# DENGUE FEVER ANTICIPATION: INSIGHTS FROM ADVANCED ML AND DEEP LEARNING MODELS

**BY**

**A. K. M FAZLUL KOBIR SIAM**
**ID: 201-15-3432**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

**Md. Aynul Hasan Nahid**
Lecturer
Department of CSE
Daffodil International University


**DAFFODIL INTERNATIONAL UNIVERSITY**

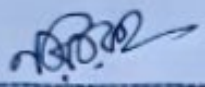
**DHAKA, BANGLADESH**

**JANUARY 2024**

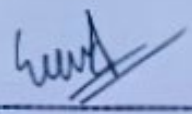
## APPROVAL

The project named Dengue Fever Anticipation: Insights from Advanced ML and Deep Learning Models,' presented by **A. K. M FAZLUL KOBIR SIAM**, ID: **201-15-3432**, to the Department of Computer Science and Engineering at Daffodil International University, is deemed acceptable for the partial fulfillment of the B.Sc. degree in Computer Science and Engineering. It has been approved for its style and content. The presentation has been held on 24 January 2024 at 9:30 AM.

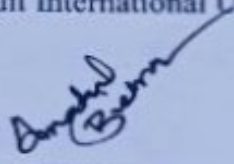
## BOARD OF EXAMINERS



**Narayan Ranjan Chakraborty** — **Chairman**
**Associate Professor & Associate Head**
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



**Md. Sazzadur Ahamed** — **Internal Examiner**
**Assistant Professor**
Department of CSE
Faculty of Science & Information Technology
Daffodil International University



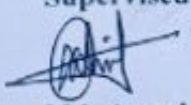
**Amatul Bushra Akhi** — **Internal Examiner**
**Assistant Professor**
Department of CSE
Faculty of Science & Information Technology
Daffodil International University

**Dr. Md. Zulfiker Mahmud** — **External Examiner**
**Associate Professor**
Department of CSE
Jagannath University

## DECLARATION

I declare that this project was completed under the guidance of **Md. Aynul Hasan Nahid, Lecturer**, Department of CSE, and Daffodil International University. I further affirm that neither this project nor any portion of it has been presented elsewhere for the attainment of any degree or diploma.

**Supervised by:**



**Md. Aynul Hasan Nahid**
**Lecturer**
Department of CSE
Daffodil International University

**Submitted by:**

Siam

**A. K. M FAZLUL KOBIR SIAM**
ID: -201-15-3432
Department of CSE
Daffodil International University

# ACKNOWLEDGEMENT

I begin by extending my heartfelt gratitude to the Almighty for blessing me and enabling me to successfully complete my final year Research Based Project.

My sincere appreciation goes to **Md. Aynul Hasan Nahid** of the Department of Computer Science and Engineering at Daffodil International University, Dhaka. Their extensive knowledge and keen interest in the field of "Machine Learning" were instrumental in guiding us throughout this project. Their unwavering patience, scholarly guidance, continuous encouragement, dedicated supervision, constructive criticism, valuable advice, and thorough review of multiple drafts at every stage played a crucial role in the completion of this project.

I would also like to express my deepest thanks to Professor Dr. Sheak Rashed Haider Noori, the Head of the Department of CSE, for his invaluable assistance in bringing my project to fruition. My gratitude extends to all the faculty members and staff of the CSE department at Daffodil International University.

I am grateful to my fellow classmates at Daffodil International University, who engaged in discussions and provided support during the course of my project.

Lastly, I would like to acknowledge and show my utmost respect for the unwavering support and patience of my parents.

# ABSTRACT

Recently, there has been a significant increase in the prevalence of physical illnesses, including dengue disease, drawing considerable attention due to its impact on a large population. The severity of the illness can be better understood by analyzing differences between normal and affected diagnostic reports. With numerous studies focused on understanding dengue disease, there are promising opportunities for advancing diagnostic techniques. In this study, I used the utilization of algorithmic models for early identification and raising awareness of potential threats. My straightforward approach is suitable for predicting simple cases of dengue disease illness in real-world scenarios. I have collected the dataset from Jamalpur Sadar Hospital. I employed various classifiers, including Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Convolution Neural Network (CNN), Long Short-Term Memory Network (LSTM), Bi-Directional Long Short-Term Memory Network (BLSTM), Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), K-Nearest Classifier (KNN), Adaboost Classifier (ABC), Decision Tree (DT), Support Vector Machine (SVM), Quadratic Discriminant Analysis (QDA), Ridge Classifier (RC), Passive Aggressive (PA), Gaussian Naïve Bayes (GNB) and ensemble techniques. Notable results were achieved, with the Ridge Classifier (RC) standing out as the most accurate, achieving an impressive accuracy rate of 96%. I implemented hyperparameter tuning to optimize the performance of each classifier. Through an experimental investigation and a review of recent findings, I confirmed that the bagging classifier Ridge Classifier (RC) performed exceptionally well, accurately predicting dengue disease with an accuracy rate of 96%.



**Keywords: Dengue Disease, Algorithm, Model, Accuracy.**

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1

# INTRODUCTION

## 1.1 Introduction

Living with dengue condition marked by hormone deficiencies and performance deterioration, poses daily challenges. Early detection of this prevalent issue remains a critical concern, with timely diagnosis being paramount. Machine learning emerges as a promising tool for predicting dengue by analyzing a wealth of authorized health data and patient diagnostics. Our study delved into patient medical records to uncover crucial indicators of the condition, leveraging these findings to identify dengue. While collaborative efforts among academics have aimed to develop machine learning algorithms for this purpose, their methods have often proven unreliable. I propose an alternative approach to enhance illness prediction, differentiating between supervised learning, which relies on labeled data to generate outputs from inputs, and unsupervised learning, which uncovers hidden patterns and information using unlabeled data.

## 1.2 Motivation

Numerous academic institutions have embarked on creating machine learning algorithms aimed at disease identification within the human body, including conditions like dengue. However, it became evident that their methods lacked accuracy and smoothness in predicting dengue. In response, I propose my innovative approach to enhance the body's ability to forecast illnesses. These machine learning methods fall into two distinct categories, with supervised learning relying on labeled data to generate outputs from inputs through input-output pairings, while unsupervised learning leverages unlabeled data to uncover hidden patterns and information. My developed technique focuses on anticipating the onset of dengue disease in individuals with suspected or ongoing conditions, aiming to provide a more reliable solution.

## 1.3 Rationale of the Study

I have developed a prediction model to detect dengue in humans as a result of my study, a condition increasingly affecting my society. Recognizing the scarcity of diagnostic

resources and information, particularly in our economically challenged nation, where assessing symptoms and diagnosing dengue prove to be expensive, I have turned to machine learning as a potential solution.

### 1.4 Research Question

- What was the aim of this study?

- What is the probability of successful dengue detection for individuals, whether afflicted by the condition or not?

- What methods can be employed to anticipate onset about dengue?

- In what real-life scenarios can this research find application?

- What is the expected project timeline and progression?

### 1.5 Expected output

As the prevalence of dengue continues to rise, uncertainty surrounds its presence in individuals. By scrutinizing diagnostic reports, I offer a proactive approach to predict and identify this condition. My method not only aids in detecting dengue but also enhances decision-making and ensures accurate evaluation of outcomes. Furthermore, it has the potential to measure life satisfaction and address associated issues while simultaneously increasing public awareness of dengue. The efficiency of my model allows for rapid assessment of the condition, providing a valuable tool in addressing this health concern.

### 1.6 Project Management and Finance

Project management and finance are intricately linked, playing a vital role in the successful execution of initiatives. Effective project management involves planning, organizing, and overseeing resources to achieve specific goals within a specified timeframe. Financial considerations, such as budgeting, cost estimation, and resource allocation, are fundamental aspects of project management. A robust financial framework ensures that projects stay within budgetary constraints while delivering value. Project managers must have a keen understanding of financial principles to make informed decisions, manage risks, and optimize resource utilization. The synergy between project management and finance is essential for achieving project objectives while maintaining fiscal responsibility and overall project success.

## 1.7 Report Layout

The layout of the study encompasses the following key sections:

**Background Study:** Providing an in-depth exploration of the context and relevant research in the field of dengue.

**Research Methodology:** Detailing the approach, tools, and techniques used to conduct the study and develop the proposed model.

**Experimental Results and Discussion:** Presenting the findings, outcomes, and a comprehensive discussion of the research results.

**Summary, Conclusion, and Future Analysis:** Summarizing the key takeaways, drawing conclusions, and outlining potential directions for future research.

**References:** Citing the sources and literature used to support the study and its findings.

# CHAPTER 2

# BACKGROUND STUDY

## 2.1 Preliminaries

Machine learning methods play a pivotal role in identifying the distinct patterns of dengue disease architecture. My focus lies in the evaluative examination of patients' diagnostic reports within this domain. To accomplish this, I employ a range of techniques, including ANN, CNN, RNN, LSTM, BLSTM, SVM, GNB, RF, LR, GB, KN, ABC, RC, PA, QDA and DT. This section delves into the exploration of these machine learning models, drawing upon the collective research efforts of various experts in the field, as elaborated in the following.

## 2.2 Related Works

The application of machine learning (ML) in the detection of heart illness has shown promise. Because of their efficacy in this field, machine learning algorithms—which frequently use tree topologies for decision models—are implemented. Dhiman, Sohrab, et al. [1] presented a unique machine learning method for dengue fever prediction. A patient dataset was created by combining data from the patient's symptoms, medical history, and diagnostic report. The dataset was created using real-time raw data samples from different kinds of dengue fever patients treated at Chittagong Medical College Hospital and Dhaka Medical College Hospital in Bangladesh. The whole dataset was split into a 70:30 ratio, with 70% going toward training and 30% going toward testing, to make training and testing the model easier. The decision tree (DT) and random forest (RF) are two machine learning techniques used in the suggested classification model. The decision tree outperformed the random forest, with a median accuracy of 79%, according to the data.

Diagnostic recommendations were developed by Mayuna, Palakorn, et al. [2] to aid in the early diagnosis of severe dengue infections. It examined 302 patients' retrospective data that had been ICD-10-categorized. The guidelines were converted into a disease severity rating system by using multivariate analysis to find independent diagnostic factors of severe dengue infection. With a minimum score of more than 14, the derived

scores varied from 0 to 38.6. The study showed that a straightforward score sheet with prediction value for assessing the severity of dengue illness may be created by combining many diagnostic factors.

In [3], Padet, Atchara, et al. Dengue forecasting approaches such as the K-H model, SVM, and ANN are frequently employed to estimate incidence rates. The time series analytic techniques ARIMA and SARIMA are applied in epidemiological studies of dengue illness and other infectious illnesses. To enhance forecast accuracy in comparable climates, other factors are needed as present approaches do not account for other variables. This might reduce the ability to forecast epidemics.

Machine learning methods such two boosted decision trees, two-class Bayes point machines, multiclass decision forests, and boosted decision tree regression were employed by Md Habibur, Omar Faroque, and Farhana [4]. Surveys of those afflicted with or currently experiencing dengue fever provided the dataset. To gauge the machine learning model's performance, tenfold cross-validation is employed. Azure Machine Learning Studio is a tool for data evaluation and prediction. With a 95% accuracy rate, the research is the first of its kind to be based in Bangladesh and identify dengue illness. The purpose of this study is to increase knowledge about the illness and motivate appropriate action.

Dengue fever is a mosquito-borne illness that affects 3.9 billion people worldwide and is endemic in Malaysia, especially in Selangor, according to Nurul Azam, Yap bee wah, et al. [5]. In order to find the best machine learning model for epidemic prediction, researchers looked at five areas in Selangor, Malaysia. We used climate factors such as temperature, humidity, wind speed, and rainfall. The best prediction performance was displayed by the SVM model, which had 70.6 percent accuracy, 14% sensitivity, 95% specificity, and 56% precision. The most significant predictor was the week of the year. The study recommends that in the future, dengue prediction models should be enhanced or utilized with algorithms inspired by nature.

To find the best accurate model, A Siddiq, N Shukla, and B Pradhan [6] examined linear and nonlinear models. The Support Vector Classification model obtained 76% accuracy based on temperature and humidity, as well as DF instances from Jeddah, Saudi Arabia.

The accuracy of other models was 52%, 55%, and 57% for decision trees, random forests, and linear regression, respectively.

According to Sandeep Kumar, Arpita Nath, et al. [7], dengue fever is a serious pandemic in tropical and subtropical areas and is brought on by mosquitoes that feed on human blood. The study suggests a machine learning-based Dengue Fever Expert System (DFES-MLA) to accurately forecast the illness using symptomatic data. To address the unbalanced nature of the dengue dataset, the model employs four oversampling strategies, Decision Tree and Random Forest classifiers, and data preparation processes.

Qanita, Dalya, and Alanoud [8] evaluated and predicted the average per week of dengue fever episodes in San Juan and Iquitos using machine learning regression approaches. Performance is measured using the Mean Absolute Error (MAE), and the sources and treatments of the virus are predicted using the Poisson Regression Model.

According to Vipul, Nikhil, and Ajaymon [9], dengue brought on by Aedes Aegypti mosquitoes is a menace to international health. Climate and meteorological factors, including temperature, precipitation, and humidity, are positively connected with dengue cases. A unique double linear regression model was developed in a study on machine learning-based dengue forecasting, and it outperformed earlier models by 19.81 average absolute error.

According to Son T Mai and Ha T Phi [10], millions of individuals are impacted by dengue fever (DF), an infectious illness spread by mosquitoes that is on the rise. In order to forecast DF outbreaks for several regions and future time steps, the Proximity Ensemble (PT-Ensem) architecture is presented, with an emphasis on the impact of climate information on DF outbreaks. The event-to-event probabilistic framework, proximity graph, ensemble prediction method, data aggregation scheme, closeness propagation step, and temporal propagation step are the six main parts of PT-Ensem.

## 2.3 Comparative Analysis and Summary

Table 2.1. The comparison between existing and our proposed method

| SL | Authors | Algorithms | Best Accuracy |
|---|---|---|---|
| 1. | Dhiman, Sohrab, et al [1] | RF, DT | DT = 79% |
| 2. | Mayuna, palakorn, et al [2] | MLR | MLR = 90.2% |
| 3. | Md Habibur, Omar Faroque, Farhana [4] | DT, GNB, DF, DT | DT = 95% |
| 4. | Nurul azam, Yap bee wah, et al. [5] | SVM | SVM= 70% |
| 5. | A Siddiq, N Shukla, B pradhan[6] | SVC | SVM = 76% |
| 6. | Models Applied | ANN, RNN, CNN, LSTM, BLSTM, RF, LR, GB, KNN, AbC, DT, SVM, QDA, RC, PA, GNB. | Bagging RC = 96% |

## 2.4 Scope of the Problem

The task at hand revolved around streamlining and simplifying the diagnosis process for dengue. Given the extensive body of machine learning-related research associated with our proposed model, my primary objective was to maximize accuracy. Despite the limited room for refinement within the existing procedure, the concept was to implement user-friendly technology in order to reduce the frequency of dengue diagnoses, making the process more accessible and efficient.

## 2.5 Challenges

The material proved exceptionally user-friendly and immensely practical in our use. Upon completing the data collection phase, a meticulous manual examination of the dataset for any missing information becomes necessary.

# CHAPTER 3

# RESEARCH METHODOLOGY

## 3.1 Research Subject and Instrumentation

With my data set, I used a wide range of hybrid models and methods to optimize accuracy. Essential to my efforts was cutting-edge configuration tools complemented by top-tier GPUs, ensuring optimal performance. My toolkit incorporated the Python programming language, alongside associated tools like Jupyter Notebook, Google Colaboratory, and Anaconda. This suite of resources empowered us to seamlessly develop and execute Python code directly within the browser, enhancing the efficiency and versatility of our data analysis and model implementation.

## 3.2 Data Collection Procedure

I have taken the dataset from Jamalpur Sadar Hospital, comprising 250 rows and 18 columns. Among these columns, the diagnostic attribute played a pivotal role in categorizing the prevalence of dengue disease, while each individual trait proved crucial for identifying this condition. Patients were classified into two groups denoted by 0 and 1, representing the occurrence and absence of dengue, as depicted in Figure 3.1. The dataset was further divided into two segments: the training set and the test set. In the training set, 80% of the applicants were selected, while the remaining 20% constituted the test set, facilitating comprehensive model development and assessment for dengue prediction.

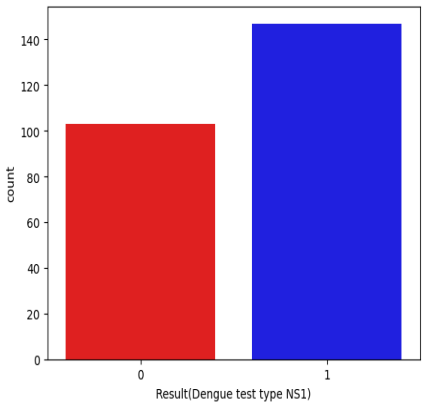



Figure 3.1: Target values

The dataset only included nominal values. and it was not without its share of missing or inaccurate data. For a comprehensive understanding of the dataset and its scope, please refer to Table 3.1, which provides a detailed breakdown of its contents.

Table 3.1: Details of the dataset

| Attributes | Types of Values | Reference Value |
|---|---|---|
| Age | Integer | Patient's age |
| Hemoglobin(g/dl) | Float | Male: 13-18, Female: 11.5-16.5, Child: 10-13 I: 8-10 g/dl |
| Lymphocytes(%) | Integer | Child: 52-62%, Adult: 20-50% |
| Monocytes(%) | Integer | Child: 3-7%, Adult: 2-10% |
| Eosinophils(%) | Integer | Child: 1-3%, Adult: 1-6% |
| RBC | Float | Male: 4.5-6.5, Female: 3.8-5.5 |
| HCT(%) | Float | M: 40-54%, F: 37-47% |
| MCV(fl) | Float | 76-94 |
| MCH(pg) | Float | 27-32 |
| MCHC(g/dl) | Float | 29-34 |
| RDW-CV(%) | Float | 10-16 |
| Total Platelet Count(/cumm) | Integer | 150000-450000 |
| MPV(fl) | Float | 7-11 |
| PDW(%) | Float | 10-18 |
| PCT(%) | Float | 0.1-0.2 |
| Total WBC count(/cumm) | Integer | A: 4000-11000 C: 5000-15000 I: 6000-18000 |
| Gender | Object | Patient's gender. |

| | | |
|---|---|---|
| Result (Dengue test type NS1) | Object | Positive or Negative. |

### 3.2.1 Categorical Data Encoding

It is a crucial process in our research, involves converting categorical data into numerical values. Given that machine learning relies on numeric input and output, the utilization of this technique was indispensable. To facilitate the application of the categorical data encoding method, I needed to transform the dataset columns, enabling the integration of categorical data into my analytical framework.

### 3.2.2 Feature Scaling

This entails scale of multiple data, which includes addressing less then zero. As part of this process, modifications were implemented to ensure consistent scaling. Additionally, a columns, Neutrophils(%) was deemed necessary for removal from the dataset.

## 3.3 Statistical Analysis

In section  also essential component projects, and in my case, it played a pivotal role in the development and evaluation of the algorithms employed. I opted to utilize a CSV file as the foundation for my dataset, necessitating several preparatory steps before it could be effectively used. These preparations encompassed various actions, including pre-processing and data collection. Within my model, I implemented a range of classifiers, namely ANN, CNN, RNN, LSTM, BLSTM, SVM, GNB, RF, LR, GB, KN, ABC, RC, PA, QDA and DT methods, to predict dengue disease. Each classifier was rigorously assessed for accuracy, yielding the following results: ANN got 86%, RNN 84%, CNN 94%, LSTM 74%, BLSTM 78%, RF 94%, LR 94%, GB 92%, KN 92%, ABC 92%, GNB 94%, SVM 94%, QDA 94%, RC 96%, PA 94% and DT 93%. To further enhance predictive accuracy, I employed ensemble techniques, where the Bagging classifier Ridge Classifier (RC) emerged as the frontrunner, bagging a remarkable accuracy rate of 96%, in boosting Logistic Regression (LR) got 94%. In addition, the Stacking Classifier achieved an accuracy of 92%, and the Voting Classifier hard voting VOTA demonstrated a respectable accuracy of 94% and Soft voting VOTB

got 90%. Through meticulous hyperparameter tuning, I fine-tuned the parameters of each classifier to maximize their predictive capabilities.

### 3.4 Proposed Methodology

**Flow chart:**

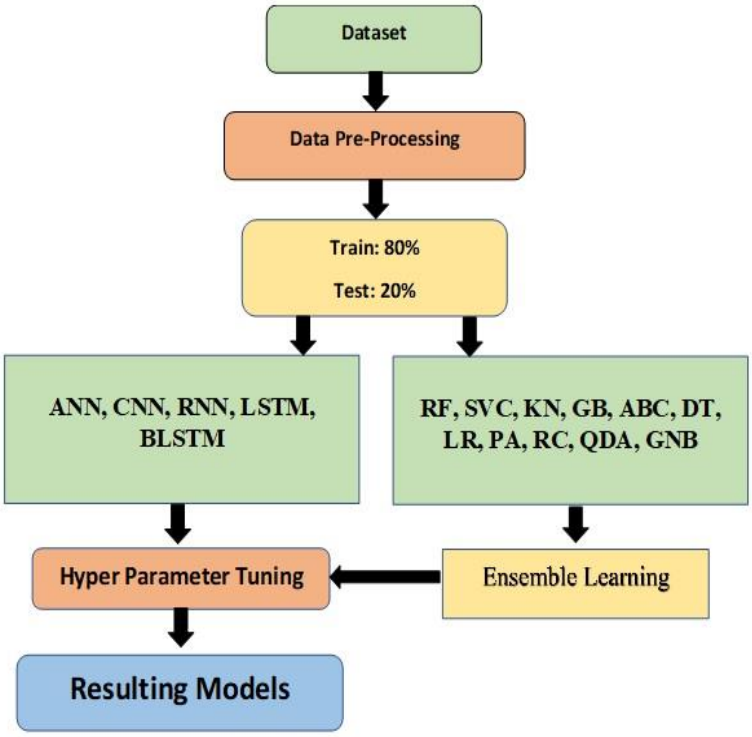



Figure 3.2: Methodology of Dengue Disease

I harnessed the power of a process diagram to predict dengue disease effectively. My initial steps revolved around the presentation of the training and testing datasets for my system, followed by the implementation of critical data cleaning methods like scaling, Categorical to Numeric conversion, and Feature Selection. The allocation of 80% -20% for training and testing ensured a robust evaluation process. Subsequently, I executed various deep learning and machine learning algorithms and meticulously assessed their results. To elevate my predictive accuracy to its maximum potential, I turned to ensemble algorithms, encompassing techniques like bagging, boosting, stacking, and voting. This allowed me to extract the most from the combined algorithms and derive results that were comprehensively analyzed. The models employed in this phase were subjected to outcome analysis to determine their effectiveness in predicting dengue

disease. The model, illustrated in Figure 3.2, encapsulated our research journey, offering insights into the most effective techniques employed in our study.

The intricate connections between two variables were explored in a correlation subplot, which illuminated how one variable's behavior shifted in response to changes in another. The degree of interdependence between variables played a crucial role in determining the likelihood of one factor being accurately predicted from another. This deepened understanding of the dataset has improved our ability to identify the key factors that influence dengue [9]. Figure 3.3 presented a comprehensive view of all the traits associated with the predicted property "Dengue Disease," shedding light on the interrelationships within the dataset and paving the way for more accurate predictions.

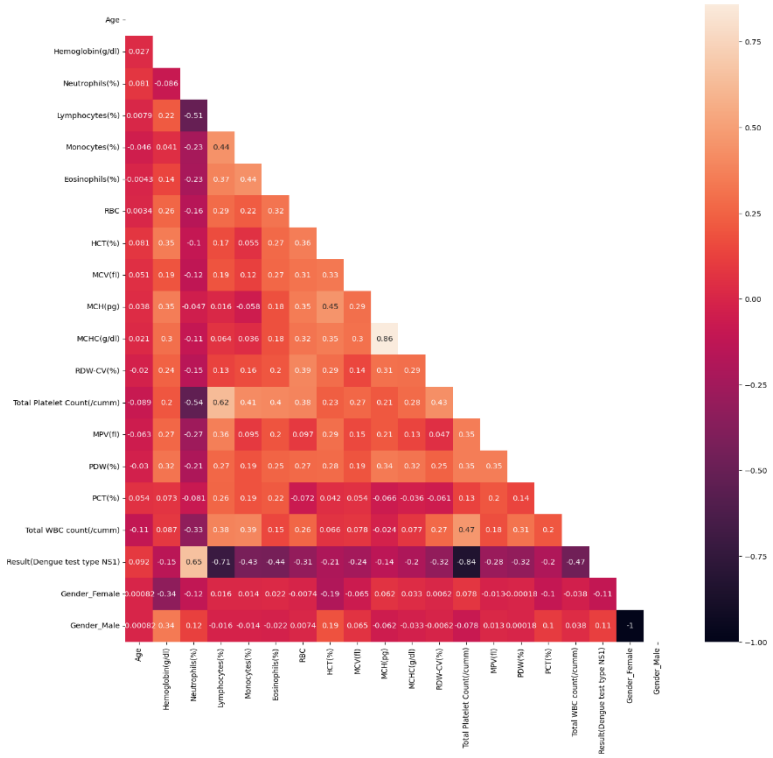



Figure 3.3: Correlated Features of Dengue Disease Dataset

## 3.5 Implementation Requirements

For apply models effectively, a reliable source of data is imperative. The initial step involves the meticulous cleaning of the dataset to ensure smooth operations. The dataset undergoes a comprehensive cleansing process employing various filtering methods,

culminating in a pristine dataset ready for analysis. Subsequently, vital data pre-processing methods are implemented, including the application of the Standard Scaler Transform for normalization and the conversion of categorical data into numerical format. The dataset is then partitioned into two subsets, with 80% allocated for training and the remaining 20% for rigorous testing. This rigorous testing involves the practical implementation of diverse deep learning and machine learning algorithms, which are meticulously evaluated to determine their predictive efficacy. To maximize forecast accuracy, we turn to ensemble algorithms, including the Voting ensemble, to combine the strengths of multiple algorithms and extract the most accurate predictions. The outputs generated by these ensemble algorithms undergo comprehensive assessment, and the results are further validated through the fine-tuning of hyperparameters to optimize their performance. Subsequently, the models employed in the process are subjected to a thorough outcome analysis to determine their effectiveness in predicting the target variable. Moving forward, the data analysis stage is essential to lay the foundation for the learning process. Model learning and the fitting of predictive techniques are integral components, paving the way for the next crucial step: model evaluation through voting. Also meticulous selection process ensures the most effective model is chosen for deployment, ultimately maximizing the model's performance and its utility in practical applications.

# CHAPTER 4

# Experimental results and discussion

## 4.1 Experimental Setup

In this paper, model was used as supervised. The technique was classification, where the algorithm learned patterns and relationships within the data. Subsequently, the trained model was applied to the testing dataset to predict outcomes or classify new instances. The specific deep learning and machine-learning algorithm employed in this study will be elaborated upon in the subsequent sections.

### 4.1.1 Classifier Algorithms

ANN, CNN, RNN, LSTM, BLSTM, SVM, GNB, RF, LR, GB, KN, ABC, RC, PA, QDA and DT was applied.

**Artificial Neural Network (ANN)**

Artificial Neural Networks (ANNs) are a fundamental component of machine learning, inspired by the intricate structure and functioning of the human brain. ANNs are versatile models that excel in tasks ranging from pattern recognition to complex decision-making. Comprising interconnected nodes, or artificial neurons, organized into layers, ANNs process information through weighted connections, mimicking the synaptic strengths in biological neural networks. In an ANN, the input layer receives data, and subsequent hidden layers transform this input using learned weights. The output layer then produces the final prediction or classification. What sets ANNs apart is their ability to adapt and learn from data. Through a process known as training, ANNs adjust the weights between neurons based on the provided data and the desired output. This adaptability enables ANNs to generalize patterns, unseen. DL, a subset of machine learning, has gained prominence with the development of deep neural networks, characterized by multiple hidden layers. This depth allows ANNs to automatically extract hierarchical features from data, making them powerful tools for tasks such as image and speech recognition. ANNs have proven effective in diverse domains, from natural language processing to medical diagnostics, showcasing their significance in advancing artificial intelligence and solving complex real-world problems.

**Recurrent Neural Network (RNN)**

Recurrent Neural Networks (RNNs) represent a category of artificial neural networks designed for processing sequential and temporal data. What sets RNNs apart from traditional feedforward neural networks is their unique ability to capture dependencies and patterns within sequences, making them particularly effective for tasks involving time-series data, natural language processing, and speech recognition. At the core of RNN architecture is the concept of recurrent connections, allowing information to persist within the network across different time steps. This recurrence enables RNNs to maintain a memory of previous inputs, making them well-suited for tasks where context and sequential relationships are crucial. Despite their conceptual strength, traditional RNNs suffer from challenges such as the vanishing gradient problem, limiting their ability to capture long-range dependencies effectively. To address this, variations like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been introduced. These variants incorporate sophisticated gating mechanisms, facilitating better information flow over extended sequences. RNNs find applications in diverse domains, ranging from natural language processing tasks like language modeling and machine translation to time-series analysis in finance and healthcare. While effective, the evolving landscape of neural network architectures continues to refine and extend the capabilities of RNNs, ensuring they remain instrumental in modeling sequential data and understanding temporal relationships.

**Convolutional Neural Network (CNN)**

Convolutional Neural Networks (CNNs) represent a class of deep learning models designed for processing structured grid data, particularly images. They have gained immense popularity for their remarkable success in image recognition, classification, and feature extraction tasks. Unlike traditional neural networks, CNNs are equipped with specialized layers, such as convolutional and pooling layers, which enable them to automatically and adaptively learn hierarchical representations of input data. The core innovation of CNNs lies in the convolutional layers, where filters or kernels systematically slide across input images, capturing local patterns and features. This spatial hierarchy allows CNNs to recognize complex patterns by learning low-level features in the initial layers and progressively combining them to form higher-level abstractions in subsequent layers. Pooling layers further contribute to translation

invariance by reducing spatial dimensions while retaining essential information. Their uses demonstrate the adaptability and effectiveness of CNN designs and go beyond image processing to domains like speech recognition and natural language processing. CNNs are a key component of contemporary computational developments, as seen by their effectiveness in pushing the frontiers of neural networks and artificial intelligence.

**Long Short-Term Memory (LSTM)**

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to address the challenge of learning and remembering long-term dependencies in sequential data. Introduced by Hochreiter and Schmidhuber in 1997, LSTMs have become a cornerstone in the field of deep learning, particularly for tasks involving sequential information, such as natural language processing and time series prediction. What sets LSTMs apart from traditional RNNs is their unique architecture, featuring memory cells with self-connected gates. These gates enable the network to regulate the flow of information, selectively remembering or forgetting past states, thereby mitigating the vanishing gradient problem associated with standard RNNs. The architecture includes input, forget, and output gates, allowing LSTMs to capture and retain relevant information over extended sequences. LSTMs excel in modeling temporal dependencies, making them well-suited for tasks where understanding context and capturing long-range dependencies is crucial. Their ability to effectively handle vanishing gradient issues has made LSTMs instrumental in diverse applications, from speech recognition to machine translation. With their capacity to retain contextual information over extended periods, LSTMs have significantly contributed to the advancement of deep learning models for sequential data analysis.

**Bidirectional Long Short-Term Memory (BLSTM)**

Bidirectional Long Short-Term Memory (BLSTM) is a sophisticated neural network architecture designed to capture intricate dependencies and patterns in sequential data, making it particularly effective in tasks involving time series or sequential informationBy processing incoming data in both forward and backward directions, the BLSTM network, a variation of the Long Short-Term Memory (LSTM) network, improves predictive capabilities. The key innovation lies in its bidirectional nature, enabling the model to consider past and future context simultaneously. BLSTM is

ideally suited for applications including speech recognition, natural language processing, and clinical time series analysis because of its bidirectional processing, which is essential for comprehending temporal links and interdependence within a sequence. By incorporating memory cells and gating mechanisms, BLSTM can effectively capture long-range dependencies in sequential data, mitigating issues like vanishing gradients that often hinder traditional recurrent neural networks. This bidirectional approach allows the model to learn from past and future context, improving its ability to predict and analyze sequential patterns. BLSTM has proven valuable in various domains where understanding the context of data over time is essential, making it a powerful tool in the realm of deep learning and sequential data analysis.

**Random Forest**

The Random Forest classifier is a powerful and versatile machine learning algorithm that has gained immense popularity for both classification and regression tasks. It works by building a collection of decision trees, each of which is built using a subset of the features that are accessible and a randomly selected portion of the initial training data. This technique introduces variability and decorrelates the individual trees, mitigating overfitting and improving the model's generalization performance. In classification, the Random Forest combines the results from these decision trees through a majority vote, while in regression, it computes the average of the individual tree predictions. One of the key advantages of Random Forest lies in its ability to handle high-dimensional data, maintain robustness against outliers, and provide feature importance for model interpretability. The algorithm is less prone to overfitting compared to single decision trees, thanks to its inherent bagging (Bootstrap Aggregating) and feature bagging components. In comparison to other methods, Random Forest is less susceptible to hyperparameter modification, making it especially helpful when working with complicated and noisy datasets. Additionally, the Random Forest can identify influential features and provide insights into their contribution to the model's predictive power. Its robust performance, scalability, and flexibility have made it a popular choice across various domains, including finance, healthcare, and image analysis. However, the trade-off for its power and versatility is increased computational cost and complexity, which can be a consideration for real-time or resource-constrained

applications. Nonetheless, the Random Forest remains a reliable workhorse in machine learning, delivering accurate predictions and valuable insights for diverse problem-solving scenarios [19].

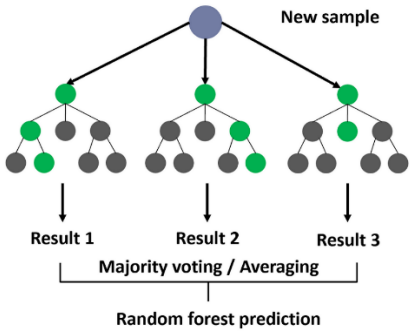



Figure 4.1: Random Forest

**Decision Tree**

In order to classify an instance, we first look at the characteristic that the tree node's base represents. Next, we follow an extension of the structure based on that feature's value. The Decision Tree technique, which just needs two number Classes, is one of the most effective and well-known prediction techniques. Each inner node of a decision tree, a structure of data with an ordered structure where every node in the leaf hierarchy denotes a distinct class, represents an attribute test. On the basis of decision trees, a tree structure known as DT is frequently utilized. The approach may be used to solve classification and regression issues. As the tree grows from the root node, the "splitting" procedure is utilized to select the "Best Features" or "Best Attributes" from the prospective characteristics pool. It is typical to compute two extra metrics, "Entropy," as indicated in (4.1), and "Data Gain," as mentioned in (4.2), in order to find the "Best Attribute" [15]. Entropy analyzes the consistency of a dataset, whereas collecting data measures the pace at changes that occur in the volatility of attributes. The notion is depicted in Fig. 4.2 below.

$$E(D) = -P\ (positive) log2P\ (positive) - P(negative)\log 2P\ (negative) \quad (4.1)$$

$$Gain\ (Attribute\ X) = Entropy\ (decision\ Attribute\ Y) - Entropy\ (X, Y) \quad (4.2)$$

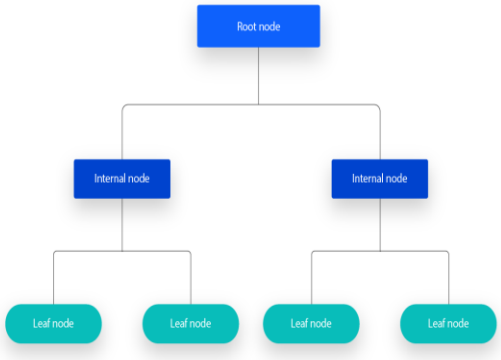



Figure 4.2: Decision Tree

**Naïve Bayes**

The term "GNB" refers to a group of Bayes' Theorem-based algorithms for classification that calculate the probability of an event happening given the probability that another event could also happen. Each algorithm in this group is predicated on the fundamental tenet that any two attributes being identified are unrelated to each other (equation 4.3). The concept is shown in Fig 4.3 below.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad (4.3)$$

The constant value is taken to represent a Gaussian distribution for every characteristic in Gaussian NB. The term “Normal distribution” is often used interchangeably w (i^th).

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma y^2}} \exp\left(-\frac{(x_i-\mu_y)^2}{2\pi\sigma y^2}\right) \qquad (4.4)$$

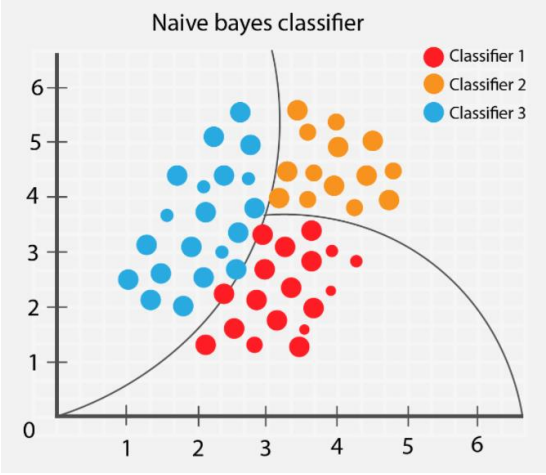



Figure 4.3: Gaussian NB Classifier

**Logistic Regression**

Logistic Regression is a widely utilized and interpretable machine learning classifier that excels in binary and multiclass classification tasks. Unlike linear regression, which predicts continuous values, logistic regression models the probability of an instance belonging to a particular class using the logistic function (sigmoid). It estimates the odds of an event occurring and maps them to a range between 0 and 1, allowing it to provide clear class separation. The model is trained by minimizing the logistic loss or cross-entropy loss through iterative optimization techniques like gradient descent. Logistic Regression is advantageous for its simplicity, quick training, and ease of interpretation. It can handle both linear and non-linear relationships between features and the target variable through polynomial or interaction terms. Although it is basically a binary classifier, one-vs-rest or softmax regression techniques may be used to expand it to multiclass issues. One limitation is its susceptibility to overfitting when dealing with high-dimensional data or complex relationships, which can be mitigated through regularization techniques like L1 (Lasso) or L2 (Ridge) regularization. Despite its simplicity, logistic regression is a valuable tool in various domains, including healthcare (predicting disease outcomes), finance (credit risk assessment), and natural language processing (text classification), and it serves as a foundational model in many machine learning pipelines due to its transparency and effectiveness [17] [18]. The concept is shown in Fig 4.4 below.

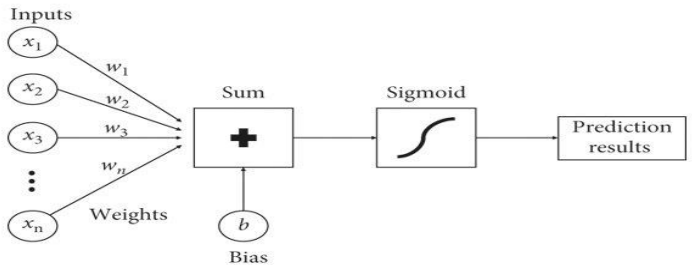



Figure 4.4: Logistic Regression Classifier

**Support Vector Machine**

Regression and classification problems may both be resolved using the Support Vector Classifier (SVC). However, categorization issues are where artificial intelligence is most frequently applied. The SVM approach looks for a straight line, or judgment limit, that divides the region into categories in variables properly fresh. A hyperplane is this

highest utility bound. Using SVM, which chooses the most extreme locations and vectors, a hyperplane may be created. As a result, the word "support vector," which is used to describe these severe situations, is where the technique's name, "support vector machine," comes from [16]. The Support Vector Classifier (SVC)'s working procedure is depicted in Fig 4.5.

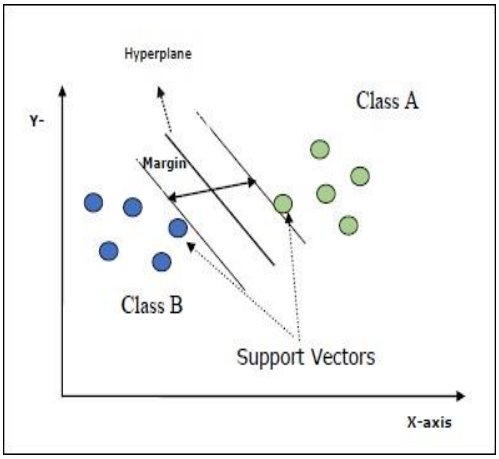



Figure 4.5: SVC classifier

**Gradient Boosting**

The Gradient Boosting Classifier is a powerful and versatile machine learning algorithm that excels in predictive modeling, particularly in classification tasks. It operates by iteratively building a strong predictive model through the combination of multiple weak models, typically decision trees, in a sequential manner. At each iteration, the algorithm focuses on the misclassified data points from the previous stage, assigning them greater importance. This iterative process allows the algorithm to continuously refine its predictions, ultimately creating a robust ensemble model. One of the key advantages of the Gradient Boosting Classifier is its ability to handle complex, high-dimensional data and capture intricate relationships between variables. By combining the outputs of multiple weak learners, it can achieve superior predictive performance. However, this power comes at a computational cost, and training a Gradient Boosting model can be more time-consuming compared to some other algorithms. To mitigate the risk of overfitting, careful hyperparameter tuning and cross-validation are essential when implementing Gradient Boosting. The choice of the learning rate, the number of boosting iterations (trees), and the maximum depth of trees are critical factors that influence the model's performance. Gradient Boosting is a

frequently utilized technique in many domains, such as information mining, banking, and biology, since it can effectively handle complicated classification problems and yield precise answers. Its versatility and robustness make it a valuable tool for both beginners and experienced data scientists aiming to tackle a wide range of classification tasks [22]. The Gradient Boosting (GB)'s working procedure is depicted in Fig 4.6.

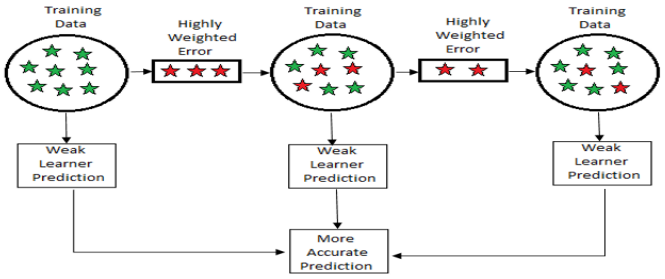



Figure 4.6: Gradient Boosting Classifier

**K-Nearest**

The K-Nearest Neighbors (KN) classifier is a widely used and intuitive machine learning algorithm for classification tasks. It operates on the principle that similar data points tend to belong to the same class. In the KN algorithm, an input data point is classified based on the majority class among its K nearest neighbors in the feature space. The choice of K, the number of neighbors to consider, is a critical hyperparameter that impacts the algorithm's performance. KN is non-parametric and does not make strong assumptions about the underlying data distribution, making it applicable in various scenarios. Its simplicity and ease of implementation make it a popular choice for introductory machine learning tasks. However, KN's computational efficiency can be a limitation for large datasets, as it requires calculating distances between the data point in question and all other data points in the dataset. Moreover, KN's performance is metric, and the curse of dimensionality can affect its accuracy as the number of features or dimensions increases. To address these challenges, techniques such as feature selection, dimensionality reduction, and careful hyperparameter tuning are often employed in conjunction with KN. Despite its limitations, KN remains a valuable tool for many classification problems, particularly when the dataset is manageable in size and the algorithm's assumptions align well with the underlying data distribution. Fig 4.7, which is below, illustrates the idea.

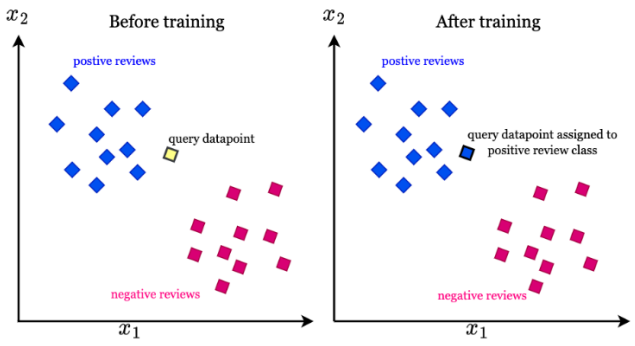



Figure 4.7: K-Nearest Classifier

**Adaboost**

The AdaBoost (Adaptive Boosting) classifier is a powerful ensemble learning method designed to enhance the performance of weak classifiers by combining them into a robust and accurate model. AdaBoost operates iteratively, sequentially adjusting the weight of each training instance based on the accuracy of the previous weak classifiers. This means that instances that are misclassified receive higher weights, allowing subsequent weak classifiers to focus on them and improve their classification accuracy. The final prediction is then made by combining the weighted outputs of these weak classifiers. One of AdaBoost's strengths lies in its adaptability to different classification problems, as it can work with a wide range of base classifiers, typically decision stumps or shallow decision trees. It's particularly effective in addressing complex datasets and overcoming issues such as overfitting, as it gives more emphasis to challenging data points during training. Moreover, AdaBoost is known for its ability to handle high-dimensional feature spaces effectively. While AdaBoost is a powerful algorithm, it's not immune to outliers or noisy data, which can adversely affect its performance. However, its capacity to mitigate these issues is strengthened by its sequential learning process. By leveraging AdaBoost's combination of weak learners, it often results in a strong and accurate classifier that is widely used in various fields, including face detection, text classification, and bioinformatics, where high performance and adaptability are essential.

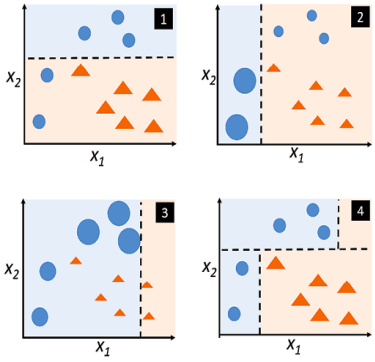



Figure 4.8: Adaboost

**Quadratic Discriminant Analysis**

Quadratic Discriminant Analysis (QDA) is a statistical classification technique used in machine learning and pattern recognition. It is an extension of Linear Discriminant Analysis (LDA) and is particularly applicable when the assumption of equal covariance matrices among classes is not met. In QDA, each class is characterized by its own covariance matrix, providing a more flexible model that can better capture the underlying distribution of the data. The goal of QDA is to find the decision boundaries that best separate different classes by estimating the probability distributions of the input features for each class. QDA models the likelihood of a data point belonging to a specific class using a quadratic decision boundary, allowing for more complex relationships between variables compared to linear boundaries. QDA involves estimating the mean and covariance matrix for each class and then using Bayes' rule to calculate the posterior probability of a data point belonging to each class. During classification, the class with the highest posterior probability is assigned to the data point. While QDA can be effective in capturing non-linear decision boundaries, it requires estimating more parameters, and if the number of features is large, it may lead to overfitting. The fundamental shape of information and the particular covariance matrix assumptions determine which of the two Bayesian distribution algorithms to use: QDA or LDA. Overall, QDA is a valuable tool for classification problems when class-specific covariances are unequal, and it offers a more flexible approach compared to LDA.

**Ridge Classifier**

Ridge Classifier is a linear classification algorithm that extends the traditional linear models by incorporating L2 regularization, also known as Ridge regularization. This regularization term is added to the standard linear regression cost function, aiming to

prevent overfitting and improve the generalization of the model. In the context of classification, Ridge Classifier is often used for binary or multiclass classification tasks. It applies Ridge regularization to the coefficients of the linear decision boundary, encouraging them to be small. This regularization term is proportional to the squared L2 norm of the coefficients, penalizing large values. The regularization term introduces a trade-off between fitting the training data well and keeping the model parameters small. The strength of regularization is controlled by a hyperparameter, commonly denoted as alpha. Higher values of alpha increase the regularization strength, leading to a simpler model with smaller coefficients. The Ridge Classifier is part of a family of classifiers that leverage regularization techniques to enhance model stability and prevent overfitting. where working with data that have convergence problems—that is, where characteristics are correlated—it is very helpful. Ridge regularization helps stabilize the model by distributing the impact of correlated features more evenly. Scikit-learn, a popular machine learning library in Python, provides an implementation of Ridge Classifier, making it accessible for practitioners to apply in various classification scenarios.

**Passive Aggressive Classifier**

The Passive-Aggressive (PA) Classifier is an online learning algorithm designed for dynamic and large-scale datasets. Operating in a lazy learning fashion, it updates its model incrementally as it encounters new data, making it suitable for scenarios with evolving information. The algorithm processes one training example at a time and updates its model when mistakes occur during predictions. The update rule, guided by an aggressiveness parameter, adjusts the model to rectify errors, with higher aggressiveness leading to quicker adaptations. This versatility allows the Passive-Aggressive algorithm to be applied to both classification and regression tasks. Its applications span various domains, including natural language processing and text classification, making it a valuable tool for scenarios with continuous and substantial data influx. Different variants, such as PA-I and PA-II, offer flexibility in adapting to specific characteristics of the data and learning requirements.

**Ensemble Learning Algorithms**

Ensemble learning refers to the technique of combining multiple machine learning models to improve overall predictive performance and robustness [23].

**Bagging Classifier**

Bagging is a powerful technique that reduces variance and improves the stability of machine learning algorithms, with a particular focus on decision tree algorithms. By creating multiple subsets of the training data through bootstrapping and training separate models on each subset, bagging helps mitigate issues like overfitting and handling missing variables. The predictions from these individual models are then combined using techniques such as majority voting or averaging to generate an ensemble model. This ensemble model, formed through the combination of diverse model predictions, exhibits enhanced performance and robustness. Bagging is a valuable tool for improving the reliability and accuracy of machine learning algorithms, providing a more robust solution for classification tasks [25]. Fig 4.9, which is below, illustrates the idea.

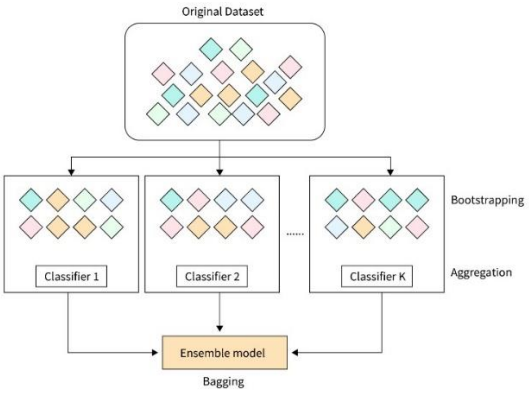



Figure 4.9: Bagging

**Boosting Classifier**

Boosting is a powerful technique that leverages a weighted average to combine multiple algorithms, transforming weak learners into strong learners and enhancing the accuracy of independent models. This technique focuses on creating loss functions that guide the learning process of the individual models. The concept of boosting is illustrated in highlighting the iterative nature of the algorithm. In our investigation, we utilize the boosting method for both training and testing phases to develop a hybrid model that leverages the strengths of each individual model. The equation for the boosting algorithm, which captures the iterative nature of the model construction, is depicted below. By utilizing boosting, we can effectively improve the accuracy and performance of our models by iteratively adjusting the weights and combining the predictions of

multiple weak learners into a more robust and accurate ensemble model [24] [26]. Below, in Fig 4.10, is an illustration of the concept.

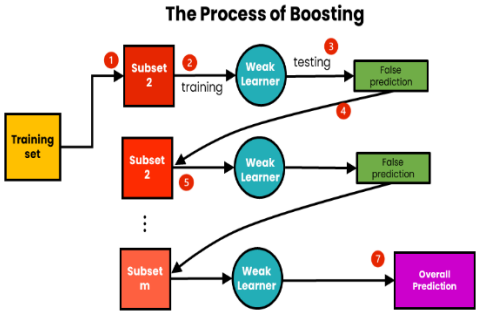



Figure 4.10: Ensemble Boosting

**Stacking Classifier**

Stacking, also known as stacked generalization, is a distinctive approach in machine learning. It involves exploring multiple models for solving the same problem. The core idea is to address a learning problem by employing various models, with each model focusing on a specific aspect of the problem rather than the entire problem. The crucial aspect is that each of these individual models can produce intermediate predictions. Consequently, we can train a second model that learns the same target using these intermediate predictions. This second model, as the name suggests, is intended to be "stacked" on top of the others. The ultimate goal is to enhance overall performance and typically achieve a model that outperforms each individual intermediate model. Ultimately, stacking trains a single model that aggregates the outputs of multiple algorithms and generates a new prediction. In terms of efficiency, stacking often outperforms any single model [20]. It can be illustrated using logistic regression as a combiner approach to integrate all the existing predictions into a final prediction, as depicted in Fig 4.11 below.

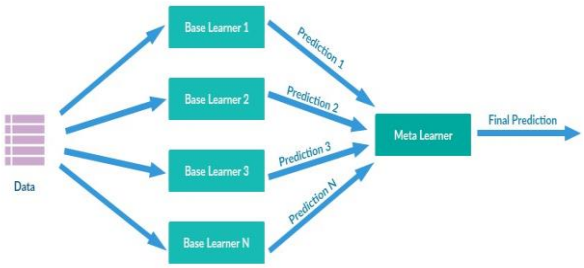



Figure 4.11: Stacking

**Voting Classifier**

A voting classifier is a combination of multiple individual algorithm which works with majority, effectively making predictions through a "majority rules" approach. This technique involves developing several models that predict outcomes, and the final prediction is based on the collective votes from these models. The concept is illustrated in Fig 4.12 below. The specific algorithm used for the voting classifier is depicted in the calculations provided in references [21] [27] [28].

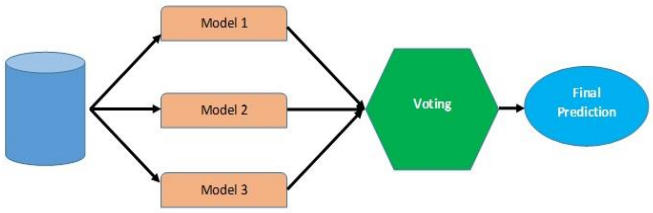



Figure 4.12: Voting

## 4.2 Experimental Results & Analysis

In this phase of the study, the evaluation of existing models played a pivotal role in assessing the efficiency of the proposed model targeting dengue using the designated dataset [25]. The process commenced with the initial implementation of the chosen dataset, followed by a rigorous examination to identify and rectify missing or erroneous data points, ensuring the dataset's integrity. A diverse range of machine learning algorithms was subsequently deployed, and their performances meticulously analyzed. For the proposed algorithms, a comprehensive assessment was conducted through confusion matrices, which included key metrics such as Accuracy, Precision, Recall, and F-1 Score, Specificity providing a holistic view of their predictive capabilities. Additionally, traditional algorithms underwent the same scrutiny, further enabling a comparative analysis. The evaluation extended to exploring the potential of different ensemble techniques, incorporating bagging, boosting, stacking, and voting, to leverage the collective strengths of multiple models for enhanced prediction accuracy. A total of five deep learning and eleven distinct traditional classifiers were harnessed, and the resulting outcomes, thoroughly assessed, facilitated the identification of the most effective approaches for predicting dengue disease. This comprehensive evaluation

process served as a critical step in gauging the performance of the proposed model and fine-tuning its predictive accuracy for practical application.

I looked at how well deep learning algorithmic performance. The table presents the evaluation metrics for five different algorithms. In terms of accuracy, the Convolutional Neural Network (CNN) outperforms others with 94.00%, followed closely by the Artificial Neural Network (ANN) at 86.00%. CNN exhibits superior precision (99.99%) and recall (91.66%), leading to a high F-1 Score of 95.65%. On the other hand, Long Short-Term Memory (LSTM) shows lower accuracy (74.00%) and recall (69.44%), indicating challenges in capturing all relevant instances. Bidirectional LSTM (BLSTM) strikes a balance with accuracy (78.00%), precision (90.32%), and recall (77.77%). These metrics collectively provide a comprehensive overview of each algorithm's performance, guiding their suitability for specific tasks. The visualization is shown in Table 4.1 and Figure 4.13.

Table 4.1. Performance Evaluation of Deep Learning

| Algorithm | Accuracy | Precision | Recall | F-1 Score | Specificity |
|---|---|---|---|---|---|
| ANN | 86.00 | 96.77 | 83.33 | 89.55 | 92.85 |
| RNN | 84.00 | 96.66 | 80.55 | 87.87 | 92.85 |
| CNN | 94.00 | 99.99 | 91.66 | 95.65 | 99.99 |
| LSTM | 74.00 | 92.59 | 69.44 | 79.36 | 85.71 |
| BLSTM | 78.00 | 90.32 | 77.77 | 83.58 | 78.57 |

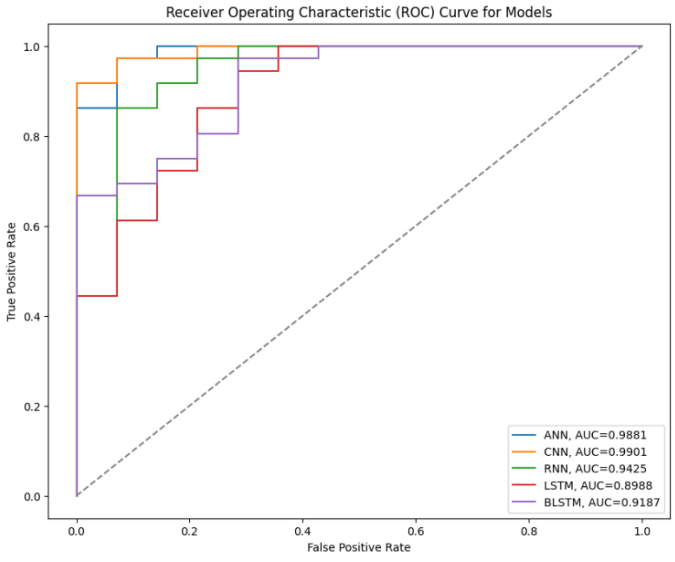



Figure 4.13: AUC-ROC Curve Analysis of Deep Learning

Curve shown in Figure 4.13 also revealed that CNN got the highest score, coming in at 99.01%. ANN, RNN, LSTM, BLSTM had accumulated, respectively, 98.81%, 94.25%, 89.88%, 91.87%. In our evaluation process, we have measured the compilation time, illustrated in Table 4.2, was carefully assessed, with specific attention to the time taken by individual algorithms. Notably, the BLSTM algorithm exhibited the highest compilation time, requiring 4.60 seconds. This analysis provided valuable insights into the computational efficiency of each algorithm, aiding in the selection and optimization of algorithms for our model, and enhancing the overall efficiency of the proposed method.

Table 4.2. Compilation Time of Deep Learning

| Algorithm | Train Time | Test Time |
|---|---|---|
| ANN | 2.39 | 0.1864 |
| RNN | 2.52 | 0.3505 |
| CNN | 2.47 | 0.2137 |
| LSTM | 2.62 | 0.2881 |
| BLSTM | 4.60 | 0.4275 |

The evaluation metrics for various machine learning algorithms are presented in the table. Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), Support Vector Machine (SVM), K-Nearest Neighbors (KN), Adaboost Classifier (ABC), GNB, QDA, Ridge Classifier (RC), and Passive Aggressive Classifier (PA) were assessed. Ridge Classifier stands out with the highest accuracy (95.18%) and precision (91.30%), achieving an exceptional F-1 Score (98.76%). The Convolutional Neural Network (CNN) in the previous table demonstrated competitive performance, emphasizing the importance of selecting algorithms tailored to specific tasks and datasets. The visualization is shown in Table 4.3 and Figure 4.14

Table 4.3. Performance Evaluation of Traditional Algorithms

| **Algorithm** | **Accuracy** | **Precision** | **Recall** | **F-1 Score** | **Specificity** |
|---|---|---|---|---|---|
| **LR** | 94.12 | 86.95 | 96.96 | 94.73 | 90.00 |
| **RF** | 92.37 | 82.60 | 95.98 | 93.10 | 87.09 |
| **DT** | 93.73 | 80.60 | 94.32 | 93.10 | 87.00 |
| **GB** | 92.55 | 82.60 | 95.97 | 93.10 | 87.09 |
| **SVM** | 94.46 | 86.95 | 91.96 | 94.73 | 90.00 |
| **KN** | 93.02 | 80.60 | 93.98 | 93.10 | 87.00 |
| **ABC** | 92.39 | 82.60 | 94.99 | 93.10 | 87.09 |
| **GNB** | 94.66 | 91.30 | 98.97 | 94.13 | 92.85 |
| **QDA** | 94.00 | 86.95 | 93.96 | 94.73 | 90.00 |
| **RC** | 95.18 | 91.30 | 98.76 | 96.42 | 93.10 |
| **PA** | 94.33 | 86.95 | 94.72 | 94.73 | 90.00 |

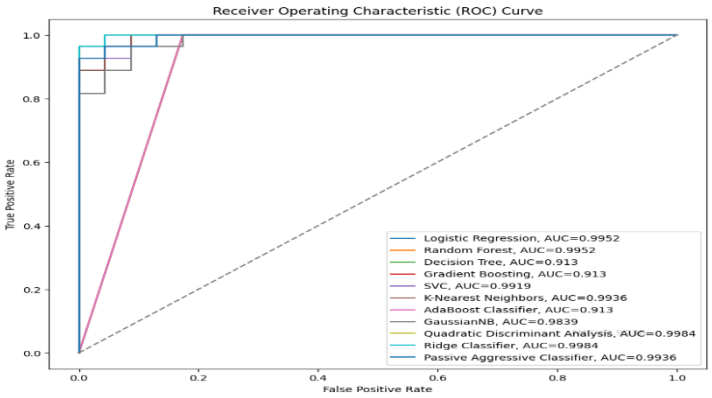



Figure 4.14: AUC-ROC Curve Analysis of Traditional Algorithm

Curve shown in Figure 4.14 likewise demonstrated that QDA and RC had the highest score, coming in at 99.84%. Attained percentages of 99.52%, 99.52%, 91.3%, 99.19%, 91.3%, 98.39%, 99.36%, 99.36% and 94.36%, respectively, were RF, LR, GB, SVC, ABC, GNB, PA, KN, and DT.

I have measured the compilation time, illustrated in Figure 4.18, was carefully assessed, with specific attention to the time taken by individual algorithms. Notably, the RF algorithm exhibited the highest compilation time, requiring 0.17 seconds. Following closely, the Gradient Boosting (GB) algorithm displayed the second-highest compilation time, totaling 0.10 seconds. This analysis provided valuable insights into the computational efficiency of each algorithm, aiding in the selection and optimization of algorithms for our model, and enhancing the overall efficiency of the proposed method.

Table 4.4. Compilation Time of Traditional Algorithms

| **Algorithm** | **Train Time** | **Test Time** |
|---|---|---|
| **LR** | 0.01 | 2E-04 |
| **RF** | 0.17 | 0.004 |
| **DT** | 0.0001 | 0.007 |
| **GB** | 0.10 | 6E-04 |
| **SVM** | 0.02 | 0.003 |
| **KN** | 0.0001 | 0.12 |
| **ABC** | 0.01 | 5E-04 |
| **GNB** | 0.001 | 4E-04 |
| **QDA** | 0.01 | 4E-04 |
| **RC** | 0.02 | 0.004 |
| **PA** | 0.002 | 3E-04 |

The bagging ensemble classifiers, including Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), Support Vector Machine (SVM), K-Nearest Neighbors (KN), Adaboost Classifier (ABC), GNB, QDA, Ridge Classifier (RC), and Passive Aggressive Classifier (PA), were evaluated on various metrics. Ridge Classifier demonstrated superior performance with the highest accuracy (96.00%), precision (93.10%), and F-1 Score (87.98%). The results emphasize the efficacy of the bagging ensemble technique, with Ridge Classifier standing out as a robust model for the given classification task. The visualization is shown in Table 4.5.

Table 4.5. Performance Evaluation of Bagging

| Algorithm | Accuracy | Precision | Recall | F-1 Score | Specificity |
|---|---|---|---|---|---|
| **LR** | 92.00 | 87.09 | 97.96 | 93.10 | 82.60 |
| **RF** | 90.22 | 84.37 | 89.98 | 91.52 | 78.26 |
| **DT** | 93.24 | 84.37 | 94.99 | 91.52 | 78.26 |
| **GB** | 89.10 | 85.00 | 98.97 | 91.50 | 78.00 |
| **SVM** | 94.45 | 84.37 | 95.96 | 91.52 | 78.26 |
| **KNN** | 92.00 | 87.09 | 94.98 | 93.10 | 82.60 |
| **ABC** | 89.00 | 85.00 | 94.99 | 91.50 | 78.00 |
| **GNB** | 92.00 | 89.65 | 98.97 | 92.85 | 86.95 |
| **QDA** | 90.71 | 84.37 | 92.96 | 91.52 | 78.26 |
| **RC** | 96.00 | 93.10 | 87.98 | 96.42 | 91.30 |
| **PA** | 92.44 | 87.09 | 91.7 | 93.10 | 82.60 |

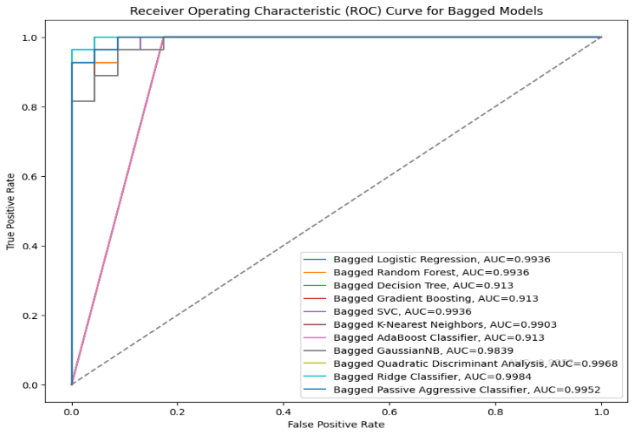



Figure 4.15: AUC-ROC Curve Analysis of Bagging

Figure 4.15's Curve similarly demonstrated that RC had the highest score, coming in at 99.84%. Attained percentages of 99.36%, 99.36%, 91.3%, 99.36%, 91.3%, 98.39%, 99.52%, 99.03%, 99.68% and 91.3%, respectively, were RF, LR, GB, SVC, ABC, GNB, PA, KN, QDA and DT.

I have measured the compilation time, illustrated in Table 4.6, was carefully assessed, with specific attention to the time taken by individual algorithms. Notably, the RF algorithm exhibited the highest compilation time, requiring 0.658 seconds. Following closely, the Gradient Boosting (GB) algorithm displayed the second-highest compilation time, totaling 0.596 seconds. This analysis provided valuable insights into the computational efficiency of each algorithm, aiding in the selection and optimization of algorithms for our model, and enhancing the overall efficiency of the proposed method.

Table 4.6. Compilation Time of Bagging

| **Algorithm** | **Train Time** | **Test Time** |
|---|---|---|
| **LR** | 0.088 | 0.002 |
| **RF** | 0.658 | 0.025 |
| **DT** | 0.047 | 0.002 |
| **GB** | 0.596 | 0.004 |
| **SVM** | 0.07 | 0.003 |
| **KNN** | 0.037 | 0.04 |
| **ABC** | 0.065 | 0.005 |
| **GNB** | 0.046 | 0.004 |
| **QDA** | 0.037 | 0.003 |
| **RC** | 0.05 | 0.003 |
| **PA** | 0.048 | 0.003 |

In the evaluation of boosting algorithms, several classifiers, including Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), GB, SVM, Adaboost Classifier (ABC), and GNB, were assessed. SVM demonstrated high accuracy (95.05%) and recall (87.09%), indicating its effectiveness in correctly identifying positive cases. Random Forest and Gradient Boosting also exhibited strong overall performance. Notably, Gaussian Naïve Bayes had limited accuracy, emphasizing its unsuitability for this classification task. The results underscore the varied performance of boosting algorithms, with Logistic Regression standing out as particularly robust. The visualization is shown in Table 4.7

Table 4.7. Performance Evaluation of Boosting

| Algorithm | Accuracy | Precision | Recall | F-1 Score | Specificity |
|---|---|---|---|---|---|
| LR | 94.20 | 90.00 | 91.96 | 94.73 | 86.95 |
| RF | 92.01 | 87.09 | 93.72 | 93.10 | 82.60 |
| DT | 93.40 | 89.65 | 94.22 | 92.85 | 86.95 |
| GB | 94.67 | 82.60 | 98.97 | 93.10 | 87.09 |
| SVM | 95.05 | 87.09 | 94.96 | 93.10 | 82.60 |

| | | | | | |
|---|---|---|---|---|---|
| ABC | 91.50 | 82.60 | 86.28 | 93.10 | 87.09 |
| GNB | 90.32 | 90.99 | 94.89 | 91.00 | 92.00 |

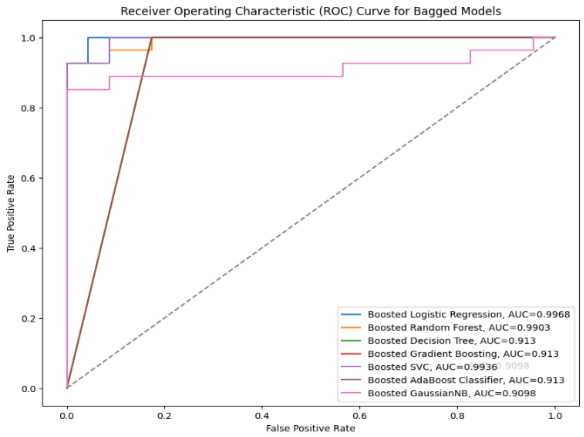



Figure 4.16: AUC-ROC Curve Analysis of Boosting

Figure 4.16's Curve similarly demonstrated that LR had the highest score, coming in at 99.68%. Attained percentages of 99.03%, 91.3%, 99.36%, 91.3%, 90.98% and 91.3%, respectively, were RF, GB, SVC, ABC, GNB and DT.

I have measured the compilation time, illustrated in Figure 4.8, was carefully assessed, with specific attention to the time taken by individual algorithms. Notably, the SVM algorithm exhibited the highest compilation time, requiring 0.107 seconds. Following closely, the RF algorithm displayed the second-highest compilation time, totaling 0.0838 seconds. This analysis provided valuable insights into the computational efficiency of each algorithm, aiding in the selection and optimization of algorithms for our model, and enhancing the overall efficiency of the proposed method.

Table 4.8. Compilation Time of Boosting

| **Algorithm** | **Train Time** | **Test Time** |
|---|---|---|
| LR | 0.0369 | 0.0027 |
| RF | 0.0838 | 0.0035 |
| DT | 0.0037 | 0.005 |
| GB | 0.053 | 0.0006 |
| SVM | 0.107 | 0.0042 |
| ABC | 0.0094 | 0.0009 |
| GNB | 0.0258 | 0.0036 |

In the evaluation of ensemble methods, including Stack Generalization (STA), Hard Voting (VOTA), and Soft Voting (VOTB), the models demonstrated varying

performance. Hard Voting achieved high accuracy (94.00%) and recall (99.7%), showcasing its effectiveness in correctly identifying positive cases. Soft Voting, while maintaining good accuracy, exhibited slightly lower performance compared to Hard Voting. Stack Generalization also showed robust results, with accuracy comparable to Hard Voting. These ensemble methods leverage the strengths of multiple models, contributing to improved predictive outcomes for in-hospital mortality. The visualization is shown in Table 4.9.

Table 4.9. Performance Evaluation of Stacking and Voting

| Algorithm | Accuracy | Precision | Recall | F-1 Score | Specificity |
|---|---|---|---|---|---|
| STA | 92.00 | 87.09 | 96.99 | 93.10 | 82.60 |
| VOTA | 94 | 90 | 99.7 | 94.736 | 86.95 |
| VOTB | 90 | 84.37 | 98 | 91.52 | 78.26 |

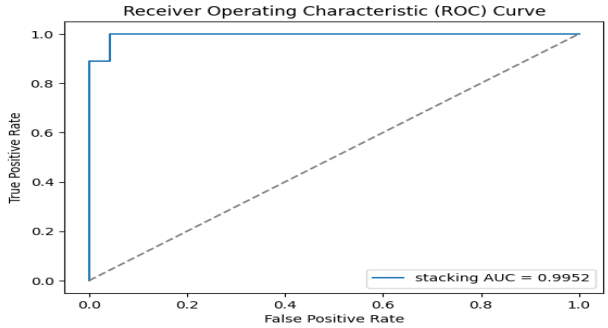



Figure 4.17: AUC-ROC Curve Analysis of Stacking

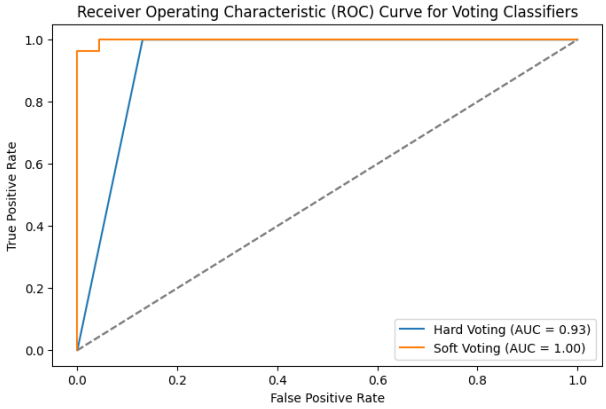



Figure 4.18: AUC-ROC Curve Analysis of Voting

Figure 4.18's AUC-ROC curve similarly demonstrated that attained percentages of 99.52%, 93% and 100%, respectively, were Stacking, VOTA and VOTB.

I have measured the compilation time, illustrated in Table 4.10, was carefully assessed, with specific attention to the time taken by individual algorithms. Notably, the Stacking algorithm exhibited the highest compilation time, requiring 0.6688 seconds. Following closely, the Voting algorithm displayed the second-highest compilation time, totaling 0.1054 seconds. This analysis provided valuable insights into the computational efficiency of each algorithm, aiding in the selection and optimization of algorithms for our model, and enhancing the overall efficiency of the proposed method.

Table 4.10. Compilation Time of Stacking and Voting

| Algorithm | Train Time | Test Time |
|---|---|---|
| STA | 0.6688 | 0.0064 |
| VOTA | 0.1054 | 0.0064 |
| VOTB | 0.1054 | 0.0091 |

## 4.3 Discussion

During this stage, I will elucidate the assessment framework for my proposed model, taking into account crucial performance metrics such as accuracy, precision, recall, and F-1 score.

### 4.3.1 Accuracy

Accuracy is a measure of the model's correctness, comparing its predictions to the actual real-world measurements. It focuses on a single variable and primarily addresses intentional errors, making it one of the most straightforward and widely used evaluation techniques for any model. Ensuring the accuracy of our models is a crucial aspect of model validation and performance assessment.

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

### 4.3.2 Precision

This section addresses precision, which measures the positively predicted with actually occurred. It reflects the true positive rate, highlighting the actual percentage of instances when the model correctly predicted true positive outcomes. While a strong recall is desirable for many models, it can sometimes be misleading if not considered in the context of precision and other performance metrics.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

### 4.3.3 Recall

This section discusses recall, which is actual data points which was correct by model. Recall is crucial in determining the model's ability to capture true positive instances, and it establishes the ratio of all positive labels to the predicted positives. While high accuracy is generally desirable, it's essential to recognize that it can sometimes be misleading if not assessed alongside other important metrics like recall.

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

### 4.3.4 F-1 Score

This section discusses the evaluation metrics of accuracy and recall, emphasizing their relevance in assessing a model's performance. Key metrics to consider are the recall and accuracy ratios, which provide insights into the model's ability to correctly identify relevant instances and overall accuracy. It's important to note that if the mean of the harmonic mean of these metrics is relatively low, it may indicate that the model's performance is not optimal, warranting further improvements.

$$F - 1\ Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

### 4.3.5 Training and Testing Time

The evaluation of training and testing time in machine learning is pivotal for assessing the efficiency and applicability of a model. During training, considerations include scalability with dataset size, algorithm complexity, available computational resources, and the impact of hyperparameter tuning. Scalability ensures the model's adaptability to larger datasets, while efficient algorithms, proper hardware utilization, and optimization techniques influence training time. On the testing side, inference speed is critical, especially in real-time applications, where models with faster testing times are preferred. The complexity of the model, choice between batch and online processing, and adherence to latency requirements are crucial factors influencing testing time.

Striking a balance between these considerations is essential for developing machine learning models that are both accurate and practically deployable in diverse applications.

### 4.3.6 Specificity

Specificity, in the context of machine learning and classification models, is a performance metric that measures the ability of a model to correctly identify negative instances out of the total actual negative instances. It is often used in binary classification problems and provides insights into the model's accuracy in recognizing true negatives. Specificity is calculated as the ratio of true negatives to the sum of true negatives and false positives. A higher specificity indicates a lower rate of false positives, which is particularly important in situations where the cost or consequences of misclassifying negative instances is significant. Specificity is complementary to sensitivity (recall) and, together, they offer a comprehensive understanding of a model's performance, especially in scenarios with imbalanced class distribution or varying costs associated with different types of errors.

### 4.3.7 AUC and ROC

The AUC-ROC (Area Under the Receiver Operating Characteristic) curve is a performance metric widely used in machine learning for binary classification models. This curve visually represents the trade-off between the true positive rate (sensitivity) and the false positive rate at different classification thresholds. The AUC-ROC score quantifies the area under this curve, providing a single numerical measure of the model's ability to discriminate between positive and negative instances. The score ranges from 0 to 1, with a higher value indicating better model performance. A score of 0.5 suggests random guessing, while a score of 1 indicates perfect classification. The AUC-ROC score is especially valuable in scenarios with imbalanced class distribution, where accuracy might be misleading. It facilitates model comparison, with higher scores indicating better discriminatory power. In summary, the AUC-ROC score is a robust metric offering both a graphical representation and a concise numerical assessment of a binary classification model's performance. It is particularly useful when faced with imbalanced datasets and provides a meaningful basis for model selection.

# CHAPTER 5

# IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

## 5.1 Impact on Society

Recommended strategy presents significant benefits, from an economic and social perspective. Our model was meticulously crafted and discern the critical and results of individuals afflicted with dengue. This research bestows a multitude of societal advantages, foremost among them being the capacity to educate and raise awareness about the prevalence of dengue and available preventive measures. Our model's precision in diagnosis and regular monitoring facilitates the early initiation of treatment, enhancing individuals' ability to make informed healthcare decisions and anticipate potential affliction. Notably, the streamlined and efficient nature of our method significantly reduces time and computational demands, simplifying disease prediction with remarkable accuracy. Our comprehensive data analysis employs advanced diagnostic techniques to uncover the underlying factors contributing to dengue. On a societal level, we aspire to witness the widespread acceptance and implementation of our recommended approach. By disseminating knowledge and promoting proactive healthcare practices, we aim to create a more informed and health-conscious society. The ultimate goal is to empower individuals to take charge of their well-being, thereby mitigating the impact of dengue and other health-related challenges. In summary, our model offers a promising avenue for not only precise disease prediction but also for the betterment of public health and healthcare awareness on a broader scale.

## 5.2 Impact on Environment

The proposed methodology holds exceptional promise for remote and underserved areas, offering simplified diagnostic methods that can effectively reduce complexity and save time. Its straightforward and non-invasive nature ensures that the environment will reap the benefits without any adverse effects. With our model, individuals in remote regions need not travel to urban centers to determine whether they have dengue or not, making healthcare more accessible and convenient. This predictive model, which also forecasts likely outcomes, can seamlessly complement a patient's diagnostic

report, alleviating concerns about the cost of local treatment or affordable dengue identification. Its user-friendly design ensures that people at any skill level can utilize it with ease.

Through the implementation of our recommended model, the potential to ascertain the presence of dengue in a patient becomes a reality, significantly enhancing the political and social healthcare landscape. We firmly believe that the adoption of our proposed model will usher in a substantial advancement in the realm of medical scientific technology, ultimately improving the quality of healthcare services and medical technology across the board.

### 5.3 Ethical Aspects

Before implementing our system, it is imperative to take ethical precautions to safeguard against the inadvertent disclosure outcomes, or unintended. The diagnosis of dengue, both in the real world and in forthcoming research endeavors, stand to benefit from our recommended approach, as this issue transcends geographical boundaries and affects a global population. Our method empowers individuals, whether they are patients or well-informed individuals, to anticipate the onset and progression of their dengue condition, offering a valuable tool for proactive healthcare management on a global scale.

### 5.4 Sustainability Plan

We are confident that our proposed model can seamlessly integrate into the global technology landscape for diagnosing and researching dengue illness. We believe that our recommended approach will be particularly beneficial for women at risk of developing dengue. With the necessary resources and support, we are enthusiastic about extending our assistance to underserved rural areas. Our proposed paradigm is designed to be practical and enduring, making a lasting impact on healthcare accessibility and dengue management.

# CHAPTER 6

# SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

## 6.1 Summary of the Study

In this thought-provoking essay, we harness the power of algorithms to gauge the impact potential of individuals, offering a reliable means to foresee future developments through our model. The diagnostic technique employed holds significant promise, enabling the prediction of an individual's potential influence. This foresight is not only beneficial for individuals who might mistakenly believe they need to be dengue-aware, but also for understanding the various stages of dengue. Our recommended methodology extends its advantages to the field of medical diagnosis, bolstered. This multifaceted approach empowers individuals and healthcare professionals alike, fostering a deeper understanding of health-related concerns and their potential impact.

## 6.2 Conclusions

In our present-day world, characterized by its blend of simplicity and advanced technology, access to cutting-edge innovations is virtually universal. My proposal leverages this technological landscape to make the process of predicting dengue disorder in individuals remarkably swift and straightforward. With the potential to benefit individuals worldwide, I am committed to ensuring the practicality and continued enhancement of my model, with plans to introduce additional features and address broader healthcare concerns in the future. This vision, founded on the current state of technology, sets the stage for an ever-evolving and progressive approach to healthcare and disease prediction.

## 6.3 Implication for Further Study

I have collected 250 data from the hospital. If the dataset is large, then RC model mightn’t be able to provide better accuracy. As humans, mortality is an inherent part of our existence, and we grapple with numerous illnesses daily. While dengue disease

affects many of us, some possess the tools to combat it. Residing in a developing nation, we have access to advanced and precise diagnostic and therapeutic technologies. These advancements have streamlined the process of diagnosing dengue illness, making it faster and more efficient. In my pursuit to provide innovation, I aspire to see my approach embraced by others. I have continually refined existing algorithms for enhanced performance and are committed to expanding my offerings in the future, fostering progress in healthcare and disease management.

## Reference:


[1] Sarma, D., Hossain, S., Mittra, T., Bhuiya, M. A. M., Saha, I., & Chakma, R. (2020, December). Dengue prediction using machine learning algorithms. In 2020 IEEE 8th R10 humanitarian technology conference (R10-HTC) (pp. 1-6). IEEE.

[2] Srisuphanunt, M., Puttaruk, P., Kooltheat, N., Katzenmeier, G., & Wilairatana, P. (2022). Prognostic indicators for the early prediction of severe dengue infection: a retrospective study in a University Hospital in Thailand. Tropical medicine and infectious disease, 7(8), 162.

[3] Siriyasatien, P., Phumee, A., Ongruk, P., Jampachaisri, K., & Kesorn, K. (2016). Analysis of significant factors for dengue fever incidence prediction. BMC bioinformatics, 17(1), 1-9.

[4] Habibur Rahman, M., Omar Faroque, M., & Tithi, F. S. (2022). Dengue Fever Prediction. In Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces (pp. 709-718). Springer Singapore.

[5] Salim, N. A. M., Wah, Y. B., Reeves, C., Smith, M., Yaacob, W. F. W., Mudin, R. N., ... & Haque, U. (2021). Prediction of dengue outbreak in Selangor Malaysia using machine learning techniques. Scientific reports, 11(1), 939

[6] Siddiq, A., Shukla, N., & Pradhan, B. (2021, December). Predicting Dengue Fever Transmission Using Machine Learning Methods. In 2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) (pp. 21-26). IEEE.

[7] Rana, S. K., Boruah, A. N., Biswas, S. K., Chakraborty, M., & Purkayastha, B. (2022, May). Dengue fever prediction using machine learning analytics. In 2022 International conference on machine learning, big data, cloud and parallel computing (COM-IT-CON) (Vol. 1, pp. 126-130). IEEE.

[8] Baker, Q. B., Faraj, D., & Alguzo, A. (2021, May). Forecasting dengue fever using machine learning regression techniques. In 2021 12th International Conference on Information and Communication Systems (ICICS) (pp. 157-163). IEEE.

[9] Mishra, V. K., Tiwari, N., & Ajaymon, S. L. (2019, December). Dengue disease spread prediction using twofold linear regression. In 2019 IEEE 9th International Conference on Advanced Computing (IACC) (pp. 182-187). IEEE.

[10] Mai, S. T., Phi, H. T., Abubakar, A., Kilpatrick, P., Nguyen, H. Q., & Vandierendonck, H. (2022, November). Dengue fever: From extreme climates to outbreak prediction. In 2022 IEEE International Conference on Data Mining (ICDM) (pp. 1083-1088). IEEE.

[11] Li, C., Zhang, Z., Ren, Y., Nie, H., Lei, Y., Qiu, H., ... & Pu, X. (2021). Machine learning based early mortality prediction in the emergency department. International Journal of Medical Informatics, 155, 104570

[12] "What is Correlation in Machine Learning?",Accessed: August 6, 2020, Available: https://medium.com/ analytics -vidhya/what-is-correlation-4fe0c6fbed47

[13] Lahoura, V., Singh, H., Aggarwal, A., Sharma, B., Mohammed, M. A., Damaševičius, R., ... & Cengiz, K. (2021). Cloud computing-based framework for breast cancer diagnosis using extreme learning machine. Diagnostics, 11(2), 241

[14] Nahar, N., & Ara, F. (2018). Liver disease prediction by using different decision tree techniques. International Journal of Data Mining & Knowledge Management Process, 8(2), 01-09

[15]Aljahdali, Sultan, and Syed Naimatullah Hussain. "Comparative prediction performance with support vector machine and random forest classification techniques." International journal of computer applications 69, no. 11 (2013).

[16]L. Mary Gladence, M. Karthi, V. Maria Anu. “A statistical Comparison of Logistic Regression and Different Bayes Classification Methods for Machine Learning” ARPN Journal of Engineering and Applied Sciences, ISSN 1819-6608, Vol -10, No-14, August 2015.

[17]“Logistic Regression for Machine Learning”, Accessed: August 6, 2021, Available: https://www. capitalone.com /tech/machine-learning/what-is-logistic-regression/

[18]Ghosh, Pronab, Asif Karim, Syeda Tanjila Atik, Saima Afrin, and Mohd Saifuzzaman. "Expert cancer model using supervised algorithms with a LASSO selection approach." International Journal of Electrical and Computer Engineering (IJECE) 11, no. 3 (2021): 2631

[19]Shorove Tajmen, Asif Karim, Aunik Hasan Mridul, Sami Azam, Pronab Ghosh, Alamin Dhaly, Md Nour Hossain. “A Machine Learning based Proposition for Automated and Methodical Prediction of Liver Disease”. In April 2022 The 10th International Conference on Computer and Communications Management in Japan.

[20]Aunik Hasan Mridul, Md. Jahidul Islam, Mushfiqur Rahman, Mohammad Jahangir Alam, Asifuzzaman Asif. “A Machine Learning-Based Traditional and Ensemble Technique for Predicting Breast Cancer”, In December, 2022. Conference: 22th International Conference on Hybrid Intelligent Systems (HIS 2022) online, 2022At: Auburn, Washington, USA.

[21]Bentéjac, Candice, Anna Csörgő, and Gonzalo Martínez-Muñoz. "A comparative analysis of gradient boosting algorithms." ArtificialIntelligence Review 54, no. 3 (2021): 1937-1967.

[22] Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. CRC press

[23]emmens, Aurélie, and Christophe Croux. "Bagging and boosting classification trees to predict churn." Journal of Marketing Research43, no. 2 (2006): 276-286.

[24]Wang, Yizhen, Somesh Jha, and Kamalika Chaudhuri. "Analyzing the robustness of nearest neighbors to adversarial examples." In International Conference on Machine Learning, pp. 51335142. PMLR, 2018.

[25]Drucker, Harris, Corinna Cortes, Lawrence D. Jackel, Yann LeCun, and Vladimir Vapnik. "Boosting and other ensemble methods." Neural Computation 6, no. 6 (1994): 1289-1301.

[26]Sharma, Ajay, and Anil Suryawanshi. "A novel method for detecting spam email using KNN classification with spearman correlation as distance measure." International Journal of Computer Applications 136, no. 6 (2016): 28-35.

[27]Pasha, Maruf, and Meherwar Fatima. "Comparative Analysis of Meta Learning Algorithms for Liver Disease Detection." J. Softw. 12, no.12 (2017): 923-933.

[28]Islam, Rakibul, Abhijit Reddy Beeravolu, Md Al Habib Islam, Asif Karim, Sami Azam, and Sanzida Akter Mukti. "A Performance Based Study on Deep Learning Algorithms in the Efficient Prediction of Heart Disease." In 2021 2nd International Informatics and Software Engineering Conference (IISEC), pp. 1-6. IEEE, 2021.

# DENGUE FEVER ANTICIPATION: INSIGHTS FROM ADVANCED ML AND DEEP LEARNING MODELS

ORIGINALITY REPORT

| 18% | 13% | 8% | 11% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Grambling State University<br>Student Paper | 1% |
| 2 | Submitted to University of Greenwich<br>Student Paper | 1% |
| 3 | Submitted to Jacksonville University<br>Student Paper | 1% |
| 4 | Submitted to Loomis-Chaffee High School<br>Student Paper | 1% |
| 5 | www.coursehero.com<br>Internet Source | 1% |
| 6 | fastercapital.com<br>Internet Source | 1% |
| 7 | www.researchgate.net<br>Internet Source | <1% |
| 8 | "Hybrid Intelligent Systems", Springer Science and Business Media LLC, 2023<br>Publication | <1% |
| | Submitted to University of Westminster | |