**AI DENGUE DETECTION USING CBC DATA**

**CERTIFICATE**

**DECLARATION**

**ACKNOWLEDGEMENT**

**ABSTRACT**

Dengue fever is a tropical mosquito-transmitted disease spread through the Aedes mosquito, where the human body works as the primary host. Each year, densely populated countries such as Bangladesh, Thailand, and India, particularly in the Southeast Asian region, experience the majority of dengue outbreaks worldwide. Notably, in 2023, Bangladesh endured an unprecedented dengue outbreak, registering the highest number of cases in over two decades since 2000. This research aims to facilitate early detection of dengue from patients’ complete blood count (CBC) medical laboratory reports collected from two hospitals in Dhaka, Bangladesh. The custom-built dataset, comprising 320 samples and 14 hematology features, is used to evaluate diverse artificial intelligence techniques. This dataset documents suspected dengue cases in Bangladesh from May 2023 to October 2023, reflecting a significant outbreak period, including a gender distribution ratio of 5:3 male to female patients. Various preprocessing steps, handling missing values and outliers, one-hot encoding, synthetic oversampling, and removing redundant features, are applied to the employed dataset. Five feature selection methods and diverse machine learning algorithms, along with ensemble learning and transformer-based models, are implemented. The stacking ensemble classifier achieved the highest performance, with an accuracy of 96.88% and an F1 score of 0.9646. The stacking technique has been built using the LightGBM meta-classifier and XGBoost, Logistic Regression, and Multilayer Perceptron base learners.

INDEX TERMS Complete blood count, dengue prediction, explainable AI, feature selection, machine learning, ensemble learning, transformer model.

**CONTENTS**

1. **Certificate**
2. **Declaration**
3. **Acknowledgment**
4. **Abstract**
5. **List of Figures**
6. **List of Tables**
7. **List of Abbreviations**

**Chapter 1: Introduction**

* 1.1 Overview of the Domain Chosen
* 1.1.1 Artificial Intelligence in Healthcare
* 1.1.2 Role of CBC Data in Dengue Diagnosis
* 1.2 Objective
* 1.3 Problem Formulation
* 1.4 Scope
* 1.5 Limitations / Feasibility
* 1.5.1 Data Availability and Quality
* 1.5.2 Computational Requirements
* 1.6 System Requirements
* 1.6.1 Software Requirements
* 1.6.2 Hardware Requirements

**Chapter 2: Literature Survey**

* 2.1 Objective of the Literature Survey
* 2.1.1 Understanding Existing Authentication Methods
* 2.1.2 Fingerprint Authentication in Banking
* 2.2 Importance of the Literature Survey
* 2.3 Literature Review
* 2.3.1 Traditional ATM Security Systems
* 2.3.2 Fingerprint-Based ATM Systems
* 2.4 Research Gaps
* 2.5 Selected Literature Sources

**Chapter 3: Project Objectives**

* 3.1 Introduction
  + 3.1.1 Background of the Study
  + 3.1.2 Research Problem and Identified Gaps
* 3.2 Project Objectives
* 3.3 Project Justification

**Chapter 4: Methodology**

* 4.1 Introduction
  + 4.1.1 Justification for AI-Based Approach
  + 4.1.2 Overview of Key Methodology Components
* 4.2 Proposed Methodology
* 4.3 Proposed Algorithm (Pseudocode & Implementation)
* 4.4 Methodology Design (Flowchart/Dataflow Diagram)

**Chapter 5: Results**

**Chapter 6: Conclusion**

**References**

**Appendix**

**LIST OF FIGURES**

1. **Figure 1**: Loading Required Libraries for Dengue Detection Model
2. **Figure 2**: Reading the CBC Dataset (CBC Report.csv)
3. **Figure 3**: Data Preprocessing (Handling Null Values, Dropping Duplicates)
4. **Figure 4**: Data Visualization (Dengue Test Distribution by Gender)
5. **Figure 5**: Dataset Splitting (Training and Testing Data - 80:20 Ratio)
6. **Figure 6**: Random Forest Model Performance (Accuracy, Precision, Recall, F1-Score)
7. **Figure 7**: Decision Tree Model Performance (Highest Accuracy Among All Models)
8. **Figure 8**: MLP Classifier Performance (Lowest Accuracy Compared to Other Models)
9. **Figure 9**: Comparative Performance of All Models in a Bar Graph
10. **Figure 10**: Dengue Detection Input (WBC, RBC, Hemoglobin, etc.)
11. **Figure 11**: Dengue Test Prediction Result (Dengue Positive/Negative)

**LIST OF ABBREVIATIONS**

* **AI** – Artificial Intelligence
* **ANN** – Artificial Neural Network
* **AUC** – Area Under the Curve
* **CBC** – Complete Blood Count
* **CNN** – Convolutional Neural Network
* **GRU** – Gated Recurrent Unit
* **IgM/IgG** – Immunoglobulin M/Immunoglobulin G
* **KNN** – K-Nearest Neighbors
* **LIME** – Local Interpretable Model-agnostic Explanations
* **LSTM** – Long Short-Term Memory
* **ML** – Machine Learning
* **MLP** – Multilayer Perceptron
* **NS1** – Non-Structural Protein 1
* **PCR** – Polymerase Chain Reaction
* **RBC** – Red Blood Cell
* **SMOTE** – Synthetic Minority Oversampling Technique
* **SVM** – Support Vector Machine
* **TPU** – Tensor Processing Unit
* **WHO** – World Health Organization
* **WBC** – White Blood Cell

**CHAPTER 1**

**INTRODUCTION**

**1.1 Overview of the Domain Chosen**

Dengue fever is a rapidly spreading mosquito-borne viral disease that poses a significant public health threat, particularly in tropical and subtropical regions. The World Health Organization (WHO) reports millions of infections annually, with severe cases leading to hospitalization and fatalities. Early detection is critical to reducing mortality rates and improving patient outcomes.

With advancements in **Artificial Intelligence (AI) and Machine Learning (ML)**, healthcare analytics has seen significant improvements in disease prediction and diagnosis. AI-based models can analyze **Complete Blood Count (CBC) data** to detect early signs of dengue, allowing timely medical intervention.

**1.1.1 Artificial Intelligence in Healthcare**

AI has revolutionized healthcare by enabling automated diagnostics, predictive modeling, and personalized treatment recommendations. AI-driven models trained on hematological data can accurately classify infections, reducing dependence on expensive and time-consuming manual diagnostic methods.

**1.1.2 Role of CBC Data in Dengue Diagnosis**

CBC tests measure various blood components, including white blood cells, red blood cells, and platelet counts, which are key indicators of dengue infection. AI-based systems can analyze CBC parameters to distinguish between dengue and other febrile illnesses, ensuring early and precise diagnosis.

**1.2 Objective**

The objective of this study is to develop an accurate early detection model for dengue using complete blood count data from patients in Dhaka, Bangladesh. The primary objective of this study is to develop an **AI-based early detection system for dengue using CBC data**. By leveraging **machine learning and deep learning models**, the research aims to:

* **Improve early diagnosis** of dengue to reduce complications.
* **Enhance accuracy** in dengue detection using AI-based classification techniques.
* **Develop a cost-effective and scalable solution** for healthcare facilities, especially in low-resource settings.
* **Compare multiple AI models** to identify the most effective one for dengue classification.

**1.3 Problem Formulation**

Dengue diagnosis currently relies on **NS1 antigen tests, PCR tests, and IgM/IgG antibody tests**, which can be expensive and have varying sensitivity. The lack of early detection methods often leads to delayed treatment, increasing morbidity and mortality rates. The challenge is to create a **machine learning-based model** that can analyze **routine blood test data (CBC)** and provide an early and reliable indication of dengue infection.

Key challenges include:

* **Variability in blood test parameters** due to individual immune responses.
* **Imbalanced datasets**, as non-dengue cases outnumber dengue cases.
* **Ensuring real-time prediction capabilities** for practical clinical use.

**1.4 Scope**

The proposed AI-based system focuses on:

* **Analyzing hematological patterns** in CBC data to detect early dengue infection.
* **Developing an automated classification model** using machine learning techniques.
* **Comparing traditional diagnostic methods** with AI-based predictions.
* **Deploying the model in hospitals and diagnostic centers** for practical use.
* 1. **Data Analysis:** The study focuses on analyzing a custom dataset of blood test results from dengue-suspected patients in Dhaka, Bangladesh, during a significant outbreak period.
* 2. **AI Techniques:** It explores and evaluates various machine learning algorithms, including ensemble learning and transformer-based models, for early detection of dengue.
* 3. **Feature Selection:** The research applies different feature selection methods to improve model performance by reducing redundancy and handling missing or outlier values.
* 4. **Accuracy Enhancement:** The scope includes optimizing models for high accuracy and F1 scores, aiming to develop a reliable tool for dengue diagnosis in resource-limited settings.

The research will primarily focus on dengue detection but can be extended to identify other vector-borne diseases in the future.

**1.5 Limitations / Feasibility**

**1.5.1 Data Availability and Quality**

* CBC datasets must be **comprehensive and well-annotated** to train AI models effectively.
* **Limited access to hospital records** may restrict model generalization.

**1.5.2 Computational Requirements**

* **Deep learning models require high computational power**, which may limit implementation in resource-constrained settings.
* **Real-time deployment challenges** in clinical environments due to data privacy concerns.

**1.6 System Requirements**

To implement the AI-based dengue detection system, the following requirements are considered:

**1.6.1 Software Requirements**

* **Programming Language:** Python (TensorFlow, PyTorch, Scikit-learn)
* **Development Environment:** Jupyter Notebook, Google Colab
* **Machine Learning Libraries:** Pandas, NumPy, Matplotlib, Seaborn
* **Database Management:** MySQL / Firebase
* **Web Framework (for deployment):** Flask / Django

**1.6.2 Hardware Requirements**

* **Processor:** Minimum Intel i5 / Ryzen 5 or higher
* **RAM:** At least 8GB (16GB recommended for deep learning models)
* **GPU:** NVIDIA GTX 1650 or higher (for deep learning model training)
* **Storage:** Minimum 100GB SSD (for dataset storage and processing)

**1.7 Introduction**

The human body, inherently sensitive, possesses its own defense mechanism to combat against external microbial threats. Nevertheless, humans frequently fall victim to viral or bacterial infections, resulting in diseases that are significantly lethal. Dengue fever, for instance, is a viral disease primarily transmitted to humans by the Aedes mosquito. Every year, millions across the globe suffer from dengue fever, with thousands falling victim to its consequences [1].

According to the World Health Organization (WHO) and the European Union, in 2023, over six million people in nearly 92 countries were affected by dengue fever. Bangladesh alone recorded more than 0.31 million cases and over 1,600 deaths from this hemorrhagic fever [2]. Dengue is most prevalent in urban or peri-urban areas within the tropical and subtropical regions of the world, attributed mainly to insufficient sanitation, haphazard development and unplanned urbanization [3].

According to the latest review by the WHO, the countries in the African, Southeast Asian, and Western Pacific regions have the highest incidence of dengue fever. Among the countries in the Southeast Asian region, Bangladesh recorded the highest number of dengue cases between June and October. The number of affected patients and fatalities due to dengue in 2023 was the highest in recent decades [4].

Hence, early, efficient, and rapid detection and response measures for this arboviral disease are crucial. The escalating trend in the number of affected individuals underscores the imperative for implementing appropriate preventive future measures to avert surpassing previous records in terms of both affected patients and fatalities [5]. While dengue symptoms primarily arise from the bite of the Aedes mosquito, the virus typically remains dormant for a period before becoming apparent.

Though not inherently fatal, dengue presents a range of debilitating symptoms similar to those of other diseases. Typically, individuals with dengue experience intense fever, excruciating bodily pain, nausea, loss of appetite, and various types of skin rashes. Despite the absence of specific symptoms in the initial two weeks post-infection, patients often experience a sudden deterioration in health [6].

Pathological tests usually reveal a decrease in platelets in the blood, indicating a critical condition. Dengue virus has four serotypes. When someone is infected with one serotype, the body develops long-term immunity to it. However, the consequences can be severe if they are subsequently infected with a different serotype. While dengue symptoms may not be severe initially, upon a second infection, dengue can lead to severe conditions like shock syndrome, internal bleeding, or multiple organ failure. Considering the recent surge in dengue infections, this study introduces artificial intelligence (AI) approaches that enable early detection of dengue by employing various critical hematologic features.

In this work, a private dataset has been collected from two local hospitals in Dhaka, Bangladesh. The dataset comprises complete blood count (CBC) data for 320 individuals, classified as dengue ‘positive’ or ‘negative,’ and 14 attributes. The dataset has been preprocessed employing diverse techniques. Various machine learning models have been applied, i.e., Logistic Regression, Random Forest, SVM, LightGBM, XGBoost, and stacking classifier. Additionally, we deployed five deep-learning models – MLP, ANN, CNN, Bi-LSTM, and GRU and two advanced transformer models, TabPFN and TabTransformer.

Hyperparameter tuning with GridSearchCV and Keras Tuner framework, and five feature selection methods have been employed to extract essential features. The pivotal role of various features in decision-making, mainly focusing on the interpretability of black-box models, is investigated using the LIME-based explainable AI approach.

The study offers several significant contributions, which can be summarized as follows: 1) A major contribution of this work is to present a private CBC hematology report-based dengue dataset comprising 320 samples and 14 characteristic features collected from two local hospitals in Dhaka, Bangladesh. 2) Stacking ensemble model constructed from LightGBM meta-classifier and XGBoost, Logistic Regression, and Multilayer Perceptron base learners has been applied. TabPFN and TabTransformer-based transformers and advanced deep learning models are implemented. 3) GridSearchCV and Keras Tuner are applied to tune the best hyperparameters of the applied machine learning and deep learning models.

Five feature selection methods have also been used to identify the most salient features. 4) Employing an explainable AI tool, LIME, this research shed light on the key features that significantly impact the most on predicting dengue positive and negative cases. 5) The novelty of this work is to apply explainable stacking ensemble and transformer-based AI models and investigate significant features employing a private blood test report-based dengue dataset.

**CHAPTER 2**

**LITEARTURE SURVEY**

## ****2.1 Objective of the Literature Survey****

The primary objective of this literature survey is to explore existing research on **AI-based dengue detection** using **Complete Blood Count (CBC) data**. This survey identifies:

* **The role of AI and machine learning models** in automating dengue diagnosis.
* **Comparison of existing AI models**, including traditional ML algorithms and advanced deep learning techniques.
* **Challenges in dataset quality, model accuracy, and real-world implementation** for dengue detection.

### ****2.1.1 Understanding AI-Based Dengue Diagnosis****

* AI has revolutionized **infectious disease diagnosis** by enabling **automated, data-driven detection models**.
* Machine learning techniques, such as **Random Forest, SVM, and XGBoost**, have been used for **blood test-based dengue prediction**.
* Deep learning models like **CNN, LSTM, and GRU** offer improved accuracy by capturing complex patterns in **CBC data**.

### ****2.1.2 Challenges in AI-Based Dengue Detection****

* **Limited availability of high-quality labeled datasets** for training ML/DL models.
* **High false-positive rates** due to overlapping symptoms with other febrile illnesses.
* **Computational constraints** in deploying AI-based solutions in resource-limited healthcare settings.

## ****2.2 Importance of the Literature Survey****

Dengue fever is a global public health concern, especially in tropical and densely populated regions. Early detection of dengue using **CBC data** can improve treatment outcomes and reduce mortality rates. This literature survey:

* **Examines the evolution of AI-based dengue detection models.**
* **Compares machine learning vs. deep learning techniques** for dengue classification.
* **Identifies research gaps** in model performance, dataset availability, and real-world implementation.

This survey is structured as follows:

1. **Review of existing AI models** used for dengue classification.
2. **Comparison of methodologies** applied in dengue detection research.
3. **Identification of gaps and limitations** in previous studies.

## ****2.3 Literature Review****

This section analyzes research papers focusing on **AI models for dengue detection**, including **CBC-based classification, blood smear image analysis, and environmental factor-based prediction**.

### ****2.3.1 Machine Learning Approaches for Dengue Prediction****

* **Sarma et al. (2020):** Used **Decision Tree** to classify dengue cases based on CBC parameters. Achieved **79% accuracy** but struggled with class imbalance.
* **Fernández et al. (2016):** Applied **Logistic Regression** on febrile illness data. Achieved **69.2% accuracy**, but sensitivity was higher at **86.2%**.
* **Mello-Román et al. (2019):** Developed an **ANN-based model** for hospital-admitted dengue patients. Achieved **96% accuracy** with strong sensitivity/specificity.

### ****2.3.2 Deep Learning and Transformer-Based Approaches****

* **Mayrose et al. (2023):** Used **CNN-based blood smear image analysis** for dengue detection. Achieved **95.74% accuracy** with a **0.96 F1-score**.
* **Abdualgalil et al. (2022):** Applied **Extra Tree Classifier** on clinical data, achieving **99.12% accuracy** and a **0.99 F1-score**.
* **Ong et al. (2023):** Combined **XGBoost with Boruta feature selection** for dengue transmission prediction using meteorological data. Achieved **81% accuracy**.

## ****2.4 Research Gaps****

Despite advancements in **AI-based dengue detection**, several limitations exist, highlighting the need for further improvements.

### ****2.4.1 Dataset Limitations and Class Imbalance Issues****

* Most studies use **small, region-specific datasets**, limiting **generalization to other populations**.
* Class imbalance in **dengue-positive vs. dengue-negative cases** affects **model training and prediction accuracy**.

### ****2.4.2 Need for Improved AI Models****

* Many models lack **real-time deployment capabilities** in clinical settings.
* Existing methods **fail to incorporate explainable AI (XAI) techniques** to make model decisions interpretable for healthcare professionals.

**2.5 Literature surveys**

In recent years, the advancement of AI has facilitated rapid and accurate diagnosis of various diseases through machine learning techniques. Machine learning enables the accurate identification of diseases such as diabetes, Parkinson’s, Alzheimer’s, cardiovascular diseases, ocular diseases, etc. Machine learning primarily involves training algorithms with new data and providing insights about patterns. As a result, with the assistance of this vast repository of data, precise disease identification becomes possible. To determine if a patient is suspected of having dengue, individuals typically visit the nearest hospital or clinic and undergo multiple tests, such as a CBC, IgM/IgG antibody test, NS1 antigen test, etc. Following blood collection, various pathological procedures are performed, and it generally takes many hours and costs a reasonable sum of money [7].

A specialized doctor then reviews the reports to assess the severity of dengue fever based on the results. These procedures can be intensive for the people of low-income countries like Bangladesh due to the scarcity of available specialized doctors and pathologists. As a result, many researchers are striving to make dengue prediction more efficient and cost-effective by presenting various approaches and ideas. This section below delivers a detailed overview of existing methods for automatically detecting dengue fever using CBC hematology samples, blood smear images, environmental factors, and other relevant parameters from recent articles.

Davi et al. [8] utilized the human genome data of 102 patients to predict dengue flavivirus using machine learning models. The authors investigated the patients at high risk of developing extreme phenotypes despite moderate symptoms. Among the applied machine learning algorithms, the ANN model demonstrated the best accuracy score of 86%, with a sensitivity of 98% by extracting features using SVM RFE.

Sarma and other researchers [9] designed an automatic dengue prediction model using machine learning algorithms based on the recent outbreak in Bangladesh. The researchers collected raw data from the patients from Dhaka and Chittagong, Bangladesh’s two largest and most densely populated cities. The decision tree algorithm achieved the highest accuracy of 79%. Fernàndez et al. [10] applied a logistic regression model to diagnose dengue fever based on the features of approximately 550 patients with febrile illness. The applied logic regression model attained 69.2% accuracy for the positive cases with 86.2% sensitivity and 0.66 AUC score.

Mayrose and researchers [11] demonstrated an automated dengue prediction model using several machine learning techniques and blood smear image samples based on the lymphocyte nucleus and platelets. The authors achieved the best performance using the SVM classifier with 95.74% accuracy and 0.96 F1 coefficient. Prome et al. [12] predicted the number of dengue cases in different areas of Bangladesh by employing machine learning approaches and a weather dataset. The SVM model attained the best performance with a mean absolute error of 3.865.

Mello-Román et al. [13] presented predictive models for dengue based on real patient data admitted to Paraguay’s health centers with dengue fever symptoms. The applied ANN attained the maximum accuracy of 96% with the highest sensitivity and specificity of 96% and 97%, respectively. Dey et al. [14] initiated to predict dengue cases based on 11 states’ data of Bangladesh. The authors empirically analyzed how environmental factors affect the rise and fall of dengue cases. The applied Support Vector Regression algorithm demonstrated the best results with an R2 score of 0.75. The Multiple Linear Regression algorithm illustrated an excellent performance with a 0.62 R2 coefficient.

Abdualgalil et al. [15] utilized clinical data from a local medical center of Yemen to predict dengue using efficient machine learning techniques. They implied five machine learning algorithms that performed efficiently on the utilized clinical data. The Extra Tree Classifier algorithm demonstrated the best performance with 99.12% accuracy and 0.99 F1 coefficient. Ong and his colleagues [16] depicted the transmission rate of dengue with meteorological data by comparing different machine learning algorithms. This study used multiple variables, algorithms, vector indices, and meteorological data. An ensemble machine learning algorithm, XGBoost, with the Boruta feature selection technique, achieved the highest accuracy (81%) and 0.815 AUC.

Chaw et al. [17] developed an AI-based automatic model that predicts if there is a chance of shock development among all dengue patients. They used physiological data from ill patients at the University of Malaya Medical Centre and trained the model based on these collected data. Among the applied machine learning models, the decision tree approach attained the maximum F1 score of 0.92 and 0.64 AUC.

Sarwar and his colleagues[18] introduced a model that can accurately predict the number of dengue-affected patients. The authors considered various environmental factors in Dhaka, including humidity, temperature, and rainfall, as these variables critically influence dengue outbreaks. After implementing statistical algorithms, the SVM algorithm achieved the highest R-squared coefficient of determination Akter et al. [19] conducted a comparative study to predict dengue fever in Dhaka city, evaluating the effectiveness of time series analysis and machine learning techniques. Based on the time series, the applied ARIMA model hypothesizes the forecasts of dengue outbreaks with a 15.29 mean absolute percentage error (MAPE).

The neural network model demonstrated superior performance, achieving the lowest MAPE of 1.15. Majeed et al. [20] executed various hybrid AI models to predict cases of dengue viral fever in five regions of Malaysia. Various hybrid LSTM models have been applied by combining stacked, temporal and spatial attention approaches. The spatial stacked attention with the LSTM technique demonstrated the best performance with the lowest RMSE of 3.17. It can be understood from the reviews of the related articles that significant works have been initiated on automatic dengue prediction employing advanced machine learning and deep learning techniques. However, most of these works did not investigate the dengue virus’s significant clinical and environmental features. Few of these articles applied state-ofthe-art explainable AI techniques to interpret the AI model’s predictions.

**CHAPTER 3**

**PROJECT OBJECTIVES**

### 3.1 Introduction

This chapter outlines the objectives of the study, detailing how they contribute to addressing the research problem and filling gaps identified in the literature. The primary focus is to develop an artificial intelligence-based approach for the early detection of dengue using CBC data, ensuring higher accuracy and reliability compared to traditional diagnostic methods. The chapter further elaborates on the specific objectives and justifications for the research.

#### 3.1.1 Background of the Study

* Dengue fever is a rapidly spreading mosquito-borne disease, particularly in Southeast Asia.
* Early detection is crucial to reducing fatality rates, yet conventional diagnostic methods can be time-consuming and resource-intensive.
* AI-driven models can provide efficient and early diagnosis using CBC data, enabling timely medical intervention.

#### 3.1.2 Research Problem and Identified Gaps

* Current dengue detection methods rely on clinical symptoms or specialized lab tests, which may be delayed.
* Limited studies have explored AI-based detection using hematology data.
* Existing AI models often lack generalizability due to small datasets and imbalanced class distributions.

### 3.2 Project Objectives

The primary goal is to develop an AI-based model for the early detection of dengue using CBC data. Specific objectives include:

1. **Dataset Preparation & Preprocessing:** Collect and clean CBC data, handle missing values, perform feature selection, and apply synthetic oversampling techniques.
2. **Model Development:** Implement machine learning and deep learning models, including ensemble learning and transformer-based techniques.
3. **Performance Evaluation:** Compare different models using accuracy, F1-score, and other relevant metrics.
4. **Optimization and Generalization:** Improve model performance through hyperparameter tuning and cross-validation.
5. **Implementation and Validation:** Validate the best-performing model on real-world datasets and assess its feasibility for clinical use.

### 3.3 Project Justification

* **Addressing Research Gaps:** This study contributes by integrating AI with hematology-based dengue detection, an area with limited prior research.
* **Enhancing Early Diagnosis:** The proposed model offers a rapid, cost-effective, and scalable approach to detecting dengue in resource-limited settings.
* **Improving Predictive Accuracy:** The use of stacking ensemble techniques enhances classification performance compared to conventional models.
* **Practical Impact:** The findings can aid healthcare professionals in making data-driven diagnostic decisions, potentially reducing mortality rates.

Bottom of Form

**CHAPTER 4**

**METHODOLOGY**

This chapter presents the methodology adopted for developing an AI-based dengue detection system using CBC data. It explains the chosen approach, justifies its selection over alternative methods, and outlines a structured workflow. The methodology encompasses data preprocessing, feature selection, model training, evaluation, and optimization.

### ****4.1.1 Justification for AI-Based Approach****

* Traditional dengue diagnostic methods rely on clinical symptoms or serological tests, which can be time-consuming and costly.
* AI-driven models offer faster and more reliable predictions using routine CBC data, reducing the need for specialized tests.
* Machine learning algorithms, including ensemble methods and transformers, have shown superior performance in disease detection tasks.

### ****4.1.2 Overview of Key Methodology Components****

* **Dataset Collection & Preprocessing:** Handling missing values, outliers, feature selection, and data augmentation using synthetic oversampling.
* **Model Selection & Training:** Evaluating various AI techniques, including stacking ensemble models and deep learning approaches.
* **Performance Analysis:** Comparing models using key metrics like accuracy, precision, recall, and F1-score.
* **Optimization & Validation:** Hyperparameter tuning and cross-validation to improve generalizability.

## ****4.2 Proposed Methodology****

The proposed methodology consists of multiple phases:

1. **Data Collection:** CBC reports collected from hospitals in Dhaka.
2. **Data Preprocessing:** Cleaning, normalizing, and handling class imbalance.
3. **Feature Selection:** Five feature selection techniques applied to extract the most relevant features.
4. **Model Development:** Implementing stacking ensemble models using XGBoost, LightGBM, Logistic Regression, and MLP.
5. **Performance Evaluation:** Assessing models on accuracy, precision, recall, and F1-score.

## 

## ****4.3 Proposed Algorithm (Pseudocode & Implementation Details)****

The stacking ensemble method is proposed to enhance detection accuracy. Below is a simplified pseudocode representation:

### ****Pseudocode for Stacking Ensemble Model****

markdown

CopyEdit

Input: CBC dataset (X, Y)

Preprocessing:

- Handle missing values

- Normalize features

- Perform feature selection

- Apply synthetic oversampling

Train Base Models:

- Model1 = Train(XGBoost, X\_train, Y\_train)

- Model2 = Train(Logistic Regression, X\_train, Y\_train)

- Model3 = Train(MLP, X\_train, Y\_train)

Stacking Ensemble:

- Meta\_model = LightGBM

- Stacked\_predictions = Combine outputs of Model1, Model2, Model3

- Train Meta\_model on Stacked\_predictions

Evaluate:

- Compute Accuracy, Precision, Recall, F1-score

Return: Trained Stacking Model

### ****Implementation Details****

* **Feature Engineering:** One-hot encoding applied to categorical features.
* **Data Augmentation:** Synthetic oversampling used to balance dengue and non-dengue cases.
* **Hyperparameter Tuning:** Grid search applied for optimal parameters.

## ****4.4 Methodology Design (Dataflow Diagram)****

A structured flowchart is used to illustrate the step-by-step methodology:

### 

### ****Explanation of the Flowchart:****

1. **Input Stage:** CBC data collection.
2. **Processing Stage:** Data cleaning, feature selection, and balancing.
3. **Training Stage:** AI models are trained using the processed dataset.
4. **Prediction Stage:** The stacking ensemble model predicts dengue presence.
5. **Evaluation Stage:** The model’s accuracy and effectiveness are analyzed

**CHAPTER 5**

**RESULT**

In Dengue detection using Machine learning algorithms, we have used Random forest, Decision tree and MLP classifier. By using python 3.7.0 and jupyter editor, code is executed.



Fig: Loading Libraries

First step is loading the important and required packages.

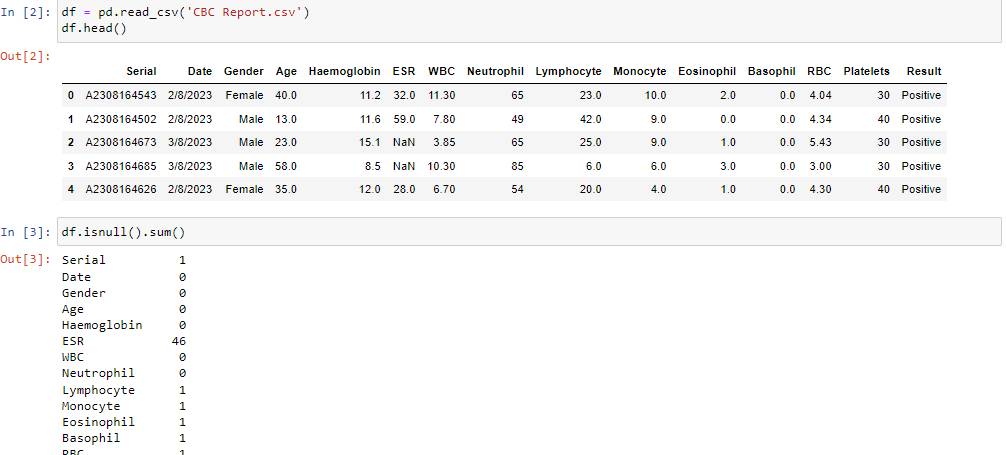


Fig: Read the dataset

‘CBC Report.csv’ is dataset is loaded and displaying the first five records of dataset. After that checking null values.

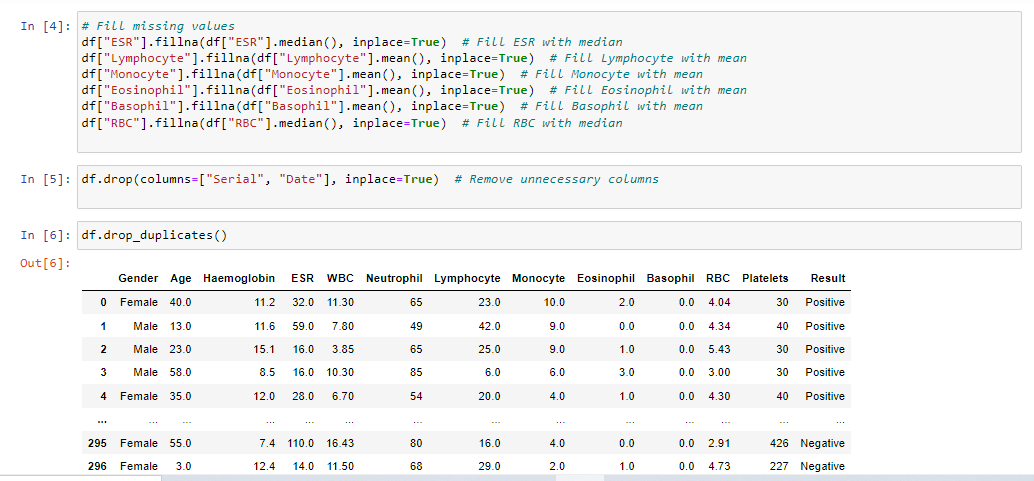


Fig: Data Preprocessing

Data processing means cleaning the data for training the model. Here dropping unnecessary columns , dropping duplicate values and filling missing values process is done.

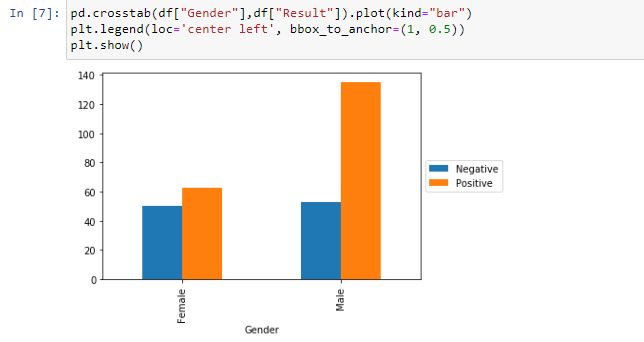


Fig: Data Visualization

Here we are checking the distribution of Dengue test with gender.

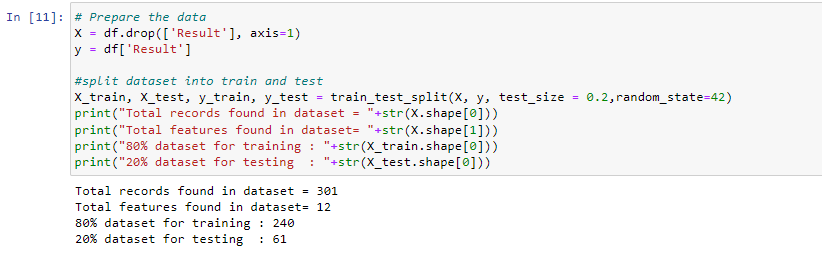


Fig: Dataset Splitting

Dataset is split into training and testing the data.

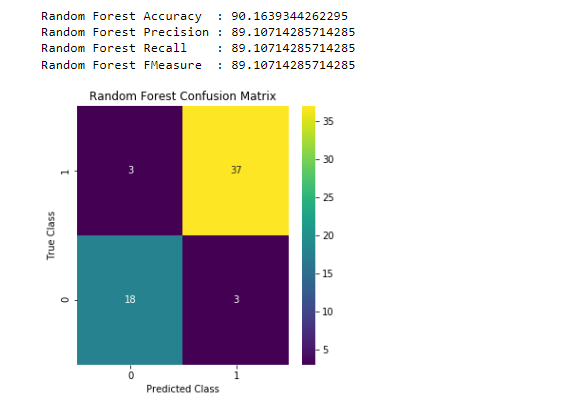


Fig: Random Forest Performance

Random forest is trained with 80% data and given the accuracy, precision ,recall and fscore. Confusion matrix is displaying true values and predicted values.

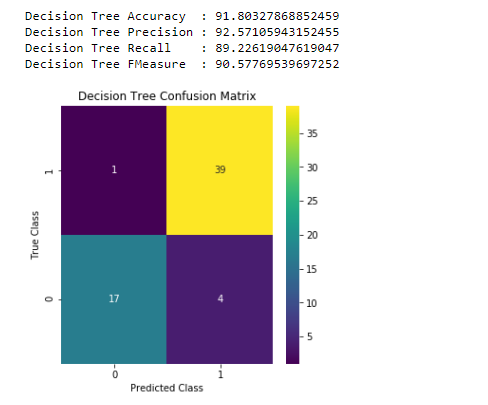


Fig: Decision Tree Performance

DT is giving highest accuracy among all three algorithms.

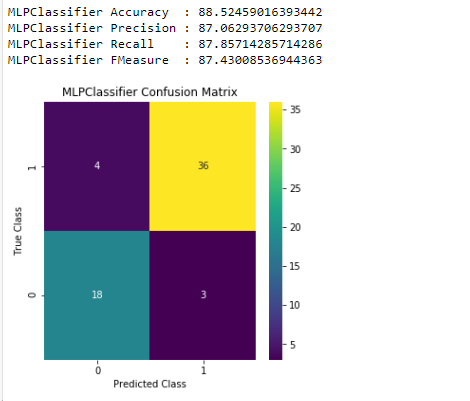


Fig: MLP Performance

MLP is giving lowest accuracy among all three algorithms so considered as existing method.

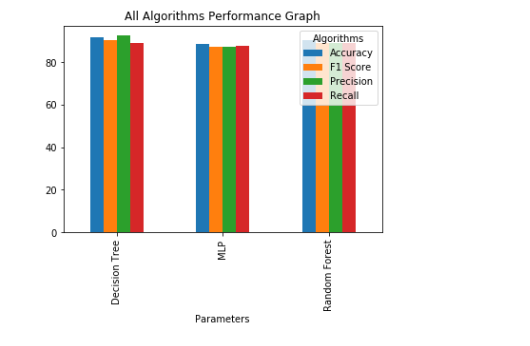


Fig: All algorithms performance in bar graph

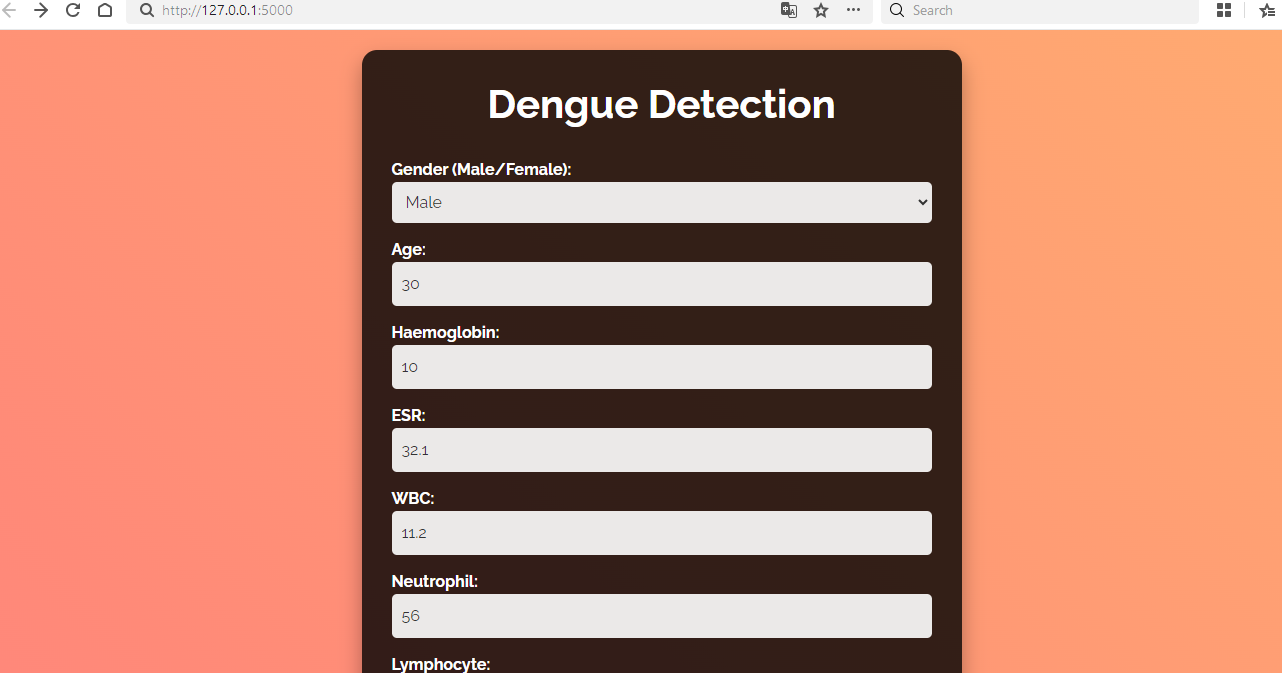
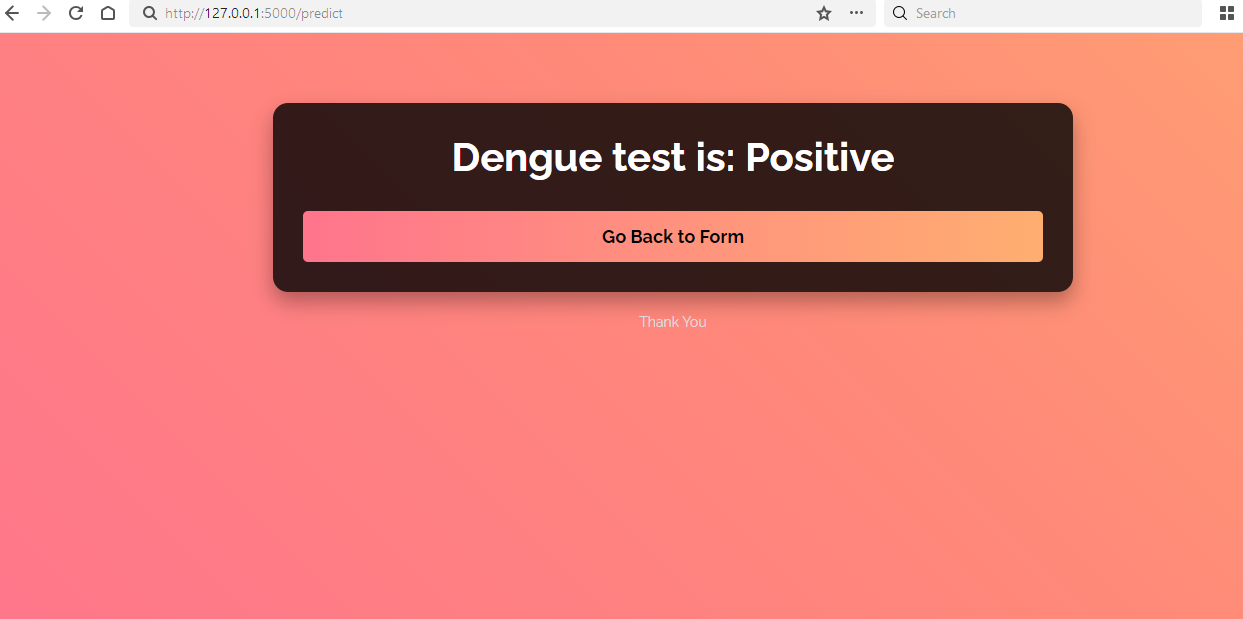


Fig: Dengue Detection

Entering the values for dengue test detection such as WBC, RBC, Haemoglobin,others



Fig; Dengue test prediction

After entering all values dengue test is predicted as positive.

**CHAPTER 6**

**CONCLUSION**

This research applies AI techniques to predict dengue infection using a dataset of 320 CBC samples and 14 hematology features from Dhaka, Bangladesh. Data preprocessing steps include handling missing values, one-hot encoding, synthetic oversampling, and feature standardization. Various machine learning, deep learning, and transformer-based models are tested, with hyperparameter optimization using GridSearchCV and Keras Tuner. The stacking ensemble model (LightGBM meta-classifier with XGBoost, Logistic Regression, and MLP base learners) performs the best. The MLP neural network is the top deep learning model. The LIME XAI approach interprets the model's predictions. Future work may include expanding the dataset and adding multimodal data. This research introduces various AI techniques to predict the dengue virus employing a private CBC report dataset. The dataset comprises 320 samples and 14 hematology features collected from local hospitals in Dhaka, Bangladesh. Diverse dataset preprocessing steps are implemented to the dataset, i.e., handling missing values and outliers, onehot encoding, feature standardization, synthetic oversampling, and removing redundant features. Various machine learning, deep learning and transformer-based models are applied to predict positive and negative dengue cases. The hyperparameters of the applied models are optimized by employing the GridSearchCV and Keras Tuner frameworks. A stacking ensemble approach constructed with LightGBM meta-classifier and XGBoost, Logistic Regression, and MLP base learners accomplishes the best performance among the machine learning models. The MLP neural network model performs best among the deep learning models. Finally, the LIME XAI approach has been applied to investigate the salient features and interpret the predictions provided by the stacking classifier. In the future, the employed dataset can be expanded by adding new data from a larger cohort of patients. Multimodal architecture can be applied using blood smear images for the same patient data. A multiclass problem can be defined using mild, moderate, severe positive, and negative dengue case samples.

**REFERENCES**

[1] M. B. Khan, Z.-S. Yang, C.-Y. Lin, M.-C. Hsu, A. N. Urbina, W. Assavalapsakul, W.-H. Wang, Y.-H. Chen, and S.-F. Wang, ‘‘Dengue overview: An updated systemic review,’’ J. Infection Public Health, vol. 16, no. 10, pp. 1625–1642, Oct. 2023.

[2] N. Ali, ‘‘The recent burden of dengue infection in bangladesh: A serious public health issue,’’ J. Infection Public Health, vol. 17, no. 2, pp. 226–228, Feb. 2024.

[3] S. Roy, A. Biswas, M. T. A. Shawon, S. Akter, and M. M. Rahman, ‘‘Land use and meteorological influences on dengue transmission dynamics in Dhaka city, Bangladesh,’’ Bull. Nat. Res. Centre, vol. 48, no. 1, pp. 1–24, Mar. 2024, doi: 10.1186/S42269-024-01188-0.

[4] N. Sharif, N. Sharif, A. Khan, and S. K. Dey, ‘‘The epidemiologic and clinical characteristics of the 2023 dengue outbreak in Bangladesh,’’ Open Forum Infectious Diseases, vol. 11, no. 2, pp. 1–29, Feb. 2024, doi: 10.1093/OFID/OFAE066. 112366 VOLUME 12, 2024 N. J. Riya et al.: Artificial Intelligence-Based Early Detection of Dengue Using CBC Data

[5] M. E. H. Kayesh, I. Khalil, M. Kohara, and K. Tsukiyama-Kohara, ‘‘Increasing dengue burden and severe dengue risk in Bangladesh: An overview,’’ Tropical Med. Infectious Disease, vol. 8, no. 1, p. 32, Jan. 2023.

[6] D. C. Kajeguka, F. M. Mponela, E. Mkumbo, A. N. Kaaya, D. Lasway, R. D. Kaaya, M. Alifrangis, E. Elanga-Ndille, B. T. Mmbaga, and R. Kavishe, ‘‘Prevalence and associated factors of dengue virus circulation in the rural community, Handeni district in Tanga, Tanzania,’’ J. Tropical Med., vol. 2023, pp. 1–9, Nov. 2023.

[7] M. A. Kabir, H. Zilouchian, M. A. Younas, and W. Asghar, ‘‘Dengue detection: Advances in diagnostic tools from conventional technology to point of care,’’ Biosensors, vol. 11, no. 7, p. 206, Jun. 2021.

[8] C. Davi, A. Pastor, T. Oliveira, F. B. d. L. Neto, U. Braga-Neto, A. W. Bigham, M. Bamshad, E. T. A. Marques, and B. Acioli-Santos, ‘‘Severe dengue prognosis using human genome data and machine learning,’’ IEEE Trans. Biomed. Eng., vol. 66, no. 10, pp. 2861–2868, Oct. 2019.

[9] D. Sarma, S. Hossain, T. Mittra, Md. A. M. Bhuiya, I. Saha, and R. Chakma, ‘‘Dengue prediction using machine learning algorithms,’’ in Proc. IEEE 8th R10 Humanitarian Technol. Conf., Dec. 2020, pp. 1–6.

[10] E. Fernández, M. Smieja, S. D. Walter, and M. Loeb, ‘‘A predictive model to differentiate dengue from other febrile illness,’’ BMC Infectious Diseases, vol. 16, no. 1, pp. 1–7, Dec. 2016.

[11] H. Mayrose, G. M. Bairy, N. Sampathila, S. Belurkar, and K. Saravu, ‘‘Machine learning-based detection of dengue from blood smear images utilizing platelet and lymphocyte characteristics,’’ Diagnostics, vol. 13, no. 2, p. 220, Jan. 2023.

[12] S. Sabrina Prome, T. Basak, T. Islam Plabon, and R. Khan, ‘‘Prediction of dengue cases in Bangladesh using explainable machine learning approach,’’ in Proc. Int. Conf. Inventive Comput. Technol. (ICICT), Apr. 2024, pp. 1–5.

[13] J. D. Mello-Román, J. C. Mello-Román, S. Gómez-Guerrero, and M. García-Torres, ‘‘Predictive models for the medical diagnosis of dengue: A case study in Paraguay,’’ Comput. Math. Methods Med., vol. 2019, pp. 1–7, Jul. 2019.

[14] S. K. Dey, M. M. Rahman, A. Howlader, U. R. Siddiqi, K. M. M. Uddin, R. Borhan, and E. U. Rahman, ‘‘Prediction of dengue incidents using hospitalized patients, metrological and socio-economic data in Bangladesh: A machine learning approach,’’ PLoS One, vol. 17, no. 7, Jul. 2022, Art. no. e0270933.

[15] B. Abdualgalil, S. Abraham, and W. M. Ismael, ‘‘Early diagnosis for dengue disease prediction using efficient machine learning techniques based on clinical data,’’ J. Robot. Control (JRC), vol. 3, no. 3, pp. 257–268, May 2022.

[16] S. Q. Ong, P. Isawasan, A. M. M. Ngesom, H. Shahar, A. M. Lasim, and G. Nair, ‘‘Predicting dengue transmission rates by comparing different machine learning models with vector indices and meteorological data,’’ Sci. Rep., vol. 13, no. 1, pp. 1–10, Nov. 2023.

[17] J. K. Chaw, S. H. Chaw, C. H. Quah, S. Sahrani, M. C. Ang, Y. Zhao, and T. T. Ting, ‘‘A predictive analytics model using machine learning algorithms to estimate the risk of shock development among dengue patients,’’ Healthcare Anal., vol. 5, pp. 1–17, Jul. 2024. [18] M. T. Sarwar and M. A. Mamun, ‘‘Prediction of dengue using machine learning algorithms: Case study Dhaka,’’ in Proc. 4th Int. Conf. Electr., Comput. Telecommun. Eng. (ICECTE), Dec. 2022, pp. 1–6.

[19] T. Akter, M. T. Islam, M. F. Hossain, and M. S. Ullah, ‘‘A comparative study between time series and machine learning technique to predict dengue fever in Dhaka city,’’ Discrete Dyn. Nature Soc., vol. 2024, pp. 1–12, May 2024.

[20] M. A. Majeed, H. Z. M. Shafri, Z. Zulkafli, and A. Wayayok, ‘‘A deep learning approach for dengue fever prediction in Malaysia using LSTM with spatial attention,’’ Int. J. Environ. Res. Public Health, vol. 20, no. 5, p. 4130, Feb. 2023.

**APPENDIX**

**PYTHON**

**1.1 Introduction**

\* One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

\*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

\* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

\* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

* Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
* Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
* Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
* Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
* Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
* Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
* Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
* Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
* Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

**1.2 Installation**

To install Python on your computer, follow these basic steps:

* Step 1: Visit the Python website Go to the official Python website at <https://www.python.org/>.
* Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
* Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
* Step 4: Install Python 3.7.0 using Cmd.
* Step 5: Install the all libraries that required to run the project
* Step 6: Run

**1.3 Python Features:**

1. **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2. **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3. **Open Source:** Python is a free online programming language since it is open-source.
4. **Portable:** Python is portable because the same code may be used on several computer standard
5. **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6. **GUI:** It stands for GUI (Graphical User Interface)
7. **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

**1.4 Python GUI (Tkinter)**

* Python provides a wide range of options for GUI development (Graphical User Interfaces).
* Tkinter, the most widely used GUI technique, is used for all of them.
* The Tk GUI toolkit offered by Python is used with the conventional Python interface.
* Tkinter is the easiest and quickest way to write Python GUI programs.
* Using Tkinter, creating a GUI is simple.
* A part of Python's built-in library is Tkinter. The GUI programs were created.
* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

**1.5 Python IDLE**

* Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
* Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
* IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
* IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
* The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
* Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

* Open the command prompt (Windows) or terminal (macOS/Linux).
* Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
* Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

* Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
* Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
* Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
* Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
* Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
* Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
* Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

**1.6 Libraries**

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

**Libraries in Python offer various advantages:**

* Code Reusability:
* Efficiency:
* Collaboration
* Domain-Specific Functionality
* To use a Python library, you need to install it first.

There are some libraries following:

* **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

* It has functions for analysing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
* Pandas allow us to analyse big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

* **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
* NumPy stands for Numerical Python.
* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
* The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.
* **Matplotlib:**

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

* Plotting Functions
* Customization Options
* Multiple Interfaces
* Integration with NumPy and pandas
* Subplots and Figures:
* Saving and Exporting
* **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

* Easy-to-Use Interface:
* Broad Range of Algorithms:
* Data Pre-processing and Feature Engineering:
* Model Evaluation and Validation:
* Integration with NumPy and pandas:
* Robust Documentation and Community Support:
* **Keras:**

\* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

* User-Friendly API
* Multi-backend Support
* Wide Range of Neural Network Architectures
* Pre-trained Models and Transfer Learning:
* Easy Model Training and Evaluation:
* GPU Support:
* **h5py:**

\* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

* + HDF5 File Access
  + Dataset Handling:
  + Group Organization:
  + Attributes:
  + Compatibility with NumPy
  + Performance
* **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

* Neural Network Framework:
* Computational Graphs
* Automatic Differentiation
* GPU and TPU Support
* Distributed Computing
* Deployment Capabilities
* **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

* Cross-Platform Compatibility
* Simple and Easy-to-Use
* Widgets and Layout Management
* Event-Driven Programming
* Customization and Styling
* Integration with Other Libraries
* **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

* Text Processing
* Part-of-Speech Tagging
* Named Entity Recognition
* Chunking and Parsing
* Sentiment Analysis:
* WordNet Integration:
* **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

Here are some key features and functionalities of SciPy:

* Numerical Integration:
* Optimization and Root Finding
* Linear Algebra
* Signal and Image Processing
* Statistics