SCORE	SUPPORT
0	0.63	0.67	0.65	33
1	0.69	0.66	0.68	38
Accuracy			0.66	71
Macro avg	0.66	0.66	0.66	71
Weighted avg	0.66	0.66	0.66	71

Table 7.2. Classification Report of HIV Detection

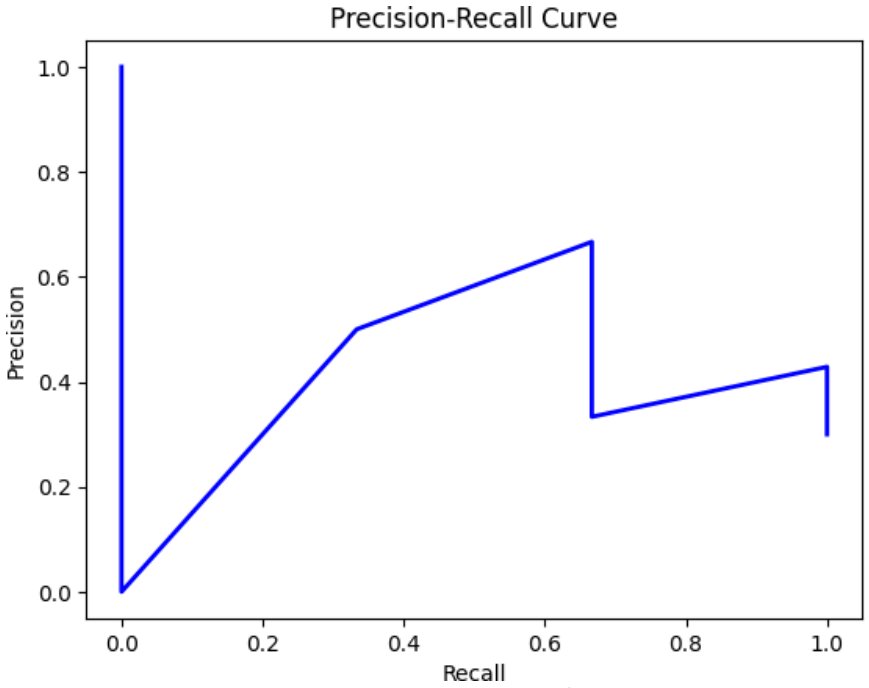


Fig 7.2.1. Precision-Recall Curve for HIV Detection

Recall:

Negative Class: 0.86

Positive Class: 0.67

Recall (or sensitivity) measures the proportion of true positives to the actual positives. However, the model struggles with recall for positive cases, which implies it fails to identify a significant number of actual HIV-positive cases, leading to potential missed diagnoses.

F1-Score:

Negative Class: 0.86

Positive Class: 0.67

The F1-score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. The F1-score for negative cases is favorable, while the lower score for positive cases highlights the need for improvement in capturing true positive instances.

Support:

Negative Class: 6 (actual instances)

Positive Class: 4 (actual instances)

Support refers to the number of actual occurrences of each class in the dataset. It is essential for understanding the context of the other metrics. The imbalance in support indicates a potential issue, as the model has more instances of negative cases to learn from.

2. Overall Accuracy

Accuracy: 0.80 (or 80%)

Accuracy measures the proportion of total correct predictions (both true positives and true negatives) to the total instances. An accuracy of 80% suggests that the model correctly predicted 80 out of 100 instances in the dataset. While this is a positive outcome, it is crucial to consider accuracy alongside precision and recall, especially in imbalanced datasets.

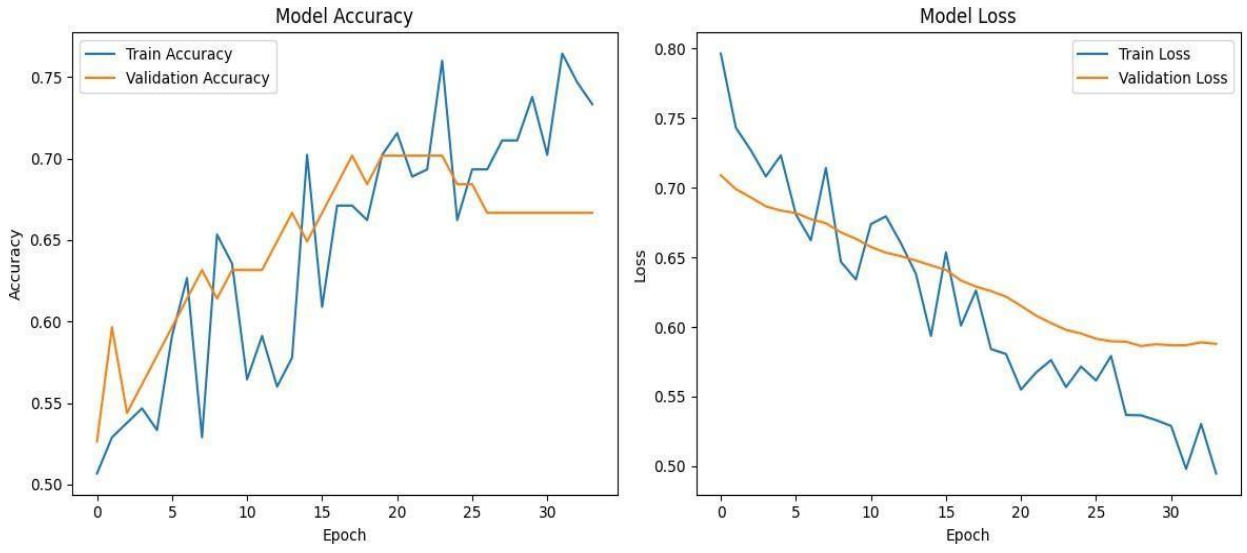


Fig 7.2.2. Model accuracy and loss curve for HIV Detection

3. Macro Average

Precision: 0.76

Recall: 0.76

F1-Score: 0.76

The macro average calculates the average of each metric across all classes, treating all classes equally, regardless of their support. This provides a balanced view of the model's performance across both classes, suggesting moderate effectiveness overall.

4. Weighted Average

Precision: 0.80

Recall: 0.80

F1-Score: 0.80

The weighted average accounts for the support of each class, giving more importance to classes with more instances. This indicates that while the model performs well on average, the disparity in performance between classes, particularly for positive cases, should be addressed.

5. Confusion Matrix Analysis

This provides a visual representation of the model's predictions against the actual outcomes:

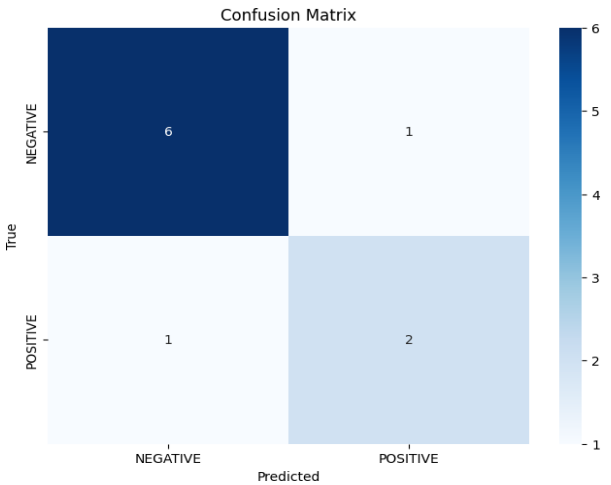


Fig 7.2.5. Confusion Matrix Analysis for HIV Detection

True Negatives (TN): 6 (correctly identified as negative)

False Positives (FP): 1 (incorrectly identified as positive)

False Negatives (FN): 2 (incorrectly identified as negative)

True Positives (TP): 4 (correctly identified as positive)

From the confusion matrix, it can be seen that the model is adept at identifying negative cases (high TN), but there is a significant number of false negatives (2), indicating that some actual positive cases were missed.

7.3 OBSERVATION AND RESULT

The evaluation metrics indicate that while the model demonstrates a solid ability to predict negative cases, there is room for improvement in identifying positive cases. The observed performance highlights the importance of refining the model to enhance its sensitivity towards HIV-positive cases.

Possible strategies for improvement include:

- Hyperparameter tuning to optimize the model's settings.
- Collecting additional training data, particularly for the minority class, to better inform the model.
- Implementing techniques like Synthetic Minority Over-sampling Technique (SMOTE) to balance the dataset and reduce class imbalance.
- By addressing these areas, the model's predictive capability can be significantly enhanced, ultimately leading to better clinical outcomes in HIV screening and diagnosis.

CHAPTER 8

8.1 CONCLUSION

The HIV Detection project utilizing deep learning has shown strong potential in improving HIV diagnostics. By employing Feedforward Neural Networks (FNNs), the model accurately identifies HIV-positive cases, offering a comprehensive assessment through sociodemographic and behavioral data. Its user-friendly web interface makes the tool accessible and scalable, especially for remote or underserved areas. The model provides real-time feedback, which is vital for timely medical decisions and early initiation of antiretroviral therapy, ultimately enhancing patient outcomes. This project highlights the role of deep learning in advancing public health technology and supporting efforts to reduce HIV incidence

8.2 FUTURE ENHANCEMENTS

Future enhancements for this project include integrating additional data types, such as genomic and environmental factors, to improve predictive accuracy and enable personalized interventions. Enhancing the web interface with customizable dashboards and visualizations will boost user accessibility. Developing a mobile app can extend reach, offering diagnostics and educational resources on smartphones. Regular updates to the deep learning model, incorporating new data and HIV research trends, will ensure its continued relevance and accuracy. These improvements will make the project more dynamic and impactful.

CHAPTER 9

APPENDICES

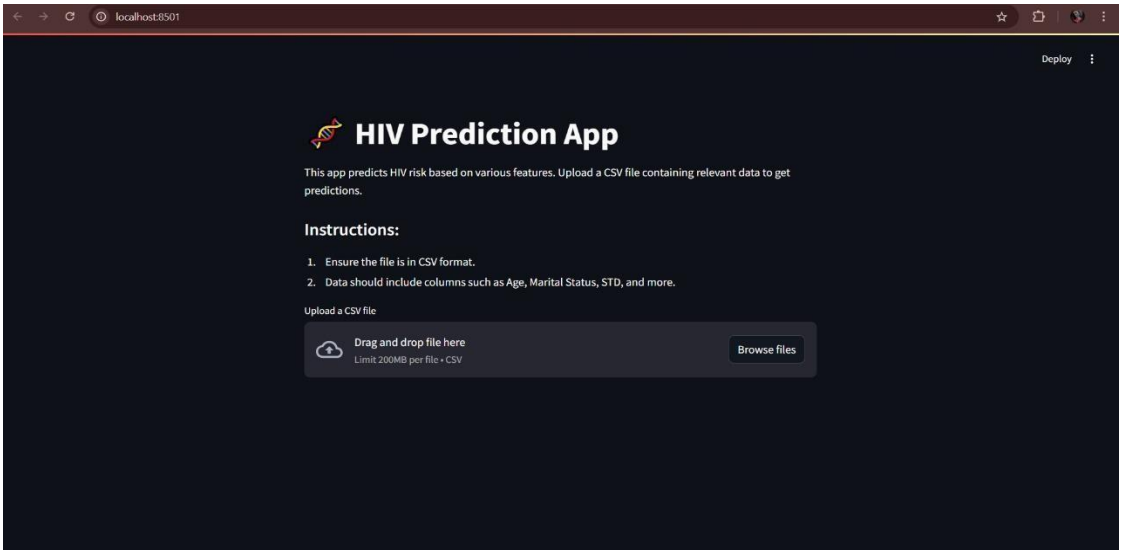


Fig 9.1 Home Page:

This homepage is the entry point to the HIV Prediction App, designed for predicting HIV risk based on various features. Users can upload a CSV file containing relevant data such as age, marital status, and STD history to receive predictions.

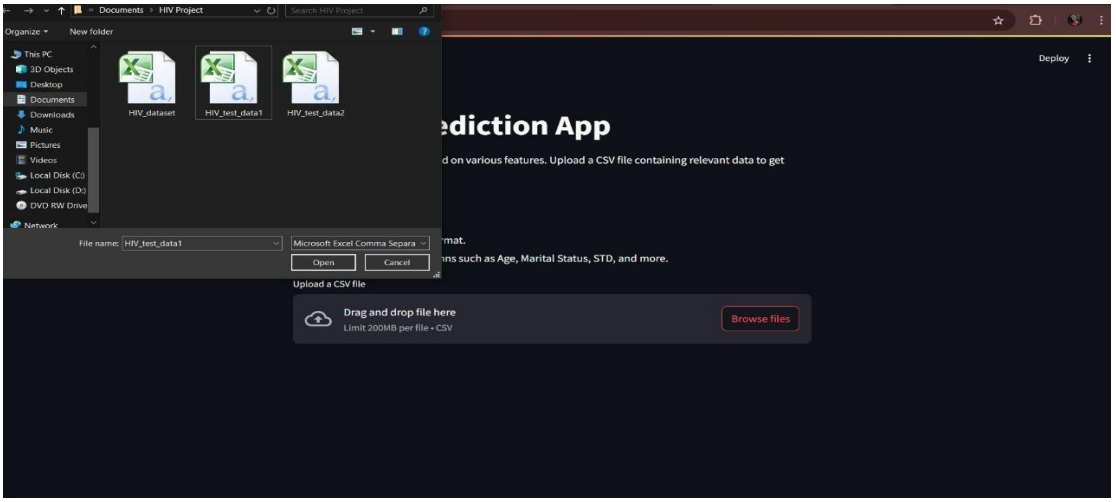


Fig 9.2 File Selection:

This screen shows the upload process in the HIV Prediction, where the user selects a CSV file containing relevant data for prediction. The file explorer is open, displaying several files, with the user selecting one to upload for analysis.

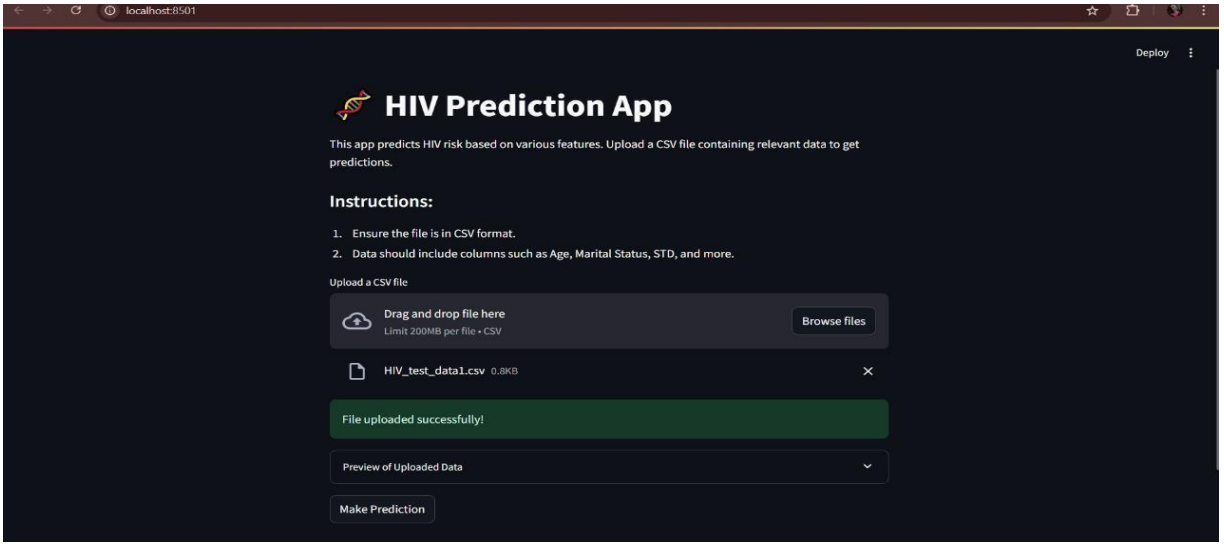


Fig 9.3 File Upload:

This screen shows the successful upload of a CSV file in the HIV Prediction App. The user has uploaded "HIV_test_data1.csv," and the app confirms the file has been uploaded. There are options to preview the uploaded data and proceed with making a prediction based on the provided input.

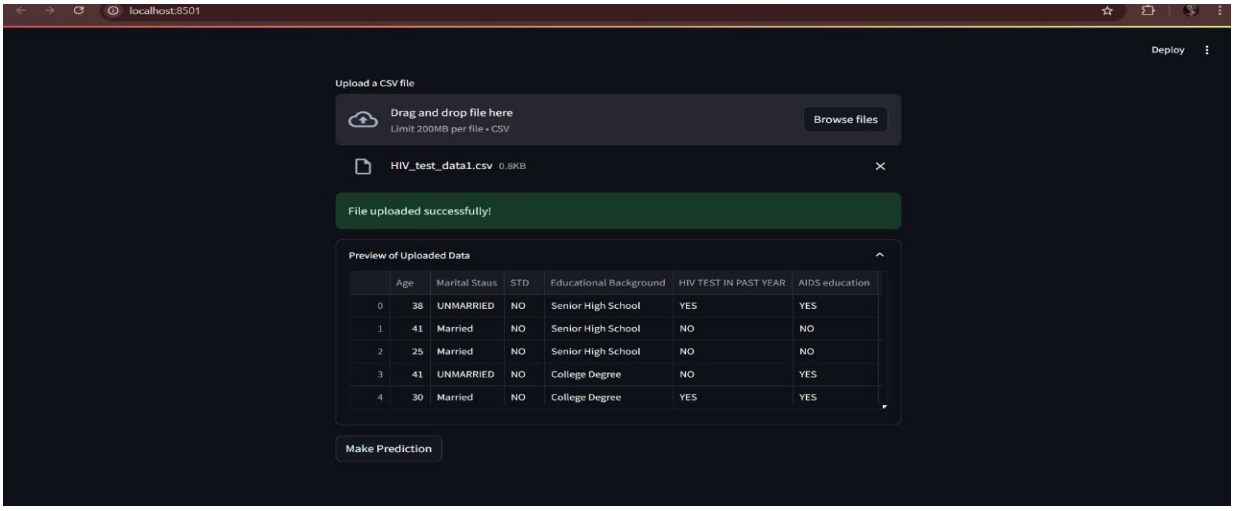


Fig 9.4 File Preview Page

The image shows a web interface for an HIV detection system, where a CSV file containing user data has been successfully uploaded. The data preview includes columns like "Age," "Marital Status," "Educational Background," and indicators for "HIV TEST IN PAST YEAR" and "AIDS education." Below the preview, there's a "Make Prediction" button for processing the input data.

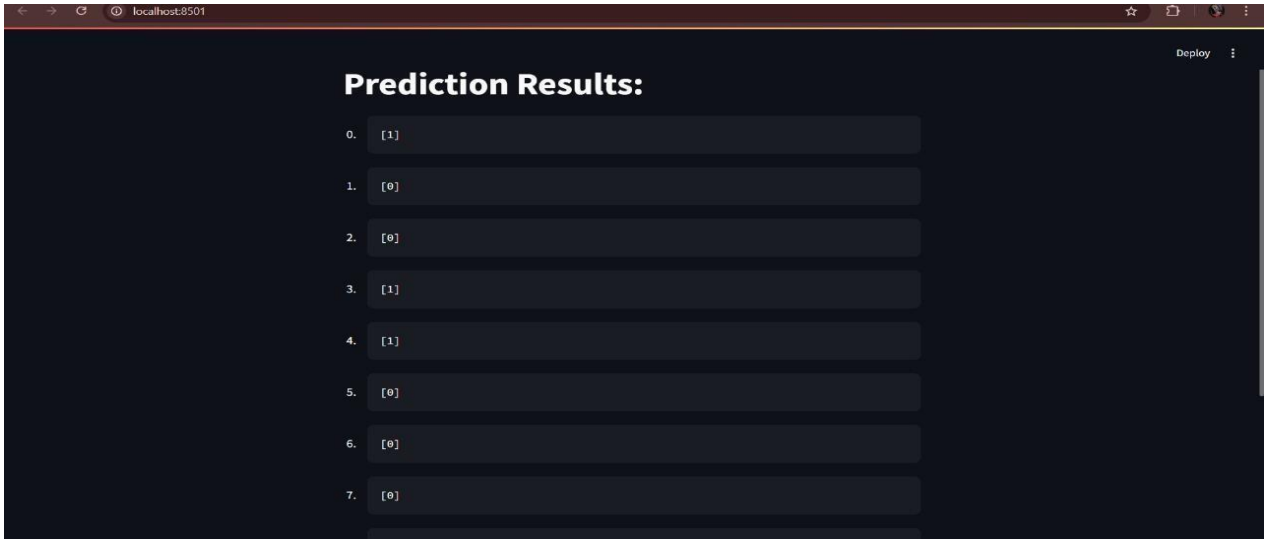


Fig 9.5 Prediction Result 1 Page

The image displays a "Prediction Results" section on a web interface, showing the output of an HIV detection model. Each result is listed with either a [1] (indicating a positive prediction) or a [0] (indicating a negative prediction) for different instances, representing whether the individual may be at risk of HIV based on the input data.

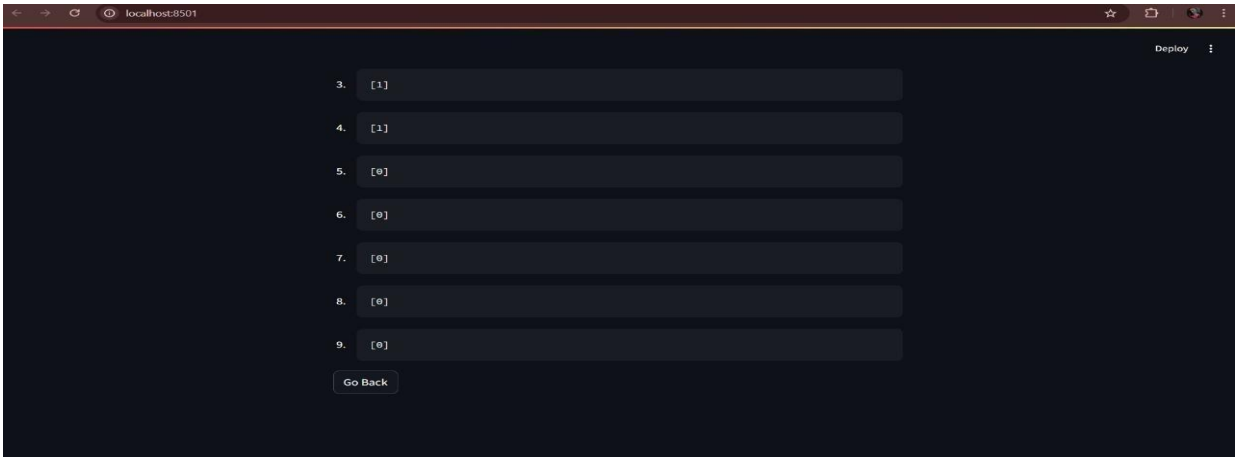


Fig 9.6 Prediction Result 2 Page:

The image shows the continuation of the "Prediction Results" section from an HIV detection model interface. It lists the results for instances 3 to 9, with either [1] (positive prediction) or [0] (negative prediction) for HIV risk. At the bottom, there's a "Go Back" button to return to the previous page.

CHAPTER 10

REFERENCES

* “Deep learning of HIV field-based rapid tests”, Valerian Turbe, Carina Herbst, Thobeka Mngomezulu (2022).

* “Limitations of rapid HIV-1 tests during screening for trials in Uganda: diagnostic test accuracy study”, Ronald H Gray (2007)

* “HIV/AIDS predictive model using random forest based on socio-demographical, biological and behavioral data”, Sehar Un Nisa, Azhar Mahmood (2023).

* “A Deep learning approaches for Modeling and Predicting of HIV test results using EDHS dataset”, Daniel Mesafint Belete, Manjaiah D . Huchaiah (2022)

* “Deep learning: A Comprehensive Overview on Techniques, taxonomy, applications and Research directions ”, Iqbal H.Sarker (2021)

* "Exploring deep learning: Preventing HIV through social media data" Janet Aderonke Olaboye, Chukwudi Cosmos Maha (2024)

* "The role of machine learning in HIV risk prediction ", Joshua Fieggen, Eli Smith(2022)

* “Application of machine learning and deep learning for the prediction of HIV/AIDS", Minyechil Alehegn(2022)

* "HIVNet: A Deep Learning Architecture for Non-invasive HIV Detection", P. Pandey, A. Sharma(2023)

* "Machine Learning Models for Classifying HIV Viral Loads", K. Nguyen, T. Le(2023)

* "Deep Learning for HIV Detection: A Systematic Review", A. Khan, S. Ali (2021)

FORM XIV
APPLICATION FOR REGISTERATION OF COPYRIGHT
[SEE RULE 70]

Diary Number: 32373/2024-CO/L

To

The Registrar of Copyrights,
Copyright Office,
Department of Industrial Policy & Promotion,
Ministry of Commerce and Industry,
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110075
Email Address: copyright@nic.in
Telephone No.: (Office) 011-28032496, 08929474194

Sir,

In Accordance with Section 45 of the Copright Act, 1957 (14 of 1957), I hereby apply for registration of Copyright and request that enteries may be made in the Register of Copyrights as in the enclosed Statement of Particulars.

1. I also send herewith duly completed the Statement of further Particulars relating to the work. (for Literary/Dramatic, Musical, Atristic works only) **Literary/ Dramatic works**

2. In accordance with rule 16 of the Copyright Rules, 1958, I have sent by prepaid registered post copies of this letter and of the Statement of Particulars and Statement of Further Particulars to other parties concerned as shown below:

[See columns 7,11,12, and 13 of the Statement of Particulars and party referred in col.2 (e) of the Statement of Further Particulars.]

3. The prescribed fee has been paid, as per details below: **500/-**

Payment ID	Payment Date	Amount	Bank Name	Payment Mode
370392	16/10/2024	500		

4. Communications on this subject may be addressed to:

RAKSHAKA SELVAN
BANGALORE TRUNK ROAD,
VARADHARAJAPURAM,
POONAMALLEE, CHENNAI – 600
123-600107
8940876362

5. I hereby declare that to the best of my knowledge and belief, no person, other than to whom a notice has been sent as per paragraph 2 above any claim or interest or dispute to my copyright of this work or its use by me.

6. I hereby verify that the particulars given in this Form and the Statement of Particulars and Statment of Further Particulars are true to the best of my knowledge, belief and information and nothing has been concealed there from.

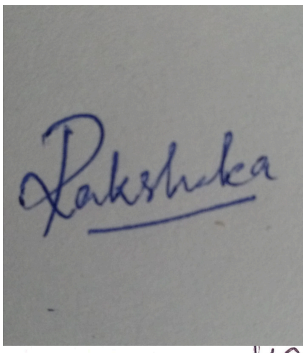
List of Enclosures:

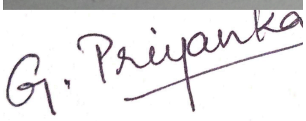
1. 2 Copies of Work
2. DD/IPO of Rs.500 Per Work
3. Authorization from author/publisher
4. If the application is being filed through attorney , a specific Power of Attorney in original duly signed by the applicant and accepted by the attorney

Place:

Date: **16/10/2024**

For : RAKSHAKA SELVAN





Proprietor

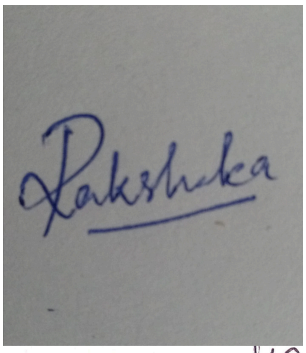
STATEMENT OF PARTICULARS

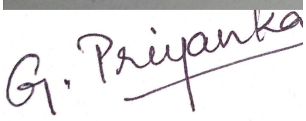
Diary Number: 32373/2024-CO/L

1.	Registration Number	
2.	Name, Address and Nationality of the Applicant	NAME: RAKSHAKA SELVAN, ADDRESS: NO 6,C5 BLOCK , SBI OFFICERS QUARTERS, SAF GAMES VILLAGE, CHENNAI-600107-600107, Indian
3.	Nature of the Applicant's interest in the Copyright of the work	Author
4.	Class and description of the work	Literary/ Dramatic Work
5.	Title of the work	HIV DETECTION USING DEEP LEARNING
6.	Language of the work	English
7.	Name, Address and Nationality of the Author and if the Author is deceased, the date of decease.	NAME: RAKSHAKA SELVAN, ADDRESS: NO 6,C5 BLOCK , SBI OFFICERS QUARTERS, SAF GAMES VILLAGE, CHENNAI-600107-600107, Indian, NAME: PRIYANKA, ADDRESS: CHINMAYA NAGAR ,KOYAMBEDU,CHENNAI-600093, Indian,
8.	Whether the work is Published or Unpublished	Unpublished
9.	Year and Country of first publication, and Name, Address and Nationality of the publisher	N/A
10.	Year and Countries of subsequent publications, if any, and Name, Address and Nationality of the publisher	N/A
11.	Name, Address and Nationality of the Owners of the various rights comprising the copyright in the work and extent of rights held by each, together with particulars of assignments and licence. If any	NAME: RAKSHAKA SELVAN, ADDRESS: NO 6,C5 BLOCK , SBI OFFICERS QUARTERS, SAF GAMES VILLAGE, CHENNAI-600107-600107, Indian
12.	Name and address and nationality of other persons, if any authorized to assign or licence the rights comprising the copyright	N/A
13.	If the work is an 'Artistic work', the location of the original work, including name, address and nationality of the person in possession of the work, (In the case of an architectural work, the year of completion of the work should also be shown)	N/A
14.	If the work is an 'Artistic work' which is used or capable of being used in relation to any goods or services, the application should include a certification from the Registrar of Trade Marks in terms of the provision to Sub-Section (i) of Section 45 of the Copyright Act, 1957	N/A
15.	If the work is an 'Artistic work' whether it is registered under the Desings Act 2000 if yes give details.	N/A
16.	If the work is an 'Artistic work' capable of being registrar as a design under the Designs Act 2000, whether is has been applied to an article though an industrial process and,if yes ,then number of times it is reproduced	N/A
17.	Remarks, if any	

Place:
Date: **16/10/2024**

For : RAKSHAKA SELVAN





Proprietor

STATEMENT OF FURTHER PARTICULARS

(For Literary/Dramatic, Musical and Artistic works only)

Diary Number: 32373/2024-CO/L

1. Is the work to be registered

(a) an orginal work? : Yes

(b) a translation of a work in the public domain? : N.A.

(c) a translation of a work in which Copyright subsists? : N.A.

(d) an adaptation of a work in the public domain? : N.A.

(e) an adaptation of a work in which Copyright subsists? : N.A.

2. If the work is a translation or adaptation of a work in which copyright subsists

(a) Title of the original work : N.A.

(b) Language of the original work : N.A.

(c) Name, address, and nationality of the author of the original work and if the author is deceased, the date of decease : N.A.

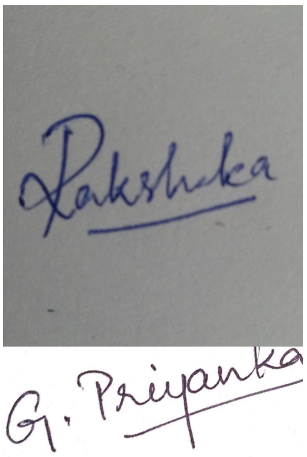
(d) Name, address, and nationality of the publisher, if any, of the original work : N.A.

(e) Name, address, and nationality of the publisher, or adaptation including the name, address and nationality of party authorizing : N.A.

3. Remarks, if any

Place:
Date: **16/10/2024**

For : RAKSHAKA SELVAN



Proprietor

Copyright Office
Government of India

Acknowledgement Slip
(Date: 16/10/2024)

Diary Number: **32373/2024-CO/L**	Form Received: **Online**
Copyright Reg. of: **Literary/ Dramatic**	Titled: **HIV DETECTION USING DEEP LEARNING**

Communication Address

Name	Address	Phone Number
RAKSHAKA SELVAN	BANGALORE TRUNK ROAD, VARADHARAJAPURAM, POONAMALLEE, CHENNAI – 600 123-600107	8940876362

Financial Details

Payment ID	Amount	Bank Name	Payment Mode	Payment Date
370392	500			16/10/2024

* For future communication please mention this DIARY No.

INSTRUCTIONS

For the purpose of processing the application, following documents are mandatory to send by post along with the acknowledgement slip(Office Copy).

1.	2 Copies of work
2.	DD/IPO of Rs.(as applicable) per work favouring Registrar Copyright Office payable at New Delhi (Not applicable for online payment)
3.	Authorization from author/publisher
4.	If the work is being used on goods or capable of being used on the goods
5.	If the application is being field through attorney, a specific power of attorney in original duly signed by the applicant and accepted by the attorney
6.	Search Certificate from Trade Mark Office(TM-60) (*Only in case of Artistic work*).
7.	Applicant must take a print out of the application, sign it and send along with the other documents.

Kindly send the above documents within 30 Days from the date of online submission on the following address given by herewith: .

Office of the Registrar of Copyrights
Copyright Office, Department for Promotion of Industry & Internal Trade
Ministry of Commerce and Industry
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110078
Email Address: copyright@nic.in
Telephone No.: 011-28032496

APPLICANT'S COPY
THANK YOU

Copyright Office
Government of India

सत्यमेव जयते

Acknowledgement Slip
(Date:16/10/2024)

Diary Number: **32373/2024-CO/L**	Form Received: **Online**
Copyright Reg. of: **Literary/ Dramatic**	Titled: **HIV DETECTION USING DEEP LEARNING**

Communication Address

Name	Address	Phone Number
RAKSHAKA SELVAN	BANGALORE TRUNK ROAD, VARADHARAJAPURAM, POONAMALLEE, CHENNAI – 600 123-600107	8940876362

Financial Details

Payment ID	Amount	Bank Name	Payment Mode	Payment Date
370392	500			16/10/2024

* For future communication please mention this DIARY No.

INSTRUCTIONS

For the purpose of processing the application, following documents are mandatory to send by post along with this acknowledgement slip(Applicant's Copy).

1.	2 Copies of work
2.	DD/IPO of Rs.(as applicable) per work favouring Registrar Copyright Office payable at New Delhi (Not applicable for online payment)
3.	Authorization from author/publisher
4.	If the work is being used on goods or capable of being used on the goods
5.	If the application is being field through attorney, a specific power of attorney in original duly signed by the applicant and accepted by the attorney
6.	Search Certificate from Trade Mark Office(TM-60) (*Only in case of Artistic work*).
7.	Applicant must take a print out of the application, sign it and send along with the other documents.

Kindly send the above documents within 30 Days from the date of online submission on the following address given by herewith: .

Office of the Registrar of Copyrights
Copyright Office, Department for Promotion of Industry & Internal Trade
Ministry of Commerce and Industry
Boudhik Sampada Bhawan,
Plot No. 32, Sector 14, Dwarka,
New Delhi-110078
Email Address: copyright@nic.in
Telephone No.: 011-28032496

OFFICE COPY
THANK YOU