GEOGRAPHICAL DISASTER ANALYSIS BASED ON MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

UPPUTURI SWATHI (21UECS0642) **(20211)**
GANADI MANEESH (21UECS0179) **(20454)**
DARUKAMALLI NITHIN REDDY (21UECS0144) **(20498)**

Under the guidance of
Mr. R. GANESAN, M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

GEOGRAPHICAL DISASTER ANALYSIS BASED ON MACHINE LEARNING

Minor project-II report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

UPPUTURI SWATHI	(21UECS0642)	**(20211)**
GANADI MANEESH	(21UECS0179)	**(20454)**
DARUKAMALLI NITHIN REDDY	(21UECS0144)	**(20498)**

Under the guidance of
Mr. R. GANESAN, M.Tech.,
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

May, 2024

CERTIFICATE

It is certified that the work contained in the project report titled “GEOGRAPHICAL DISASTER ANALYSIS BASED ON MACHINE LEARNING by “UPPUTURI SWATHI (21UECS0642), GANADI MANEESH (21UECS0179), DARUKAMALLI NITHIN REDDY (21UECS0144)” has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2024

Signature of Professor In-charge
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science and Technology
May, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
(UPPUTURI SWATHI)
Date: / /

(Signature)
(GANADI MANEESH)
Date: / /

(Signature)
(DARUKAMALLI NITHIN REDDY)
Date: / /

APPROVAL SHEET

This project report entitled "GEOGRAPHICAL DISASTER ANALYSIS BASED ON MACHINE LEARNING" by UPPUTURI SWATHI (21UECS0642), GANADI MANEESH (21UECS0179), DARUKAMALLI NITHIN REDDY (21UECS0144) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners	**Supervisor**
	Mr. R. Ganesan, M.Tech,.

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our respected **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Mr.R.GANESAN,M.Tech.,** for his cordial support, valuable information and guidance, he/she helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. U.HEMAVATHI, M.E., Ms. C. SHYAMALA KUMARI, M.E.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

UPPUTURI SWATHI	**(21UECS0642)**
GANADI MANEESH	**(21UECS0179)**
DARUKAMALLI NITHIN REDDY	**(21UECS0144)**

ABSTRACT

Flooding is the most common natural disaster on the planet, affecting hundreds of millions of people and causing between 6,000 and 18,000 fatalities every year of which 20 percent are in India. Reliable early warning systems have been shown to prevent a significant fraction of fatalities and economic damage, but many people don't have access to those types of warning systems. So, to avoid it flood prediction system based on Machine Learning are build. This enhancement of the prediction system provides cost-effective solutions and better performance. In this, a prediction model is constructed using rainfall data to predict the occurrence of floods due to rainfall. The model predicts whether flood may happen or not based on the rainfall range for particular locations. Indian district rainfall data is used to build the prediction model. The dataset is trained with various algorithms such as K-Nearest Neighbors, Decision Tree, Logistic Regression, Support Vector Machine(SVM) and so on.Floods are one of the costliest and deadliest Natural Disasters known to mankind. Due to the inconsistent nature of rain, estimation of flood becomes complex. Most of the previous works have focused on forecasting floods but limited research has been done on flash flood prediction also known as nowcasting. Since Flash floods manifest in a matter of hours, people remain unaware of the disaster leading to loss of lives. Many previous works have highlighted the time period (time difference between successive rows) of the dataset as the limitation to predict flash floods. By foreseeing the disaster as well as assessing its threat in real-time would ensure timely actions which can avoid loss of life.

Keywords:

Machine Learning, K-Nearest Neighbours, Decision Tree, Logistic Regression, Support Vector Machine

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
FF	Flood Forecasting
IEEE	Institute of Electrical and Electronics Engineers
ML	Machine Learning
SVM	Support Vector Machine
UML	Unified Modeling Language

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

Every year, India is the topmost flood-prone disaster place in the world. Mostly water logging in urban cities occurs in low-lying areas. Moreover, the increase in water logging is due to some fundamental points such as surface runoff, relative altitude, and not enough path of the water to drainage So, flood forecasting is essential at these places.In a recent year, there were many parts of countries which are prone to flood like Assam, Bihar, Goa, Odisha, Pune, Maharashtra, Tamil Nadu, Karnataka, Kerala, and Gujarat.In the year 2015 rainfall, Chennai received 1049 millimeters (mm) of rainfall in November. Since 1918, 1088 mm of precipitation was the best recorded in November. 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.Flood prediction models are of significant importance for hazard assessment and extreme event management. Robust and accurate prediction contribute highly to water recourse management strategies, policy suggestions and analysis, and further evacuation modeling.Thus, the importance of advanced systems for short-term and long-term prediction for flood and other hydrological events is strongly emphasized to alleviate damage

1.2 Aim of the Project

The main aim of this project is 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.By harnessing vast datasets encompassing meteorological variables, terrain characteristics, and historical flood records, machine learning algorithms are trained to discern patterns and relationships that can forecast the onset, severity, and spatial extent of flooding events. These predictive models serve several crucial purposes. Firstly, they enable the provision of timely warnings to at-risk populations, allowing for proactive evacuation and preparation measures. Secondly, they facilitate the identification and assessment of high-risk areas, aiding in resource allocation and infrastructure planning to minimize vulnerability. Additionally, machine learning contributes to enhancing the accuracy and reliability of flood forecasts through continuous refinement and integration of real-time data, thereby improving response strategies and reducing the socio-economic impact of floods. Ultimately, the aim of flood prediction using machine learning is to save lives, protect property, and build resilience in the face of natural disasters.By training machine learning algorithms on these data sets, the goal is to develop accurate predictive models that can anticipate floods in advance. These models can help authorities and communities take proactive measures to mitigate the impact of floods, such as evacuating residents, reinforcing infrastructure, and implementing early warning systems.

1.3 Project Domain

The domain of the project is the application of machine learning algorithms and techniques to the field of hydrology and water resources engineering. This involves analyzing data related to precipitation, river flow rates, water levels, soil moisture, and other related variables, and developing models that can accurately predict the occurrence and severity of floods in a given area. The domain requires knowledge of statistical modeling, data analysis, hydrological processes, and machine learning algorithms. The targeted subject for flood prediction using machine learning is the occurrence and severity of floods in a given area. This includes predicting the timing, extent, and impact of floods on the environment, infrastructure, and communities. It can include various variables that affect the flood occurrence, such as precipitation patterns, soil moisture, land use, topography, and river flow rates. Machine learning models can be trained on historical data of these variables and their relationship to flood occurrences, in order to develop accurate predictions for future floods.

1.4 Scope of the Project

The scope of this project is that it will try to avoid floods by predicting it from before using machine learning techniques. If proper prediction of flood is done then it much reduce the chance of happening floods in future and it will avoid any disaster to occur. Machine Learning algorithms can analyze large amounts of data from various sources such as satellite imagery, weather forecasts, river flow sensors. These models can be trained using supervised, unsupervised, or reinforcement learning techniques to predict flood events with a high degree of accuracy. Machine learning facilitates the development of predictive models capable of discerning complex patterns and relationships within these datasets, enabling forecasters to anticipate the occurrence, magnitude, and spatial extent of floods with greater accuracy and lead time. Beyond prediction, machine learning contributes to risk assessment by identifying vulnerable areas and populations, guiding resource allocation and infrastructure planning efforts. Moreover, it plays a crucial role in the development of early warning systems, providing timely alerts and advisories to at-risk communities and decision-makers. The scope extends to real-time monitoring, uncertainty estimation, and continuous model refinement to adapt to evolving environmental conditions and improve forecast accuracy.Furthermore, community engagement, capacity building, and integration with decision support systems are integral components, ensuring that the benefits of machine learning-based flood prediction are effectively communicated and utilized to enhance resilience and reduce the socio-economic impact of flooding events.

Chapter 2

LITERATURE REVIEW

[1] J. Akshya et al., (2019) have proposed A Hybrid Machine Learning Approach for Classifying Aerial Images of Flood-Hit Areas.It addresses various issues occurring in numerous parts of southern India have recently encountered severe damage to lives and properties due to floods. Floods are one among the most destructive natural hazard and recovering to normal life takes ample time. During hazards, various technologies are in use for speeding up relief operations and to minimize the amount of damage, one such being the use of drones. Many algorithms are in need for automatic analysis of remote sensing and aerial images. Nowadays, drones are being used for taking images from varied heights similar to aerial images, as they have cameras with exceptional features and effective sensors. It proposes a hybrid approach to classify whether a region in an aerial image is flood affected or not. A combination of Support Vector Machine (SVM) and k-means clustering proved capable of detecting flooded areas with good accuracy, classifying about 92 percent of flooded images correctly. Performance analysis is done by changing various kernel functions in SVM. The results show that there is a decrease in the prediction and training time when quadratic SVM is used.

[2] A.B. Ranit et al., (2018) have proposed ”Different Techniques of flood Forecasting and Their Applications” .It addresses the Flood Forecasting (FF) importance and its challenging problems in hydrology. Reliability of forecast is to provide as much advance notice as possible of an impending flood to the authorities and the general public. A forecast has increased in the modeling capabilities of hydrology and advancements in knowledge for analysis as well as improvements in data collection through satellite observations. It reviews different aspects of flood forecasting, including the models being used, techniques of collecting inputs and displaying their results, and warnings.Different types of real time flood forecasting techniques and models have been suggested and used by many investigators. These are: (i) deterministic models, (ii) stochastic and statistical models and more recently (iii) Artificial Neural Network (ANN) and fuzzy logic techniques.

[3] F. A. Ruslan et al., (2017) have proposed "Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang," .It gives information about year end usually would see many states in Malaysia would probably hits by severe floods due to Monsoon rain especially in the east coast. Many peoples suffered properties damages and economic losses. Thus, an accurate flood water level prediction model is required as an alarm system that would warn the affected area and residence to prepare for evacuation due to the upcoming severe flood. It compares the prediction performances of a developed flood prediction models that were designed using Multiple-Input Single-Output (MISO) Auto Regressive with Exogenous Input (ARX) and MISO Auto Regressive Moving Average with Exogenous Input (ARMAX) structure. The models were designed using Matlab System Identification toolbox of parametric model. The location for the case of study was at Pahang River, Temerloh, Pahang with four upstream stations and one downstream station or observed location. The data used were obtained from the Malaysian Department of Drainage and Irrigation. Simulation results showed that the prediction performance of flood prediction model designed by ARMAX structure showed better Best Fit value and smaller rms values as compared to the model designed using ARX structure.

[4] F. R. G. Cruz et al., (2018) have proposed "Flood Prediction Using Multi-Layer Artificial Neural Network in Monitoring System with Rain Gauge, Water Level, Soil Moisture Sensors," TENCON.It tells about flood which is one of the most destructive natural phenomena that happens on most part of the world. Notably in the Philippines, this was a major issue as it can lead to damage of properties, damage to infrastructures or even loss of lives. Current systems adhere to solve issues to prevent devastating disasters caused by floods. In this study, a system is developed to predict flood level based on real-time monitoring sensors and systems. The system predicts in advance the flood level based on the current data it gathered from sensors integrated in a real-time monitoring system. Multi-layered artificial neural network with the aid of MATLAB was used in the development of the prediction model.

[5] G. Kaur et al., (2019) have proposed An Efficient Automated Hybrid Algorithm to Predict Floods in Cloud Environment.It tells about the natural and environmental sciences are one of the scientific domains which seek a lot of attention as it requires accurate real time predictions. In particular, flooding induced by heavy precipitation is one of the regular risks in Eastern Indian states. In the research work, the state of Odisha, India have been selected for predicting floods because majority of the

state districts have been exposed to floods, leading to unprecedented loss of life and property. It is an optimization based feature selection Genetic Algorithm (GA) have been combined with classification algorithms to predict the occurrence of floods. The experimental results show that the GA-SVM algorithm outperforms in terms of accuracy and total execution time in comparison to other hybrid algorithms. Finally, the results are validated and compared by executing the proposed hybrid algorithm over the heterogeneous resources in Cloud environment.

[6] J. M. A. Opella et al., (2019) have proposed"Developing a Flood Risk Assessment Using Support Vector Machine and Convolutional Neural Network: A Conceptual Framework," It tells that flooding is one of the most devastating natural hazards that affect not only to infrastructures and agriculture but also to human lives. The prominent effect of global warming boasted its danger and impact in a wider range. In order to address and provide more effective measures to lessen the impact of flood hazards, it would be better to identify first the areas with such flood vulnerability.

[7] Qianyu Zhang et al., (2018) have proposed "Investigation of Image Processing based Real-time Flood Monitoring," in Proc.It tells that real-time flood monitoring using image processing techniques is a popular research topic in the field of environmental monitoring. The main idea behind this approach is to use image processing algorithms to analyze satellite or drone images to detect and monitor flood events in real-time.The first step in this process is to collect satellite or drone images of the area of interest.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The increasing growth of machine learning, computer techniques divided into traditional methods and machine learning methods. This section describes the related works of flood predictions and how machine learning methods are better than traditional methods. The existing method in this project have a certain flow and also SVM is used for model development. But it requires large memory and result is not accurate. SVMs are a type of supervised learning algorithm that can be used for classification or regression tasks. In the case of flood prediction, SVMs can be used for regression to predict the water level of a river or stream based on historical data such as rainfall, temperature, soil moisture, and other environmental factors.

Disadvantages of Existing System

Computation requirements: ML models can require significant computational resources to train and run, especially when working with large datasets. This can make the implementation of the model difficult or costly, particularly in areas with limited computing resources.

Time-consuming: Depending on the complexity of the model and the size of the dataset, ML-based flood prediction models can take a long time to train and optimize. This can delay the deployment of the model and make it less useful for realtime decision making.

Difficulty in handling and maintenance: ML models can be complex and require ongoing maintenance and updates to remain effective. This can be difficult for organizations or communities with limited technical expertise, and can also be challenging to manage over time as data and environmental conditions change.

3.2 Proposed System

A machine learning-based flood prediction system involves using data from various sources such as weather forecasts, river levels, and historical flood data to develop models that can predict the likelihood and severity of floods. The system would need to collect relevant data, preprocess it, and extract features such as rainfall intensity and soil saturation. A machine learning model, such as Decision Trees or Artificial Neural Networks(ANN), would then be trained on the preprocessed data to learn the relationships between the extracted features and the occurrence of floods

Advantages of Proposed System

High Accuracy: A machine learning-based flood prediction system can achieve high accuracy by analyzing large amounts of historical data and identifying patterns and trends that are difficult for humans to detect.

Time Saving: Machine learning-based flood prediction systems can analyze large amounts of data in a short period, saving time compared to traditional methods.

Low Complexity: Machine learning-based flood prediction systems can be designed with low complexity, meaning they are easier to understand and use. This can be especially important for non-technical stakeholders who need to interpret the results of the model and make informed decisions.

3.3 Feasibility Study

3.3.1 Economic Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased. The machine learning-based flood prediction system has the potential to provide a range of economic benefits to flood-prone areas. For example, an early warning system could save lives and reduce property damage, while reduced flood insurance claims could be beneficial for both insurers and property owners. Additionally, the improved emergency response could be beneficial in mitigating the effects of a flood. By minimizing the impact of flooding, the machine learning-based flood prediction system could stimulate economic activity in flood-prone areas, as people are more likely to invest in such areas if they are safer. Therefore, implementing this system can provide significant economic advantages while reducing the negative impact of floods on the community.

3.3.2 Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system. Additionally, the project requires robust computing infrastructure to handle the large amounts of data and perform the necessary computations. Therefore, it is important to ensure that the necessary hardware and software resources are available before embarking on such a project.

3.3.3 Social Feasibility

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system. The important aspect of social feasibility is community acceptance. It is essential to involve communities in the development and deployment of ML-based flood prediction models to ensure that the system meets their needs, addresses their concerns, and is tailored to their local conditions. Engaging with communities can increase their trust and confidence in the technology, which is crucial for its success.

3.4 System Specification

3.4.1 Hardware Specification

Processor : Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHzr
RAM : 8GB DRD4 or higher
Hard Disk : 128 GB
Key Board : Standard Windows Keyboard
Mouse : No Mouse
Monitor : Any

3.4.2 Software Specification

Operating System : Windows 10
Server-side Script : Python 3.6
IDE : PyCharm
Libraries Used : Flask,pandas,sklearn,kNeighborsclassifer

3.4.3 Standards and Policies

Flood Modeller

Flood Modeller is a software for flood modeling and prediction, developed by Jacobs. It has a command line interface that can be used to automate various tasks, such as creating and running models, generating flood maps, and producing reports.
Standard Used:ISO/IEC 27001

Delft-FEWS

Delft-FEWS (Flood Early Warning System) is an open source web application for flood prediction and early warning, developed by Deltares. It is designed to provide real-time flood forecasting and monitoring, and can be used to manage and analyze large volumes of data. Delft-FEWS is built using Java and various open source libraries, and can be customized to meet the specific needs of different users.
Standard Used:ISO/IEC 27001

Chapter 4

METHODOLOGY

4.1 Architecture for Flood Prediction

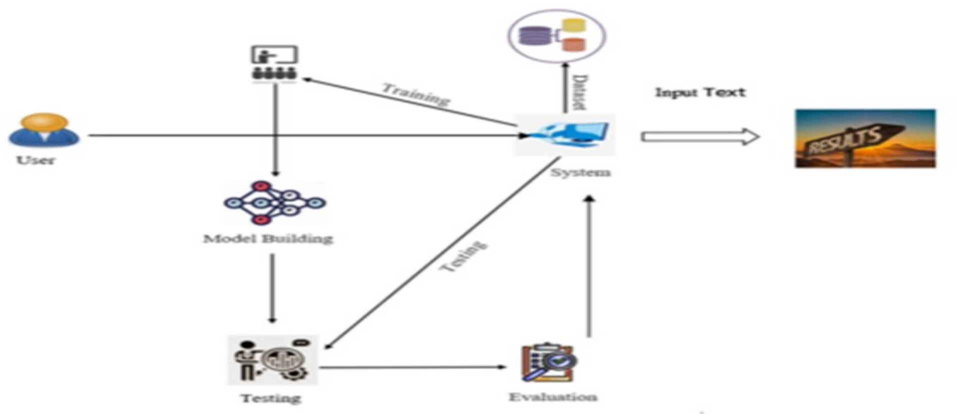

Figure 4.1: **Architecture Diagram for flood prediction**

The Figure 4.1 represents the architecture of a machine learning-based flood prediction involves

•Data Collection: The first step is collecting the necessary data.

•Data Preprocessing: Once the data is collected, it must be cleaned and preprocessed to remove any errors.

•Feature Extraction: Next, relevant features must be extracted from the preprocessed data.

•Model Training: With the preprocessed and feature-extracted data, a machine learning model is trained .

•Model Evaluation: The performance of the trained model is evaluated using metrics like accuracy, precision, recall, or F1 score.

•Deployment: Once the model is deemed to be accurate and reliable, it can be deployed in a real-world setting.

4.2 Design Phase

4.2.1 Data Flow Diagram

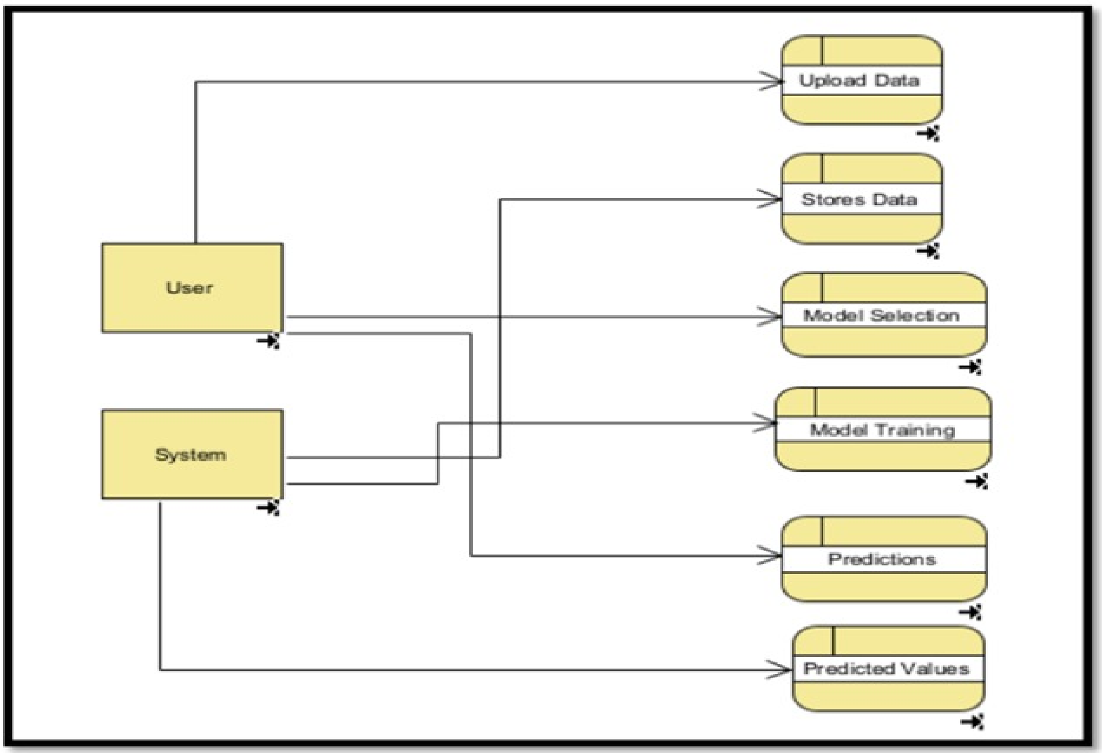

Figure 4.2: **Data Flow Diagram**

The Figure 4.2 represents the data flow diagram of a machine learning-based flood prediction involves components

•Input Data: It include a variety of data sources such as rainfall, river levels, that are relevant to predicting floods.

•Data Preprocessing: The input data is then preprocessed, which involves cleaning and transforming the data into a format that can be used by the machine learning model.

•Feature Extraction: The preprocessed data is then used to extract relevant features, such as the amount of rainfall, river levels.

•Machine Learning Model Training: The extracted features are then used to train a machine learning model.

•Model Evaluation: It is evaluated to determine its performance.

•Prediction: It can be used to predict the likelihood of flooding in real-time.

•Output: It include various forms of information, such as warning alerts.

4.2.2 Use Case Diagram

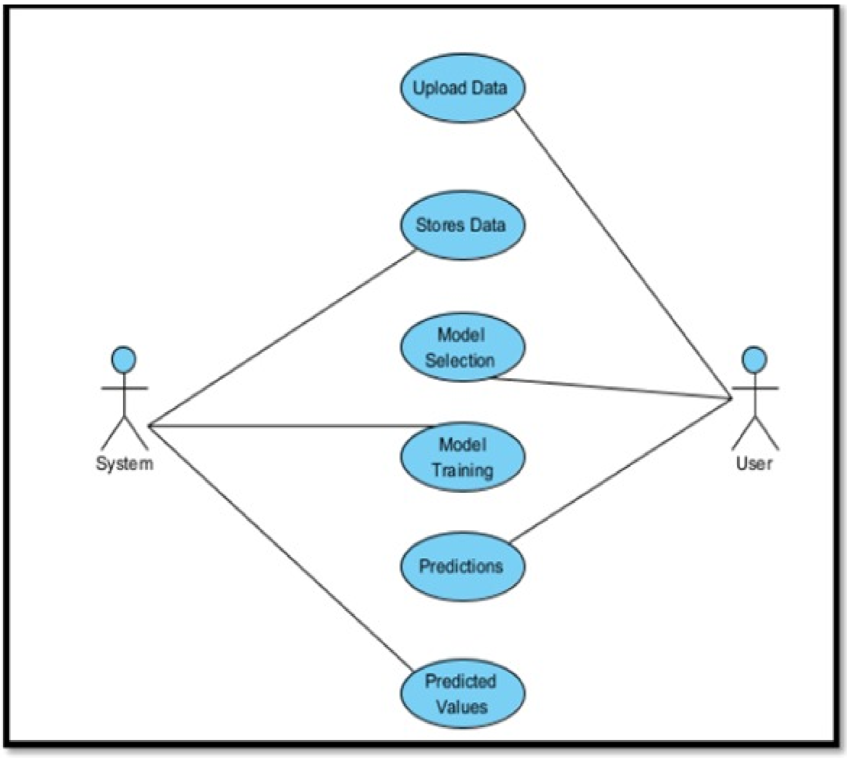

Figure 4.3: **Use Case Diagram**

The Figure 4.3 represents the use case diagram is a high-level visual representation of the potential interactions between users (actors) and a system. Here's how a use case diagram could be used for a machine learning-based flood prediction system:
•Actors: The first step is to identify the actors. In the case of a flood prediction system, actors could include weather forecasters such as system and users.
•Use Cases: Once the actors have been identified, the next step is to define the various use cases, or the specific interactions that they may have with the system. Some examples of use cases for a flood prediction system could include:
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis.
Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.
The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

4.2.3 Class Diagram

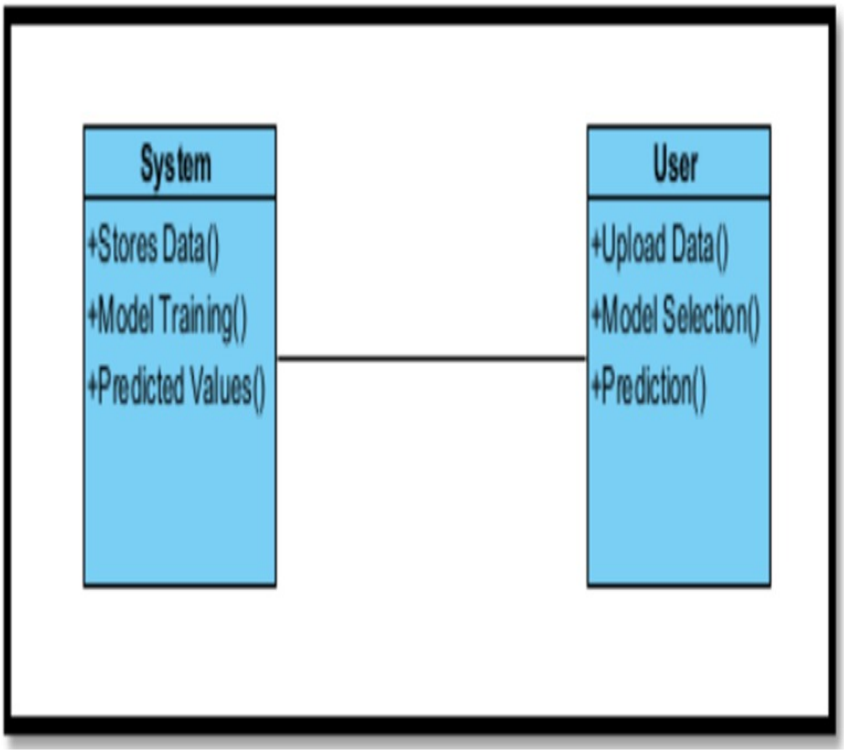

Figure 4.4: **Class Diagram**

The Figure 4.4 represents the class diagram, the terms system and user can refer to different entities or components depending on the context and scope of the diagram. In general, a system in a class diagram refers to the software system being designed or analyzed. It can represent the overall architecture, components, and interactions of the system.

On the other hand, a user in a class diagram typically refers to the human or external entity that interacts with the system. This could include end-users who use the system to perform specific tasks, as well as other systems or devices that connect to the system. For example, in a machine learning-based flood prediction system, a user could be a hydrologist who uses the system to analyze flood risk in a specific region, or a weather station that provides data to the system.

It's worth noting that the terms system and user can be used in different ways in other types of diagrams or contexts. For example, in a use case diagram, a user typically refers to an actor while a system may refer to a subsystem or module within the overall system.

4.2.4 Sequence Diagram

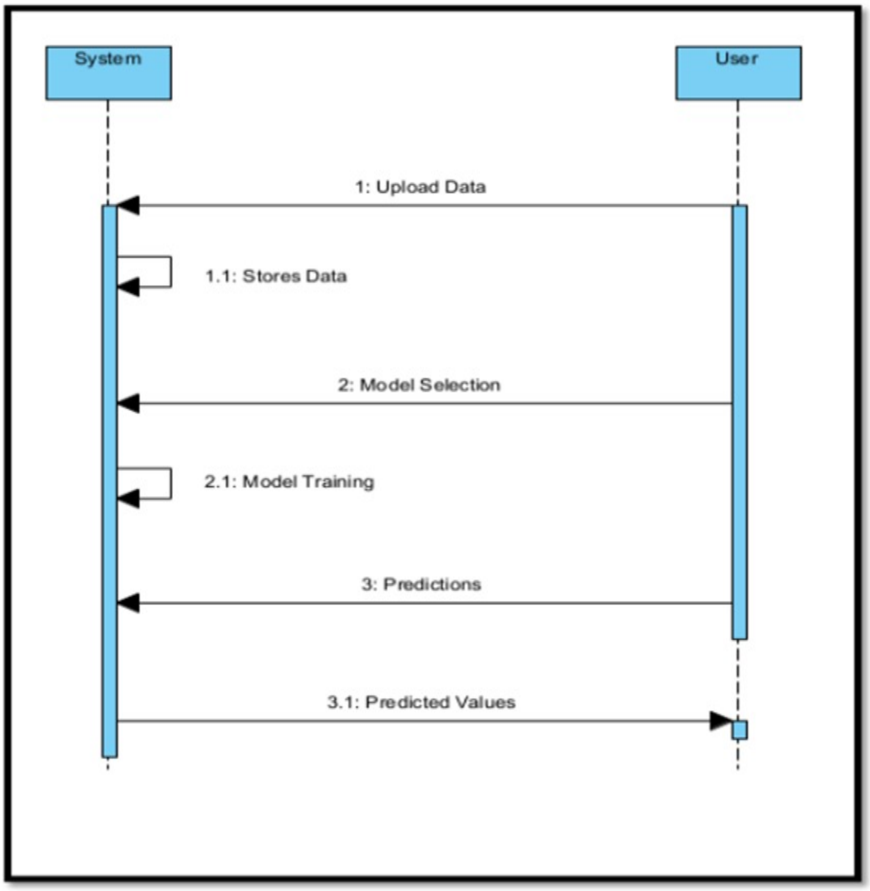

Figure 4.5: **Sequence Diagram**

The Figure 4.5 represents the sequence diagram, the terms system and user refer to two different types of actors or components that interact with each other. The system generally refers to the software or hardware components that make up the main processing or prediction engine of the system. This might include components such as data collection modules, pre-processing modules, machine learning models, and output components. The system typically receives input data from various sources, processes it using algorithms or models, and generates output or predictions that are sent back to the user.

The user refers to any external entity or actor that interacts with the system. This might include people such as end-users, administrators, or other stakeholders who need to interact with the system in some way. The user typically provides input data to the system, such as parameters or configuration settings, and receives output or predictions from the system in return.

4.2.5 Collaboration Diagram

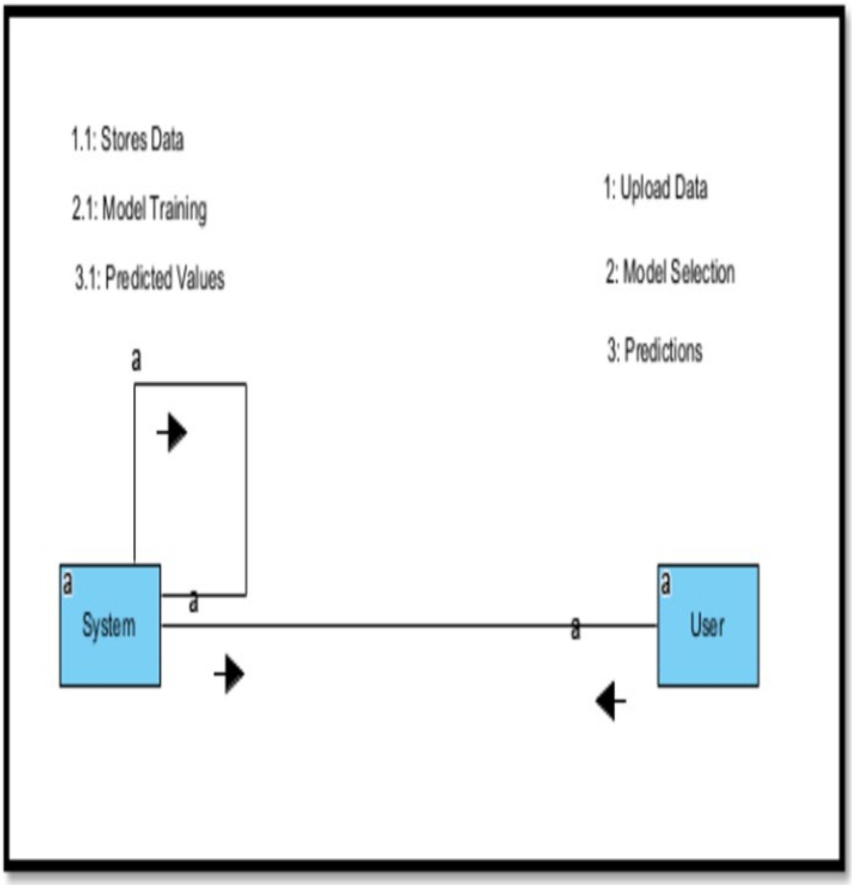

Figure 4.6: **Collaboration diagram**

The Figure 4.6 represents the collaboration diagram, the system and user components play an important role in illustrating the interactions and dependencies between different components of a system. In the context of flood prediction, the system and user components can be used to represent different aspects of the flood prediction system and the users who interact.

The system component can represent the flood prediction software or system itself, which may include various modules responsible for collecting, analyzing, and presenting data related to flood prediction.

The user component, on the other hand, can represent the various users who interact with the flood prediction system. These may include emergency responders or even individual citizens who want to stay informed about flood risks in their area.

4.2.6 Activity Diagram

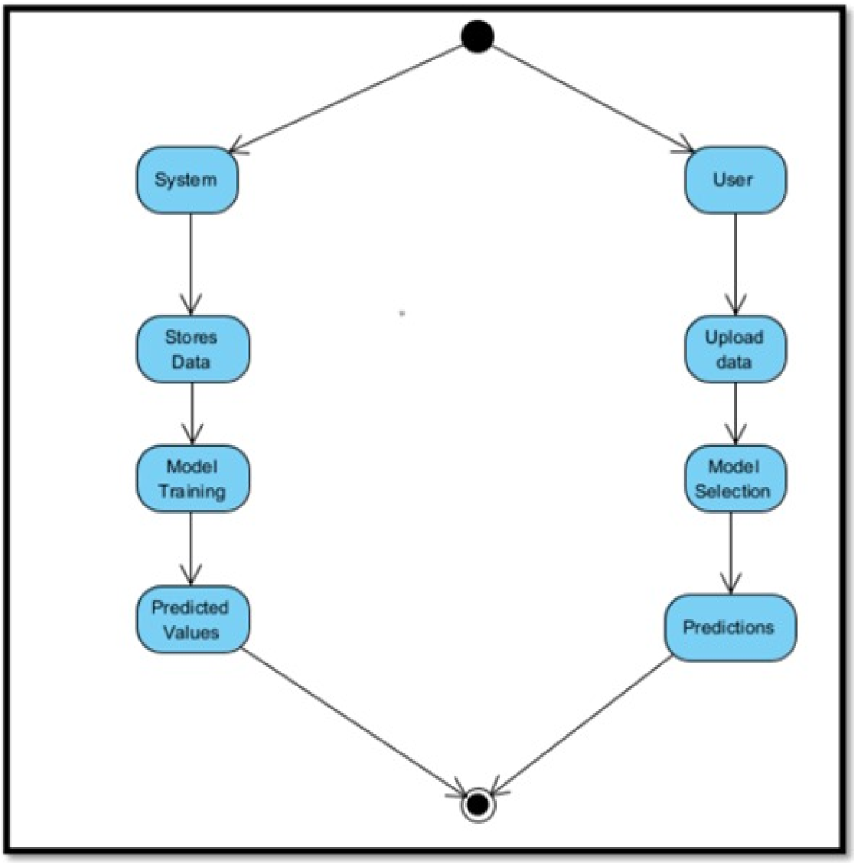

Figure 4.7: **Activity Diagram**

4.3 Algorithm & Pseudo Code

4.3.1 Enhanced SVM Algoirthm

A high-level algorithm for flood prediction using machine learning:

1.Data Collection:

Collect historical data on rainfall, water level, temperature, humidity, wind speed, and other relevant environmental factors. The data should cover a period of several years and include both flood and non-flood periods.

2.Data Preprocessing:

Clean the data by removing duplicates, filling in missing values, and normalizing the data. Divide the data into training and testing sets.

3.Feature Selection:

Select the most relevant features that are likely to influence the occurrence of floods.

4.Model Selection:

Choose a suitable machine learning algorithm such as Random Forest, SVM, ANN, Gradient Boosting, or LSTM. Train the model on the training set using the selected features.

5.Model Evaluation:

Evaluate the performance of the model on the testing set using metrics such as accuracy, precision, recall, and F1-score. Adjust the model parameters or try different algorithms if necessary.

6.Prediction:

Use the trained model to make predictions on new data. The model should be able to predict the likelihood of floods based on the input environmental factors.

7.Deployment:

Integrate the model into a system that can receive real-time data from weather stations or remote sensors. The system should be able to provide timely warnings and alerts to the relevant authorities and the public. Note that this is a simplified algorithm, and there are many details and considerations that need to be taken into account when developing a real-world flood prediction system.

4.3.2 Pseudo Code

```
# Import necessary libraries
from sklearn.ensemble import Decision Tree Classifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Load your dataset (X features, y labels)
# Assume X is a matrix of features and y is a vector of labels (flood types or categories)
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Initialize the  Decision Tree Classifier
rf_classifier = RandomForestClassifier(n_estimators=100, random_state=42)
# Train the classifier on the training data
rf_classifier.fit(X_train, y_train)
# Make predictions on the test data
y_pred = rf_classifier.predict(X_test)
# Evaluate the accuracy of the model
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
# Now, you can use the trained model to make predictions on new data
# For example, if you have new features in a variable called 'new_data'
new_predictions = rf_classifier.predict(new_data)
print("Predictions for new data:", new_predictions
```

4.4 Module Description

4.4.1 Data Pre-processing

Data Preprocessing Cleans the data by removing duplicates, filling in missing values, and normalizing the data. Divide the data into training and testing sets, ensuring that both sets have a similar distribution of the target variable (flood or non-flood).Preprocessing of data is a critical step in the model training process for flood prediction. The goal of preprocessing is to transform the raw data into a format that can be used by machine learning algorithms to make accurate predictions about potential flood events

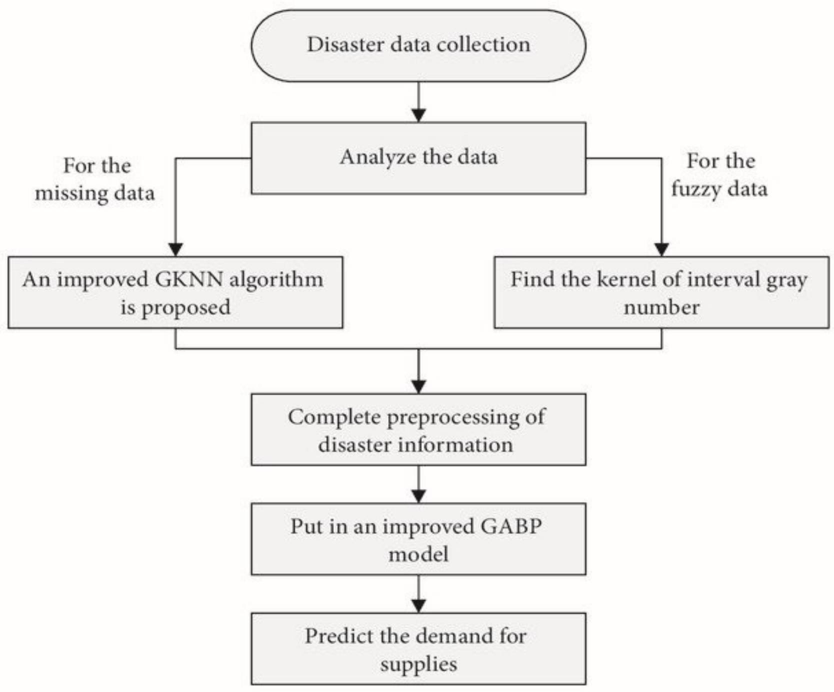

Figure 4.8: **Data Preprocessing**

4.4.2 Feature extraction

Feature extraction is a critical step in machine learning-based flood prediction as it involves selecting and transforming relevant input variables into a more informative representation that can be used by the model to make accurate predictions. Feature extraction techniques that can be used in flood prediction include wavelet analysis, Fourier transforms, and statistical techniques such as auto correlation and cross correlation analysis.

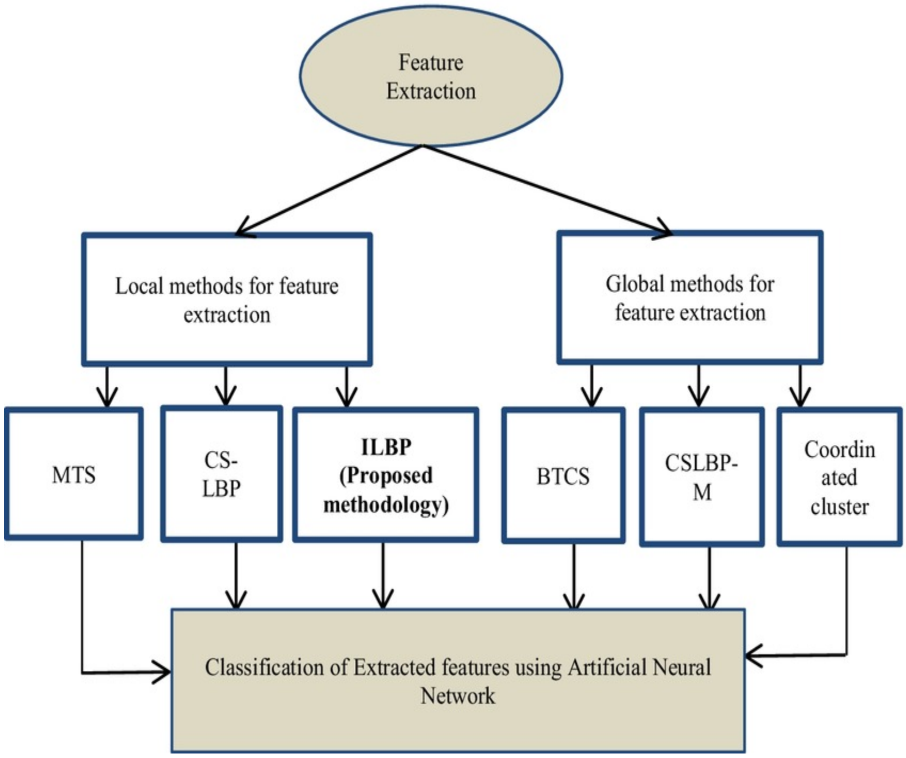

Figure 4.9: **Feature Extraction**

4.4.3 Machine Learning Based Applications

There are various Machine Learning (ML) applications that can be used for flood prediction, some of which include: Regression: Regression models can be used to predict river levels based on historical data. The model can be trained on past river levels, rainfall data, and other environmental factors to predict future river levels. Decision Trees: Decision trees can be used to predict flood occurrences based on historical data. The model can be trained on past flood events and other environmental factors such as rainfall and river flow data to predict the likelihood of future flood occurrences. Random Forests: Random forests can be used to predict flood occurrences based on a combination of multiple decision trees. This can help to improve the accuracy of the model and reduce the risk of overfitting.

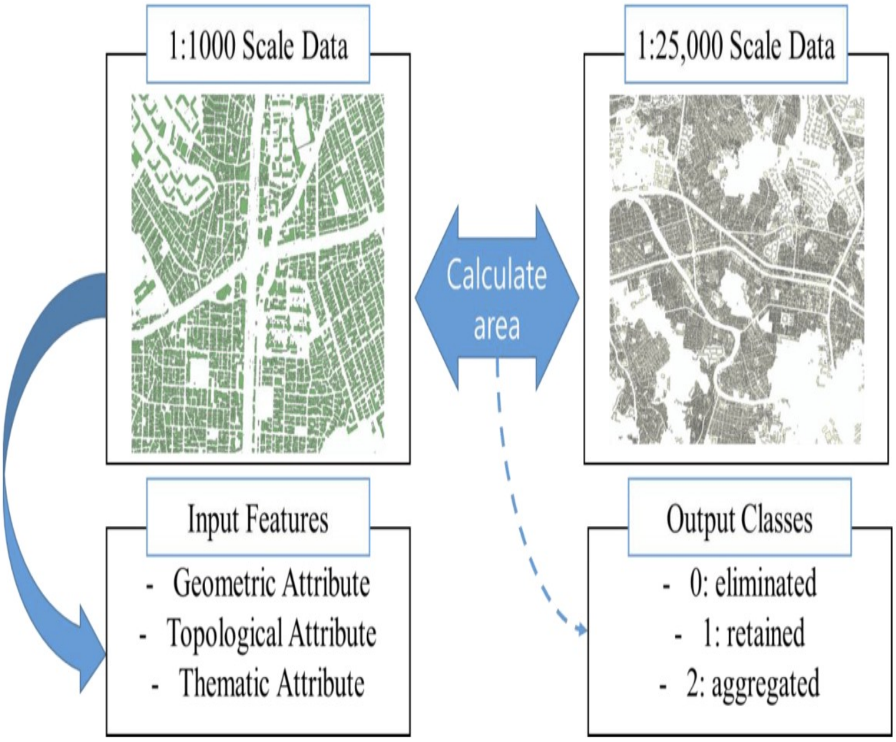

Figure 4.10: **Machine Learning Based Applications**

4.5 Steps to execute the project

4.5.1 Uploading Dataset

•Select the required dataset which contains rainfall data.
•The dataset should contain data about various state rainfall.
•Based on the analysis of rainfall data,upload dataset .

4.5.2 Model training

•After uploading dataset,try to select a model to train the dataset.
•Training refers to adding few features in order to perform prediction.
•Select a required model like knn,decision tree etc in order to train the uploaded dataset.

4.5.3 Prediction

•After training the dataset using different models,try to analyze the dataset.
•Based on the analysis of the dataset,try to predict the rainfall in previous years for every state.
•Now after prediction of rainfall based on past years,try to predict flood will occur or not.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

SUBDIVISION	YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	ANNUAL
KERALA	1901	28.7	44.7	51.6	160.0	174.7	824.6	743.0	357.5	197.7	266.9	350.8	48.4	3248.6
KERALA	1902	6.7	2.6	57.3	83.9	134.5	390.9	1205.0	315.8	491.6	358.4	158.3	121.5	3326.6
KERALA	1903	3.2	18.6	3.1	83.6	249.7	558.6	1022.5	420.2	341.8	354.1	157.0	59.0	3271.2
KERALA	1904	23.7	3.0	32.2	71.5	235.7	1098.2	725.5	351.8	222.7	328.1	33.9	3.3	3129.7
KERALA	1905	1.2	22.3	9.4	105.9	263.3	850.2	520.5	293.6	217.2	383.5	74.4	0.2	2741.6
KERALA	1906	26.7	7.4	9.9	59.4	160.8	414.9	954.2	442.8	131.2	251.7	163.1	86.0	2708.0
KERALA	1907	18.8	4.8	55.7	170.8	101.4	770.9	760.4	981.5	225.0	309.7	219.1	52.8	3671.1
KERALA	1908	8.0	20.8	38.2	102.9	142.6	592.6	902.2	352.9	175.9	253.3	47.9	11.0	2648.3
KERALA	1909	54.1	11.8	61.3	93.8	473.2	704.7	782.3	258.0	195.4	212.1	171.1	32.3	3050.2
KERALA	1910	2.7	25.7	23.3	124.5	148.8	680.0	484.1	473.8	248.6	356.6	280.4	0.1	2848.6
KERALA	1911	3.0	4.3	18.2	51.0	180.6	990.0	705.3	178.6	60.2	302.3	145.7	87.6	2726.7
KERALA	1912	1.9	15.0	11.2	122.7	217.3	948.2	833.6	534.4	136.8	469.5	138.7	22.0	3451.3
KERALA	1913	3.1	5.2	20.7	75.7	198.8	541.7	763.2	247.2	176.9	422.5	109.9	45.8	2610.8
KERALA	1914	0.7	6.8	18.1	32.7	164.2	565.3	857.7	402.2	241.0	374.4	100.9	135.2	2899.1
KERALA	1915	16.9	23.5	42.7	106.0	154.5	696.1	775.6	298.8	396.6	196.6	302.5	14.9	3024.5
KERALA	1916	0.0	7.8	22.0	82.4	199.0	920.2	513.9	396.9	339.3	320.7	134.3	8.9	2945.3
KERALA	1917	2.9	47.6	79.4	38.1	122.9	703.7	342.7	335.1	470.3	264.1	256.4	41.6	2704.8

Figure 5.1: **Input Design**

5.1.2 Output Design

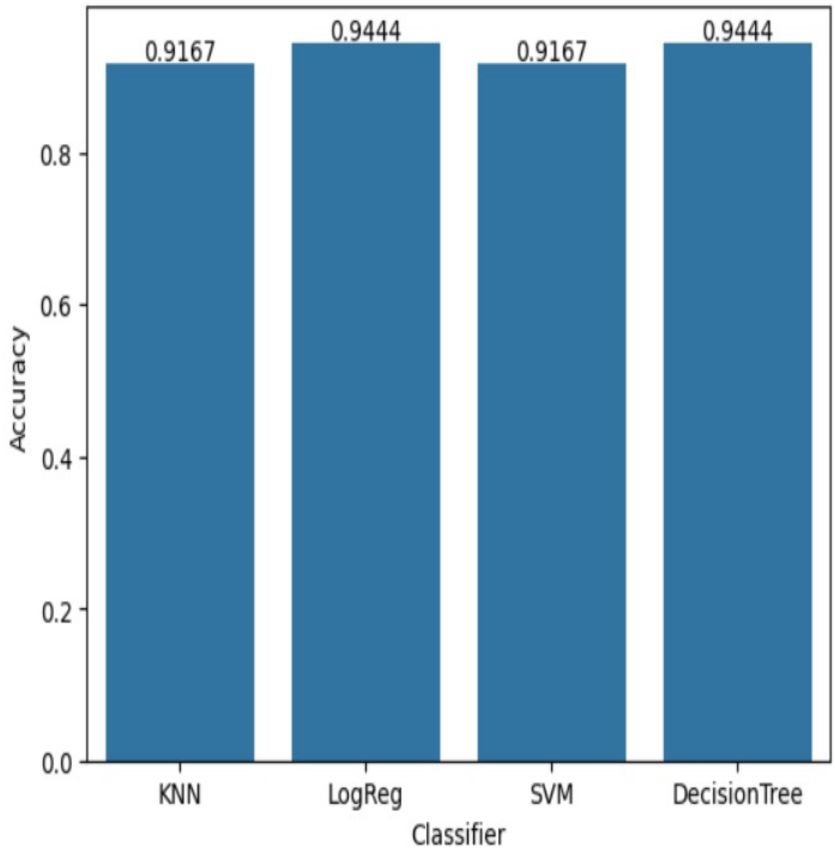

Figure 5.2: **Output Design**

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Unit testing involves the design of cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application it is done after the completion of individual unit before integration. This structural testing that relies on knowledge of its construction and is invasive.Prediction units are then subjected to tests to verify that the models produce accurate predictions for known scenarios and edge cases. Integration testing ensures that all components work together seamlessly, while robustness testing assesses the system's resilience to noise and errors. Performance testing evaluates computational efficiency and responsiveness, while documentation and logging testing ensure clarity and completeness of system documentation.

5.3.2 Integration testing

Integration tests designed to test integrated software components determine if they actually program. Testing is event driven and is more concerned with the basic outcome of fields. Integration tests demonstrate that although the components individually satisfaction shown by successfully unit testing. combination components and consistent, Integration testing from the combination of components,specifically aimed at exposing the problems that arise from the combination of components.This testing phase focuses on verifying the integration of data preprocessing, model training, prediction generation, and result dissemination modules, as well as their coordination in real-time or near-real-time scenarios.Moreover, integration testing verifies the integration of machine learning models into the prediction pipeline, assessing their ability to utilize preprocessed data to generate accurate flood forecasts.

5.3.3 System testing

System testing ensures that the entire integrated software system meets requirements. It tests configuration ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing based and flows emphasizing pre-driven process links and integration points,process descriptions.System testing also involves evaluating the performance and accuracy of machine learning models used for flood prediction, including their ability to generalize to unseen data and adapt to changing environmental conditions. Finally, usability testing may be performed to gather feedback from end-users and stakeholders on the system's user interface, functionality, and overall user experience.

5.3.4 Test Result

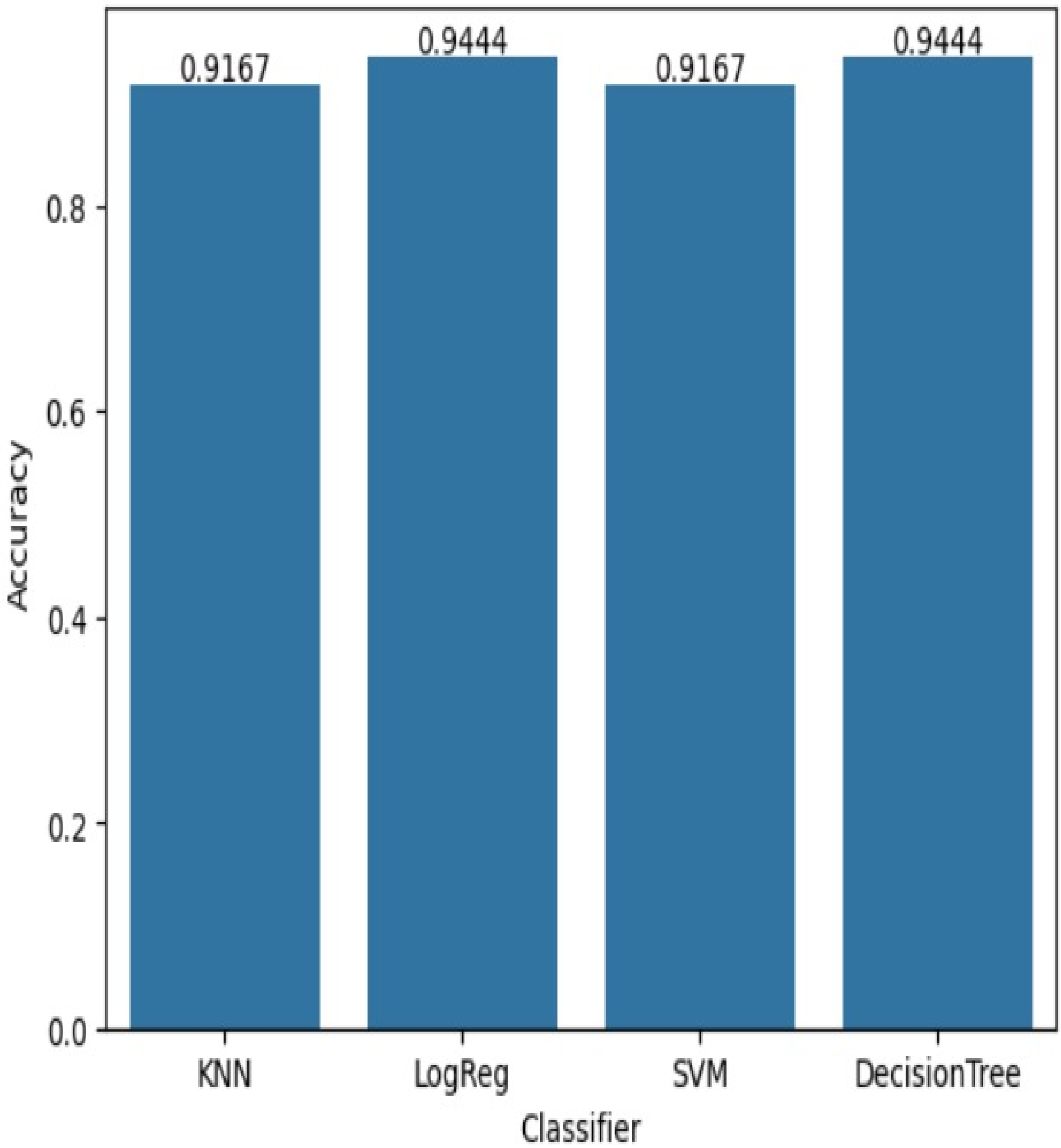

Figure 5.3: **Analysis of Accuracy Rate**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

The proposed system is based on the Random forest Algorithm that creates many decision trees. Accuracy of proposed system is done by using random forest gives the output approximately 95 percent. Random forest implements many decision trees and also gives the most accurate output when compared to the decision tree.Random Forest algorithm is used in the two phases. Firstly, the RF algorithm extracts sub-samples from the original samples by using the bootstrap resampling method and creates the decision trees for each testing sample and then the algorithm classifies the decision trees and implements a vote with the help of the largest vote of the classification as a final result of the classification. The random Forest algorithm always includes some of the steps as follows:Selecting the training dataset:Using the bootstrap random sampling method we can derive the K training sets from the original dataset properties using the size of all training set the same as that of original training dataset. Building the random forest algorithm: Creating a classification regression tree each of the bootstrap training set will generate the K decision trees to form a random forest model, uses the trees that are not pruned. Looking at the growth of the tree,31 this approach is not chosen the best feature as the internal nodes for the branches but rather the branching process is a random selection of all the trees gives the best features.Knowledge about the characteristics of a river's drainage basin, such as soil-moisture conditions, ground temperature, snowpack, topography, vegetation cover, and impermeable land area, which can help to predict how extensive and damaging a flood might become.The random-forest model showed validation accuracies of 95 percent.

6.2 Comparison of Existing and Proposed System

A decision tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. Though a commonly used tool in data mining for deriving a strategy to reach a particular goal.The increasing growth of machine learning, computer techniques divided into traditional methods and machine learning methods. This section describes the related works of flood predictions and how machine learning methods are better than traditional methods. The existing method in this project have a certain flow and also SVM is used for model development. But it requires large memory and result is not accurate.The ML method of DT is one of the contributors in predictive modeling with a wide application in flood simulation. DT uses a tree of decisions from branches to the target values of leaves. In classification trees (CT), the final variables in a DT contain a discrete set of values where leaves represent class labels and branches represent conjunctions of features labels. When the target variable in a DT has continuous values and an ensemble of trees is involved, it is called a regression tree (RT).

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction.The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds.In proposed system, we implement a Machine Learning algorithms for getting insights from the complex patterns in the data. This technique is computationally inexpensive because of its simple architecture.The random-forest model showed validation accuracies of 78.78.

6.3 Sample Code

```
import from flask import Flask , render template , request
import pandas as pd
from sklearn model selection import train test−split
from sklearn.linear model import Logistic Regression.
from sklearn neighbors import KNeighbors Classifier
from sklearn model selection import cross val score
from sklearn.metrics import accuracy score , precision.score.recall.score.classification.report
from sklearn tree import Decision TreeClassifier
import xgboost as xgb
import pickle
app.config['upload folder'] 'C:\Users\YMTS0356 Pycharm Projects\floods\upload'
app Flask (name)
@app.route("/")
return render template('index.html')
def home():
def cleaning (file):
data file [[ Mar−May, Jun−Sep.10days.june increased Rainfall , flood']]
return data
def spliting(file):
global X.y
X file drop ([flood].axis 1)
file flood
x train x test y train y test train test.split(X,y.test.size=0.2.random state 9)
return x train x test , y train y test
@app.route(/upload" methods ["POST", "GET"])
def upload():
if request. method="POST":
file request.files["file
global df
print (file)
filetype os.path.splitext (file filename)[1]
print (filetype)
if filetype.csv':
pathos.path.join(app.config['upload folder"], file filename)
file save(path)
dfpd.read.csv(path)
df.drop(['Unnamed: 01.axis 1.inplace. True)
return render template('view.html.col name df.columns.row val list (df.values.tolist (0))
else:
return render template (upload.html)
return render template("upload.html")
@app.route("/model, methods=['POST', "GET"])
def model():
if request.method "POST":
clean data cleaning (df)
x.train, s.test.y train, y test spliting(clean data)
model int (request form["model"])
```

```
if model == 1 :
knn KNeighbors Classifier()
score cross_val_score (knn.X, r, cv = 5 )
print(score)
print(score. mean())
knknn.fit (x train , y train)
pre kn.predict(x_test)
file models/knn.h5
pickle.dump(kn, open (file , 'wb'))
scores accuracy score (y_test , pre)
print(scores)
return render template('model.html', msg = 'success',score scores, Selected =^ prime KN * N'
if model 2:
dt Decision Tree Classifier()
score cross_val_score (dt.X,y.cv5)
print (score)
print(score. mean())
d=dt.fit(x_train , y train)
file models/dt.h5'
pickle.dump(d, open (file , 'wb'))
pre d. predict(x-test)
scores accuracy score (y test.pre)
print(scores)
return render template (model.html', msg 'success', score scores, Selected 'Decision
Tree Classifier')
if model == 3:
Ir = LogisticRegression ()
score cross_val.score (Ir mathcal ,X,y,cv=5
print(score)
print(score.mean())
1lr.fit (x train ,y-train)
pre 1.predict(x_test)
file models/Ir.h5
pickle.dump(1, open(file , 'wb'))
scores accuracy score (y test , pre)
print(scores)
return render template('model.html', msg = success score Regression') scores, Selected = 'Logistic
if model 4:
xg= xgb .XGBClassifier()
score cross.val.score (xg.X.y. cv = 5 )
print(score)
print(score.mean ())
x=xg. fit (x\ train ,y\ train)
pre => . predict(x-test)
file 'models/xgb.h5
pickle.dump(x, open (file , 'wb'))
scores accuracy score (y_test.pre)
print (scores).
return render template (model.html', msg 'success', score scores, Selected 'XGBoost")
return render template('model.html')
```

```
@app.route(/prediction methods ["POST". "GET"])
def prediction ():
if request.method "POST":
a = float (request.form['f1'])
bfloat (request.form [12])
c = float (request.form [^ * f 3^ * \
d = (request.form['f4'])
values [[float (a), float (b).float(c), float (d)]]
filenamr'models/xgb.h5
model pickle.load (open (filenam. "rb"))
ex = . DataFrame(values.columns=X.columns)
pred model. predict (ex)
print (pred)
return render template (prediction.html',res pred)
return render_template('prediction.html")
if name='main':
app.run (debug=True)
```

Output

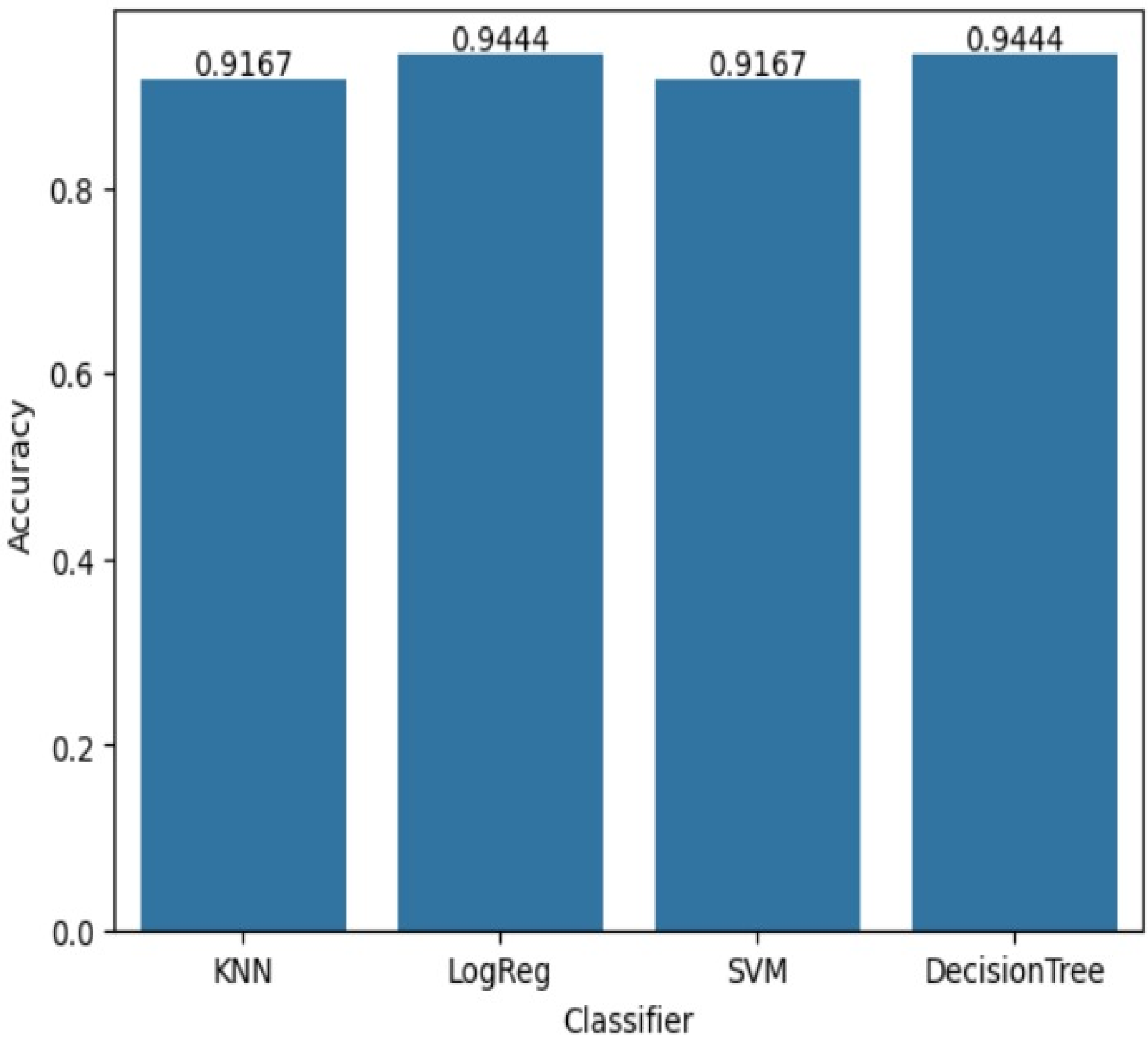

Figure 6.1: **Output for Flood prediction**

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

We have successfully developed a system to predict whether the floods will occur or not in this application. This is created in a user-friendly environment with Python programming and Flask. The system is likely to gather data from the user in order to predict whether there is a chance of flood occurring or not. In conclusion, machine learning-based flood prediction has the potential to revolutionize the way we prepare for and respond to floods. By analyzing vast amounts of historical and real-time data from weather sensors, river gauges, and other sources, machine learning algorithms can help to identify patterns and trends that can be used to predict when and where floods are likely to occur. The use of machine learning can also enable more accurate and timely predictions, which can be crucial in minimizing the damage caused by floods.

In addition, machine learning can help to improve the speed and efficiency of response efforts by providing emergency responders with real-time data and insights about flood conditions.However, it is important to note that machine learning-based flood prediction is not a panacea. While these algorithms can provide valuable insights, they are only as good as the data they are trained on. This means that it is essential to ensure that the data used to train these algorithms is accurate, reliable, and representative of the conditions that may be encountered in the real world.One of the significant advantages of machine learning-based flood prediction models is their ability to analyze vast amounts of data in real-time and provide accurate predictions. This enables emergency response teams to take necessary actions quickly and efficiently, such as evacuating people from affected areas and preparing emergency shelters.

7.2 Future Enhancements

Machine learning-based flood prediction models have already shown significant progress in predicting floods and providing early warnings to communities. However, there are several potential enhancements that can be made to these models in the future. One such enhancement is the incorporation of more data sources, such as social media data, which can provide real-time information about flood conditions and help improve the accuracy of the models. Another enhancement could be the integration of artificial intelligence algorithms that can continuously learn and adapt to new data, improving the model's predictive capabilities over time. We intend to investigate prediction approach with the revised data set and employ the most accurate and relevant machine learning algorithms for detection.

In addition, the use of advanced analytics and visualization tools could enable emergency response teams to better analyze and interpret flood data, making it easier to identify high-risk areas and take appropriate action. Furthermore, the development of more advanced sensors and IoT technologies can provide more accurate and detailed data about water levels, precipitation, and other environmental factors, further improving the accuracy of the models. Finally, collaborations between scientists, researchers, and industry leaders can help drive innovation and accelerate the development of these models, leading to more accurate and effective flood prediction and prevention strategies.there are several other potential enhancements that can be made to improve flood prediction in the future are:Improved modeling techniques, Increased use of satellite imagery and remote sensing data,Better integration of data sources,Enhanced community engagement,Development of early warning systems,Increased investment in infrastructure.Overall, a combination of advanced technology, community engagement, and infrastructure investment can help improve flood prediction and reduce the impact of flooding on communities.

Chapter 8

PLAGIARISM REPORT

Plagiarism Detector .net

Apr 23, 2024

Plagiarism Scan Report

9% Plagiarized

91% Unique

Characters:1768	Words:241
Sentences:11	Speak Time: 2 Min

Excluded URL	None

Content Checked for Plagiarism

Flood prediction is a critical area of research due to its significant impact on public safety, infrastructure, and the environment. Traditional methods of flood prediction often rely on deterministic models that have limitations in handling the complex interactions of various environmental factors. In recent years, machine learning techniques have emerged as promising tools for improving the accuracy and timeliness of flood forecasts. This paper presents an overview of the application of machine learning in flood prediction. We discuss the key steps involved in developing a machine learning-based flood prediction system, including data collection, feature selection, model training,

Figure 8.1: **Plagiarism**

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
import from flask import Flask , render template , request
import pandas as pd
from sklearn model selection import train test-split
from sklearn.linear model import Logistic Regression.
from sklearn neighbors import KNeighbors Classifier
from sklearn model selection import cross val score
from sklearn.metrics import accuracy score , precision.score.recall.score.classification.report
from sklearn tree import Decision TreeClassifier
import xgboost as xgb
import pickle
app.config['upload folder'] 'C:\Users\YMTS0356 Pycharm Projects\floods\upload'
app Flask (name)
@app.route("/")
return render template('index.html')
def home():
def cleaning (file):
data file [[ Mar-May, Jun-Sep.10days.june increased Rainfall , flood']]
return data
def spliting(file):
global X.y
X file drop ([flood].axis 1)
file flood
x train x test y train y test train test.split(X,y.test.size=0.2.random state 9)
return x train x test , y train y test
@app.route(/upload" methods ["POST", "GET"])
def upload():
if request. method="POST":
file request.files["file
global df
print (file)
filetype os.path.splitext (file filename)[1]
print (filetype)
if filetype.csv':
pathos.path.join(app.config['upload folder"], file filename)
file save(path)
```

```
dfpd.read.csv(path)
df.drop(['Unnamed: 01.axis 1.inplace. True)
return render template('view.html.col name df.columns.row val list (df.values.tolist (0))
else:
return render template (upload.html)
return render template("upload.html")
@app.route("/model, methods=['POST', "GET"])
def model():
if request.method "POST":
clean data cleaning (df)
x.train, s.test.y train, y test spliting(clean data)
model int (request form["model"])
if model == 1 :
knn KNeighbors Classifier()
score cross_val_score (knn.X, r, cv = 5 )
print(score)
print(score. mean())
knknn.fit (x train, y train)
pre kn.predict(x_test)
file models/knn.h5
pickle.dump(kn, open (file, 'wb'))
scores accuracy score (y_test, pre)
print(scores)
return render template('model.html', msg = 'success',score scores, Selected =^ prime KN * N'
if model 2:
dt Decision Tree Classifier()
score cross_val_score (dt.X,y.cv5)
print (score)
print(score. mean())
d=dt.fit(x_train, y train)
file models/dt.h5'
pickle.dump(d, open (file, 'wb'))
pre d. predict(x-test)
scores accuracy score (y test.pre)
print(scores)
return render template (model.html', msg 'success', score scores, Selected 'Decision
Tree Classifier')
if model == 3:
Ir = LogisticRegression ()
score cross_val.score (Ir mathcal ,X,y,cv=5
print(score)
print(score.mean())
llr.fit (x train ,y-train)
pre l.predict(x_test)
file models/Ir.h5
pickle.dump(l, open(file, 'wb'))
scores accuracy score (y test, pre)
print(scores)
return render template('model.html', msg = success score Regression') scores, Selected = 'Logistic
if model 4:
```

```
xg= xgb .XGBClassifier()
score cross.val.score (xg.X.y. cv = 5 )
print(score)
print(score.mean ())
x=xg. fit (x\ train ,y\ train)
pre => . predict(x−test)
file 'models/xgb.h5
pickle.dump(x, open (file , 'wb'))
scores accuracy score (y_test.pre)
print (scores).
return render template (model.html', msg 'success', score scores, Selected 'XGBoost")
return render template('model.html')
@app.route(/prediction methods ["POST". "GET"])
def prediction ():
if request.method "POST":
a = float (request.form['f1'])
bfloat (request.form [12])
c = float (request.form [ˆ ∗ f 3ˆ ∗ \
d = (request.form['f4'])
values [[float (a), float (b).float(c), float (d)]]
filenamr'models/xgb.h5
model pickle.load (open (filenam. "rb"))
ex = . DataFrame(values.columns=X.columns)
pred model. predict (ex)
print (pred)
return render template (prediction.html',res pred)
return render_template('prediction.html")
if name='main':
app.run (debug=True)
```

9.2 Poster Presentation

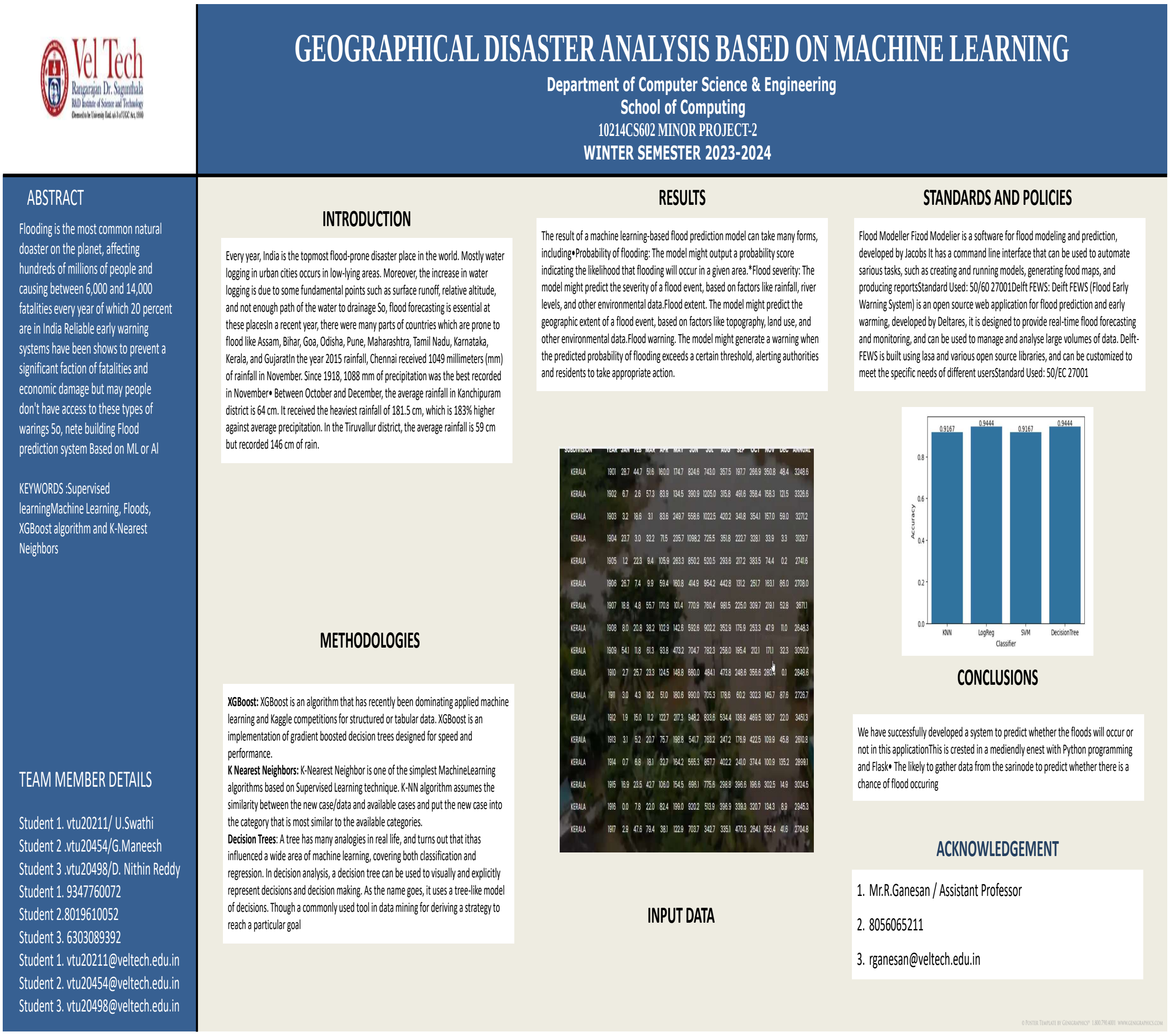

Figure 9.1: **Poster**

References

[1] J. Akshya and P. L. K. Priyadarsini, ”A Hybrid Machine Learning Approach for Classifying Aerial Images of Flood-Hit Areas,” 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), 2019, pp. 1-5, doi: 10.1109/ICCIDS.2019.8862138.

[2] A. B. Ranit and P. V. Durge, ”Different Techniques of Flood Forecasting and Their Applications,” 2018 International Conference on Research in Intelligent and Computing in Engineering (RICE), 2018, pp. 1-3, doi: 10.1109/RICE.2018.8509058.

[3] F. A. Ruslan, K. Haron, A. M. Samad and R. Adnan, ”Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang,” 2017 IEEE 13th International Colloquium on Signal Processing its Applications (CSPA), 2017, pp. 179-184, doi: 10.1109/CSPA.2017.8064947.

[4] F. R. G. Cruz, M. G. Binag, M. R. G. Ga and F. A. A. Uy, ”Flood Prediction Using Multi-Layer Artificial Neural Network in Monitoring System with Rain Gauge, Water Level, Soil Moisture Sensors,” TENCON 2018 - 2018 IEEE Region 10 Conference, 2018, pp. 2499-2503, doi: 10.1109/TENCON.2018.8650387.

[5] G. Kaur and A. Bala, ”An Efficient Automated Hybrid Algorithm to Predict Floods in Cloud Environment,” 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), 2019, pp. 1-4, doi: 10.1109/CCECE.2019.8861897.

[6] J. M. A. Opella and A. A. Hernandez, ”Developing a Flood Risk Assessment Using Support Vector Machine and Convolutional Neural Network: A Conceptual Framework,” 2019 IEEE 15th International Colloquium on Signal Processing Its Applications (CSPA), 2019, pp. 260-265, doi: 10.1109/CSPA.2019.8695980.

[7] Qianyu Zhang., Nttha Jindapetch.,Rakkrit Duangsoithong and Dujdow Buranapanichkit., ”Investigation of Image Processing based Real-time Flood Monitoring,” in Proc. of the 2018 IEEE 5th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA 2018), Songkhla, Thailand, 2018.