Implementing Machine Learning Techniques To Forecast Floods In Bangladesh Based On Historical Data

by

S.M Toufique
19201141
Sadiq Uddin Bhuiyan
19201018
Ahmed Lateef
19241016
Arman Zaman
19201005
Jubaer Bin Islam
19341002

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

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2024

Declaration

It is hereby declared that

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

Student's Full Name & Signature:

S.M Toufique
19201141

Sadiq Uddin Bhuiyan
19201018

Ahmed Lateef
19241016

Arman Zaman
19201005

Jubaer Bin Islam
19341002

Approval

The thesis/project titled "Implementing Machine Learning Techniques To Forecast Floods In Bangladesh Based On Historical Data" submitted by

1. S.M Toufique (19201141)
2. Sadiq Uddin Bhuiyan (19201018)
3. Ahmed Lateef (19241016)
4. Arman Zaman (19201005)
5. Jubaer Bin Islam (19341002)

Of Fall, 2023 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Jan 22, 2024.

Examining Committee:

Supervisor:
(Member)

Dewan Ziaul Karim

Senior Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Ethics Statement

This research paper stays faithful to a set of ethical guidelines to ensure the honesty, objectivity, and respect of all parties. This paper's research and data were collected in accordance with established ethical guidelines and regulations. There was no discrimination or disparity in the data acquisition, and the results were not manipulated for biased purposes. Throughout the research procedure, human participants provided informed consent, and their confidentiality and privacy were protected. Any potential limitations or ethical considerations associated with this research are acknowledged and discussed in an appropriate manner. Assuredly, we trust that our work demonstrates our respect for all intellectual property and contributes to the long-term prosperity of humanity.

Abstract

Flooding is a complex phenomenon that, due to its nonlinear and dynamic character, is difficult to anticipate. As a result, the prediction of floods has emerged as a critical area of study in the field of hydrology. Numerous researchers have handled this topic in various ways, spanning from physical models to image processing, however, the time steps and precision are insufficient for all applications. This report looks at machine learning approaches for forecasting weather conditions and criteria and assessing the related margins of uncertainty. The evaluated outputs enable more accurate and precise flood prediction for a variety of applications, including transportation systems.

Through the exploration of innovative approaches to flood forecasting, machine learning algorithms have emerged as a potential solution. Up-and-coming methods, including ANNs, SVMs, and Random Forests, have shown impressive performance in identifying intricate patterns and connections in both weather and hydrological data. By leveraging past weather and water information, these algorithms can generate advanced predictions of future conditions and anticipate possible flood occurrences. Responding to emergency scenarios can be made more efficient and beneficial by exploiting machine learning capabilities and advanced sensor data to more accurately predict and prepare for the devastation caused by floods, and more easily deliver aid to flood affected regions.

Keywords: Flood; Machine Learning; Prediction

Dedication

This thesis is a humble tribute to the innumerable authors whose words have sparked our imagination and fueled our passion for knowledge. Your eloquent writing, innovative ideas, and unrelenting pursuit of the truth have motivated me to undertake this scholarly endeavor. Your written works have broadened my mind's horizons and increased my comprehension of the world. We will be eternally grateful for your brilliance, originality, and unwavering commitment to the written word. This thesis demonstrates the profound influence you have had on our intellectual development. Thank you for imparting your knowledge and influencing us as writers and thinkers.

Acknowledgement

Before we get underway, we would like to mention that we are utterly grateful to Allah, the Almighty, for the opportunity, the guidance and the direction we needed to complete our thesis, as per schedule. Moreover, we are indebted to Mr. Dewan Ziaul Karim, our esteemed thesis supervisor, who never stopped encouraging, mentoring, and guiding us as we navigated a difficult subject, and therefore, we convey our profound gratitude. His steady support and continuous input allowed us to overcome the challenges. We will always be grateful to him for making time to talk to us and giving us advice on how to make our work better. Thirdly, we would like to take a moment to appreciate all of the faculty members for their assistance and encouragement throughout our time at BRAC University. And most importantly, we give thanks to our dear parents without whose unceasing prayers, passionate insight and unwavering support, we would not have made it this far.

Table of Contents

List of Figures

List of Tables

Nomenclature

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

ADT Alternating Decision Tree

ANN Artificial Neural Network

AUC Area Under Graph

$BBFT$ Bagging Best-First Decision Tree

BFT Best First Decision Tree

BPN Back Propagation Neural Network

$DBFT$ Dagging Best-First Decision Tree

DT Decision Tree

GA Genetic Algorithm

GB Gradient Boosting

KNN K Nearest Neighbour

LMT Logistic Model Tree

LR Logistic Regression

$MCQRNN$ Monotone Composite Quantile Regression Neural Network

ML Machine Learning

MLP Multilayer Perceptron

NBT Naive Bayes Tree

PCA Principal Component Analysis

REP Reduced Error Pruning

$REPT$ Reduced Error Pruning Tree

RF Random Forest

ROC Receiver Operating Characteristic

SHE Systeme Hydrologique Europeen

$SNNS$ Stuttgart Neural Network Simulator

$SWAT$ Soil Water Assessment Tool

XAJ Xinanjiang Rainfall-Runoff

Chapter 1

Introduction

1.1 Problem Statement

Floods all over the world have had a significant negative impact on the economic and social life. Vietnam has seen the death of over 90 people and the disappearance of 34 others. After the flooding of 10,000 acres of farmland, 100 villagers in Laos were devastated by the impact. In Cambodia, there were 25 fatalities and 40.000 displaced people. According to disaster management authorities, the last year alone 1 million Bangladeshis over 13 districts were affected by flooding. Numerous northern districts of the nation were flooded as a result of the country's severe rains and those in the river catchments of the neighbouring country of India. While another 7,31,958 people are still without access to safe drinking water, more than 3.3 million people have been evacuated from flood-affected areas. 41 of the 93 fatalities that have occurred since June 2020 have been children, with drowning accounting for the majority of these deaths.
Despite floods being a regular yearly occurrence the damage caused by these natural disasters, recovery is not easy. Each severe flood brings back the same problems. Floods have a wide range of effects, including both social and economic ones. It's critical for water management, environmental challenges, and social safety to predict river water levels after significant rain. For these purposes, mathematical models based on statistical analysis or physical considerations have been created. The forecasts they offer are time-consuming and imperfect in both situations. Flood management technology dates back thousands of years, from simple dams and levees to the modern day. But because floods are a natural occurrence, the inherent randomness of the natural world prevents them being totally predicted using statistical analysis. Thus machine learning provides a more effective strategy to predict floods and manage their ill effects.

1.2 Aims and Objectives

This research aims to develop and implement machine learning techniques for accurate and timely flood forecasting. Our primary goals are as follows:

- To identify the most appropriate machine learning algorithms for flood prediction based on the analysis of the collected data by developing and training machine learning models using the preprocessed data, taking into account

various algorithms including Decision Tree classifier, K-neighbour classifier, Logistic Regression, Naive Bayes and Random Forest

- To contribute to the field of flood prediction by utilising the potential of machine learning techniques to improve accuracy and provide early warnings for flood-prone areas, allowing disaster relief to take a proactive stance rather than the current reactive stance

1.3 Background Information

The South Asian population eagerly await the arrival of the monsoon season, every summer. Between June and October, the region experiences more than 70% of its annual rainfall. Rains that are unusually heavy nearly always signal calamity. Occasional droughts occur when rain falls too little or too late. When there is too much rain, vast swaths of land are washed away. This year has already been disastrous for sections of Bangladesh and India. Rivers break their banks due to unusually heavy rainfall in May and June. By June 22nd, 83% of Sylhet and 90% of Sunamganj, two districts in Bangladesh's north-east that are home to 6 million people, were entirely submerged. Authorities and humanitarian workers are desperately attempting to reach more than 9 million people throughout Bangladesh and the neighbouring Indian regions of Assam and Meghalaya. At least a hundred of them are believed to have perished, with roughly 30 of them dying in Bangladesh. In the following days, the death toll is virtually expected to rise. However, considering the intensity of the floods, it is less than what would have been predicted. Heavy rain and flash floods, for example, killed more than 180 people in Belgium and Germany in July 2021, countries far richer and less populated than Bangladesh. The number of deaths linked with such calamities has dropped considerably in Bangladesh. Cyclone Bhola killed an estimated 300,000 to 500,000 people in 1970. Cyclone Amphan, predicted to be the most severe cyclone to form in the Bay of Bengal in two decades, killed roughly 30 people in 2020. June 2020 saw massive flooding, caused by an extended and severe monsoon as well as upstream flooding, affecting 5.4 million people in the north/central regions. About 37% of the country's entire area was flooded, affecting almost 33 districts, and it was deemed the country's longest flooding event in the past 22 years.
People in numerous regions saw repeated floods till the beginning of October 2020 as a result of monsoon rains and severe rainfall upstream [16]. Housing, clean water accessibility, access to proper hygiene, and livelihoods were all severely damaged in the majority of the impacted areas. According to a report from the Bangladesh Ministry of Disaster Management and Relief (MoDMR) dated 2 August 2020, there were 1,059,295 marooned homes and 41 fatalities as a result of the prolonged floods, which affected about 5,448,271 people in 33 districts. In addition, the Ministry of Agriculture (MoA) reported that 42 million dollars' worth of crops, 125,549 hectares of agricultural land, and 83,000 hectares of paddy fields were damaged. Fisheries and cattle were also moderately to severely damaged by the floods. The Department of Livestock Service (DLS) estimates that the industry lost 16,537 hectares of grassland and USD 74.5 million in livestock. The Department of Public Health and Engineering (DPHE) reports that 100,223 latrines and 928,60 tube-wells were eliminated. In eight flood-affected districts in Rangpur division, rivers have deteriorated

3,745 hectares of land, according to the north zone office of the Water Development Board. The COVID-19 pandemic, protracted flooding, and monsoon rains all made the population's predicament worse.

How has Bangladesh mitigated the impact of harsh weather? Floods regularly inflict severe damage to numerous infrastructure and socioeconomic system elements, resulting in large direct and indirect economic losses. River flow has complicated behaviour that is influenced by soil qualities, land use, temperature, river basin, snowfall, and other geophysical factors. It is essential to correctly estimate floods and create flood mapping as a consequence to plan for emergency responses. It is currently a popular study area in natural disaster prediction and risk management. Physical, statistical, and computational intelligence/machine learning algorithms are the most popular forms of prediction models.

Chapter 2

Related Work

Due to the scarcity of research on this topic with respect to Bangladesh, our readings focused primarily on other papers that utilised machine learning algorithms for flood prediction and their implementation and findings with those algorithms.
The ANN is a computational model developed to simulate the cognitive processes of the human brain and its capacity for acquiring new skills [2]. The system is designed to acquire the ability to identify and extrapolate the correlation between a given set of input variables and corresponding output values. In recent years, there have been advancements in ANN technology, transforming it into an applied mathematical technique that exhibits certain resemblances to the human brain [4]. ANNs possess two fundamental attributes that resemble those of the human brain: the capacity to acquire knowledge and the ability to extrapolate from limited data. Sulafa [15] developed a computational model of a neuron that is capable of performing basic computations. The neuron gets information from its input links and utilises these values to calculate the activity level. A neuron establishes connections with other neurons through its input and output synapses. Every individual neuron that receives input possesses an activity value, whereas each link that transmits information between neurons is assigned a corresponding weight. Chonglin and Kwok [7] proposed the development of ANN-GA model. This approach aims to leverage the unique qualities of both the ANN and GA methods, potentially resulting in improved performance. The incorporation of an ANN model has the potential to expedite the convergence process and improve the local search capabilities of a GA model. The approach utilises GA to optimise the initial parameters of ANN as an initial phase, afterwards followed by training using a traditional ANN. The primary goal of the genetic algorithm sub-model is to identify the most optimal parameters that will result in the achievement of minimal cumulative errors between the measured data and the computed values. The model proposed by Marina et al. [5] is based on a feed-forward neural network architecture, employing a logistic activation function. The network operates in a feed-forward manner, meaning that the transmission of signals occurs in a unidirectional manner, without the presence of feedback loops between nodes. This model receives input information through specialised input nodes that are distinct from the processing nodes. This input information is then transmitted to a set of inner hidden nodes. The University of Stuttgart developed a software SNNS which was used to implement and calibrate this model. Li-Hua and Jia [12] used four distinct algorithms: Hebb, Delta, Kohonen, and BP computation. This model incorporates both forward and backward propagation during the learn-

ing process. In this paper, the researchers normalised the original data to accelerate convergence.
Khabat et al. [20] conducted a comparative analysis of four DT algorithms in the context of flash floods. The LMT algorithm integrates both DT and LR techniques where it is used at the internal nodes and leaves respectively [3]. In addition, the researchers implemented the REPT technique which is a combination of REP and DT methods. It produced high accuracy as it has the ability to simplify the structure of the tree and reduce overfitting issues [10]. Additionally, the researchers conducted a comparison of NBT, which is well recognized as a popular categorization technique due to its simplicity and interpretability. The aforementioned approach necessitates minimal computer memory and exhibits a rapid learning capability when trained on a specific dataset [18]. Moreover, the ADT algorithm is used to boost the growth of the tree for numeric prediction [17]. It is simple and robust as it formulates a prediction based on a single input feature. BFT is a model presented by Binh et al. [30]. In this model, the node that results in the greatest drop in impurity between the existing nodes for splitting is extended first, therefore being referred to as the best node. This advantage enables the exploration of novel pruning techniques that employ cross-validation for the purpose of selecting the optimal number of expansions. Peyman et al. [26] also used this model in their research with the addition of other two models, BBFT and DBFT, to forecast the chances of flooding.
SVM is a linear regression approach which is only limited to linear functions as the name suggests. There are specific assumptions which are trained according to optimization theory. Compared to other learning machines, this one is adjusted in order to maximize the capacity of the system to generalize. The goal of this approach is to find a specific linear function which best dissolves a collection of training points. The method tends to cut down the total squared variance of the data and the parameters are chosen accordingly. Some limits are introduced to allow for some divergence between the ultimate objectives and the function. Of all the linear regression methods, the most used one is 1-SV regression, where points outside the hypothesis functions are taken as slack variables which increase in value away from the function tube. A flood prediction model consisting of two stages is presented by Gwo-Fong et al. [13] working with SVM. The first step concerns forecasting rainfall, taking into consideration the relative features and observation of rainfall. The next stage concerns forecasting runoff where the observation of runoff and the predicted rainfall are worked with. 16 Taiwanese typhoons contributed to the final dataset. The SVM model forecasted correct rainfall and runoff with a 1 to 6 hours lead time, particularly for peak runoff values. Flood forecast performance improves significantly with a 4 to 6 hours advance time. Finally, the SVM model gives an operational benefit during typhoon situations by boosting prediction lead time. In a different work, Jun Wan [21] sought to integrate SVM models in order to create another forecasting model of urban floods. The SVM model gathered its final data from a specific numerical model. The results from the SVM model were differentiated from the other and the final extent of the difference was put to value. The real-time urban flood forecast system was constructed using this technique with minimum monitoring data and expense. Han et al. [8] provide a look at the watershed of Bird Creek with SVM and tackle several critical challenges in the development and implementation of SVM regarding flood forecasting. This research demonstrates that choosing the best input combinations and parameters from a huge number of

options is a serious issue for any modeler utilizing SVMs. Comparisons with several benchmarking models have been made. These illustrate that SVM can outperform all of them in the test data set, although at a high cost in terms of effort and time. In contrast to other results which were found beforehand, this work demonstrates that kernel functions, both linear and non-linear can outperform one another under various conditions. It also reveals an intriguing conclusion to how SVM reacts to different rainfall inputs. High and low rainfall inputs were seen to produce responses that were not similar. This could be a highly valuable technique to demonstrate the behavior of a model run by SVM.

Logistic regression analysis takes on two variables, a dichotomous and polychotomous response variable and a polychotomous predictor variable, y and x respectively and figures out the association between them. The approach is based on an ordinal or normal scale with a few categorical factors coming into play. 'Success' and 'Failure' are the two categories for the response variable y. The success class is assigned a value of 1 whereas, the failure class is assigned a value of 0. The Bernoulli Distribution is followed for every specific observation. Logistic regression makes way for a route of link between the causes of floods and the occurrences themselves. Logistic regression assesses the floods and picks the most suitable model to link the variables, both dependent and independent. Ahmed et al. [19] aims for flood susceptibility mapping in the Gaza Strip's southern parts with the use of logistic regression. The logistic regression makes the fundamental assumption that a flood will occur in the future. The association between flood incidence and its dependency on many independent conditioning factors was determined using logistic regression. The logistic regression fitted parameters were utilized in the Geographic Information Systems (GIS) to generate a flood susceptibility map. In a similar research, Jati [23] studied to offer information about the primary cause elements and anticipate locations in Indonesia that are likely to experience flooding is required. The study employs a logistic regression analysis mathematical model and the use of GIS. It concerns the usage of certain variables that lead to flooding, such as flow accumulation, land use, slope and most importantly, rainfall. The prediction of flood disasters reached an accuracy rate ranging from 85% to 94% approximately, with the usage of logistic regression on the model. In addition to the above two, Lee and Kim [29] presented a method to predict the real-time extent of floods. This was solely done to decrease the amount of time required to issue an alarm after a flood occurs. This technique aims to determine a discriminant, one for each regional grid, for probabilities of floods with the use of logistic regression before forecasting extents of floods in Korea based on rainfall runoff. This calculation was done by a two-dimensional flood inundation model. A value of 1 was produced if a grid was flooded and a value of 0 for when it was not. The discriminant was filled with these values to calculate predicted danger of flood in a grid. With the use of this method, scenario rainfall reached an accuracy rate of 84% whereas for real rainfall, it was slightly lower on 75%.

The K-Nearest Neighbor (K-NN) model is a machine learning technique that is frequently used for classification and regression problems and is motivated by the idea of similarity between data points. Based on the average of the K-nearest data points in the feature space or the majority class, this model seeks to generate predictions. In the worlds of statistics and pattern recognition techniques, nearest-neighbor techniques have been thoroughly researched. Despite their inherent simplicity, nearest-

neighbor algorithms are considered versatile and robust [9]. The capability to create predictions based on proximity to known data points and the capacity to assimilate knowledge from the dataset are two fundamental characteristics that K-NN inherits from human cognition. The weighted K-NN technique put out by Altman [1] is a major expansion of the conventional K-NN paradigm. This variation gives each neighbor a weight based on how close they are to the query point, giving closer neighbors a bigger impact on the prediction. The predicted accuracy of K-NN models has been improved thanks to Altman's method, particularly in situations where some neighbors are more important than others. The work of Deegalla et al. [11] shows how K-NN and dimensionality reduction methods can be combined. To enhance the algorithm's performance on high-dimensional datasets, they presented a technique that combines Principal Component Analysis (PCA) with K-NN. By lowering the dimensionality of the feature space while retaining crucial data, this method solves the curse of dimensionality, a problem that frequently arises in K-NN. K-NN has found practical use in the context of time series forecasting, as shown by Wu et al. [14]. Their research focuses on modifying K-NN to forecast time series data while taking into account the sequential character of time series data. The flexibility of K-NN in handling various data sources was illustrated by their sliding window-based technique, which uses historical time series segments to forecast future values. Additionally, the use of K-NN models with imbalanced datasets has been investigated. A method for addressing class imbalance issues was introduced [6] that uses the Synthetic Minority Over-sampling Technique (SMOTE) in conjunction with K-NN. For the purpose of classifying minorities, this combination has been shown to improve K-NN performance.

Gradient Boost Algorithm utilizes an ensemble which is used to simplistically eliminate bias, noise, and variance which dilute the effectiveness of the prediction model [22]. Its relevance extends to practical applications where accuracy and model interpretability are paramount. Two key characteristics of gradient boosting are its iterative model construction process, where each new tree builds upon the mistakes of the preceding one, and its capacity to represent intricate, non-linear relationships in data [32]. These qualities support its adaptability in handling a range of jobs. Gradient Boosting reduces the errors caused by earlier decision trees by building decision trees one after the other and fine-tuning their parameters. Gradient descent and a loss function are used in this technique to direct the optimization process, producing ever-improved models [32]. Researchers have investigated Gradient Boosting in combination with oversampling, undersampling, and cost-sensitive learning approaches to address class imbalance issues [31]. These methods seek to equalize the attention paid to each class during model training. Gradient Boosting has advanced machine learning from its origin to the present, always adapting to suit the needs of complicated data processing. It is a great asset for predictive modeling and data-driven decision-making due to its ongoing relevance and adaptability. Qian et al. [24] trained 14 ML models based on 10 and 2 features where the 2 features are only derived from a PCA. In the 10 features, Fine KNN obtained the least accuracy while Subspace Discriminant, Linear SVM and Quadratic SVM had equally the highest accuracy. For the other modification, Medium tree produced the worst accuracy and in this case, only Linear SVM and Subspace Discriminant models had the best accuracy. They have constructed both the SVM and Subspace ensemble models in Matlab using functions like fitcsvm and fitcensemble respectively. Kruti et al. [28]

used algorithms like DT, GB and RF to train their models to give the best possible forecasting flood primarily occurring in certain regions of India. Then they utilised this data to alert residents via Android applications for any risks. Suresh et al. [25] used the Deep Neural Network as one of their algorithms to train their model. This algorithm gave better results in comparison to the SVM and KNN on the basis of their data set.

Chapter 3

Methodology

3.1 Workflow

We initiate the workflow by collecting and then organizing the data. After the formation of the collected data, it is pre-processed. Algorithms like Logistic Regression, Decision Tree, k-Nearest Neighbor, Random Forest and Gaussian Naive Bayes are implemented which is then later evaluated based on accuracy, precision, error etc. to put into comparison which one of the algorithm produces the best output to give a better result. Therefore, we can determine the optimal model for flood prediction based on historical data. Figure 3.1 is a block diagram illustrating the methodology's workflow in detail.

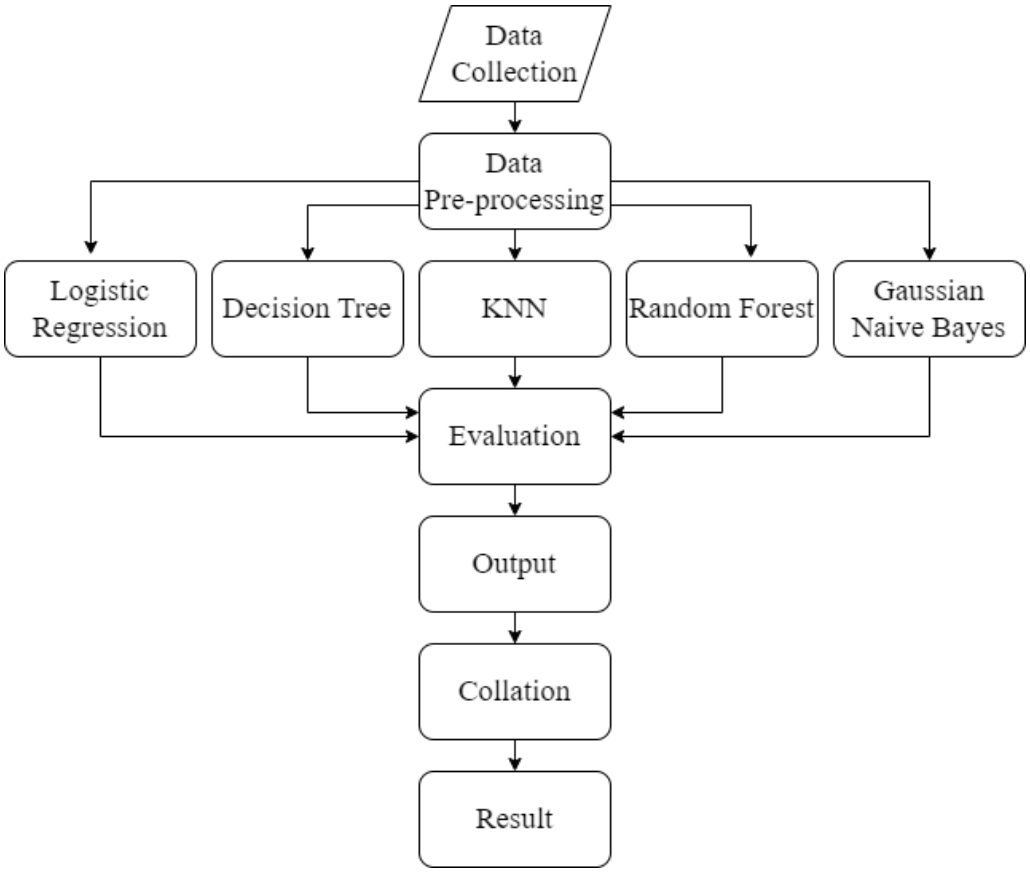


Figure 3.1: Top Level Overview of the Proposed Method

3.2 Dataset Description

The dataset used in this study, comprising weather data from multiple weather observation stations across Bangladesh since at least 1990, underwent a rigorous cleaning process to ensure data quality and reliability. The dataset consisted of approximately 200,000 data points, encompassing various weather parameters relevant to flood prediction.
The weather data chosen for this research was obtained from Bangladesh Meteorological Department (BMD) and Bangladesh Water Development Board (BWDB). The dataset obtained from BMD consisted of most of the following parameters: rainfall, minimum temperature, maximum temperature, average temperature, cloud coverage, wind speed, wind direction, Humidity. The BWDB provided the maximum, minimum and average water-levels for chosen districts. The dataset was purchased through official channels and government portals after submitting documentation of motivation and acknowledgement from the university. After approval of the concerned government entities, the data was made available for purchasing for research purposes.

Table 3.1: Variables Used in the Dataset and their Respective Units

Rainfall	Rainfall in mm.
Humidity	Relative Humidity in %
Tmax	Maximum temperature in °C
Tmin	Minimum temperature in °C
Tavg	Average temperature in °C
Wind_spd	Average Wind Speed in Kmph
Wind_dir	Wind Direction
Cloud_amt	Cloud Amount
Area_threshold	Danger-level in m.
max_wl	Maximum Water Level in mPWD
min_wl	Minimum Water Level in mPWD
avg_wl	Average Water Level in mPWD
Date	Day-Wise timestamp from 1/1/1990 to 31/12/2022
Flood	Flag of 1 or 0

BWDB and other government departments refer water levels to the Public Works Datum (PWD). PWD is a horizontal datum believed originally to have zero at a determined Mean Sea Level (MSL) at Calcutta. PWD is located approx. 1.5 ft below the MSL established in India under the British Rule and brought to Bangladesh during the Great Trigonometric Survey. Flooding occurs when water level exceeds danger level.1 representing danger level at a river location is the level above which it is likely that the flood may cause damages to nearby crops and homesteads. In a river having no embankment, danger level is about the annual average flood level. In an embanked river, danger level is fixed slightly below design flood level of the embankment. The danger level at a given location needs continuous verification. Flood Label indicated Flood occurred when 1 and 0 representing no flood occurred. The districts chosen were based on the random sampling across the four basins of Bangladesh according to the annual flood report by Bangladesh Water Development Board (BWDB) [33]. The Basins are Brahmaputra Basin, Ganges Basin, Meghna

Basin, South East Hill Basin and about districts from each basin were chosen across the basin with the most stations and day-wise historical data for the last thirty years.

The data obtained from the BMD was parameter-wise. After the purchase, the data was made available in a compressed format, from which the extracted dataset for weather parameters consisted of CSV files for each of the parameters. Each Parameter would have selected districts as multiple headers throughout the file and timestamps as columns with day wise data in rows. The data was formatted differently for each of the CSV files present in the folder. On the other hand, the data received from BWDB consisted of CSV files with the water-level, river and station name for each district and the data was row wise for the last thirty years for the respective district. The total dataset consisted of 219,602 rows and upon grouping Brahmaputra basin consisted 86,004 rows, Ganges basin consisted of 36,828 rows, Meghna basin consisted of 35,912 rows, South East Hill Basin consisted of 60,858 rows.

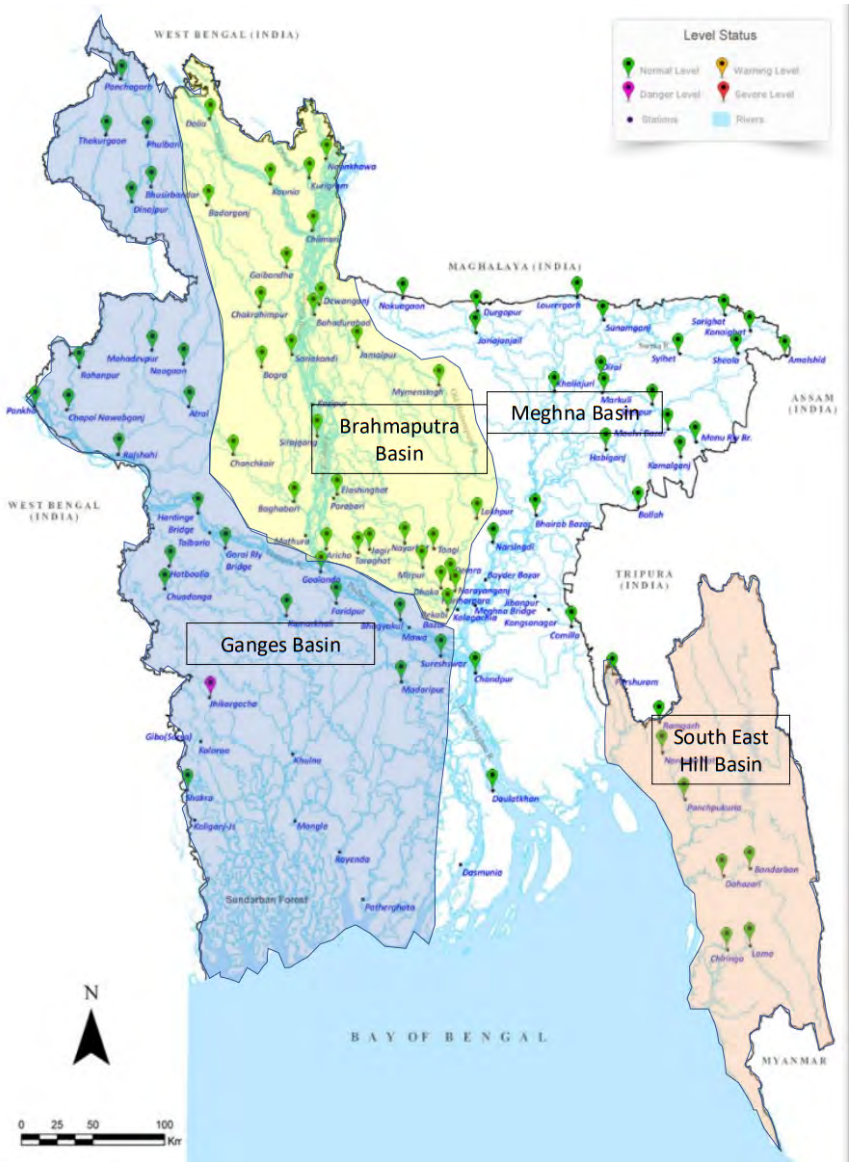


Figure 3.2: Basin Map of Bangladesh with Water Level Gauge Stations [34]

3.3 Data Preprocessing

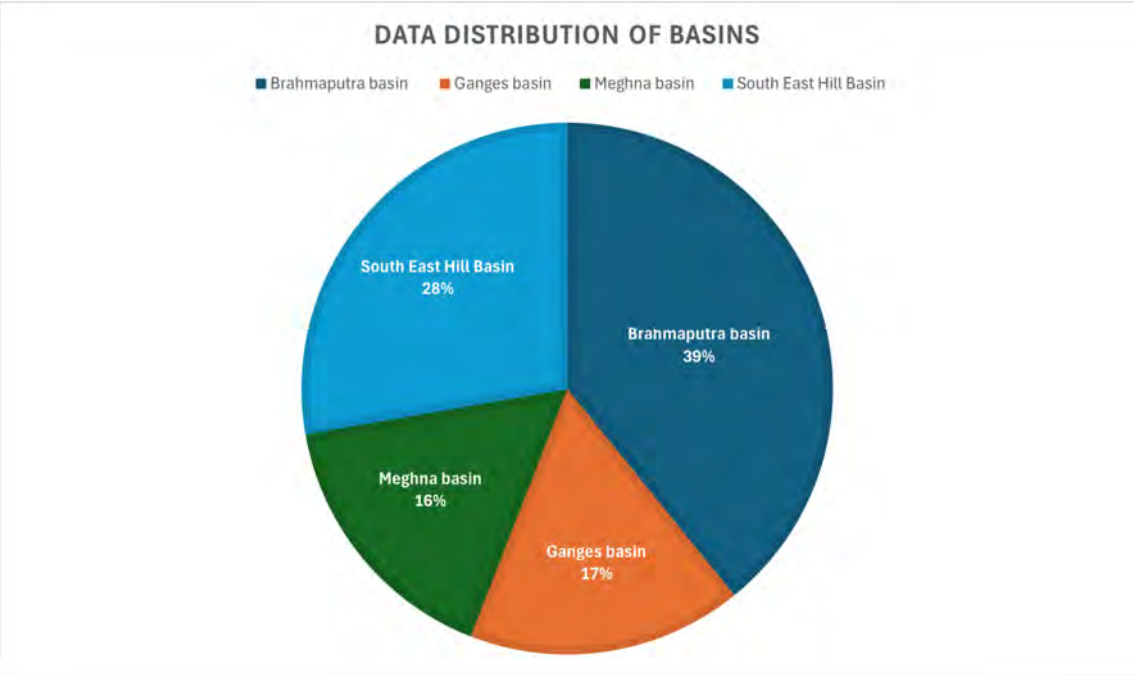


Figure 3.3: Distribution Across the Basins

In order to run machine learning models the data was needed to be converted to a flat file which is a single table of data which consists of all the attributes for flood prediction such as rainfall, minimum temperature, maximum temperature, average temperature, cloud coverage, humidity, wind speed, wind direction, water-level and a flag of whether flood occurred or not as a label. Each csv file was parsed individually to form a combined dataset. The combined dataset consisted of date, attributes and water level as features for the dataset. To flag whether flood occurred or not, a threshold water-level was extracted from the annual flood report of BWDB and if the water-level for an instance of the day crossed that threshold, the instance was recorded as flooding.

To enhance data integrity, irrelevant parameters with no significant correlation to the flood prediction results, such as weather station coordinates, were removed from the dataset. Furthermore, all weather data variables were standardized to a consistent format and unit of measurement, ensuring comparability across different features.

During the cleaning process, duplicate data instances were identified and eliminated to prevent any bias in the analysis. Outliers, which could potentially distort the modeling results, were detected using appropriate statistical techniques and either removed from the dataset or corrected to maintain data integrity.

Within the dataset only wind direction feature which contains categorical dataset. To train machine learning models the categorical values must be converted to numerical values. Using label encoder wind direction values were encoded and integrated into the dataset.

A portion of the dataset contained missing values. To deal with these, we used the SimpleImputer from the sklearn library to impute these missing values. Mean based imputation was used for numerical values such as temperature and water level, while mode based imputation was used for quantitative values such as wind direction.

Furthermore, a copy of the complete dataset before imputation was taken and divided into four smaller datasets, each corresponding with the different flood basins in Bangladesh. Having been divided, the missing values were filled in using those imputation techniques. By employing exploratory data analysis and implementing these data cleaning procedures, we ensured the reliability and quality of the dataset for subsequent analysis. The cleaned dataset was then utilized to train and evaluate various machine learning models for flood prediction in Bangladesh. However, the cleaned combined dataset and basin dataset were imbalanced, which were resolved using stratifying during training.
The dataset was split into 2 portions, train and test sets. The train set received 75% of the data while the test set was 25%. To ensure that the ratio of positive and negative labels was retained, stratification was implemented. Stratification Is used to sort data to ensure that the proportion of label values is the same across the train and test sets. Since the dataset contains unequal numbers of positive and negative labels, without stratification, a training set could be produced that does not accurately portray the qualities of the entire dataset.

3.4 Classification

The dataset utilized in this study primarily consisted of weather data, which is a crucial factor in flood prediction. The choices in the classifiers used were made with certain considerations in mind:

3.4.1 Logistic Regression

Logistic regression excels at classifying data where there is a binary outcome, in our case whether or not there will be a flood. Logistic regression provides interpretable coefficients, allowing us to assess the influence of different predictors on flood prediction. Moreover, logistic regression assumes a linear relationship between the parameters of our dataset and the log-odds of the flood occurrence, which is a reasonable assumption for many flood-related variables. This characteristic of logistic regression aligns well with the objectives of our study, enabling us to analyze the impact of individual predictors on the flood likelihood accurately and effectively.

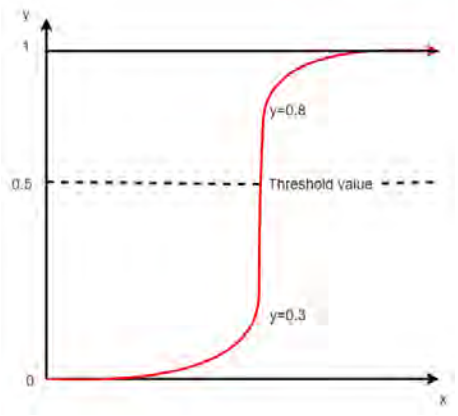


Figure 3.4: Logistic Regression Mechanism

3.4.2 Decision Tree Classifier

Decision trees, on the other hand, offer distinct advantages for analyzing complex relationships within the flood prediction dataset. They excel at capturing nonlinear relationships that may exist between predictors and the target variable. By employing decision trees, we can uncover intricate patterns and interactions among the variables, thereby enhancing our understanding of the underlying decision-making process. Moreover, decision trees provide an easily interpretable structure in the form of if-else rules, allowing us to gain meaningful insights into the factors influencing flood prediction.
Another strength of decision trees lies in their ability to naturally handle categorical and ordinal features without the need for explicit encoding. This capability makes decision trees particularly well-suited for datasets that include such types of variables, as they can efficiently incorporate them into the modeling process.

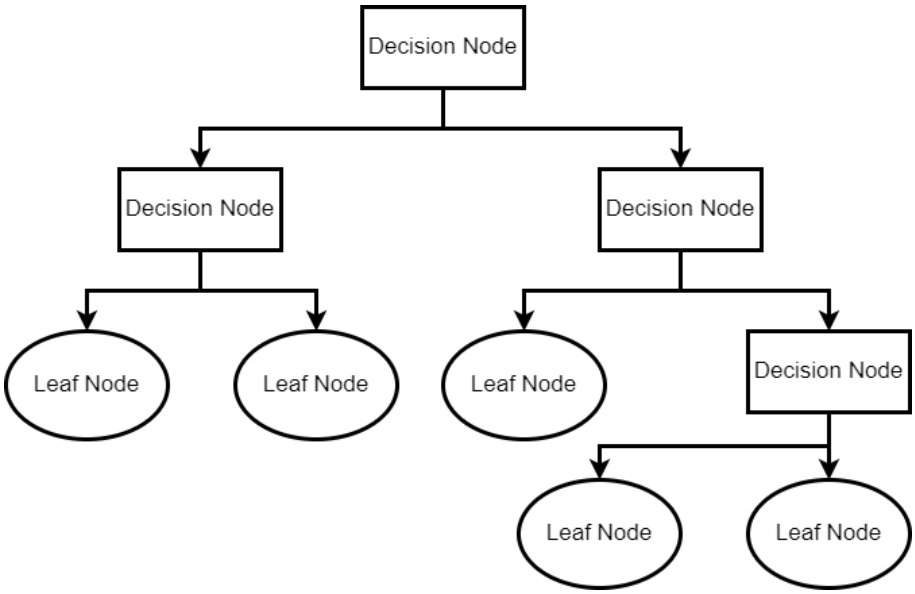


Figure 3.5: Decision Tree Classifier Mechanism

3.4.3 KNN

This results of these algorithms were further supported by the implementation of the K-Nearest Neighbour (KNN) algorithm. The KNN model was also trained using the value of weather parameters and flood occurrence as the label. The KNN classifier fit the data points to the training dataset and the performance of the model is compared using the accuracy scores from the test dataset [27].
By leveraging the strengths of logistic regression, decision tree classifier and K-Nearest Neighbour our aim was to create a comprehensive flood prediction model that considers both linear and nonlinear relationships among predictors. This approach allows us to capitalize on the interpretability of logistic regression while harnessing the complexity-capturing capabilities of decision trees and K-Nearest Neighbour. Through this combined methodology, we can develop a robust and accurate flood prediction model that not only accounts for various predictors but also maintains interpretability, essential for effective communication of research findings in the academic realm.

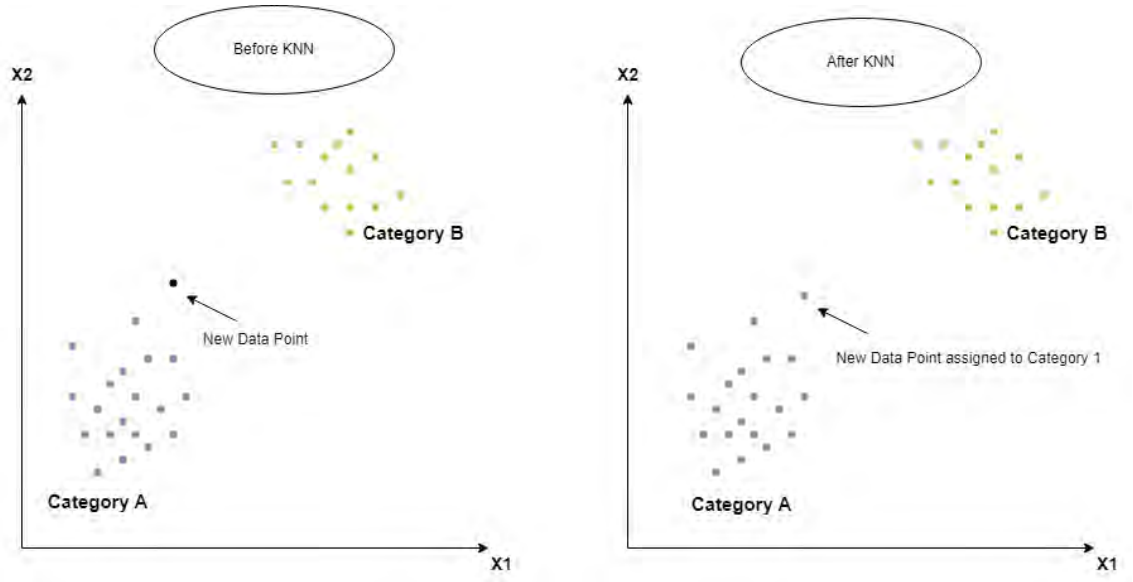


Figure 3.6: K-Nearest Neighbor Mechanism

3.4.4 Random Forest

Like forests consist of numerous trees, the random forest algorithm comprises of multiple decision trees to float through the branch and leaf nodes before coming to a final conclusion. The sole purpose of this algorithm in our flood prediction is to significantly improve our overall accuracy by taking advantage of the ability of the algorithm to work with complex datasets. Random forest efficiently handles both classification and regression problems simultaneously, enabling us to deal with both the continuous and the categorical variables of our dataset. Each decision tree of the forest is not concerned with all the features of the flood data, thereby cutting down feature space, increasing diversity and boosting stability. These benefits were vital in shaping up the model in order to accurately predict an incoming flood.

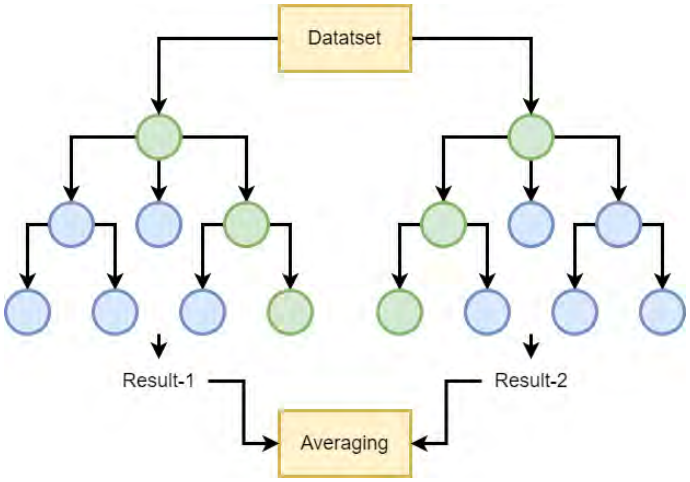


Figure 3.7: Random Forest Mechanism

3.4.5 Gaussian Naive Bayes

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{3.1}$$

Gaussian Naive Bayes is an excellent option because the features of our dataset are continuous. In general, Naive Bayes is renowned for being simple and effective. It can function well even with a minimal amount of data and is quick and simple to implement. In addition to class predictions, Gaussian Naive Bayes also produce probabilistic predictions, which are useful for figuring out how confident the model is in its predictions. The Gaussian Naive Bayes algorithm has an accuracy of 0.692 and an error of 0.308 for our model.

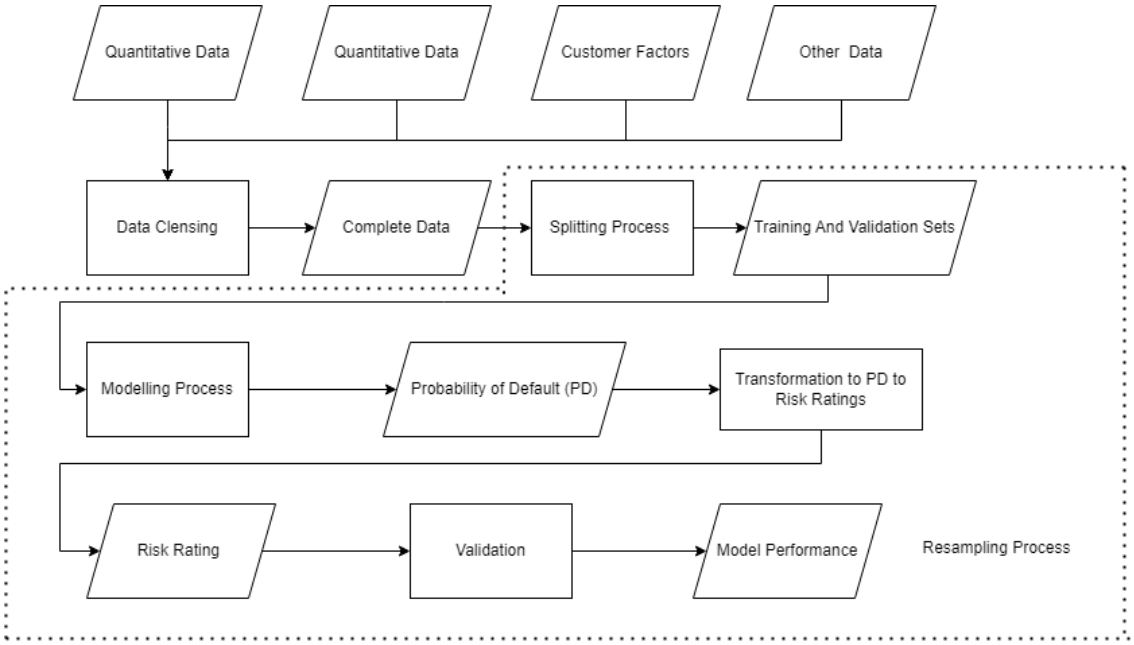


Figure 3.8: Gaussian Naive Bayes Mechanism

Chapter 4

Experimental Results and Analysis

4.1 Experimental Setup

This experiment's training and testing methodologies are developed using Python libraries including pandas, matplotlib, numpy and sklearn. The Google Colaboratory environment is utilized for training and accessing the models. Each of the five machine learning algorithms was run on the full dataset and then on the four different flood basin datasets.

4.2 Performance Analysis

Several performance evaluation metrics, including accuracy, precision, and F1-score, are used to compare and validate the performance of the model. Although accuracy is the most common metric used in classification tasks, we evaluated our model using multiple metrics from various angles. The following equations can be used to express the various evaluation metrics used in this study.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

$$F1 - Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4.4}$$

4.2.1 Decision Tree Classifier

The decision tree classifier (Table 4.1) yielded a testing accuracy of 82.9%, a testing recall of 87.5% and a testing precision of 87.4% for the full dataset. The testing F1 score was a solid 87.4%, costing an error of 17.1% along the way. Pointing out, the classifier also gave us a highest possible 100% accuracy, recall, precision and F1 score, without any error, for the Meghna Basin. Decision tree also scored highly for the other 3 basins with accuracies of 78.5%, 97.7% and 96.2% for Brahmaputra, Ganges and South East Hill Basin respectively. Overall, the algorithm was the most

reliable right after Random Forest, which we will analyze later, with the second lowest error and the second highest accuracy, precision and F1 score among the five algorithms applied.

Table 4.1: Performance Comparison of the Basins Employing Decision Tree Classifier

Decision Tree Classifier	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)	Error (%)
Brahmaputra Basin	78.5	80.0	81.7	80.84	21.5
Ganges Basin	97.7	37.4	41.7	39.43	2.3
Meghna Basin	100	100	100	100	0
South East Hill Basin	96.2	74.4	77.2	75.77	3.8
Entire Dataset	82.9	87.5	87.4	87.4	17.1

4.2.2 K-Nearest Neighbor

The results obtained from the KNN algorithm were the closest to the ones mentioned above, obtained from the decision tree classifier. The testing accuracy achieved was 80.9 percent with an error of 19.1 percent. Moreover, we obtained a testing recall of 86.2%, a testing precision of 85.8% and an F1 score of 86%. The basin datasets scored similarly, with Brahmaputra getting accuracy of 78.2%, Ganges 97.4%, Meghna 99.3% and Hill getting 92.9%. Thus, we can say that this algorithm was almost as suitable for our prediction as the decision tree classifier.

Table 4.2: Performance Comparison of the Basins Employing KNN

KNN	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)	Error (%)
Brahmaputra Basin	78.2	80.0	81.3	80.64	21.8
Ganges Basin	97.4	40.7	36.3	38.37	2.6
Meghna Basin	99.3	87.0	92.3	89.57	0.7
South East Hill Basin	92.9	53.1	55.3	54.18	7.1
Entire Dataset	80.9	86.2	85.8	86.0	19.1

4.2.3 Logistic Regression

For the Brahmaputra Basin and the Hill Basin, logistic regression(Table 4.3) displayed a stable performance with accuracies of over 82.6% and 92.0%. The F1 scores varied, with the Brahmaputra Basin scoring 86.5% and the South East Hill scoring 8.0%. Their precision and recall also varied were the precision of 77.4% and 33.3% and recall of 97.9% and 0.01% respectively. For the full dataset, logistic regression scored very poorly with an accuracy of 69.3%. While logistic regression can be dependable, it may not work well sometimes. It proved an issue with calculating the recall and precision scores for the Ganges Basin and caused major errors where no other algorithms did. However, logistic regression still managed to improve the score of the 4 basin sets over the full set quite impressively.

Table 4.3: Performance Comparison of the Basins Employing Logistic Regression

Logistic Regression	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)	Error (%)
Brahmaputra Basin	82.6	97.9	77.4	86.5	17.4
Ganges Basin	98.0	0.0	0.0	Undefined	2.0
Meghna Basin	99.0	80.4	87.7	83.9	1.0
South East Hill Basin	92.0	0.01	33.3	0.02	8.0
Entire Dataset	69.3	94.1	70.6	80.7	30.7

4.2.4 Gaussian Naive Bayes

For the Meghna Basin, Naive Bayes(Table 4.4) produced impressive results with an accuracy of 95.7%, recall of 91.4%, and F1 score of 59.0%; however, precision was lower at 43.6%. Despite an accuracy of 80.1% and a high recall of 89.5% for the Brahmaputra Basin, the precision was 78.5%, resulting in an F1 score of 83.8%. With error rates ranging from 4.3% in the Meghna Basin to 21.7% in the Ganges Basin, Naive Bayes' performance varied between the basins, suggesting that the unique features of the data from each basin can have a significant impact on how well it performs. It also did not score well with the full dataset, receiving an accuracy of 69.2%. Overall Gaussian Naive Bayes managed to score around average and thus poses to be not well suited to this type of dataset.

Table 4.4: Performance Comparison of the Basins Employing Naive Bayes

Naive Bayes	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)	Error (%)
Brahmaputra Basin	80.1	89.5	78.5	83.8	19.9
Ganges Basin	90.1	69.2	12.9	21.7	9.9
Meghna Basin	95.7	91.4	43.6	59.0	4.3
South East Hill Basin	85.9	19.4	16.7	17.9	14.1
Entire Dataset	69.2	93.1	70.8	80.4	30.8

4.2.5 Random Forest

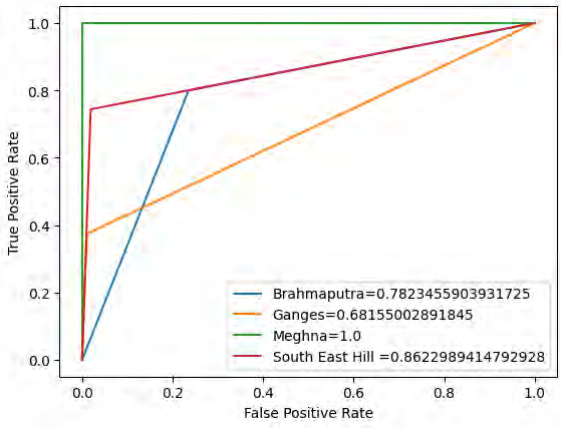
For the Meghna Basin, Random Forest(Table 4.5) received perfect scores of 100% for every threshold due to overfitting. The accuracy and recall in the Brahmaputra Basin were strong at 83.1% and 90.7%, respectively, with an F1 score of 85.9%. The level of recall for the Ganges Basin was lower, at 32.4%, indicating that Random Forest's sensitivity to different basins varies. Random Forest scored highest of all for the full dataset with accuracy of 85.6%, with a recall and precision of 87.9ând 90.6ÔOverall, Random Forest was a powerful performer with comparatively low error rates among the 5 algorithms. This algorithm's results point strongly that it is a good candidate for these sorts of datasets and may prove useful in further research.

Table 4.5: Performance Comparison of the Basins Employing Random Forest

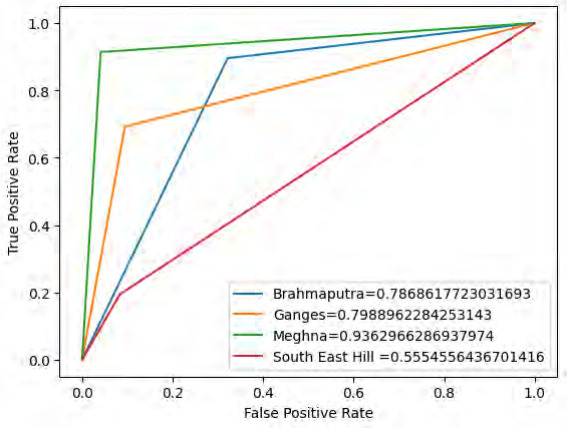
Random Forest	Accuracy (%)	Recall (%)	Precision (%)	F1 Score (%)	Error (%)
Brahmaputra Basin	83.1	90.7	81.6	85.9	16.9
Ganges Basin	98.3	32.4	62.8	42.7	1.7
Meghna Basin	100	100	100	100	0.0
South East Hill Basin	97.1	74.8	87.3	80.6	2.9
Entire Dataset	85.6	87.9	90.6	89.3	14.4

4.3 AUC-ROC Graphs

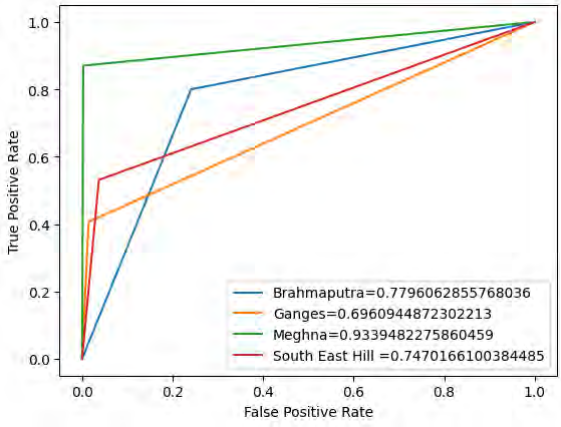
The ROC curve has two parameters: True positive rate at the y-axis and false positive rate at the x-axis. So it shows the execution of the models at the classification thresholds. The AUC measures a classifier's potentiality to differentiate amidst the classes. Due to it being both classfication-therehold and scale invariant, it provides a combined measurement of the performances. Therefore, higher the AUC-ROC score, the better the model's capacity to distinguish in-between the positive and negative classes. Here, Meghna basin achieved the highest AUC scores in all the models meaning it has the best predictions. The other basins also had decent AUC scores as most of the value were greater than 0.5 which represents there were some overlapping but not completely overlapped, otherwise it would have been useless. Most scores came out to be moderate which ranges from 0.70 to 0.90.



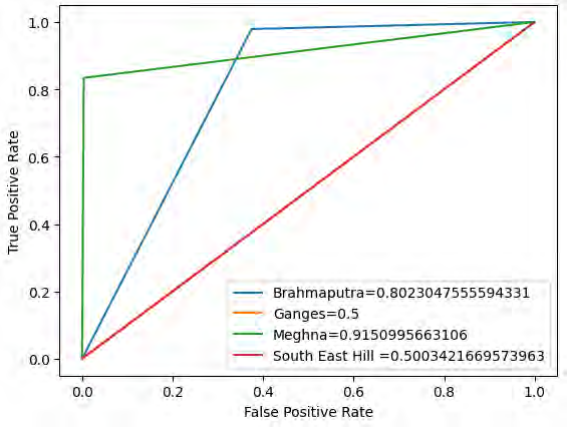
(a) Decision Tree Classifier



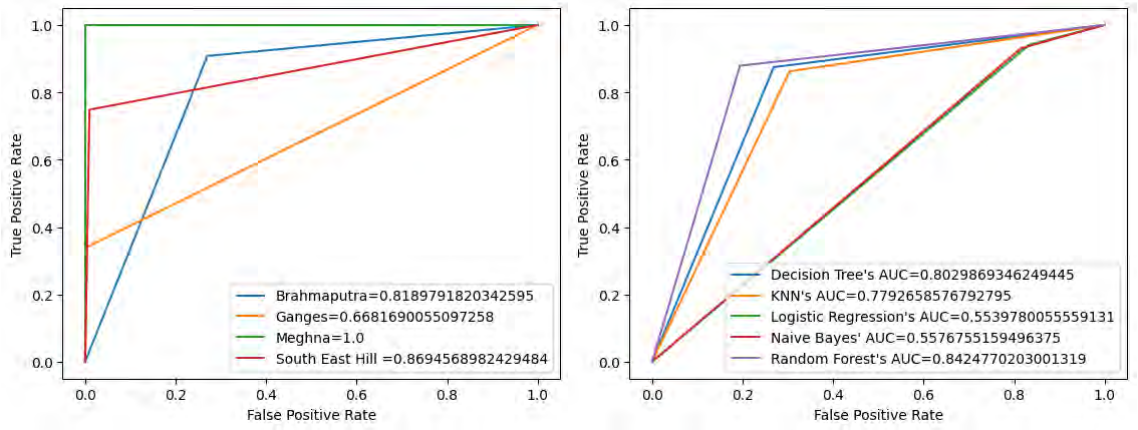
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor



(d) Logistic Regression



(a) Random Forest (b) Full Dataset

Figure 4.2: AUC-ROC Graphs

4.4 Confusion Matrix

Confusion matrix is a table of visualization that represents the performance of the algorithms, which is usually used for classification problems. True Positives (TP): These are cases in which the model predicted 'Flood', and the actual class was also 'Flood'. True Negatives (TN): These are cases in which the model predicted 'No Flood', and the actual class was also 'No Flood'. False Positives (FP): These are cases in which the model predicted 'Flood', but the actual class was 'No Flood'. False Negatives (FN): These are cases in which the model predicted 'No Flood', but the actual class was 'Flood'.

4.4.1 Combined Dataset

The confusion matrices for the combined dataset depict the Decision tree was the most accurate out of the five models, following Random Forest. Decision tree gives 52076 true positives and only 1109 false negatives and 110899 true negatives and only 617 false negatives whereas Gaussian Naive Bayes and Logistic regression gives 14329 and 14632 false positives respectively. Random Forest also showed promising results with only 3450 false positives and 4540 false negatives. K-Nearest Neighbor performed better with only 5336 false positives and 5140 false negatives.

4.4.2 Brahmaputra Basin

Upon grouping the dataset into basins, the results for the Brahmaputra basin significantly improved where the decision tree results show 27764 true positives, 135 false positives and 36283 false positives and 321 false negatives. Results for -Nearest Neighbor and Random forest also improved, showing only 228 and 261 false negatives, similarly 361 and 195 false negatives respectively. The accuracy of the models improved since most of the data is obtained from the stations in the Brahmaputra basin.

4.4.3 Ganges Basin

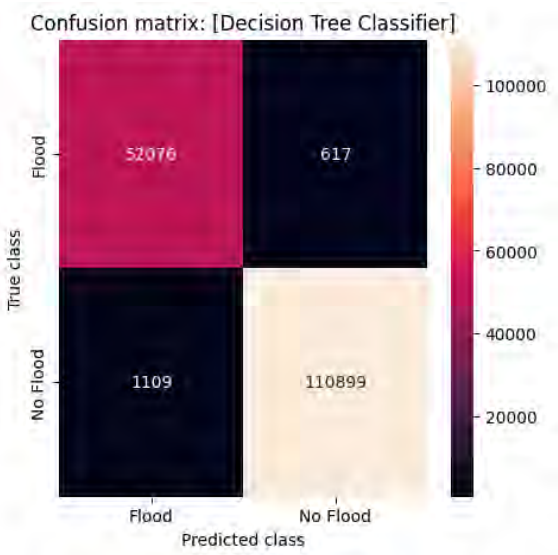
For the Ganges basin In the case of Decision tree, there are 27,073 true positives. These are cases in which the model predicted 'No Flood', and the actual class was also 'No Flood'. In this case, there are 501 true negatives. In this case, there are 0 false positives. In this case, there are 47 false negatives. K-Nearest Neighbor and Random Forest also showed similar results, but Logistic regression shows 0 false positives and true negatives while Gaussian Naive Bayes shoes some false positives and false negatives.

4.4.4 Meghna Basin

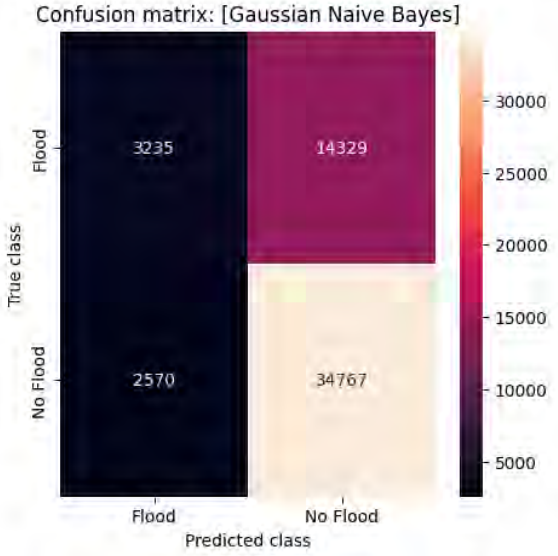
Confusion Matrices of Decision Tree, Random Forest and K-Nearest Neighbor for the Meghna Basin shows 0 false positives and false negatives with, 26031 true positives and 903 true negatives which is most likely due to overfitting of the data. Gaussian Naive Bayes and logistic regression performs better in this regard, showing a spread of 1154 and 103 false positives respectively.

4.4.5 South East Hill Basin

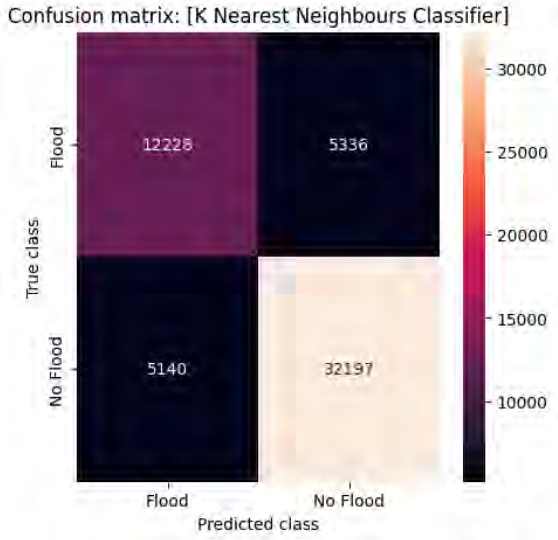
For the South East Hill Basin In the case of Decision tree, there are 41966 true positives with 52 false positives and 52 false positives. K-Nearest Neighbor, Random Forest and Logistic Regression gave similar results for positives but Logistic regression and Gaussian Naive Bayes gave significant results for true negatives with 3610 and 29887 respectively whereas other models showed a range of 302 to 366.
The model has high accuracy in predicting flood conditions, with a significant number of true positives and relatively few false negatives. However, it seems to struggle with predicting 'No Flood' conditions accurately, as indicated by the low number of true negatives and the absence of false positives.
This might be due to the model being well-tuned to recognize patterns associated with flooding conditions, but may miss some instances due to noise or other unpredictable factors. It could also be that the 'No Flood' class is underrepresented in the training data, leading to a bias in the model towards predicting floods. This is a common challenge in machine learning and can be addressed through techniques like resampling or using different evaluation metrics that are more robust to class imbalance.
From the available data, it appears that the Gaussian Naive Bayes model has a balance in predicting both 'Flood' and 'No Flood' events, but with some errors. On the other hand, the Logistic Regression model perfectly predicts 'Flood' events but fails to predict any 'No Flood' events.
This might be due to the models being trained differently or having different assumptions about the data. For example, the Gaussian Naive Bayes model assumes that the features are independent given the class, while the Logistic Regression model does not have this assumption. If this assumption is violated in the data, it could lead to poorer performance for the Gaussian Naive Bayes model. On the other hand, if the Logistic Regression model is overfitting to the 'Flood' class, it might fail to generalize well to the 'No Flood' class. These are just potential explanations and would need to be investigated further.



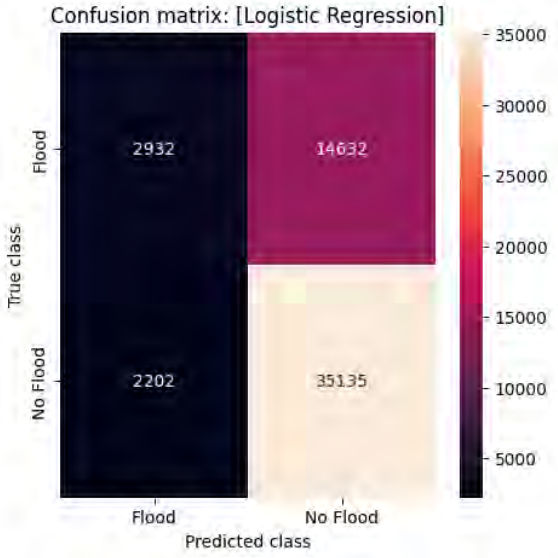
(a) Decision Tree Classifier



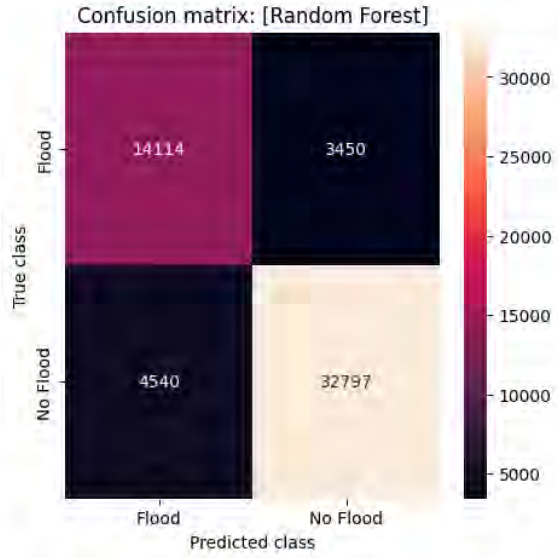
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor

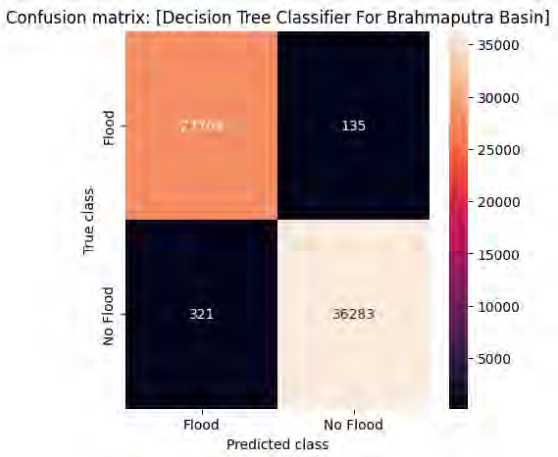


(d) Logistic Regression

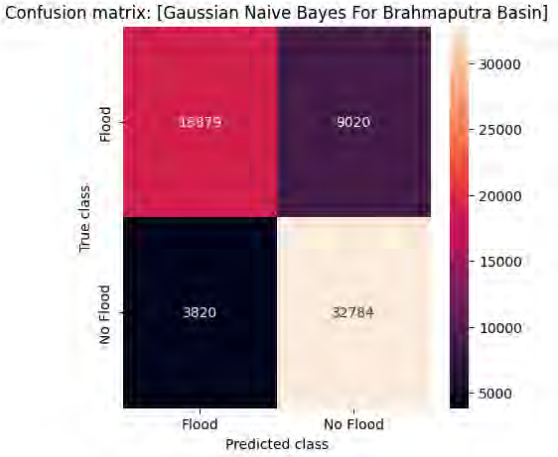


(e) Random Forest

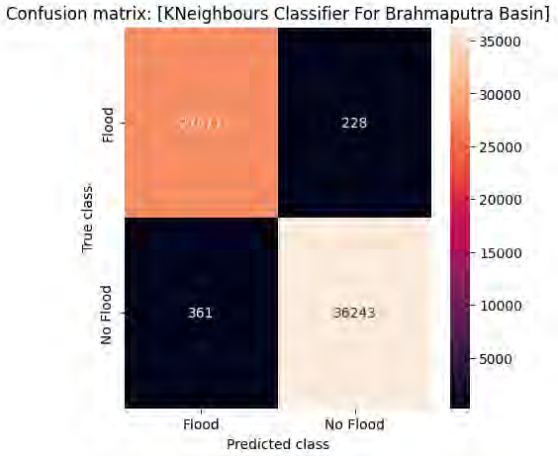
Figure 4.3: Confusion Matrices of the Entire Dataset



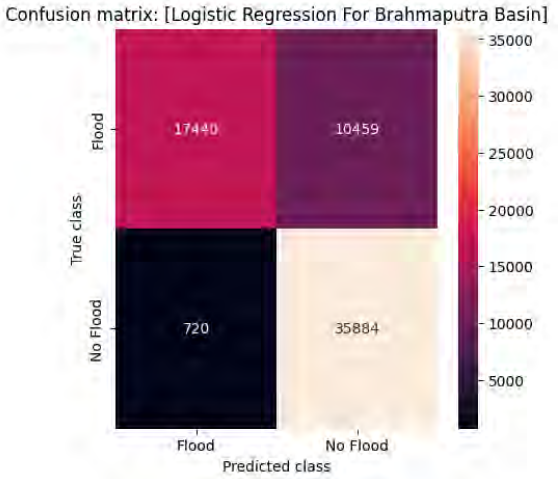
(a) Decision Tree Classifier



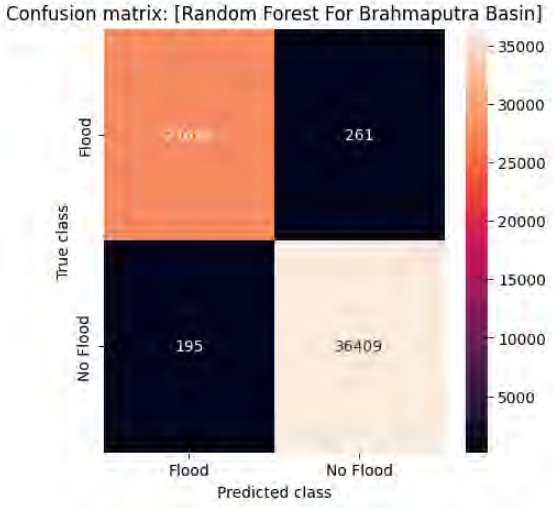
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor

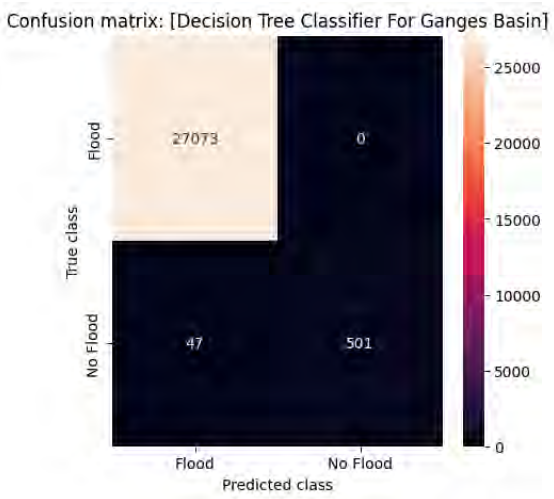


(d) Logistic Regression

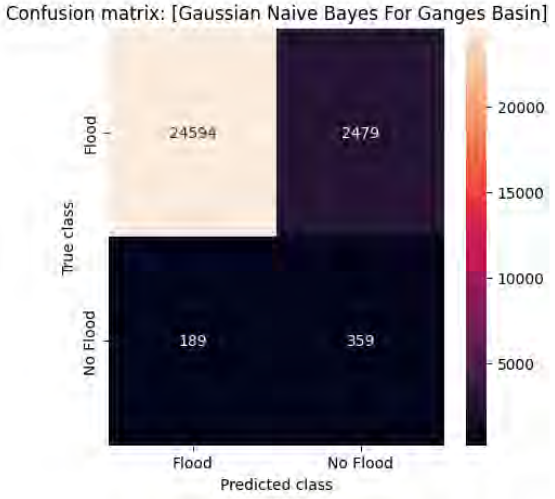


(e) Random Forest

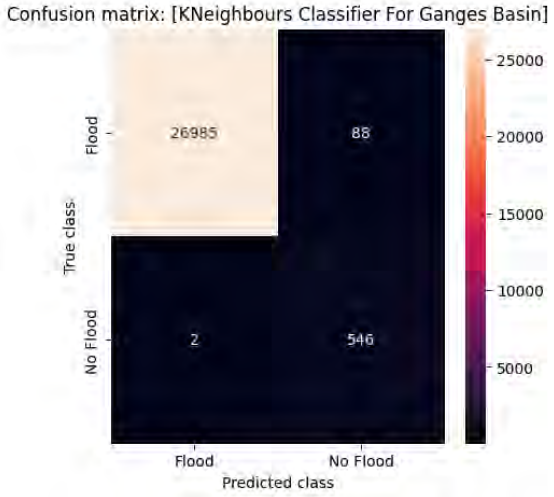
Figure 4.4: Confusion Matrices of Brahmaputra Basin



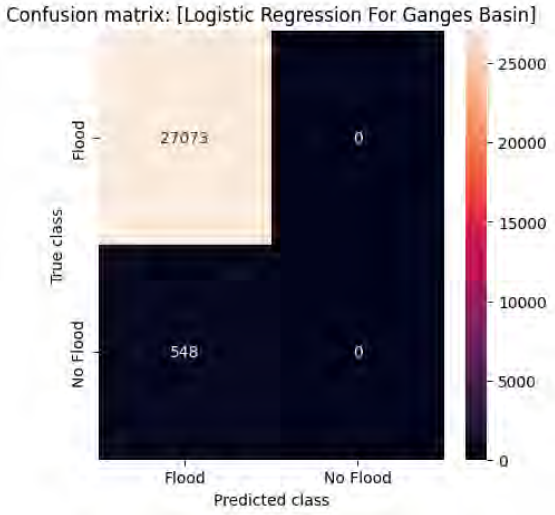
(a) Decision Tree Classifier



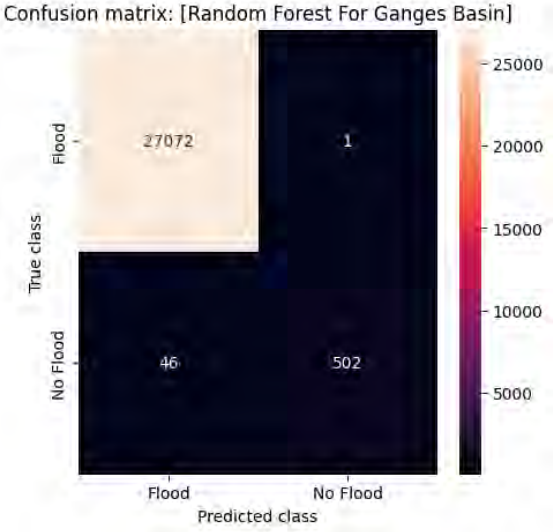
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor

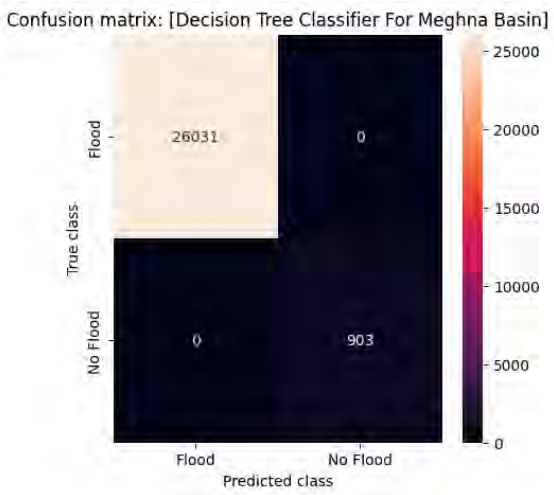


(d) Logistic Regression

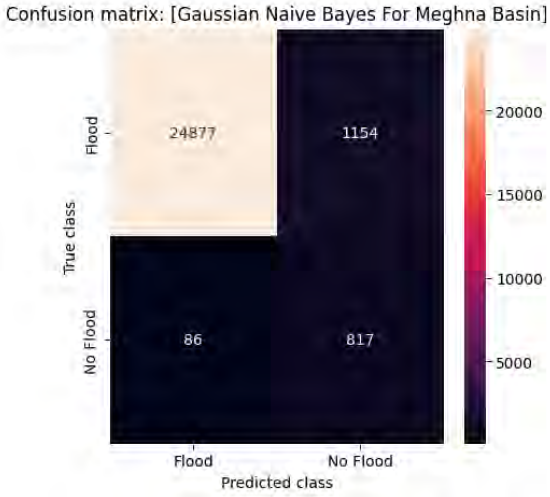


(e) Random Forest

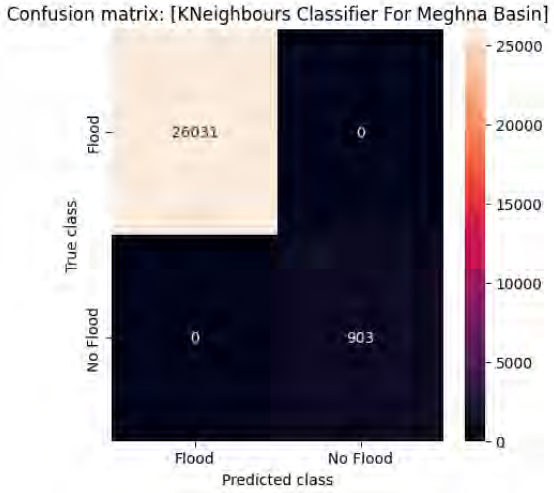
Figure 4.5: Confusion Matrices of Ganges Basin



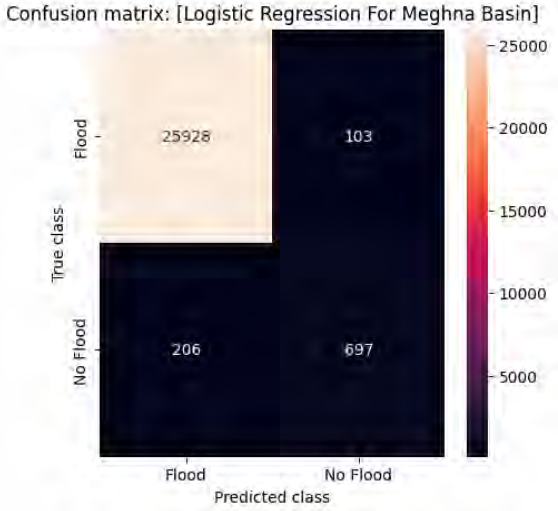
(a) Decision Tree Classifier



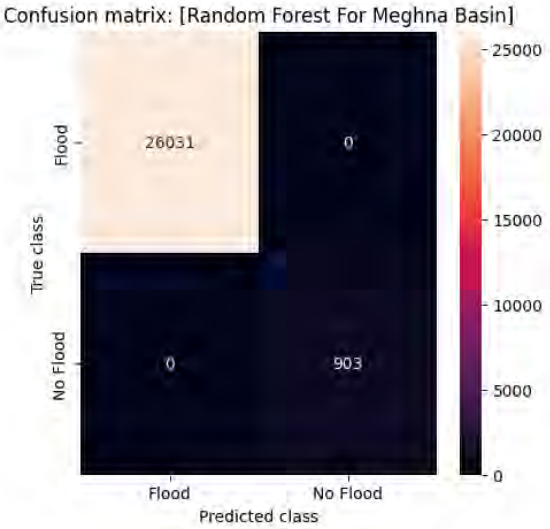
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor

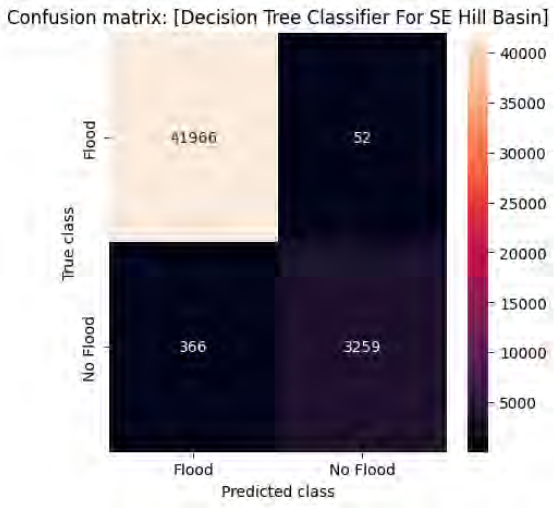


(d) Logistic Regression

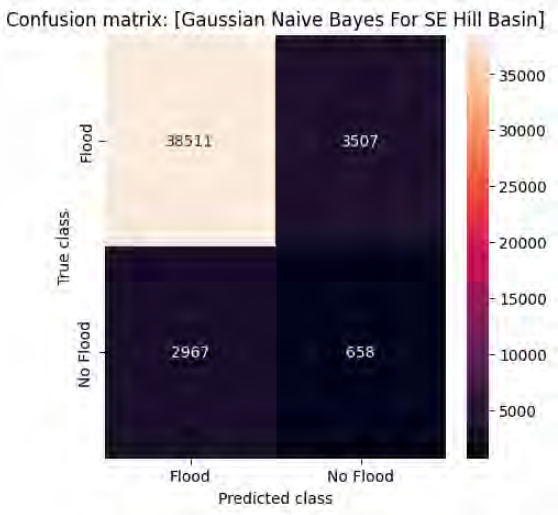


(e) Random Forest

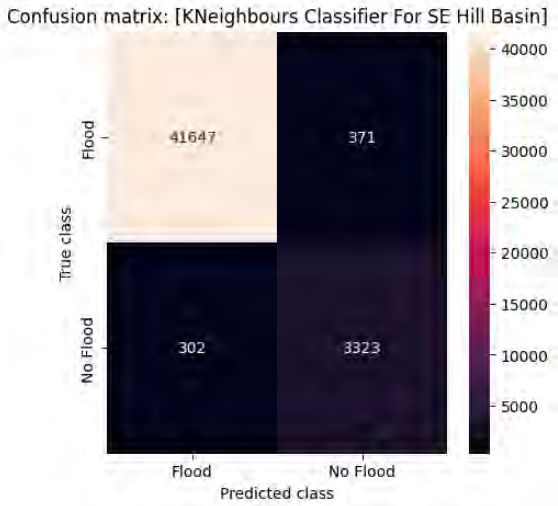
Figure 4.6: Confusion Matrices of Meghna Basin



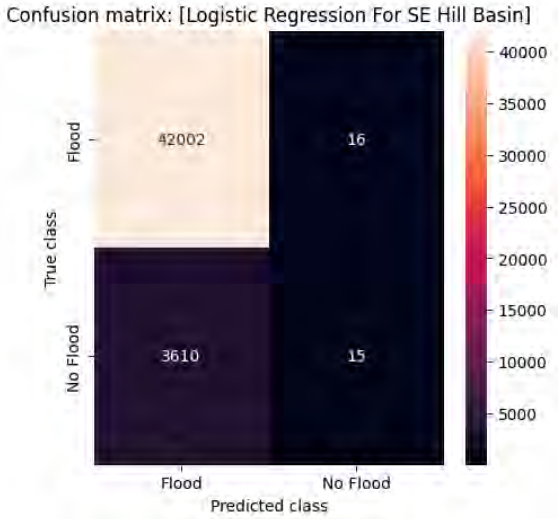
(a) Decision Tree Classifier



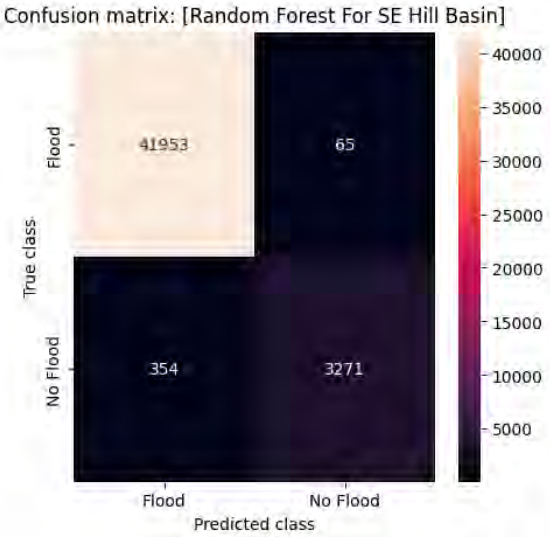
(b) Gaussian Naive Bayes



(c) K-Nearest Neighbor



(d) Logistic Regression



(e) Random Forest

Figure 4.7: Confusion Matrices of South East Hill Basin

Chapter 5

Conclusion

5.1 Conclusion

B. T. Pham et al. [30] used map data to run Decision Tree algorithms on Vietnam flood history, achieved a result of 93.4%. This is slightly higher than ours but not by a substantial amount and can be attributed to the difference in data types and dataset size. However this points to the reliability and usefulness of Decision Tree as a flood prediction algorithm. Logistic regression was used in the 2018 work by Al-Juaidi et al.[19] published in the Arabian Journal of Geosciences to map the southern Gaza Strip's flood risk. Using variables including rainfall, land use, soil type, and altitude, it examined 140 flood areas. With a prediction success rate of 76% and 81%, the model offered helpful data for reducing flood damage in the area. Comparatively speaking, the results were not as accurate as we had anticipated. Using this specific model in our work, we were able to get accuracy ranging from 69.3% to 90.9%.

The dataset that comprised the whole of Bangladesh acts as a control group for the purposes of testing the different machine learning algorithms that were used. As imputation and other preprocessing methods take the dataset in its entirety into account, the effects of distant weather stations on each other is high and leads to a large amount of noise in the dataset which greatly hampers the performance of the algorithms. When the dataset is split into four separate subsets that are grouped respective of their geographical location, the noise is greatly reduced and thus we see that every algorithm is more accurate when using the divided data compared to the data in whole. This, however, revealed an issue. The Meghna dataset performs too well, scoring 99-100% on nearly every algorithm. This is an outlier and must be discarded. Due to the fact that the Meghna basin readings comprised the smallest portion of the full dataset and that as well due to the very low number of stations in that region to take readings in the first place, the algorithms were overfitted to the dataset and do not accurately reflect their usefulness. Only Gaussian Naive Bayes managed to not overfit the data and gave a proper result. Taking the outliers out the equation we can see the following: Decision Tree Classifier for the basins has an average accuracy of 90.8% which is an increase of 7.1% compared to the full dataset; K Nearest Neighbours saw its accuracy rise from 80.9% to 89.6% an increase of 8.6%; Logistic Regressions' accuracy rose greatly from 69.3% to 90.9%; Naive Bayes had an average accuracy of 87.9% which is up 18.7%; and finally Random Forest saw great increases from 85.6% to 92.8%, which is the highest average accuracy amongst

all the algorithms. Thus we see that Random Forest is a prime candidate for use in flood prediction, with Logistic Regression and Decision Tree tied in second place.

5.2 Future Work

For future work and further improvements in flood prediction using machine learning in Bangladesh, several avenues can be explored to enhance the effectiveness and applicability of the models. Integrating data from other sources is crucial. Examples of these include satellite images, real-time meteorological data, and geographical information system (GIS) data that includes topographical features and soil moisture content. These many data sources can greatly enhance the prediction power of the model.Because deep learning approaches can handle complex time series data, they should be taken into consideration. In particular, Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), such as Long Short-Term Memory (LSTM) networks, are strong candidates. Flood forecasts made with these cutting-edge techniques might be more precise and trustworthy. It may also be possible to achieve a balance between understanding and prediction capability by creating hybrid models that integrate the advantages of several methods, such as decision trees and neural networks.
Better feature engineering efforts can result in the discovery of new or more important features, which will enhance the performance of the model. It would also be beneficial to focus on real-time prediction systems, which would allow for processing incoming data for accurate and fast flood forecasts. Equally vital are the implementation's practical features, which focus on flexibility and compatibility with current disaster management systems. This involves taking into account how the models will be updated and maintained when new data becomes available. To better adapt the models to particular geographical demands and increase their practical usefulness, cooperation with Bangladeshi local government agencies and communities is crucial. Additionally, this partnership may yield constructive criticism for additional improvements. Finally, it is critical to take into account how flood patterns are affected by climate change. The models must to be flexible enough to adjust to changing circumstances and able to make reliable forecasts in light of shifting climatic patterns. To guarantee the long-term sustainability and efficiency of flood prediction models, this kind of foresight is required. A wider range of data sources should be combined, advanced machine learning and deep learning techniques should be used, the clarity of the models should be improved, and the models' practical applicability and societal influence should be guaranteed. Making these flood prediction models an effective tool for disaster management and mitigation in Bangladesh will also require close cooperation with local authorities and careful assessment of the effects of climate change.

Bibliography

[1] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[2] D. Anderson and G. McNeill, "Artificial neural networks technology," *Kaman Sciences Corporation*, vol. 258, no. 6, pp. 1–83, 1992.

[3] J. R. Quinlan, "C4. 5: Programming for machine learning. morgan kauffmann (1993)," *URL: https://dl. acm. org/doi/10.5555/152181*, 1993.

[4] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[5] M. Campolo, P. Andreussi, and A. Soldati, "River flood forecasting with a neural network model," *Water resources research*, vol. 35, no. 4, pp. 1191–1197, 1999.

[6] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[7] C. Wu and K. W. Chau, "A flood forecasting neural network model with genetic algorithm," *International journal of environment and pollution*, vol. 28, no. 3-4, pp. 261–273, 2006.

[8] D. Han, L. Chan, and N. Zhu, "Flood forecasting using support vector machines," *Journal of hydroinformatics*, vol. 9, no. 4, pp. 267–276, 2007.

[9] M. Sharif and D. H. Burn, "Improved k-nearest neighbor weather generating model," *Journal of Hydrologic Engineering*, vol. 12, no. 1, pp. 42–51, 2007.

[10] J.-L. Polo, F. Berzal, and J.-C. Cubero, "Class-oriented reduction of decision tree complexity," in *Foundations of Intelligent Systems: 17th International Symposium, ISMIS 2008 Toronto, Canada, May 20-23, 2008 Proceedings 17*, Springer, 2008, pp. 48–57.

[11] S. Deegalla and H. Bostrom, "Fusion of dimensionality reduction methods: A case study in microarray classification," in *2009 12th International Conference on Information Fusion*, IEEE, 2009, pp. 460–465.

[12] L.-H. Feng and J. Lu, "The practical research on flood forecasting based on artificial neural networks," *Expert systems with Applications*, vol. 37, no. 4, pp. 2974–2977, 2010.

[13] G.-F. Lin, Y.-C. Chou, and M.-C. Wu, "Typhoon flood forecasting using integrated two-stage support vector machine approach," *Journal of Hydrology*, vol. 486, pp. 334–342, 2013.

[14] X.-L. Wu, X.-H. Xiang, C.-H. Wang, X. Chen, C.-Y. Xu, and Z. Yu, "Coupled hydraulic and kalman filter model for real-time correction of flood forecast in the three gorges interzone of yangtze river, china," *Journal of Hydrologic Engineering*, vol. 18, no. 11, pp. 1416–1425, 2013.

[15] S. H. Elsafi, "Artificial neural networks (anns) for flood forecasting at dongola station in the river nile, sudan," *Alexandria Engineering Journal*, vol. 53, no. 3, pp. 655–662, 2014, ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2014.06.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016814000660.

[16] G. K. Devia, B. Ganasri, and G. Dwarakish, "A review on hydrological models," *Aquatic Procedia*, vol. 4, pp. 1001–1007, 2015, INTERNATIONAL CONFERENCE ON WATER RESOURCES, COASTAL AND OCEAN ENGINEERING (ICWRCOE'15), ISSN: 2214-241X. DOI: https://doi.org/10.1016/j.aqpro.2015.02.126. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214241X15001273.

[17] H. K. Sok, M. P.-L. Ooi, and Y. C. Kuang, "Sparse alternating decision tree," *Pattern Recognition Letters*, vol. 60, pp. 57–64, 2015.

[18] S. Wang, L. Jiang, and C. Li, "Adapting naive bayes tree for text classification," *Knowledge and Information Systems*, vol. 44, pp. 77–89, 2015.

[19] A. E. Al-Juaidi, A. M. Nassar, and O. E. Al-Juaidi, "Evaluation of flood susceptibility mapping using logistic regression and gis conditioning factors," *Arabian Journal of Geosciences*, vol. 11, pp. 1–10, 2018.

[20] K. Khosravi, B. T. Pham, K. Chapi, *et al.*, "A comparative assessment of decision trees algorithms for flash flood susceptibility modeling at haraz watershed, northern iran," *Science of the Total Environment*, vol. 627, pp. 744–755, 2018.

[21] J. Yan, J. Jin, F. Chen, G. Yu, H. Yin, and W. Wang, "Urban flash flood forecast using support vector machine and numerical simulation," *Journal of Hydroinformatics*, vol. 20, no. 1, pp. 221–231, 2018.

[22] A. Y. Felix and T. Sasipraba, "Flood detection using gradient boost machine learning approach," in *2019 International conference on computational intelligence and knowledge economy (ICCIKE)*, IEEE, 2019, pp. 779–783.

[23] M. I. H. Jati, P. B. Santoso, *et al.*, "Prediction of flood areas using the logistic regression method (case study of the provinces banten, dki jakarta, and west java)," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1367, 2019, p. 012 087.

[24] Q. Ke, X. Tian, J. Bricker, *et al.*, "Urban pluvial flooding prediction by machine learning approaches–a case study of shenzhen city, china," *Advances in Water Resources*, vol. 145, p. 103 719, 2020.

[25] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, and A. Krishnan, "Flood prediction based on weather parameters using deep learning," *Journal of Water and Climate Change*, vol. 11, no. 4, pp. 1766–1783, 2020.

[26] P. Yariyan, S. Janizadeh, T. Van Phong, *et al.*, "Improvement of best first decision trees using bagging and dagging ensembles for flood probability mapping," *Water Resources Management*, vol. 34, pp. 3037–3053, 2020.

[27] N. Gauhar, S. Das, and K. S. Moury, "Prediction of flood in bangladesh using k-nearest neighbors algorithm," in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, 2021, pp. 357–361. DOI: 10.1109/ICREST51555.2021.9331199.

[28] K. Kunverji, K. Shah, and N. Shah, "A flood prediction system developed using various machine learning algorithms," in *Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021)*, 2021.

[29] J. Lee and B. Kim, "Scenario-based real-time flood prediction with logistic regression," *Water*, vol. 13, no. 9, p. 1191, 2021.

[30] B. T. Pham, A. Jaafari, T. Van Phong, *et al.*, "Improved flood susceptibility mapping using a best first decision tree integrated with ensemble learning techniques," *Geoscience Frontiers*, vol. 12, no. 3, p. 101 105, 2021.

[31] D. Devi, S. K. Biswas, and B. Purkayastha, "Correlation-based oversampling aided cost sensitive ensemble learning technique for treatment of class imbalance," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 34, no. 1, pp. 143–174, 2022.

[32] S. B. Nadkarni, G. Vijay, and R. C. Kamath, "Comparative study of random forest and gradient boosting algorithms to predict airfoil self-noise," *Engineering Proceedings*, vol. 59, no. 1, p. 24, 2023.

[33] S. Bangladesh Water Development Board (BWDB), *Flood forecasting warning centre*. [Online]. Available: http://www.ffwc.gov.bd/index.php/definitions.

[34] S. Bangladesh Water Development Board (BWDB), *Flood forecasting warning centre, annual flood report 2020*. [Online]. Available: http://www.ffwc.gov.bd/images/annual20.pdf.