VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Belagavi – 590018

A Project Report on

FLOOD PREDICTION USING AI

Submitted by

AJAY KARTHIK	4SN17CS700
CHAITANYA THEKKUNJA	4SN17CS711
KISHAN KASHYAP	4SN17CS714
SHUBHANG	4SN17CS723

In partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Under the Guidance of

Dr. SHIVAKUMAR GS

Head of the Department

Department of Computer Science & Engineering

SRINIVAS INSTITUTE OF TECHNOLOGY

(NAAC ACCREDITED)

MANGALURU - 574143, KARNATAKA.

2020 - 2021

SRINIVAS INSTITUTE OF TECHNOLOGY

(NAAC ACCREDITED)

MANGALURU - 574143, KARNATAKA.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

*Certified that the project work entitled "**FLOOD PREDICTION USING AI"** is a bona fide work carried out by*

AJAY KARTHIK	**4SN17CS700**
CHAITANYA THEKKUNJA	**4SN17CS711**
KISHAN KASHYAP	**4SN17CS714**
SHUBHANG	**4SN17CS723**

*in partial fulfillment for the award of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE & ENGINEERING** of the **VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI** during the year 2020 – 2021. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the Bachelor of Engineering Degree.*

Dr. Shivakumar G. S.
Head of the Department

Dr. Shrinivasa Mayya D.
Principal

Name of the Examiners **Signature with Date**

1.

2.

ABSTRACT

Floods are among the most destructive natural disasters, which are highly complex to model. The research on the advancement of flood prediction models contributed to risk reduction, policy suggestion, minimization of the loss of human life, and reduction the property damage associated with floods. To mimic the complex mathematical expressions of physical processes of floods, during the past two decades, machine learning (ML) methods contributed highly in the advancement of prediction systems providing better performance and cost-effective solutions. Due to the vast benefits and potential of ML, its popularity dramatically increased among hydrologists. Researchers through introducing novel ML methods and hybridizing of the existing ones aim at discovering more accurate and efficient prediction models.

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely fortunate to have got their support all along the completion of our project.

We take this opportunity to express our profound gratitude and deep regards to our Project Guide and Project Coordinator **Dr. Shivakumar G. S.**, Head of the Department, Department of Computer Science and Engineering, for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. He has communicated various ideas for improving the project and has been especially enthusiastic in giving his opinions and critical reviews in a constructive manner. We will remember his contribution forever.

We express a deep sense of gratitude to **Dr. Shivakumar G. S.**, Head of the Department, Computer Science and Engineering, for his cordial support, valuable information and guidance, which helped us in completing this project through various stages.

We also express our heartfelt gratitude to our Principal **Dr. Shrinivasa Mayya D.** for his kind co-operation and encouragement which helped us in the completion of this project.

We also thank our **Management** who helped us directly and indirectly for the successful completion of our project.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all the **Teaching and Non-teaching staff** of Department of Computer Science and Engineering who helped us in successfully completing our project.

Lastly, we would like to thank our **Parents** for their moral support and our Friends with whom we shared our day-to-day experiences and received lots of suggestions that improved our quality of work.

Ajay Karthik

Chaitanya Thekkunja

Kishan Kashyap

Shubhang

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Flood prediction is the use of forecasted precipitation and stream-flow data in rainfall-runoff and stream-flow routing models to forecast flow rates and water levels for periods ranging from a few hours to days ahead, depending on the size of the watershed or river basin. Flood prediction can also make use of forecasts of precipitation in an attempt to extend the lead-time available. Flood prediction is an important component of flood warning, where the distinction between the two is that the outcome of flood prediction is a set of forecast time-profiles of channel flows or river levels at various locations, while "flood warning" is the task of making use of these forecasts to tell decisions on warnings of floods.

Real-time flood prediction at regional area can be done within seconds by using the technology of artificial neural network. Effective real-time flood prediction models could be useful for early warning and disaster prevention.

1.1 Problem Statement

Among the natural disasters, floods are the most destructive, causing massive damage to human life, infrastructure, agriculture, and the socioeconomic system. Governments, therefore, are under pressure to develop reliable and accurate maps of flood risk areas and further plan for sustainable flood risk management focusing on prevention, protection, and preparedness.

1.2 Existing System

Physically based models were long used to predict hydrological events, such as storm, rainfall/runoff, shallow water condition, hydraulic models of flow, and further global circulation phenomena, including the coupled effects of atmosphere, ocean, and floods. Although physical models showed great capabilities for predicting a diverse range of flooding scenarios, they often require various types of hydro-geomorphological monitoring datasets, requiring intensive computation, which prohibits short-term prediction. Numerous studies suggest that there is a gap in short-term prediction capability of physical models. For instance, on many occasions, such models failed to predict properly.

1.3 Proposed System

This system aims to collect data from all the states of India and form a generalized dataset. A machine learning algorithm is applied to the labelled dataset, and patterns are extracted, which, in turn, obtain maximum accuracy with real-time input. In general, the dataset collected for predicting is split into a Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model is then created using a Multi-Layer Perceptron Classifier, and the resulting data set is then passed through it for prediction.

1.4 Objective

The objective of Flood Prediction using AI is to design a incremental model to predict floods based on rainfall levels, channel flows, river levels etc. An approach of prediction is using Artificial Neural Networks that has very good working efficiency produces the accurate results. The system helps to improve the performance. This method can be used to predict both long term and short term floods in a given region. Maintaining the project is easy and manageable.

CHAPTER 2

LITERATURE SURVEY

[1] Prediction Analysis of Floods Using Machine Learning Algorithms (NARX & SVM)

The changing patterns and behaviors of river water levels that may lead to flooding are an interesting and practical research area. They are configured to mitigate economic and societal implications brought about by floods. Non-linear (NARX) and Support Vector Machine (SVM) are machine learning algorithms suitable for predicting changes in levels of river water, thus detection of flooding possibilities. The two algorithms employ similar hydrological and flood resource variables such as precipitation amount, river inflow, peak gust, seasonal flow, flood frequency, and other relevant flood prediction variables. In the process of predicting floods, the water level is the most important hydrological research aspect. Prediction using machine-learning algorithms is effective due to its ability to utilize data from various sources and classify and regress it into flood and non-flood classes. This paper gives insight into mechanism of the two algorithms in perspective of flood estimation.

[2] Streamflow Prediction Using Deep Learning Neural Network

The most important motivation for streamflow forecasts is flood prediction and longtime continuous prediction in hydrological research. As for many traditional statistical models, forecasting flood peak discharge is nearly impossible. They can only get acceptable results in normal year. On the other hand, the numerical methods including physics mechanisms and rainfall-atmospherics could provide a better performance when floods coming, but the minima prediction period of them is about one month ahead, which is too short to be used in hydrological application. In this study, a deep neural network was employed to predict the streamflow of the Yangtze River. This method combined the Empirical Mode Decomposition (EMD) algorithm and Encoder Decoder Long Short-Term Memory (En-De-LSTM) architecture. Owing to the hydrological series prediction problem usually contains several different frequency components, which will affect the precision of the longtime prediction. The EMD technique could read and decomposes the original data into several different frequency components. It will help the model to make longtime predictions more efficiently.

[3] Forecasting Daily Precipitation Using Hybrid Model of Wavelet-Artificial Neural Network

Recently artificial neural network (ANN) as a nonlinear interextrapolator is extensively used by hydrologists for precipitation modeling as well as other fields of hydrology. In the present study, wavelet analysis combined with artificial neural network and finally was compared with adaptive neurofuzzy system to predict the precipitation in Iran. For this purpose, the original time series using wavelet theory decomposed to multiple subtime series. Then, these subseries were applied as input data for artificial neural network, to predict daily precipitation, and compared with results of adaptive neurofuzzy system. The results showed that the combination of wavelet models and neural networks has a better performance than adaptive neurofuzzy system, and can be applied to predict both short- and long-term precipitations.

[4] Rainfall Prediction Using Hybrid Adaptive Neuro-Fuzzy Inference System (ANFIS) and Genetic Algorithm

The current rainy season is erratic and very difficult to predict the rain. It requires a method that can predict rainfall with the smallest error as possible. Adaptive Neuro-Fuzzy Inference System (ANFIS) is one of the prediction methods that are quite reliable because it is equipped with a network that can learn. The ANFIS uses Sugeno FIS in its architecture. To improve the prediction results, the Sugeno FIS will be optimized in boundaries of membership function and coefficient consequent rule before it goes into the process of training with ANFIS. A genetic algorithm is used for the optimization process. The results of rainfall prediction using hybrid ANFIS-GA are proven to produce smaller RMSE of rainfall prediction method that has never been done before. With two optimization process in the boundaries of membership function with genetic algorithm and the training process with ANFIS, RMSE values obtained from the rainfall prediction becomes lower. It can be concluded that the results of rainfall prediction using the hybrid method ANFIS-GA produce smaller RMSE compared to the previous methods such as GSTAR-SUR, Tsukamoto FIS, and hybrid Tsukamoto FIS with GA.

[5] Bayesian flood forecasting method

Bayesian forecasting system (BFS) offers an ideal theoretic framework for uncertainty quantification that can be developed for probabilistic flood forecasting via any deterministic hydrologic model. It provides suitable theoretical structure, empirically validated models and reasonable analytic-numerical computation method, and can be developed into various Bayesian forecasting approaches. This paper presents a comprehensive review on Bayesian forecasting approaches applied in flood forecasting from 1999 till now. Results show that the Bayesian flood forecasting approach is an effective and advanced way for flood estimation, it considers all sources of uncertainties and produces a predictive distribution of the river stage, river discharge or runoff, thus gives more accurate and reliable flood forecasts. Some emerging Bayesian forecasting methods were shown to overcome limitations of single model or fixed model weight and effectively reduce predictive uncertainty. In recent years, various Bayesian flood forecasting approaches have been developed and widely applied, but there is still room for improvements.

[6] Comparison of random forests and support vector machine for rainfall forecasting

This study aims to compare two machine learning techniques, random forests (RF) and support vector machine (SVM), for real-time radar-derived rainfall forecasting. The real-time radar-derived rainfall forecasting models use the present grid-based radar-derived rainfall as the output variable and use antecedent grid-based radar-derived rainfall, grid position (longitude and latitude) and elevation as the input variables to forecast 1-h to 3-h ahead rainfalls for all grids in a catchment. Grid-based radar-derived rainfalls of six typhoon events during 2012–2015 in three reservoir catchments are collected for model training and verifying. Two kinds of forecasting models are constructed and compared, which are single-mode forecasting model (SMFM) and multiple-mode forecasting model (MMFM) based on RF and SVM. The SMFM uses the same model for 1-h to 3-h ahead rainfall forecasting; the MMFM uses three different models for 1-h to 3-h ahead forecasting. According to forecasting performances, it reveals that the SMFMs give better performances than MMFMs and both SVM-based and RF-based SMFMs show satisfactory performances for 1-h ahead forecasting. However, for 2- and 3-h ahead forecasting, it is found that the RF-based SMFM underestimates the observed radar-derived rainfalls in most cases and the SVM-based SMFM can give better performances than RF-based SMFM.

CHAPTER 3

SOFTWARE REQUIREMENT ANALYSIS

Software Requirement Analysis in the field of systems engineering and software engineering, encompasses those tasks that are used for determining the needs or conditions to meet for a new or altered product or project, taking account of the possibly conflicting requirements of the various stakeholders, analysing, documenting, validating and managing software or system requirements.

3.1 Feasibility Study

The main objective of the feasibility study is to treat the technical, operational and economic feasibility of developing the application. Feasibility is the determination of whether or not project is worth doing. The process followed in making this determination is called feasibility study. All systems are feasible, given unlimited resources and infinite time. The feasibility study to be conducted for this project involves:

- Technical Feasibility
- Operational Feasibility
- Economic Feasibility

3.1.1 Technical Feasibility

It is the measure of the specific technical solution and the availability of the technical resources and expertise. It is one of the first studies that must be conducted after a project has been identified. A technical study of feasibility is an assessment of the logistical aspects of business operation. This is considered with specifying equipment and software that will successful satisfy the user requirement. The technical needs of the system may vary considerably but should include the facility to produce outputs in a given time, response time under certain conditions and the ability to process a certain amount of transaction at a certain speed.

The proposed system is developed by using Juypter Notebook software. The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. The "notebook" term can colloquially make reference to many different entities, mainly the Jupyter web application, Jupyter Python web server, or Jupyter document format depending on context. A Jupyter Notebook document is a JSON document, following a versioned schema, containing an ordered list of input/output cells which can contain code, text, mathematics, plots and rich media.

3.1.2 Operational Feasibility

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented, whether there will be resistance from the users which will affect the possible application benefits. It is the ability to utilize, support and perform the necessary tasks of a system or program. It includes everyone who creates, operates or uses the system. It is the measure of how well a proposed system solves the problem and takes advantages of the opportunities identified during the scope definition and problem analysis phases. This system helps in many ways. It reduces the burden of maintaining bulk of records of all the rainfall data. Maintenance of the project is also easy and understandable and no major training and new skills are required.

3.1.3 Economic Feasibility

Economic feasibility is the most frequently used method for evaluating the effectiveness of the new system. Economic feasibility is the measure of the cost effectiveness of an information system solution. Without a doubt, this measure is most often and important one of the three. Information systems are often viewed as capital investments for the business, and, as such, should be subjected to the same type of investment analyses as other capital investments.

Economic analysis is used for evaluating the effectiveness of the proposed system. In economic feasibility, the most important is cost-benefit analysis. This project is economical as it mainly depends on the software components which are freely available.

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

The main purpose of System Requirement Specification is to translate the ideas in the minds of a client into a formal document. Through System Requirement Specification the client clearly describes what it expects from the proposed system and the developer clearly understands what capabilities are required to build the system. It includes a variety of elements (see below) that attempts to define the intended functionality required by the customer to satisfy their different users. The purpose of this document is to serve as a guide to developers and testers who are responsible for the development of the system.

4.1 Functional Overview

- Rainfall data is collected from meteorological websites.
- An algorithm is used for creating a model.
- The chances of a flood are predicted.

4.2 Operating Environment

Operating environment involves minimum software and hardware requirements required by the system.

4.2.1 Software Requirements

- Operating System : Windows 7 or above.
- Tools used : Jupyter 6.0 or above.
- Programming Language : Python Programming Language.

4.2.2 Hardware Requirements

- Processor : Pentium 4 or above
- RAM : 6GB or above
- Hard Disk : 10GB or above
- Input device : Standard Keyboard and Mouse.
- Output device : High Resolution Monitor

4.3 Functional Requirements

Functional Requirements defines the function of a system or its component. A function is described as a set of inputs, the behavior and outputs. Functional requirements specify particular results of a system. Functional requirements drive the application architecture of a system. Following are the functional requirements used in the project.

- Process data in any required format i.e., xls,csv etc.
- Import the data and store its metadata relating to the segmentation without corrupting the contents.
- Concepts and target extraction.
- Predicting the outcome.

4.4 Non-Functional Requirements

Non-Functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. Non-Functional requirements are often called as quality attributes of a system. The following are the nonfunctional requirements of the application.

- Availability: The System will work as required according to the specified requirement.
- Reliability: The System has the ability to consistently perform the intended or required functions.
- Maintainability: The maintenance of a functional unit can be performed in accordance with the prescribed requirements
- Accessibility: The System can be accessed by any appropriate users.
- Operability: The System has the ability to keep itself safe and reliable, according to predefined operational requirements.
- Usability: The System is ease of use and learnable by the users.
- Responsiveness: The system has ability to respond to the user very fast as soon as the input is fed.
- Cost: The system is economically feasible.

4.5 Performance Requirements

This application system will avoid use of papers and going through each and every detail for each state or district. Thus, this application results in efficient usage of time. This application will be useful for individuals, city planners, researchers etc. Application calculates the result automatically and displays it to the user. Interface is designed in such a way that user can understand very easily. Maintenance of the project is easy and understandable.

CHAPTER 5

SYSTEM DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirement document. The design of a system is perhaps the most critical factor affecting the quality of the software, and has a major impact on the later phases, particularly testing and maintenance. The output of this phase is the design document. The design activity is often divided into two separate phases. They are system design and detailed design.

5.1 High Level Design

High-level design which is sometimes also called system design, aims to identify the modules that should be in the system, the specifications of these modules, and how they interact with each other to produce the desired results. At the end of system design all the major data structures, file formats, output formats, as well as the major modules in the system and their specifications are decided.

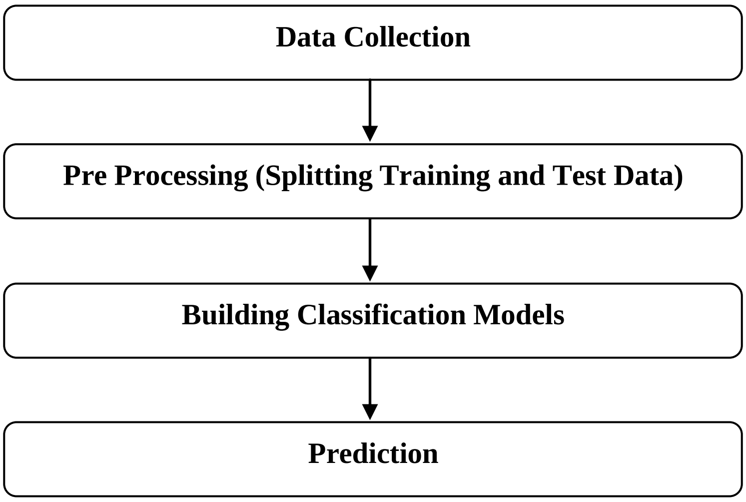

Figure 5.1: *Process Flow Diagram*

Figure 5.1 shows the process flow of Flood Prediction using AI. To get the required data, the system uses a request module which requests for the data from a repository using an API. The API will provide the data in a JSON format which is then converted to CSV format to be processed by the pandas library. The data is then cleaned up, such that any information that is not relevant, missing values, duplicate values etc. are ignored and only the relevant data is used for the next step which is splitting the data into two parts, training data and testing data. Here a ratio of 80:20 is used, where 80% of the data is used for training the model and 20% of the data is used for testing purpose. Then we use different algorithms to prepare a model for the given data. The models are then used to predict the probability of the flooding in a given region for a given month.

5.2 Detailed Design

During detailed design the internal logic of each of the modules specified in system design is decided. During this phase further details of the data structures and algorithmic design of each of the modules is specified. The logic of module is usually specified in a high-level design description language, which is independent of target language in which the software will eventually be implemented.

5.2.1 Use Case Diagram of Fruit Recognition using Image Processing

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. A use case diagram is a dynamic or behaviour diagram in UML. The use cases are represented by either circles or ellipses. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. Use case diagrams are valuable for visualizing the functional requirements of a system that will translate into design choices and development priorities. They also help identify any internal or external factors that may influence the system and should be taken into consideration.

There is only one actor that is user in the proposed system of Flood Prediction using AI. The user enters the desired region and time for the prediction of flood. After reading this data, the system will load these as parameters for pre-processing where only the data regarding the user’s choice is selected from the dataset and is used for processing. This is to increase the efficiency of the system such that it will use only the data which is relevant to the particular use case and nothing else. The data which is now pre-processed to remove irrelevant fields is then passed to each of the four algorithms, K – Nearest Neighbour, Logistic Regression, Support Vector Machine and Decision Tree. Each algorithm will output its prediction values along with the accuracy of the prediction.

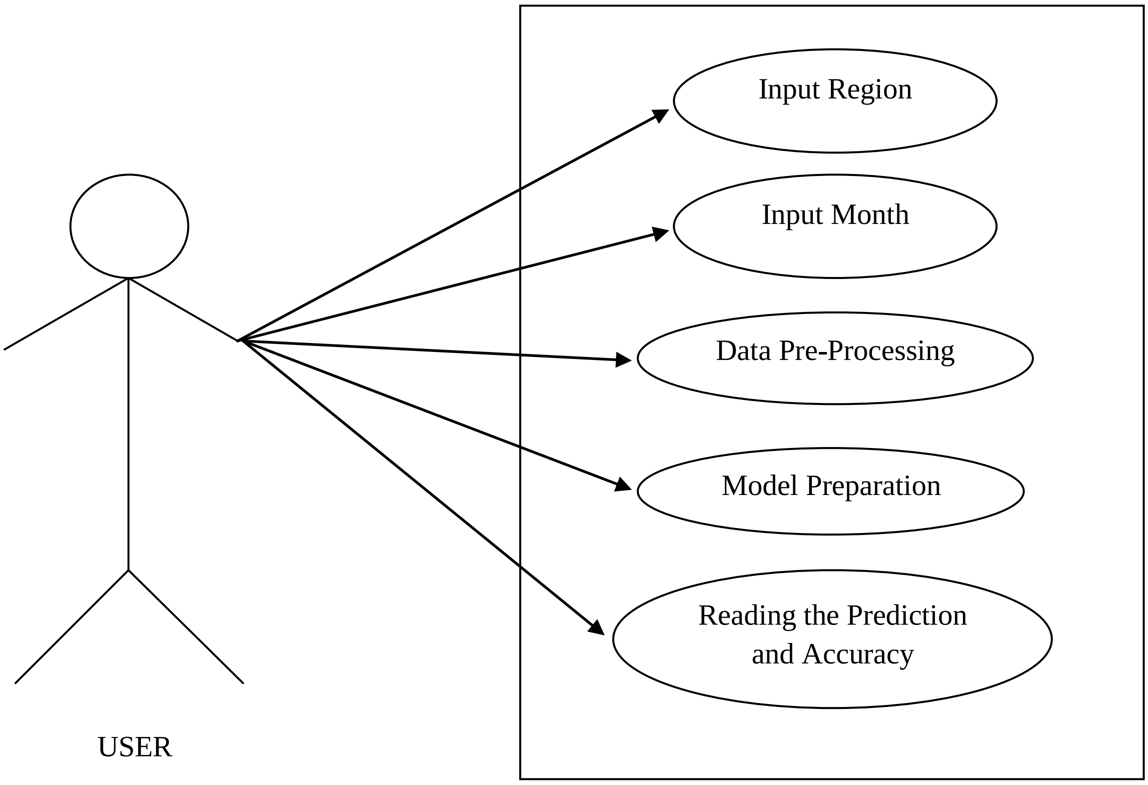

Figure 5.2: *Use Case Diagram for Floor Prediction using AI*

5.2.2 Data Flow Diagram of Fruit Recognition using Image Processing

A Data Flow Diagram (DFD) is a graph showing flow of data values from their sources in objects through processes that transform them to destination in other objects. A DFD also known as "bubble chart" has the purpose of clarifying the system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. The bubbles represent data transformations and the lines represent data flows in the system. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

Figure 5.3 shows the data flow diagram for Flood Prediction using AI. User can give input in the form of region and time of year. These values are captured and are passed to the system where it will pick the data related to the chosen region and time of year. Data is extracted from the dataset and is passed to the models for prediction. The algorithms take the input data and predict the output values which are then compared with the test data for validation and to measure accuracy of the algorithm. The predication and the accuracy are then shown to the user as output.

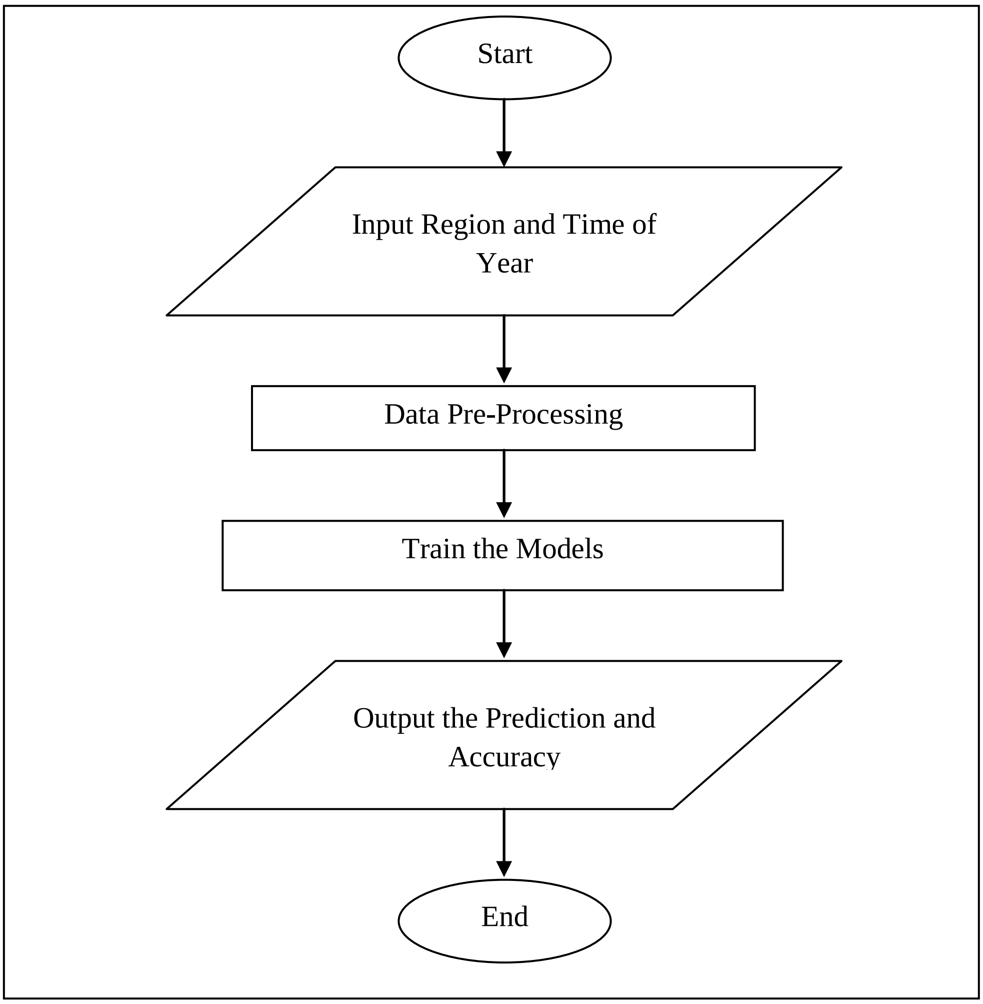

Figure 5.3: *Data Flow Diagram for Flood Predication using AI*

CHAPTER 6

SYSTEM IMPLEMENTATION

System Implementation is the stage where the theoretical design is converted into a working system, the new system may be totally new, replacing an existing manual, or automated system or it may be a major modification to an existing system. The system is implemented using Python and rainfall data set.

6.1 Programming Languages and Libraries Used

Python

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta-programming and meta-objects (magic methods)). Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Requests

Requests is a python module that allows us to send HTTP/1.1 requests extremely easily. There’s no need to manually add query strings to your URLs, or to form-encode your PUT & POST data.

We use this module to make a API request to the data set provider and get the dataset as a response.

Scikit-learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

Scikit-learn uses NumPy extensively for high-performance linear algebra and array operations. Furthermore, some core algorithms are written in Cython to improve performance. Support vector machines are implemented by a Cython wrapper; logistic regression and linear support vector machines by a similar wrapper. Scikit-learn integrates well with many other Python libraries, such as Matplotlib and plotly for plotting, NumPy for array vectorization, Pandas dataframes, SciPy, and many more.

Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

Pyqt5

PyQt5 is the latest version of a GUI widgets toolkit developed by Riverbank Computing. It is a Python interface for Qt, one of the most powerful, and popular cross-platform GUI library. PyQt5 is a blend of Python programming language and the Qt library. It is a popular C++ framework for writing GUI applications for all major desktop, mobile, and embedded platforms (supports Linux, Windows, MacOS, Android, iOS, Raspberry Pi, and more).

6.2 Methods for Flood Prediction

KNN (K-Nearest Neighbor)

Classification KNN is a nearest neighbor classification model where you can change both the distance matrix and the number of nearest neighbor. It stores training data, can use the model to compute the resubstitution prediction. This model can be convenient because training a classifier occurs in one step and classification in other steps.

Logistic Regression

Logistic Regression is a machine learning algorithm that predicts the probability of a categorical dependent variable. It is a statistical way of analyzing a set of data that comprises more than one independent variable that determines the outcome. The outcome is then measured with a dichotomous variable. The goal of this algorithm is to find the best model to describe the relationship between a dichotomous characteristic of interest and a set of independent variables.

SVM (Support Vector Machine)

It is a supervised learning algorithm which can used for binary classification or regression. It is a coordinate of individual observations. It is based on decision planes which defines decision boundaries. It also separated the set of objects having different class. a. This classifier is chosen as it is incredibly versatile in the number of different kernel functions that can be applied, and this model can yield a high predictability rate. SVM is one of the most popular and widely used clustering algorithms. It belongs to a group of generalized linear classifiers and is considered as an extension of the perceptron.

Decision Tree

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal, but are also a popular tool in machine learning.

6.3 Procedure for Flood Prediction

Step 1: Start

Step 2: Input the region of which flood needs to be predicted from the drop down.

Step 3: Input the time of year for the prediction.

Step 4: Based on the given data, select the region and time of year from the dataset.

Step 5: Pre-process the data to remove any duplicates, null values etc.

Step 6: Display graphs and other statistics related to the selected information.

Step 7: Split the data into two sets, training and test set.

Step 8: Input the data into the different models.

Step 9: Get the predictions from the models and output the prediction along with the accuracy.

Step 10: End

CHAPTER 7

TESTING

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. This includes the process of executing the program or application with the intent of finding errors. Quality is not an absolute; it is value to some person. With that in mind testing can never completely establish the correctness of arbitrary computer software; testing furnishes a criticism or comparison that compares the state and behaviour of the product against a specification.

Testing forms the first step in determining the errors in a program. Clearly the success of testing in revealing errors in programs depends critically on the test cases. Because code is the only product that can be executed and whose actual behaviour can be observed, testing is the phase where the errors remaining from all the previous phases must be detected.

The program to be tested is executed with a set of test cases and the output of the program for the test cases are evaluated to determine if the programming is performing as expected.

7.1 Testing Methodologies

The following are the testing methodologies:

- **Unit Testing:** This is the first phase of testing; the different modules or components are tested individually, often performed by coder himself.
- **Integration Testing:** In this type of testing many unit tested modules are combined into subsystems, which are then tested. The goal here is to see if the modules can be integrated properly.
- **System Testing:** Here the entire software system is tested. The reference document for this process is the requirement specification and the goal is to see if the software meets the requirements. This form of testing is popularly known as black box testing.

7.2 Testing Criteria

Table 7.1: *Test cases for Flood Prediction using AI*

Sl. No	Test Procedure	Expected Result	Actual Result	Passed/ failed
1	Click ‘Go’ with state selected as “Please Select” and month selected as “Please Select”	Error saying “State Cannot be Please Select”	Error saying “State Cannot be Please Select”	Passed
2	Click ‘Go’ with month selected as “Please Select”	Error saying “Month Cannot be Please Select”	Error saying “Month Cannot be Please Select”	Passed
3	Select a state and month	Application runs successfully	Application runs successfully and outputs the result	Passed

CHAPTER 8

SCREENSHOTS

8.1 Loading Page

The Figure 8.1 shows the loading page of the application. The system will show this screen when the user starts the application.

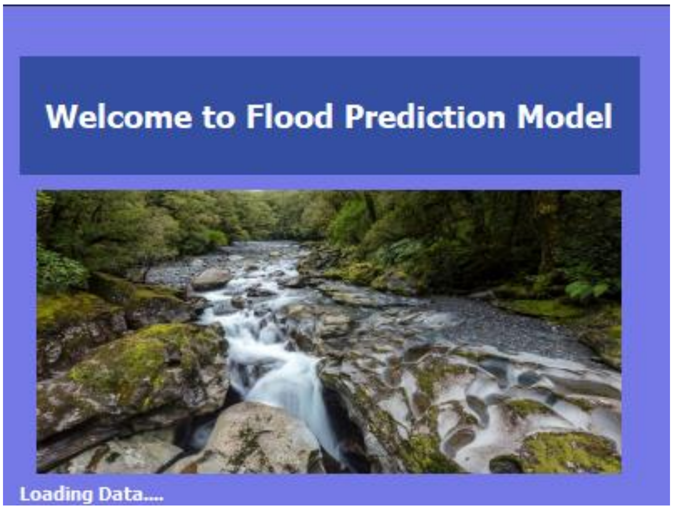

Figure 8.1: *Loading Page*

8.2 Home Page

The figure 8.2 shows the home page of the application. This is where the user will see details of the current weather of his/her location. The user will also select the state and the month for the flood prediction.

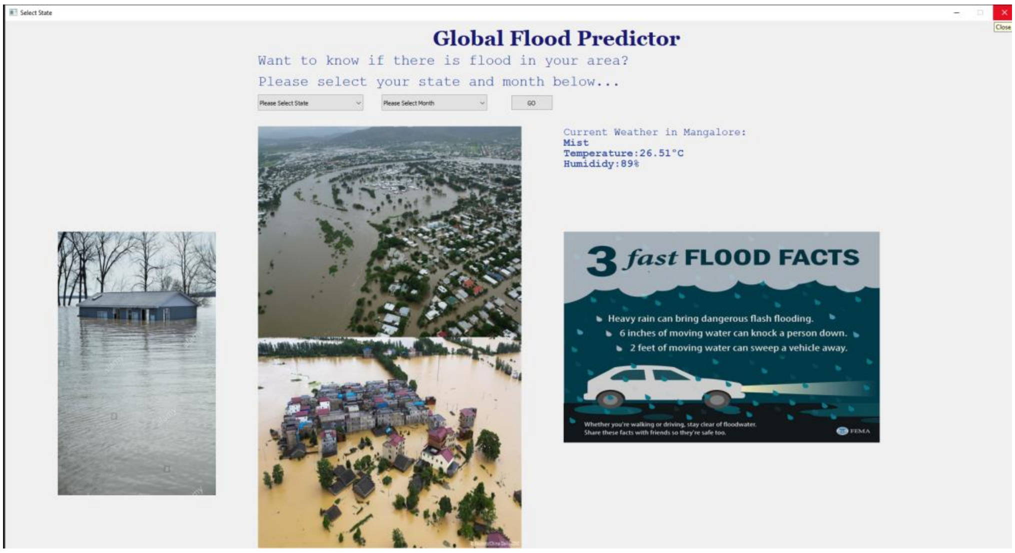

Figure 8.2: *Home Page*

8.3 Select State Error

The figure 8.3 shows the error when user does not select a state and clicks Go. This is an expected error and is handled accordingly.

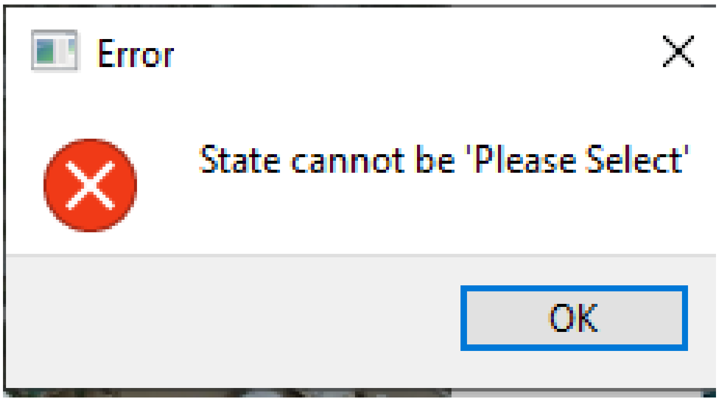

Figure 8.3: *State not selected error*

8.4 Select Month Error

The figure 8.4 shows the error when user does not select a month and clicks Go. This is an expected error and is handled accordingly.

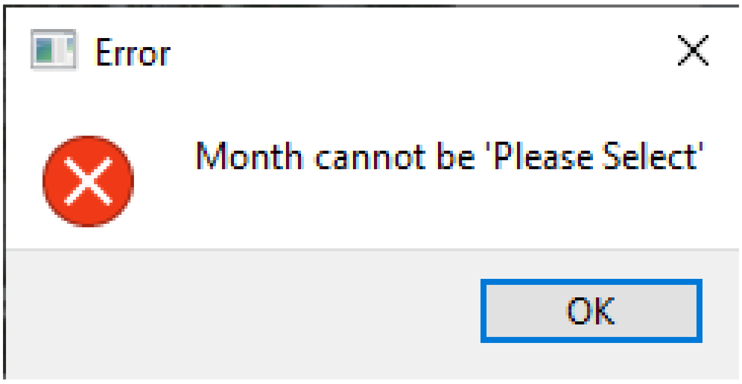

Figure 8.4: *Month not selected error*

8.5 Graphical Information

The figure 8.5 shows the rainfall information in a graphical form. In the x-axis the month is shown and y-axis represents the amount of rainfall in millimetres.

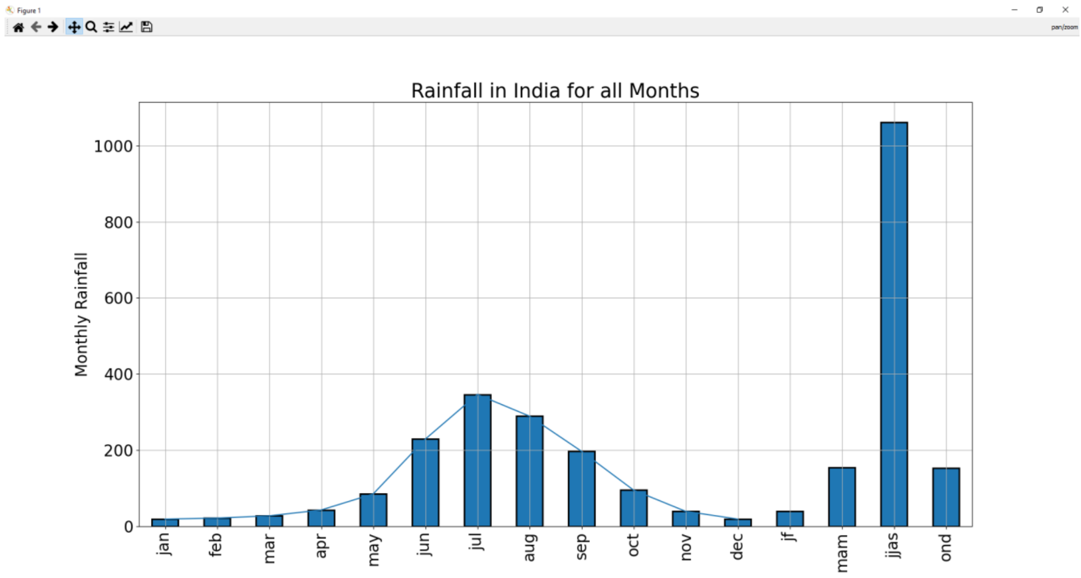

Figure 8.5: *Graphical Information about Rainfall*

8.6 Final Output

The figure 8.6 shows the final output of the application where the prediction of rainfall is shown along with the accuracy of the prediction using different algorithms.

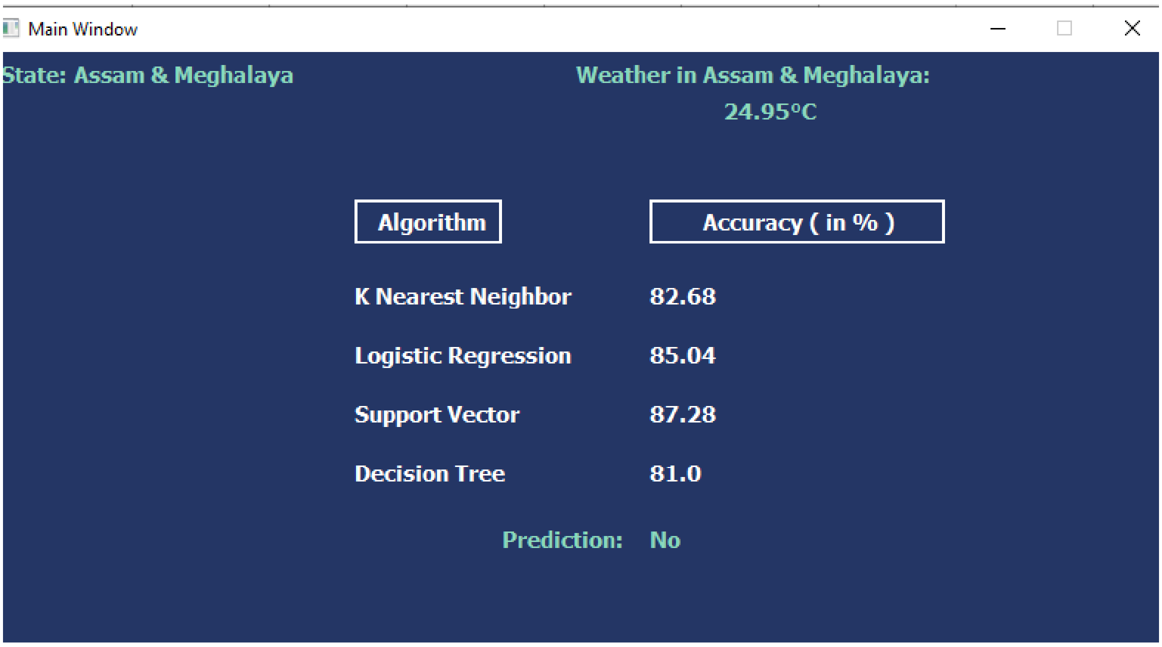

Figure 8.6: *Final Output*

CHAPTER 9

RESULT ANALYSIS

True Positive: It is an outcome where the model correctly predicts the positive class. The outcome is considered as true positive when the system can correctly predict that an incident has indeed occurred.

True Negative: It is an outcome where the model correctly predicts the negative class. The outcome is considered as true negative when the system can correctly predict that the particular incident has not occurred.

False Positive: False Positive is an accuracy measure where the model mispredicts the positive class. The outcome is considered as False Positive when the system cannot correctly predict that the particular incident has occurred.

False Negative: False Negative is an accuracy value where the model mispredicts the negative class. The outcome is considered as False Negative when the system cannot correctly predict that the particular incident has not occurred.

Specificity: Specificity is defined as the measure of the proportion of true negative, which is the actual number of negative cases that are predicted as negative. Simultaneously, another proportion of actual negative values, which got predicted as positive, are termed as a false positive rate. The sum of specificity and false positive rate value will always be 1. Mathematically, specificity can be calculated as the following:

Specificity = (True Negative) / (True Negative + False Positive)

The higher value of specificity would mean a higher value of true negative and lower false-positive rate. The lower value of specificity would mean a lower value of the true negative and higher value of false positive.

Precision: The proportion of positive predictions that is actually correct.

Precision = TP / (TP + FP)

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model correctly predicts)

Recall = TP / (TP + FN)

Algorithm	Recall	ROC	Accuracy (%)
KNN	84.44	87.15	88.28
Logistic Regression	85.03	88.10	86.64
Support Vector	84.72	87.28	88.20
Decision Tree	81.24	85.25	84.27

Table 9.1: Comparison of Accuracy Results

CONCLUSION AND FUTURE WORK

The analytical process started with data cleaning and processing, finding missing value, exploratory analysis, and finally, model building and evaluation. This is not straightforward owing to the variety of flood forecasting and warning systems. This research concludes that the statistical uncertainty quantification methods can answer the question: what is the probability of the forecasts being accurate based on past performance? This method should be applied when the users require an estimation of the uncertainty of a flood forecast as probabilistic bands based on the historical rainfall data. We predict the flood using machine learning algorithms, which gave different results. From the above observations and analysis, the best algorithm for flood prediction is the Support Vector Algorithm (87%).

For future work, we plan to consider more explanatory variables such as the humidity of the air, tides, precipitation rate, river basin levels etc. to have more variables in the prediction. Conducting a survey on spatial flood prediction using machine learning models is highly encouraged. Nevertheless, the recent advancements in machine learning models for spatial flood analysis revolutionized this particular realm of flood forecasting, which requires separate investigation.

REFERENCES

[1] https://encyclopedia.pub/73

[2] https://www.irjet.net/archives/V7/i5/IRJET-V7I51189.pdf

[3] https://csfjournal.com/volume-2-issue-5/2020/3/11/an-innovative-flood-prediction-system-using-improved-machine-learning-approach

[4] https://www.riverbankcomputing.com/static/Docs/PyQt5/

[5] https://matplotlib.org/stable/tutorials/index.html

[6] https://docs.python-requests.org/en/master/