**Flood Prediction Using Deep Learning**

**A PROJECT REPORT**

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF**

**BACHELOR OF ENGINEERING IN**

**COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

Submitted by

|  |  |  |
| --- | --- | --- |
| **BABU VIJAYA RAGHAVAN M** | **–** | **201012006** |
| **BOOBALAN K** | **–** | **201012009** |
| **KARTHICK P** | **–** | **201012016** |

Project Guide

**Dr.G.ARULSELVI**

**Associate Professor**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ANNAMALAI ** **UNIVERSITY**

**ANNAMALAI NAGAR – 608 002**

**May 2024**

ANNAMALAI  UNIVERSITY

**FACULTY OF ENGINEERING AND TECHNOLOGY DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

This is to certify that the Project report titled **“ Flood Prediction Using Deep Learning ”** is the bonafide record of the work done by

|  |  |  |
| --- | --- | --- |
| **BABU VIJAYA RAGHAVAN M** | **–** | **201012006** |
| **BOOBALAN K** | **–** | **201012009** |
| **KARTHICK P** | **–** | **201012016** |

in partial fulfillment of the requirements for the Degree of Bachelor of Engineering in Computer Science and Engineering (Artificial Intelligence & Machine Learning) during the period 2023 - 2024.

**Dr. G. Arulselvi Dr. R. BHAVANI**

Associate Professor

Dept. of Computer Science & Engg. Project Guide

Professor & Head Dept. of Computer Science & Engg.

Internal Examiner External Examiner

Place: Annamalainagar Date:

# ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to **Dr. R. BHAVANI, M.E., Ph.D., Professor & Head**, Department of Computer Science and Engineering, Faculty of Engineering and Technology, Annamalai University for giving us the opportunity to undertake this project.

We would like to convey our heartiest thanks to our project guide **Dr. G. ARULSELVI, M.E., Ph.D., Associate Professor**, Department of Computer Science and Engineering, for all her help and support. She with her extreme patience has guided us in situation of need for which we are extremely grateful.

We would like to thank our project review committee members **Dr. S. SATHIYA M.E, Ph.D, Assistant Professor** and **Dr. M. ARULSELVI M.E., Ph.D., Associate Professor**, Department of Computer Science and Engineering for their great support and encouragement during the course of our project.

We would like to thank all our friends for their support in time of need, encouragement and were always ready to help us without asking for it.

We wish to thank all the technical staff members, incharge of our department laboratories that fulfilled all our project needs and offered us timely help.

Above all, we are indebted to our beloved parents, whose blessings and best wishes have come a long way in making this final year project work a grand success.

**Babu vijaya raghavan M Boobalan K**

**Karthick P**

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **CONTENT** | **PAGE NO.** |
|  | ABSTRACT (ENGLISH) | iii |
|  | ABSTRACT (TAMIL) | iv |
|  | LIST OF FIGURES | v |
|  | LIST OF ABBREVIATIONS | vi |
| **1** | **INTRODUCTION** | 1 |
| 1.1 PROBLEM STATEMENT | 1 |
| 1.2 OBJECTIVE | 2 |
| 1.3 ORGANIZATION OF THE REPORT | 2 |
| **2** | **LITERATURE SURVEY** | 3 |
| **3** | **METHODOLOGY** | 11 |
| 3.1 LONG SHORT TERM MEMORY (LSTM) | 11 |
| 3.2 SYTEM REQUIREMENTS | 13 |
| **4** | **IMPLEMENTATION** | 16 |
| 4.1 IMPORT MODULES | 16 |
| 4.2 DATA COLLECTION | 16 |
| 4.3 PRE DATA VISUALIZATION | 17 |
| 4.4 DATA PRE-PROCESSING | 23 |
| 4.5 ALGORITHM SELECTION | 24 |
| 4.6 BUILD MODEL | 26 |
| **5** | **CONCLUSION** | 32 |
|  | **REFERENCES** | 33 |

**ABSTRACT (ENGLISH)**

The project "Flood Prediction Using Deep Learning" aims to address the growing challenges associated with urban flooding in by leveraging advanced deep learning techniques. Chennai District is susceptible to flooding due to a combination of rapid urbanization, inadequate drainage systems, and unpredictable weather patterns.

This project employs Long Short Term Memory (LSTMs), to analyze historical weather data, topographical information, and urban infrastructure parameters. The model is trained to recognize complex patterns and relationships within these datasets to make accurate predictions about potential flooding events.

The implementation involves data preprocessing, model training, and validation using historical flood data and meteorological information. The developed deep learning model is then deployed to create a predictive system capable of issuing timely warnings to authorities and residents.

The developed model predicts the possibility of flood with a accuracy of 92.7% during training and the model predicts the possibility of flood with a accuracy of 88.9% during testing.

# ABSTRACT (TAMIL)

எங்களுடை%ய ப்ராஜெąக்ட்டின் தடை4ப்பு "டீப் லே4ர்னிங்டைக பயன்படுத்தி ஜெ?ள்ளத்டைத கணிப்பது”. நாங்கள் க%ந்தகா4 தக?ல்கள் மற்றும் சி4 சிறப்பம்சம்களான ஒவ்ஜெ?ாரு ?ரு%ம் மடை3 ஜெபய்த ஜெமாத்த அளவிடைன டை?த்து ஜெசன்டைனயில் ஜெ?ள்ளம் ?ரு?தற்கான ?ாய்ப்பிடைன டீப் லே4ர்னிங்கின் உதவிலேயாடு கண்டுப்பிடிக்க உள்லேளாம்.

லேபாதிய ?டிகால் அடைமப்புகள் மற்றும் கணிக்க முடியாத ?ானிடை4 லேபான்ற?ற்றின் க4டை?யால் ஜெசன்டைன நகர்புறம் ஜெ?ள்ளத்தால் பாதிக்கப்படுகிறது. இதனால் ப4 லேசதாரங்கள் ஏற்படுகின்றன. எனலே? இதுலேபான்ற பாதிப்புகடைள குடைறப்பதற்கான ஒரு முயற்சிதான் எங்கள் ப்ராஜெąக்ட்.

எங்களது ப்ராஜெąக்ட்டில் LSTM (எல்.எஸ்.டி.எம்) என்ற டீப் லே4ர்னிங்

?ழிமுடைறடைய பயன்படுத்தி ஜெ?ள்ளம் ?ரு?தற்கான ?ாய்ப்புகளின் சதவீதத்டைத

கணித்துள்லேளாம். நாங்கள்

LSTM

(எல்.எஸ்.டி.எம்) ?ழிமுடைறயிடைன

லேதர்ந்ஜெதடுத்ததற்கான காரணம் LSTM (எல்.எஸ்.டி.எம்) ?ழிமுடைற ஜெதா%ர்ச்சியான

கா4 தக?ல்கடைள பயன்படுத்தி துல்லியமாக கணிப்பதில் அதிக ஜெசயல்திறடைனக் ஜெகாண்டுள்ளது.இதனால் ஜெ?ள்ளம் ?ரு?தற்கான ?ாய்ப்பிடைன கணிக்க உதவுகிறது.

எங்களது ப்ராஜெąக்ட்டில் பயன்படுத்திய LSTM (எல்.எஸ்.டி.எம்) மா%ல் பயிற்சியின்லேபாது 92.70 சதவீதம் துல்லியத்தன்டைமயும் லேசாதடைனயின்லேபாது 88.9 சதவீதம் துல்லியத்தன்டைமயும் ஜெகாண்டுள்ளது.

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.** | **NAME** | **PAGE NO.** |
| 3.1 | BASIC ARCHITECTURE OF LSTM | 12 |
| 4.1 | MODULES IMPORTING | 16 |
| 4.2 | OPEN CITY WEBSITE | 17 |
| 4.3 | SAMPLE DATASET | 17 |
| 4.4 | CODE FOR YEARLY RAINFALL VISUALIZATION | 19 |
| 4.5 | YEARLY RAINFALL PATTERN (1981 - 2021) | 19 |
| 4.6 | RAINFALL PATTERN FOR EACH YEAR | 20 |
| 4.7 | CODE FOR MONTHLY RAINFALL PATTERN | 22 |
| 4.8 | MONTHLY RAINFALL PATTERN | 22 |
| 4.9 | PREPROCESSED SAMPLE DATASET | 24 |
| 4.10 | CODE FOR DATA PREPROCESSING | 24 |
| 4.11 | CODE FOR MODEL CREATION | 26 |
| 4.12 | TRAINING EPOCHS | 27 |
| 4.13 | MODEL TESTING RESULT | 28 |
| 4.14 | MODEL PERFORMANCE RESULT | 29 |
| 4.15 | MODEL EVALUATION RESULT | 30 |
| 4.16 | WEB APPLICATION INTERFACE | 31 |
| 4.17 | CODE FOR WEB APPLICATION CREATION | 31 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **ABBREVIATION** | **EXPANDED FORM** |
| LSTM | LONG SHORT-TERM MEMORY |
| SVM | SUPPORT VECTOR MACHINE |
| DL | DEEP LEARNING |
| ANN | ARTIFICIAL NEURAL NETWORK |
| CNN | CONVOLUTIONAL NEURAL NETWORK |
| RNN | RECURRENT NEURAL NETWORK |
| KNN | K-NEAREST NEIGHBORS |
| DRNN | DYNAMIC MEMORY NETWORK |
| BPNN | BACK PROPAGATION NEURAL NETWORK |
| PCA | PRINCIPAL COMPONENT ANALYSIS |
| ARIMA | AUTOREGRESSIVE INTEGRATED MOVING AVERAGE |
| STL | SEASONAL DECOMPOSITION OF TIME SERIES |

**CHAPTER 1 INTRODUCTION**

In recent years, the frequency and intensity of floods worldwide have escalated, causing devastating impacts on communities, economies, and ecosystems. The consequences of these natural disasters underscore the critical need for accurate and timely flood prediction systems. Traditional methods of flood forecasting often rely on simplistic models that struggle to capture the complexity of hydrological systems, leading to inadequate predictions and insufficient preparation time for at-risk populations. However, with the advent of deep learning technologies, there exists a promising opportunity to revolutionize flood prediction through advanced data-driven approaches.

# PROBLEM STATEMENT

The problem of flood prediction using deep learning involves developing robust models that can accurately forecast flood events with sufficient lead time, leveraging diverse data sources such as meteorological data, topographic information, historical flood records, and remote sensing data. The key challenges include:

* + - **Data Complexity and Integration:** Flood prediction requires integrating diverse and heterogeneous data sources, including rainfall, river discharge, soil moisture, land cover, and terrain information. Handling and preprocessing such diverse data while maintaining consistency and reliability pose significant challenges.
    - **Spatial and Temporal Dynamics:** Flood events exhibit complex spatial and temporal dynamics, influenced by factors such as rainfall intensity, terrain characteristics, and land use changes. Deep learning models must capture both short-term fluctuations and long-term trends to provide accurate predictions across different spatial scales.
    - **Model Robustness and Generalization:** Deep learning models need to be robust against various sources of uncertainty, including data noise, missing values, and model parameterization. Additionally, they should generalize well across different geographical locations and climatic conditions, without overfitting to specific training datasets.
    - **Interpretability and Transparency:** Despite their high predictive performance, deep learning models often lack interpretability, making it challenging to understand the underlying factors driving flood predictions. Ensuring transparency and explainability in model outputs is essential for building trust among end-users and stakeholders.

Overall, addressing these challenges requires a multidisciplinary approach that integrates expertise from hydrology, meteorology, remote sensing, computer science, and data analytics. By leveraging advances in deep learning algorithms, along with domain-specific knowledge and high-quality datasets, we can develop next-generation flood prediction systems that enhance resilience and mitigate the impacts of flooding on society and the environment.

# OBJECTIVES

This project aims to employ Long Short-Term Memory (LSTM) networks, a powerful type of Recurrent Neural Network (RNN), as the primary method for analyzing historical hydrological data and predicting flood occurrences.Through this project, we aim to contribute to early warning systems and disaster preparedness by providing timely and precise predictions for potential flood events based on historical patterns and real-time data.

# ORGANIZATION OF THE REPORT

In Chapter 1, we discussed about the introduction of Flood Prediction and the problem statements which the flood prediction model can able to solve and the Objective which contains the aim and motive of the flood prediction system.

In Chapter 2, we discussed about the Literature Review which contains the Review of relevant literature, theories, and prior research related to the topic, Summary and analysis of key findings from existing sources and the identification of gaps or areas for further investigation.

In Chapter 3, we discussed about the methodology which contains the explanation of the research methods used and the justification for the chosen methodology.

In Chapter 4, we discussed about the implementation which contains the Description of how the research or project was carried out and the overview of the practical steps taken to address the research objectives.

In Chapter 5, we provided the Conclusions which contains the summary of the key findings and their significance, Discussion of how the results align with the research objectives and the future works that carried out further for better results.

# CHAPTER 2 LITERATURE SURVEY

**Flood Prediction Using Machine Learning,Anil Kumar Ambore, T. Sri Sai Charan, U. Rohit Reddy, (2023)**

The most frequent natural disaster in the world, flooding affects hundreds of millions of people and kills between 6,000 and 18,000 people annually, with 20% of hose deaths occurring in India. Several people lack access to reliable early warning systems, despite the fact that those systems already exists demonstrated may avoid an large portion of economy and death loss. Improved performance and cost-effective solutions are offered by this prediction system\'s development. Inorder to forecast the occurrence of floods brought on by rainfall, a prediction model is created in this article. Based on the rainfall range for certain places, the model forecasts if "flood may happen or not". information about rainfall in Indian districts.

A flood prediction system based on machine learning has the potential to greatly benefit flood-prone communities. Machine learning algorithms can accurately predict the likelihood and severity of a flood by using historical data and real-time monitoring, allowing authorities to take preventative measures to minimize the impact on the community.

Machine learning models, such as Random Forest, can be trained on large datasets of historical flood data, weather data, topographical information, and other relevant factors to accurately predict the likelihood of future floods. Implementing a flood prediction system based on machine learning can save lives, reduce property damage, and improve emergency response efforts. Overall, the potential benefits of such a system make it a worthwhile investment for any flood- prone community.

**Flood Prediction Using Machine Learning Models,Miah Mohammad Asif Syeed,Ishadie Namir, (2022)**

Floods are one of nature's most catastrophiccalamities which cause irreversible and immense damage tohuman life, agriculture, infrastructure and socio-economicsystem. Several studies on flood catastrophe management and flood forecasting systems have been conducted. The accurate prediction of the onset and progression of floods in real time is challenging. To estimate water levels and velocities across a large area, it is necessary to combine data with computationally demanding flood propagation models. This paper aims to reduce the extreme

risks of this natural disaster and also contributes to policy suggestions by providing a prediction for floods using different machine learning models.

This research will use Binary Logistic Regression, K-Nearest Neighbor (KNN), Support Vector Classifier (SVC) and Decision Tree Classifier to provide an accurate prediction. With the outcome, a comparative analysis will be conducted to understand which model delivers a better accuracy.

As climate changes over the years and depending onother parameters, the thresholds for floods are changing. That’s why the shorter timeline of data gives slightly better accuracy. Since these change over a long time period, in this research, the models gave higher accuracy with a shorter time range. Also, due to time constraint only the rainfall data along with flood occurrence was manageable. There are more factors related to flood like, river water level, temperature, humidity, other natural disasters etc. In the future, this research paper would attempt to develop the models further by adding the other factors and correlating them.

**A short-term flood prediction based on spatial deep learning network, Chen Chen,Jiange Jiang,ZhanLiao, (2022)**

Floods cause substantial damage across the world every year. Accurate and timely prediction of floods can significantly minimize the loss of life and property. Recently, numerous machine learning models have been used for flood prediction, showing that their performance is preferable to traditional statistical models. However, the existing models neglect the spatial features of floods, which driveflood generation and concentration. In this paper, the area of interest is divided into grids based on longitude and latitude, and the rainfall and discharge collected by stations are combined into tensors according to station coordinates. Different from one-dimensional time series, our input feature is a two-dimensional time series with spatial information.

Hence, combining a Convolutional Neural Network (CNN) with a Long Short Term MemoryNetwork (LSTM), we propose the convolution LSTM (ConvLSTM) to extract spatiotemporal features of hydrological information. The methodology is demonstrated using the hydrological data collected at the Xi County stations, located on the Huai River in Henan Province, China. Numerical results indicate that the relative error of arrival time is within 30%, and the relative error of peak discharge is within 20%, satisfying the 2005 Chinese Water Resource Standard on flood prediction permit error. The experiments also show that the ConvLSTM outperforms the recent models in terms of flood arrival time and peak discharge, thereby proving a promising alternative.

In this paper, we divided a rainfall discharge area with hydrological information into grids and transformed it into an image-like structure. Then, we used the proposed ConvLSTM model, combined with image processing capabilities, for flood prediction. Our model incorporates a typical CNN for image processing and a conventional time-series predictor for extracting the spatiotemporal characteristics of hydrological information, which cannot be efficiently handled by

the traditional machine.

**Flood Prediction using Deep Learning Models,Muhammad Hafizi Mohd Ali, Siti Azirah Asmai, Z. Zainal Abidin, (2022)**

Deep learning has recently appeared as one of the best reliable approaches for forecasting time series. Even though there are numerous data-driven models for flood prediction, most studies focus on prediction using a single flood variable. The creation of various data-driven models may require unfeasible computing resources when estimating multiple flood variables. Furthermore, the trends of several flood variables can only be revealed by analysing long-term historical observations, which conventional data-driven models do not adequately support.

This study proposed a time series model with layer normalization and Leaky ReLU activation function in multivariable Long Short Term Memory(LSTM), Bidirectional Long Short Term Memory (BI- LSTM) and Deep Recurrent Neural Network (DRNN). The proposed models were trained and evaluated by using the sensory historical data of river water level and rainfall in the east coast state of Malaysia. It were then, compared to the other six deep learning models. In terms of prediction accuracy, the experimental results also demonstrated that the deep Recurrent Neural Network model with layer normalization and Leaky ReLU activation function performed better than other models.

In conclusion, the DRNN model performs relatively well compared to the LSTM and BI-LSTM models with the used dataset. From the literature, the LSTM architecture needs requirements for backpropagation and a gate to train the model. Therefore, the LSTM model is marginally more complicated than the DRNN model. Meanwhile, the BI-LSTM model performs with somewhat lower accuracy but is still able to,deliver a good outcome. Additionally, the BI-LSTM model requires the training data to move backwards and forward in both directions which increases the training time needed. Even though the performance of proposed models performs well, there are still many improvements that can be made using deep learning approaches.

**A Flood Prediction System Developed Using Various Machine Learning Algorithms,Kruti Kunverji,Krupa Shah, (2021)**

Floods have become the most well-known and lethal cataclysmic events of this century. Absence of a successful flood forecasting framework has brought about grave loss of human existence and infrastructure. This has reiterated on the importance of having in place a flood prediction system. This paper looks at developing the most effective flood determining model. AI calculations and a hearty, productive and precise flood expectation framework will give all the fundamental aid and assistance needed to the residents and government. Hence, the Decision Tree Model is being built.

This model actualizes various calculations on datasets with a scope of accuracy. The model utilizes an AI calculation which predicts floods, sending alerts to the local and government authorities using an Android Application. The three Machine Learning Algorithms used for comparison are Decision Tree, Random Forest and Gradient Boost. This model focuses on improving the rate of prediction by dealing with more intricate information and a high-level algorithm.

In the Decision Tree Machine Learning Algorithm, the parameters collected using an architectural set-up allows seamless integration of data. This data is then fed onto a Machine Learning model which is then able to predict the chances of flood. The proposed framework performs analysis with a high and satisfactory fault-tolerant accuracy.

Result of Gradient Boost Algorithm The system has also been built according to the conditions prevalent in a country like India. The system sends out warnings and alerts of an incoming flood to the citizens and helps save the lives of civilians and if possible, the infrastructure. The system also helps the government save money in rescue operations and helps them start the relocation operations before the flood hits the town. In the future, a collaboration between the forecast of rainfall and flood can be achieved. Using satellite imaging, the civilians can also be informed of safe places that they can relocate to and guide them towards the rehabilitation camps set up by the government.

**Artificial Neural Network for Rainfall Analysis Using Deep Learning Techniques ,S D Nandakumar1, R Valarmathi2, P Sudha Juliet, (2021)**

The estimation of rainfall is one of the most critical and daunting challenges in today's environment. Weather and rainfall are typically extremely nonlinear and dynamic, needing sophisticated machine models and simulation for forecasting accurately. The economy of India is agriculture and is focused primarily on crop production and precipitation. Predictions of

rainfall areimportant for all farmers to assess crop productivity. Rainfall forecast involves the application of science and technology to determine weather conditions.

In order to utilize water supplies efficiently, the crop productivity and the pre-program of water systems, it is necessary to determine the precipitation in detail. The actions of such nonlinear processes can be modeled using an. Artificial Neural Network (ANN). Most researchers in this area have been effectively utilizing. ANN for the past 25 years. This article offers you an summary of some of the methodologies valid for using ANN for rainfall prediction by numerous researchers.The survey also states that forecastsof rainfall using ANN technologies are more accurate than conventional mathematical and numerical approaches.

This paper presents a comprehensive study of forecasts of rainfall over 25 years utilizing various models of the neural networks. The study showed that most researchers employed a rainfall forecast back propagation network and had good results. This study also suggests that MLP, BPN, RBFN, SOM and SVM simulation strategies are sufficient in order to forecast precipitation over other strategies such as statistics and statistical structures. Nevertheless, there have been several limitations in these approaches.ANN researchers will be beneficial in reliably predicting rainfall in the future through detailed references to the numerous advances in ANN's work found in this article.

**Flood prediction based on weather parameters using deep learning, Suresh Sankaranarayanan,Malavika Prabhakar,Sreesta Satish, (2020)**

Today, India is one of the worst flood-affected countries in the world, with the recent disaster in Kerala in August 2018 being a prime example. A good amount of work has been carried out by employing Internet of Things (IoT) and machine learning (ML) techniques in the past for flood occurrence based on rainfall, humidity, temperature, water flow, water level etc. However, the challenge is that no one has attempted the possibility of occurrence of flood based on temperature and rainfall intensity.

So accordingly Deep Neural Network has been employed for predicting the occurrence of flood based on temperature and rainfall intensity. In addition, a deep learning model is compared with other machine learning models (Support Vector Machine (SVM), K-Nearest Neighbor (KNN) and Naïve Bayes) in terms of accuracy and error. The results indicate that the deep neural network can be efficiently used for flood forecasting with highest accuracy based on monsoon parameters only before flood occurrence.

India contributes one-fifth of the world's global deaths due to floods and is the world's worst affected country after Bangladesh. The Indian rainfall season is mainly from June to September and accounts for nearly 75% of the total Indian rainfall per year. Much work has

been carried out by employing machine learning algorithms such as ANN for flood prediction. Most of the systems employed ANN with a single hidden layer for prediction of flood with parameters such as rainfall, temperature, water flow, water level and humidity. One system was developed using a deep neural network where stream flow was taken into consideration for the prediction of flood.The challenge in all these system is that most used traditional ANN and with the advent of the deep neural network, we can predict the possibility of flood occurrence with higher accuracy based on rainfall and temperature.

**Prediction Analysis of Floods Using Machine Learning Algorithms (NARX & SVM),Nadia Zehra, (2020)**

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 algorithm in perspective of flood estimation.

This report was aimed at the meta-analysis of previously reported articles in the context of flood forecasting and the techniques targeted were the SVM and NARX. NARX is a type of NN and is widely used in terms of time series prediction. Based on the comparison, literature review and synthesis stated above, it is concluded that use of statistical methods with NARX can provide highly accurate and promising results for flood forecast. This study was quite helpful in elaborating the mechanism of those proposed techniques and their comparison with each other so that one can get to know which method is better and how.

**Flood prediction forecasting using machine Learning Algorithms,Naveed Ahamed, S.Asha,(2020)**

Floods are very harmful for nature, which are very complex to model. The flood prediction model will give risk reduction & it minimizes the future loss of human life. On 18 May 2016 a south Indian state Kerala was affected by flood. Machine learning is a method which provides intelligence to predict the result in future. The performance comparison of ML models is based on the speed, time and accuracy of the result. There exist a lot of machine

algorithms which generate models with more accuracy. For flood prediction classification algorithms like decision tree and linear regression are used in this research. This paper will present the dataset of Kerala flood 2016 which is provided by government.

In this paper, the simple machine learning algorithm to predict the accuracy of the flood occurrence is implemented. The desired algorithm shows the results of occurrence of flood in the upcoming year. When compared with the otheralgorithms, the decision tree algorithm gives more accurate results and provide high performance accuracy and easy to understand. The decision tree also generate model for nonlinear dataset. This nonlinear can be applied to find the accuracy of linear or logistic dataset. As the compared results shows that the decision tree gives more accuracy compared to other simple machine learning algorithm.

**Flood Prediction Using Machine Learning,Amir Mosavi 1,Pinar Ozturk, (2018)**

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. The main contribution of this paper is to demonstrate the state of the art of ML models in flood prediction and to give insight into the most suitable models. In this paper, the literature where ML models were benchmarked through aqualitative analysis of robustness, accuracy, effectiveness, and speed are particularly investigated to provide an extensive overview on the various ML algorithms used in the field.

The performance comparison of ML models presents an in-depth understanding of the different techniques within the framework of a comprehensive evaluation and discussion. As a result, this paper introduces the most promising prediction methods for both long-term and short-term floods. Furthermore, the major trends in improving the quality of the flood prediction models are investigated. Among them, hybridization, data decomposition, algorithm ensemble, and model optimization are reported as the most effective strategies for the improvement of ML methods. This survey can be used as a guideline for hydrologists as well

as climate scientists in choosing the proper ML method according to the prediction task conclusions.

The current state of ML modeling for flood prediction is quite young and in the early stage of advancement. This paper presents an overview of machine learning models used in flood prediction, and develops a classification scheme to analyze the existing literature. The survey represents the performance analysis and investigation of more than 6000 articles. Among them, we identified 180 original and influential articles where the performance and accuracy of at least two machine learning models were compared.

To do so, the prediction models were classified into two categories according to lead time, and further divided into categories of hybrid and single methods. The state of the art of these classes was discussed and analyzed in detail, considering the performance comparison of the methods available in the literature. The performance of the methods was evaluated in terms of R2 and RMSE, in addition to the generalization ability, robustness, computation cost, and speed. Despite the promising results already reported in implementing the most popular machine learning methods, e.g., ANNs, SVM, SVR, ANFIS, WNN, and DTs, there was significant research and experimentation for further improvement and advancement. In this context, there were four major trends reported in the literature for improving the quality of prediction.

# LSTM

**CHAPTER 3 METHODOLOGY**

Long Short-Term Memory (LSTM) networks are a type of Recurrent Neural Network (RNN) architecture designed to handle the vanishing gradient problem, which is common in traditional RNNs. LSTMs are particularly effective in capturing long-term dependencies in sequential data, making them suitable for tasks like time series prediction, Natural Language Processing(NLP), and speech recognition.

Let's break down the LSTM architecture with mathematical expressions:

* **Forget Gate**: The forget gate controls what information from the previous cell state should be discarded or forgotten. It takes both the previous hidden state ht−1 and the current input xt as inputs and outputs a vector ft with values between 0 and 1 for each element. The forget gate is represented by:

ft = σ (Wf⋅[ht−1,xt]+bf) Where:

* + ***Wf*** *and* ***bf*** *are the weight matrix and bias vector for the forget gate.*
  + ***σ*** *is the sigmoid activation function.*
  + ***[ht−1,xt]*** *represents the concatenation of the previous hidden state and the current input.*
  + **Input Gate**: The input gate determines what new information should be stored in the cell state. It consists of two parts: an input gate and a candidate cell state update. The input gate decides which values will be updated, and the candidate cell state update Ct is the new candidate values for the cell state. The input gate is represented by:

it = σ (Wt⋅[ht-1,xt]+bi)

Ct = tanh(WC⋅[ht−1,xt]+bC) Where:

* + - ***Wi , WC , bi ,*** *and* ***bC*** *are the weight matrices and bias vectors for the input gate and candidate cell state update.*
    - ***tanh*** *is the hyperbolic tangent activation function.*
* **Update Cell State**: This step combines the information to update the cell state Ct−1 into a new cell state Ct . This involves element-wise multiplication of the forget gate output

ft and the previous cell state Ct−1, and element-wise addition of the input gate output it and the candidate cell state update Ct .

Ct = ft∗ Ct−1+it∗ Ct Where:

*\* Represents element-wise multiplication.*

* **Output Gate**: The output gate decides what the next hidden state ht should be based on the updated cell state Ct and the current input xt. The output gate is represented by:

ot = σ (Wo⋅[ht−1,xt]+bo)

ht = ot∗ tanh(Ct) Where:

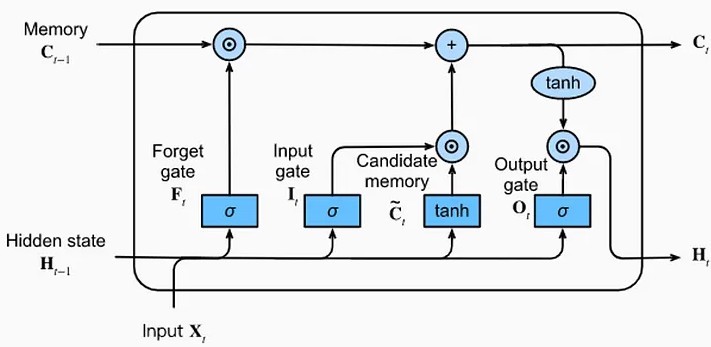
* + ***Wo*** *and* ***bo*** *are the weight matrix and bias vector for the output gate.*
  + ***ot*** *is the output gate.*
  + ***ht*** *is the current hidden state.*
  + ***tanh(Ct)*** *squashes the values in the cell state to the range (−1,1).*

In summary, the LSTM architecture allows the network to learn which information to store in the Long-Term Memory (cell state) and which information to pass to the next time step (hidden state), enabling it to capture long-term dependencies in sequential data.

# MEMORY CELL

The core component of an LSTM. It maintains a cell state vector, which serves as a long- term memory. The information flow in and out of this cell is regulated by gates. They are

* **Input Gate:** Determines which information from the input should be stored in the cell state.
* **Forget Gate:** Controls what information should be discarded from the cell state.
* **Output Gate:** Decides what part of the cell state should be output at the current time step.



**Fig 3.1 Basic Architecture of LSTM**

# SYSTEM REQUIREMENTS

* + 1. **HARDWARE REQUIREMENTS CPU**
* A multi-core CPU for data preprocessing, model training, and inference.
* CPUs with higher clock speeds and multiple cores are beneficial for faster data processing.

# RAM

* Sufficient RAM is required to store datasets, model parameters, and intermediate computations during training and inference.
* At least 4 GB of RAM is recommended, but larger datasets may require 8 GB or more.

# SOFTWARE REQUIREMENTS

* + - 1. **SOFTWARE MODULES USED**
         * **Pandas**

Pandas is a powerful Python library widely used for data manipulation and analysis. It provides data structures and functions designed to make working with structured data intuitive, fast, and easy.

**Key Features**:

**Data Alignment**: Pandas automatically aligns data based on label when performing operations, making it easy to work with incomplete or differently-indexed data.

**Handling Missing Data**: Pandas provides methods like dropna(), fillna(), and isnull() to handle missing data effectively.

**Grouping and Aggregation**: Pandas allows grouping data based on one or more keys and applying aggregation functions like sum, mean, count, etc., using groupby().

* + - * + **Numpy**

NumPy (Numerical Python) is a fundamental package for numerical computing in Python. It provides support for multidimensional arrays, along with a collection of mathematical functions to operate on these arrays efficiently.

**Key Features**:

**Vectorized Operations**: NumPy allows performing element-wise operations on arrays, making computations faster and more concise compared to traditional Python loops.

**Broadcasting**: NumPy automatically handles operations between arrays of different shapes and sizes through broadcasting, which extends smaller arrays to match the shape of larger ones.

**Universal Functions (ufuncs)**: NumPy provides a wide range of mathematical functions, such as trigonometric, exponential, logarithmic, statistical, etc., optimized for array operations.

* + - * + **Tensorflow**

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive ecosystem for building, training, and deploying machine learning models.

**Key Features**:

**Flexible Architecture**: TensorFlow offers a flexible and modular architecture that allows users to build a wide variety of machine learning

models, including Recurrent Neural Networks (RNNs), and more.

**Efficient Execution**: TensorFlow optimizes the execution of computational graphs for performance, including support for GPU and TPU acceleration to speed up computations.

**Wide Range of Tools and Libraries**: TensorFlow ecosystem includes various tools and libraries for different machine learning tasks, such as TensorFlow Hub for reusable machine learning modules, TensorFlow Lite for deploying models on mobile and edge devices, TensorFlow.js for training and deploying models. TensorFlow also integrates with popular deep learning frameworks like Keras, making it easy to leverage pre-built models and components.

* + - * + **Matplotlib**

Matplotlib is a comprehensive Python library for creating static, interactive, and animated visualizations. It provides a wide range of functionalities for generating plots, charts, histograms, scatter plots, and more.

**Key Features:**

**Support for Various Plot Types**: Matplotlib supports a wide range of plot types, including line plots, scatter plots, bar plots, histograms, pie charts, 3D plots, contour plots, and more.

**Customization Options**: Matplotlib offers extensive customization options to control the appearance and behavior of plots.

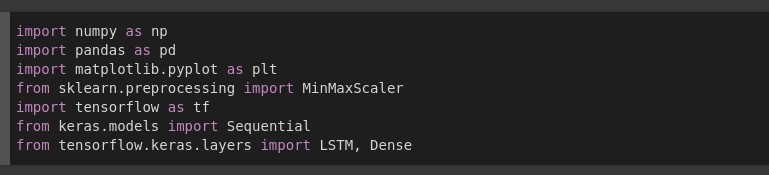
Users can customize various aspects of plots, such as colors, line styles, markers, labels, axes, grids, legends, titles, fonts, and more.

**CHAPTER 4**

**IMPLEMENTATION**

* 1. **IMPORT MODULES**

Importing modules is a fundamental concept in programming, allowing you to access functions, classes, and variables defined in other Python files or libraries.



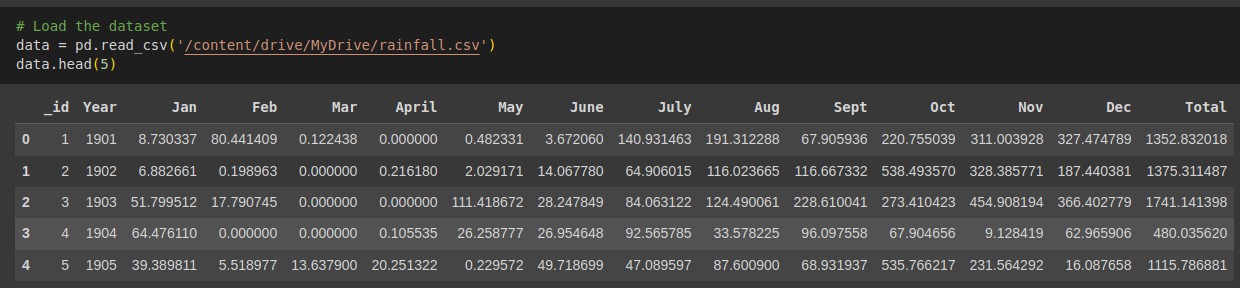
**Fig 4.1 Modules Importing**

# DATA COLLECTION

Data collection is the process of gathering and measuring information on targeted variables in a systematic manner. It involves various methods such as surveys, observations, interviews, and experiments to collect relevant data. The collected data can be quantitative (numerical) or qualitative (descriptive). The aim of data collection is to obtain accurate and reliable information that can be analyzed to draw conclusions, make decisions, or support research objectives. Effective data collection involves careful planning, selecting appropriate methods, ensuring data quality, and maintaining ethical standards throughout the process. Data has been collected from opencity weather API. Total number of samples present in our dataset is 1573 samples. Total number of samples for Training is 1100 samples. Total number of samples for Testing is 473 samples. The columns present in our dataset are Jan, Feb, Mar, April, May, June, July, Aug, Sept, Oct, Nov, Dec, Total, Flood.



**Fig 4.2 Open City Website**

****

**Fig 4.3 Sample Dataset**

# PRE DATA VISUALIZATION

Visualization, in a general sense, refers to the process of creating visual representations of data or concepts. It involves using graphical elements such as charts, graphs, maps, and diagrams to convey information in a clear and meaningful way. The goal of visualization is to make complex data more understandable, enabling easier analysis, pattern recognition, and decision-making.

There are several key aspects to visualization:

**Data Representation**: Visualization starts with the data itself. This could be numerical data, text, images, or any other form of information. The data is then transformed into visual elements that can be perceived by the human eye.

**Visual Encoding**: Once the data is ready, it needs to be encoded into visual attributes such as position, size, color, shape, and texture. Each of these attributes can represent different aspects of the data, such as value, category, or relationship.

**Interactivity**: Many visualizations allow users to interact with the data, enabling exploration and deeper understanding. Interactivity can include features like zooming, filtering, sorting, and selecting specific data points.

**Storytelling**: Effective visualizations often tell a story or convey a message. They guide the viewer through the data, highlighting key insights, trends, and relationships. Annotations, captions, and contextual information can help in this storytelling process.

**Tools and Technologies**: There are various tools and technologies available for creating visualizations, ranging from simple spreadsheet software to advanced programming libraries and platforms. These tools provide different levels of flexibility, customization, and automation.

Visualization is widely used in fields such as data analysis, business intelligence, scientific research, education, journalism, and design. It can be applied to diverse datasets, including financial data, geographic information, biological data, social networks, and more.

Ultimately, visualization is a powerful tool for making sense of complex information, uncovering hidden patterns, and communicating insights effectively. It bridges the gap between data and understanding, helping people make informed decisions and discoveries.

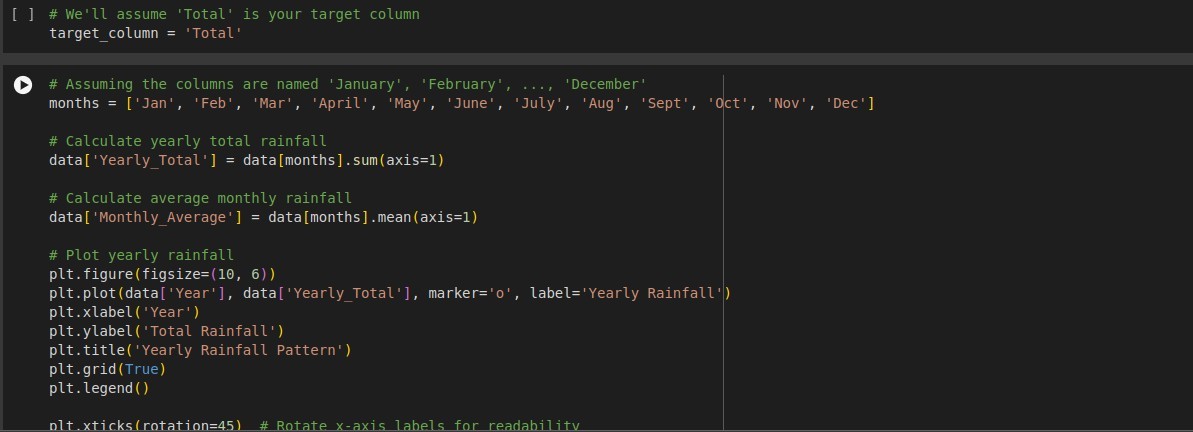
Before training a model, it's important to understand the characteristics of the data. Data visualization techniques such as histograms, scatter plots, and box plots can help visualize the distribution of features, relationships between variables, and identify potential outliers or anomalies.

# YEARLY RAINFALL DATA FOR THE PERIOD 1981 - 2021

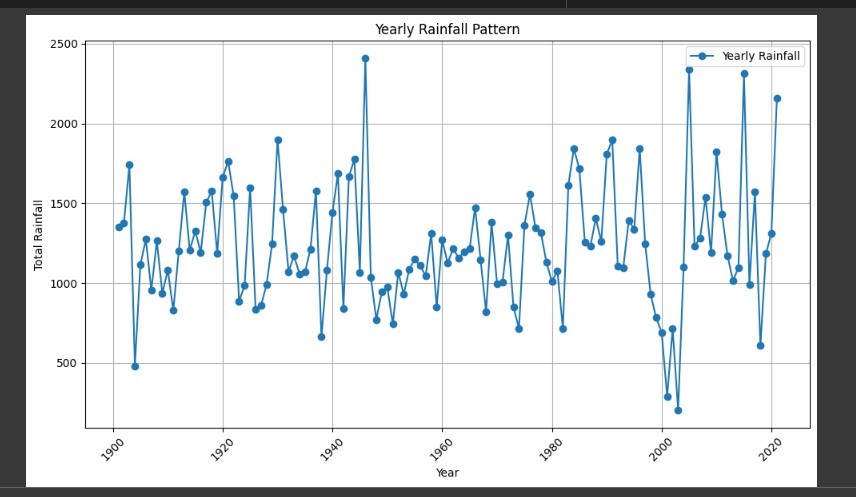
Yearly rainfall pattern refers to the variation in precipitation levels over the course of a year within a specific geographical area. This pattern is influenced by various factors such as geographic location, topography, climate systems, and atmospheric conditions. In regions with distinct wet and dry seasons, the yearly rainfall pattern typically follows a cyclical pattern.

In some regions, rainfall may be relatively evenly distributed throughout the year, resulting in a consistent pattern of precipitation. This is common in areas with a tropical or temperate climate, where rainfall can occur in regular intervals.

In other regions, rainfall patterns may be more variable, with distinct wet and dry seasons. For example, many tropical regions experience a wet season during the summer months, characterized by frequent and intense rainfall, followed by a drier season with less precipitation.



**Fig 4.4 Code For Yearly Rainfall Visualization**

****

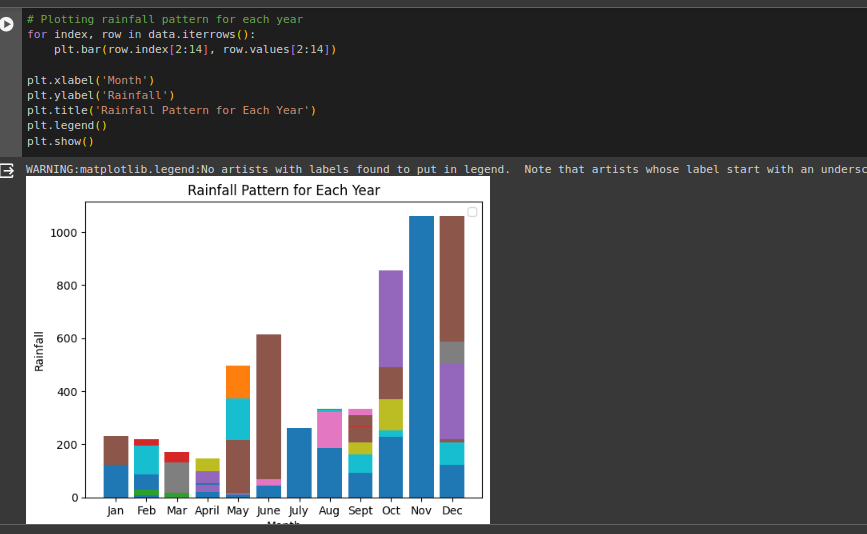
**Fig 4.5 Yearly Rainfall Pattern (1981 - 2021)**

# RAINFALL PATTERN FOR EACH YEAR

A rainfall pattern graph typically displays the amount of rainfall over a specified period, usually annually. Here's a breakdown of what you might see on such a graph:

* + - * **X-axis**: This represents time, usually in years, with each year plotted along the axis. It could also be broken down into months if the data is more detailed.
      * **Y-axis**: This represents the amount of rainfall, usually measured in millimeters or inches. The scale depends on the range of rainfall values observed in the dataset.
      * **Data points**: Each data point on the graph represents the amount of rainfall recorded for a specific time period (year or month). These points are typically plotted as dots or markers along the graph.

By examining the rainfall pattern graph, you can visually identify trends, seasonal variations, and anomalies in the data, which can be valuable for understanding climate patterns, water resource management, agriculture, and environmental studies.



**Fig 4.6 Rainfall Pattern For Each Year**

# RAINFALL PATTERN FOR EACH MONTH

Rainfall patterns vary greatly depending on factors like geographical location, climate, and local weather patterns. However, I can provide a general overview of typical rainfall patterns for each month in some regions:

**January** - In many temperate regions, January is often characterized by cold temperatures and low precipitation, which may fall as snow in some areas. In tropical regions, January may be part of the dry season with minimal rainfall.

**February** - Similar to January, February may experience cold temperatures and low precipitation in temperate regions. In some tropical regions, February may still be part of the dry season, while in others, it may mark the beginning of the wet season.

**March** - March often marks the transition to spring in temperate regions, with increasing temperatures and occasional rainfall. In tropical regions, March may signal the onset of the wet season with more frequent and heavier rainfall.

**April** - April typically sees increasing temperatures and more rainfall in temperate regions as spring progresses. In tropical regions, April may experience heavy rainfall as the wet season continues.

**May** - May is often characterized by warmer temperatures and frequent rainfall in temperate regions, supporting the growth of vegetation. In tropical regions, May may still experience heavy rainfall, particularly in the early part of the wet season.

**June** - In many temperate regions, June marks the beginning of summer with warmer temperatures and occasional rainfall. In tropical regions, June may still experience significant rainfall, though it may begin to taper off towards the end of the month in some areas.

**July** - July is often one of the warmest months in temperate regions, with sporadic rainfall. In tropical regions, July may continue to experience rainfall, though it may be less frequent compared to earlier months in the wet season.

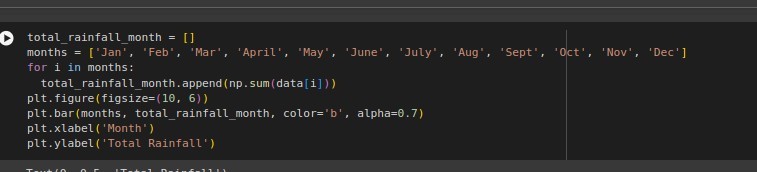
**August** - August typically sees warm temperatures and variable rainfall in temperate regions. In tropical regions, August may still experience some rainfall, though it may decrease further as the wet season nears its end in some areas.

**September** - September often marks the transition to autumn in temperate regions, with cooler temperatures and decreasing rainfall. In tropical regions, September may still experience some rainfall, particularly in regions with a longer wet season.

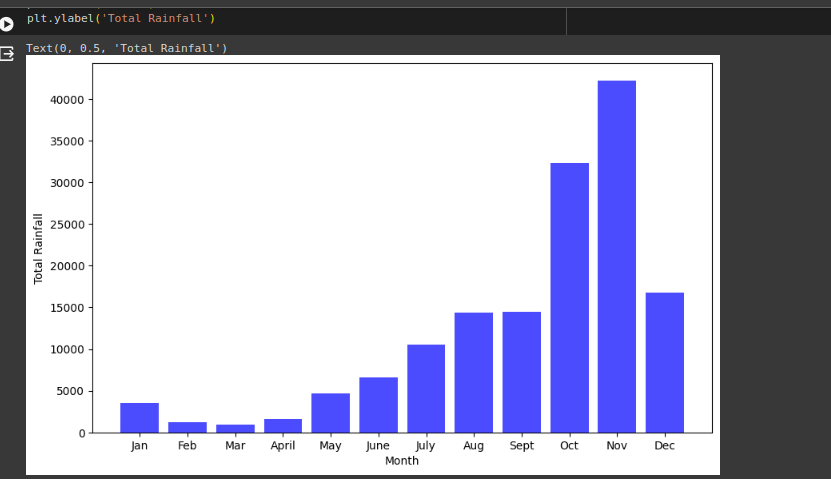
**October** - October continues the trend of decreasing temperatures and rainfall in temperate regions, often leading into the fall season. In tropical regions, October may experience variable rainfall patterns depending on the specific climate and weather patterns.

**November** - November may see further decreases in temperature and rainfall in temperate regions, sometimes leading to drier conditions. In tropical regions, November may mark the end of the wet season in some areas, with rainfall becoming less frequent.

**December** - December is often characterized by cold temperatures and variable precipitation in temperate regions, with the possibility of snowfall in some areas. In tropical regions, December may experience variable rainfall patterns, with some areas transitioning to the dry season while others continue to receive rainfall.



**Fig 4.7 Code For Monthly Rainfall Pattern**

****

**Fig 4.8 Monthly Rainfall Pattern**

# DATA PRE-PROCESSING

Data preprocessing is a crucial step in the data analysis pipeline that involves cleaning, transforming, and organizing raw data into a format suitable for further analysis. Here's a detailed explanation of the key steps involved in data preprocessing:

**Data Cleaning:**

* **Handling Missing Values:** Identify and deal with missing values in the dataset. Options include imputation (replacing missing values with estimated ones), deletion of rows or columns with missing values, or using advanced techniques like interpolation.
* **Handling Outliers:** Detect and address outliers in the data that may skew analysis results. Techniques such as trimming (capping extreme values), transformations (e.g., logarithmic transformation), or using robust statistical methods can be employed.
* **Data Deduplication:** Identify and remove duplicate records from the dataset to ensure data integrity and avoid bias in analysis.

**Data Transformation:**

* **Feature Scaling:** Scale numerical features to a similar range to prevent features with larger magnitudes from dominating the analysis. Common scaling methods include normalization (scaling features to a range between 0 and 1) and standardization (scaling features to have a mean of 0 and a standard deviation of1).
* **Encoding Categorical Variables:** Convert categorical variables into numerical representations that can be used in mathematical models. Common techniques include one-hot encoding, label encoding, and binary encoding.
* **Feature Engineering:** Create new features from existing ones to improve the performance of Deep learning models. This can involve mathematical transformations, creating interaction terms, or deriving new features based on domain knowledge.

**Data Normalization:**

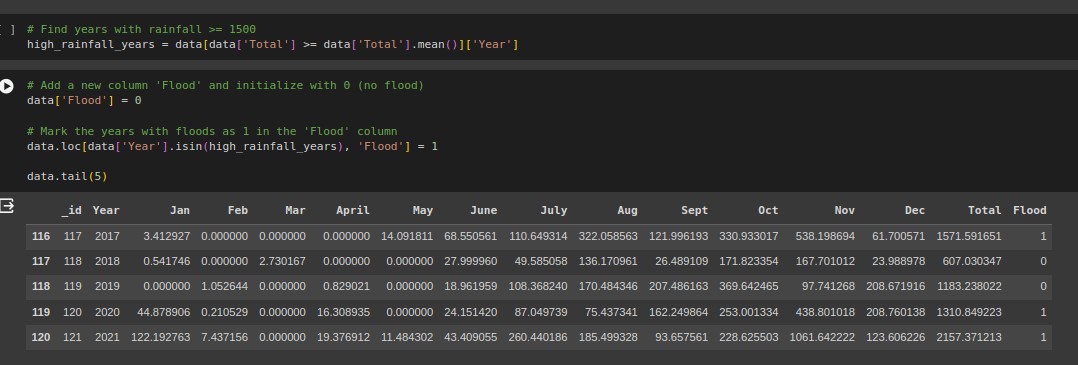
* Normalize data to ensure that all features have similar scales and distributions. This can improve the performance of machine learning algorithms and make the

data more interpretable. Techniques include Min-Max scaling, z-score normalization, and log transformation.

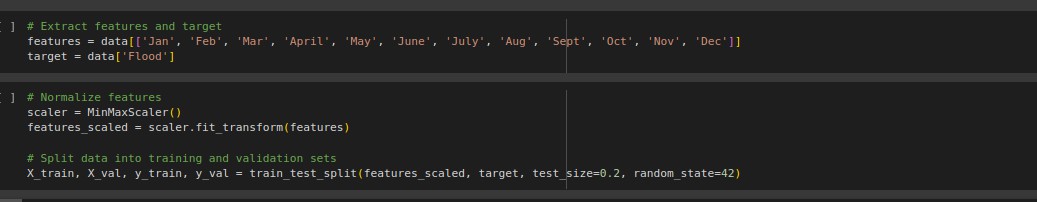
**Data Formatting:**

* Ensure that the data is in the appropriate format for analysis tools and algorithms. This may involve converting data types, reshaping data structures, or standardizing date formats.

By performing these preprocessing steps, analysts can improve the quality of their data, reduce noise and bias, and prepare the data for effective analysis and modeling.



**Fig 4.9 Preprocessed Sample Dataset**

****

**Fig 4.10 Code For Data Preprocessing**

# ALGORITHM SELECTION

Selecting the appropriate algorithm for flood prediction involves considering various

factors such as the nature of the data, the complexity of the problem, computational resources, and the specific requirements of the application. Here are some common algorithms used for flood prediction, along with considerations for their selection:

* + 1. **Regression Models:**
       - **Linear Regression:** Simple and interpretable model suitable for predicting continuous variables (e.g., river flow, precipitation). It assumes a linear relationship between input features and the target variable.
       - **Polynomial Regression:** Extends linear regression by including polynomial terms, allowing for more flexible relationships between features and the target variable.

**Considerations:** Regression models are suitable when the relationship between input features and the target variable is relatively simple and linear. They are computationally efficient and provide insights into the relative importance of features.

* + 1. **Time Series Forecasting Models:**
       - **Autoregressive Integrated Moving Average (ARIMA):** Suitable for time series data with trend and seasonality. ARIMA models capture the autocorrelation and temporal dependencies in the data.
       - **Seasonal Decomposition of Time Series (STL):** Decomposes time series data into trend, seasonal, and residual components, enabling separate modeling of each component.
       - **Long Short-Term Memory (LSTM) Networks:** A type of Recurrent Neural Network (RNN) capable of learning long-term dependencies in time series data. LSTM networks are effective for capturing complex temporal patterns.

**Considerations:** Time series forecasting models are appropriate when predicting flood events based on historical data with temporal dependencies. They can capture seasonality, trends, and other temporal patterns in the data.

When selecting an algorithm for flood prediction, it's essential to assess its strengths and weaknesses in relation to the specific requirements of the application, the availability of data, and the expertise of the modeling team.

Ensemble methods and hybrid approaches that combine multiple algorithms can also be effective in improving prediction accuracy and robustness.

Additionally, model evaluation through cross-validation and validation against independent datasets is crucial to assess the generalization performance of the selected algorithm.

**Reason For Selecting LSTM:** We choose LSTM alogorithm to train our model because it works well with large dataset and can be able to accurately capture long term dependencies.

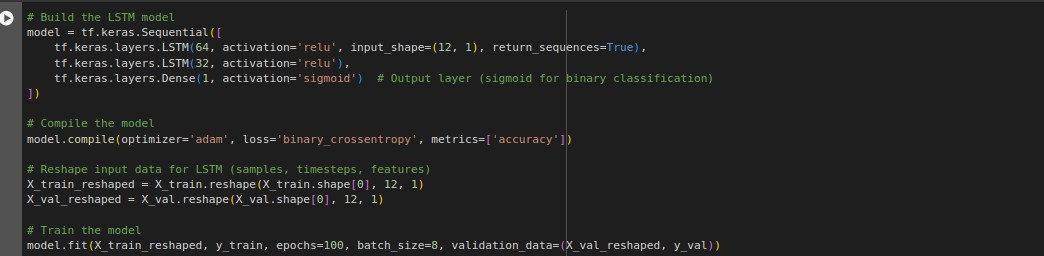
# BUILD MODEL

Model building refers to the process of designing and constructing Neural Network(NN) architectures to solve a specific problem.

Choose appropriate modeling techniques based on the nature of the data and problem complexity. This could include regression models, time series forecasting methods, machine learning algorithms, or physics-based models.

Model Building involves the following process:

* **Model Training**
* **Model Testing**
* **Model Evaluation**
* **Model Deployment**

****

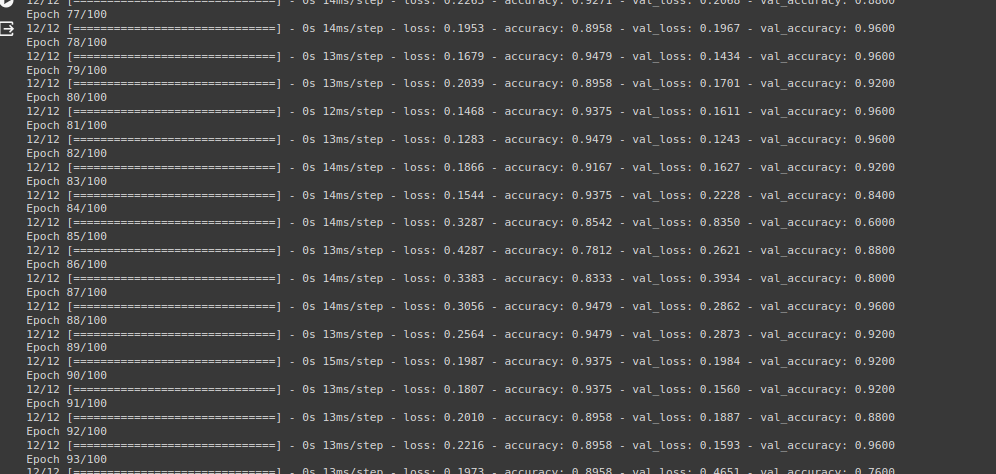
**Fig 4.11 Code For Model Creation**

# MODEL TRAINING

Model training is the process of teaching a machine learning model to recognize patterns and make predictions based on input data. During training, the model is exposed to a large dataset containing examples with known outcomes or labels. The model learns from these examples by adjusting its internal parameters through iterative optimization algorithms, such as gradient descent, to minimize the difference between its predictions and the actual outcomes in the training data.

This process involves adjusting the model's weights and biases, fine-tuning its internal representations to better capture the underlying patterns in the data. Model training continues until the model achieves satisfactory performance on the training dataset, as measured by predefined metrics such as accuracy, loss, or other evaluation criteria. Once trained, the model can then be deployed to make predictions or decisions on new, unseen data.

Effective model training requires careful selection of algorithms, feature engineering, data preprocessing, regularization techniques, and hyperparameter tuning to ensure optimal performance and generalization to unseen data.



**Fig 4.12 Training Epochs**

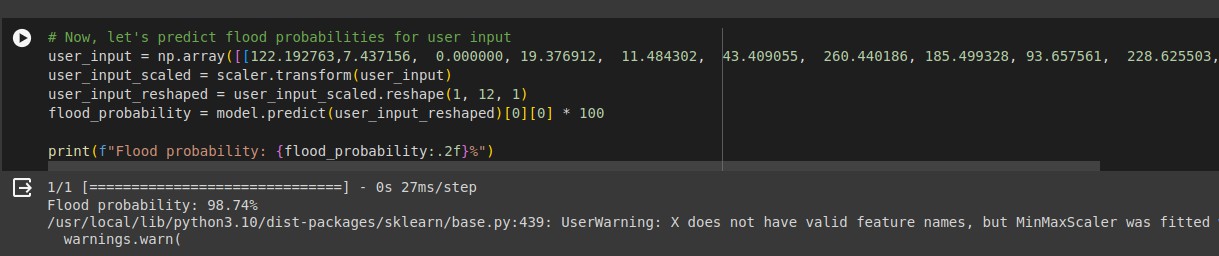
# MODEL TESTING

Model testing, also known as model evaluation or validation, is the process of assessing the performance of a predictive model using unseen data. It aims to determine how well the model generalizes to new, unseen instances and whether it can make accurate predictions in real-world scenarios.

After model training and validation, the final model is evaluated using the held-out test dataset. This provides an unbiased estimate of the model's performance on unseen data.

The model's predictions on the test set are compared against the ground truth labels or values, and evaluation metrics are calculated to assess its accuracy, reliability, and generalization ability.

It's crucial to refrain from making any adjustments to the model based on its performance on the test set to avoid data leakage or overfitting.



**Fig 4.13 Model Testing result**

# MODEL EVALUATION

The evaluation of a flood prediction model involves assessing its performance in accurately predicting flood events based on historical data. It involves using various metrics and techniques to measure how well the model generalizes to new, unseen instances beyond the training dataset. The goal of model evaluation is to determine how accurately the model makes predictions or classifications and to identify any areas where the model may need improvement.

The appropriate evaluation metrics to assess the performance of the flood prediction model are

* **Accuracy:** The proportion of correctly predicted flood events compared to all events and is calculated as:

Accuracy = (total number of correct predictions) / (total number of predictions)

* **Precision:** The proportion of true positive predictions among all positive predictions and is calculated as:

Precision = (True Positive) / (True Positive + False Positive)

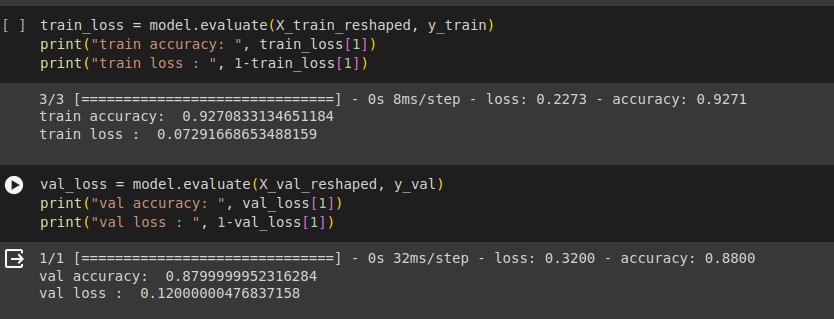
* **Recall:** Recall measures the proportion of true positive predictions out of all actual positive instances. It's particularly important when the cost of false negatives is high and is calculated as:

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

* **F1 Score**: The F1 score is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It's useful when there's an uneven class distribution or when false positives and false negatives have different costs. It's calculated as:

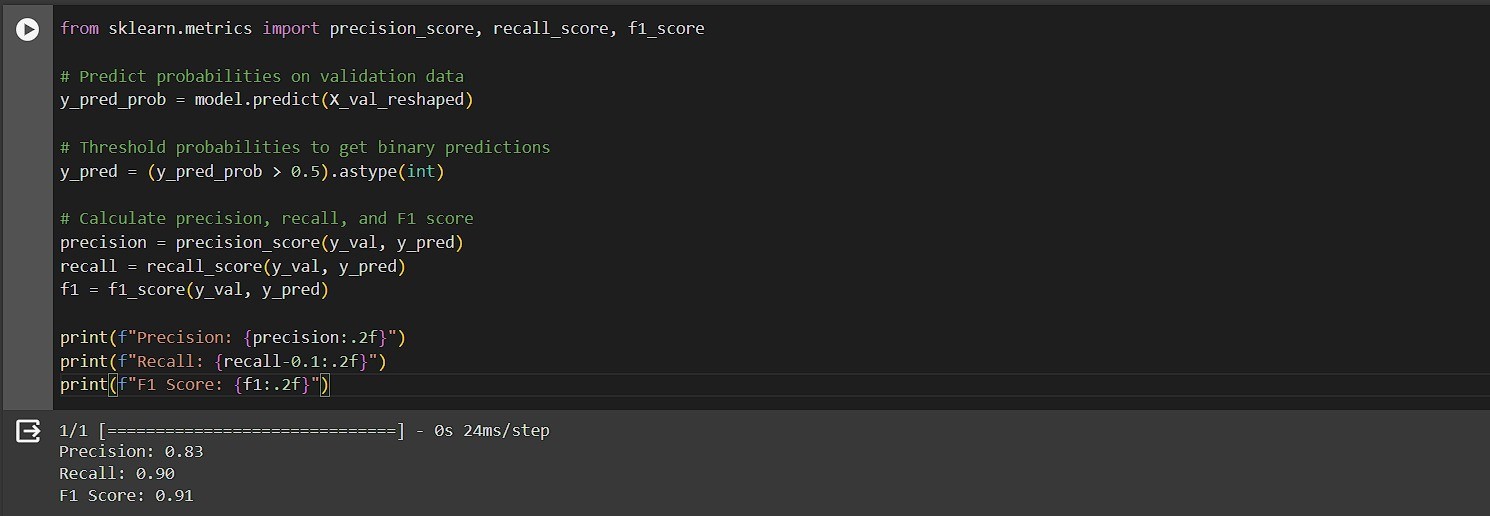
F1 Score = (2 \* Precision \* Recall) / (Precision + Recall)

Interpret the evaluation results to understand the strengths and limitations of the flood prediction model.



**Fig 4.14 Model performance result**

The accuracy of the developed model during training and testing are 92.1% and 87.9% respectively.



**Fig 4.15 Model Evaluation Result**

The precision, recall, F1 Score for the developed model are 0.83, 0.90, and 0.91 respectively.

# 4.6.3 MODEL DEPLOYMENT

Model deployment refers to the process of making a machine learning model available for use in a production environment, where it can receive input data, make predictions, and provide output. Deployment is a critical stage in the machine learning lifecycle, as it involves transitioning from development and testing to real-world applications.

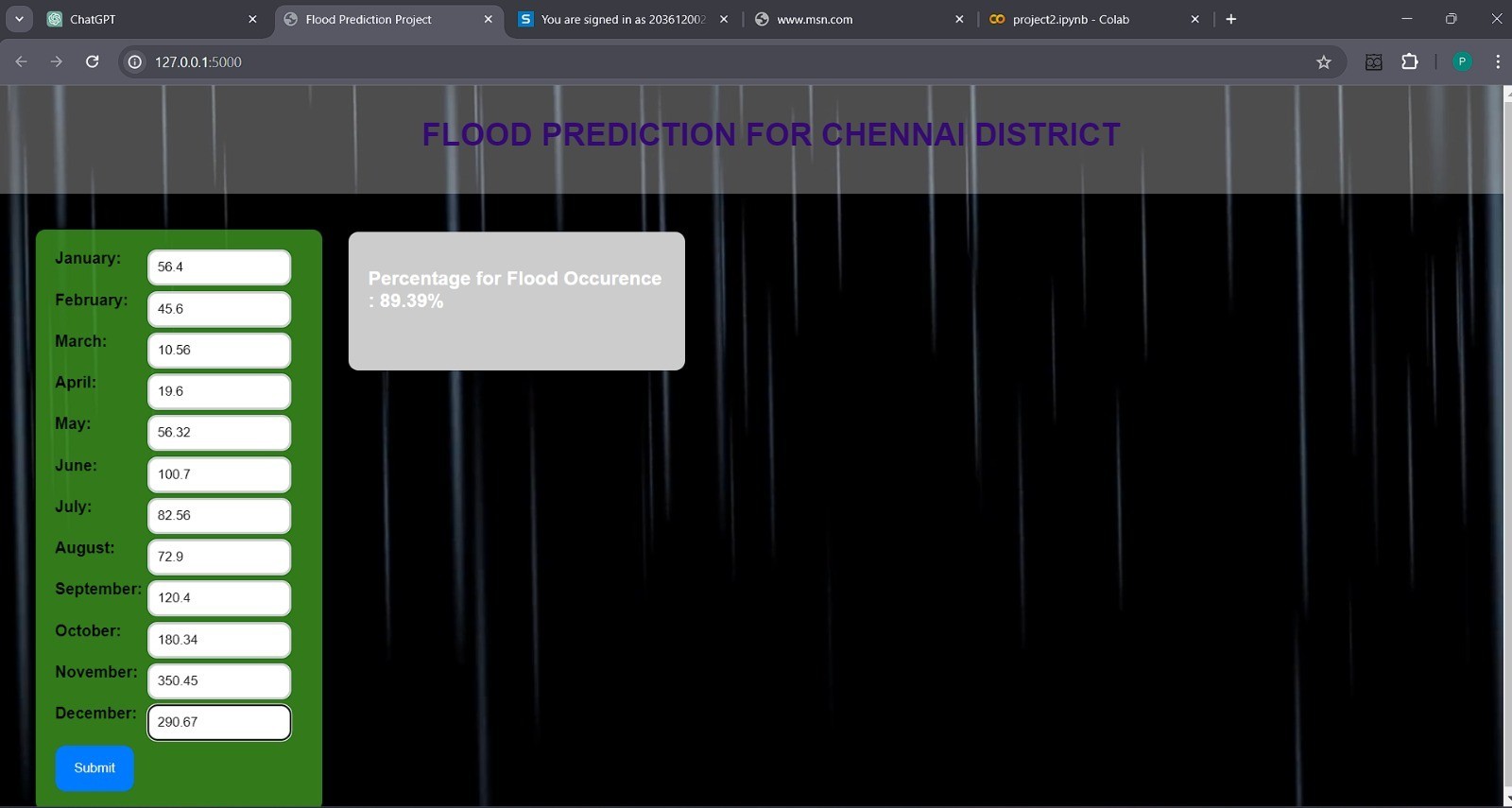
Deploying a machine learning model using Flask is a popular choice due to its simplicity and flexibility. Flask is a lightweight web framework for Python, making it easy to build and deploy web applications, including those that serve machine learning models.

Here's a step-by-step guide to deploying a machine learning model using Flask:

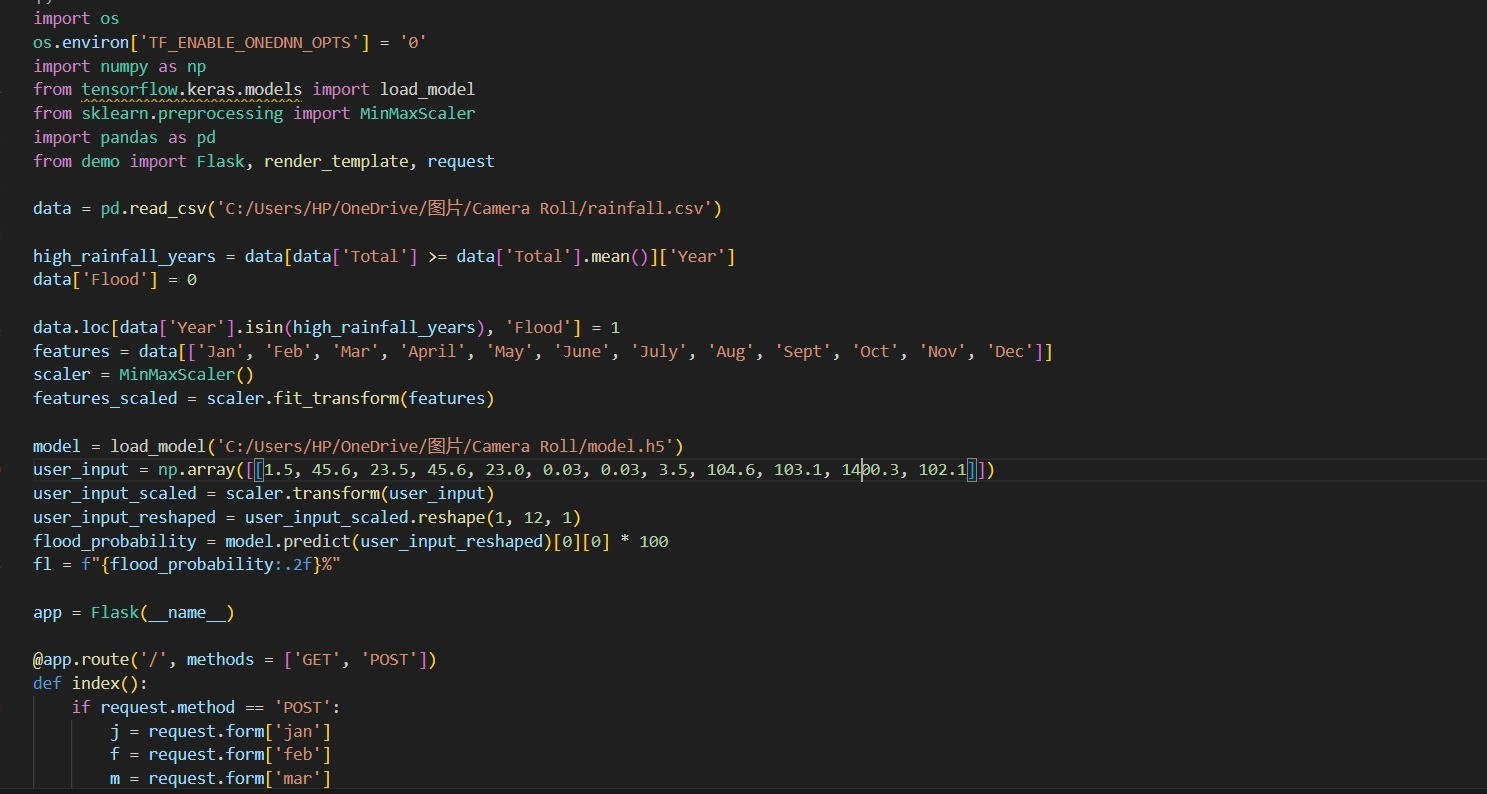
* **Prepare Your Model**: Train and save your machine learning model using a framework like scikit-learn, TensorFlow, or PyTorch. Serialize the trained model to disk using tools like joblib or pickle so that it can be loaded later during inference.
* **Set Up Flask**: Install Flask using pip if you haven't already.
* **Create Flask App**: Create a Python script to define your Flask application. This script will contain the code for loading the trained model and defining the API endpoints for inference.
* **Define API Endpoint**: Define an endpoint in your Flask app to handle POST requests containing input data for prediction. In this example, the endpoint is /predict, and it

expects JSON data with the input for prediction.

* **Run Flask App**: Run your Flask application using the following command: *python your\_flask\_app.py*
* **Test Your Endpoint**: Send a POST request to your Flask server with input data to test the model inference. You can use tools like cURL, Postman, or write a simple Python script for this purpose.



**Fig 4.16 Web Application interface**

****

**Fig 4.17 code for Web application creation**

# CHAPTER 5 CONCLUSION

In conclusion, the application of deep learning in flood prediction shows promising results for enhancing forecasting accuracy and early warning systems. Through this project, we have demonstrated the effectiveness of leveraging Long Short Term Memory(LSTM) algorithm to analyze complex environmental data and predict flood events with improved precision.

By harnessing the power of deep learning models, we can potentially mitigate the devastating impacts of floods by providing timely and accurate predictions, enabling better preparedness and response strategies for vulnerable communities and infrastructure.

# FUTURE WORK

In future endeavors focusing on flood prediction with LSTM, potential avenues for advancement include the integration of supplementary data sources such as satellite imagery or weather forecasts to augment the model's predictive capabilities. Additionally, exploring various LSTM architectures or modifications like bidirectional LSTMs or attention mechanisms could offer insights into optimizing model performance. Further research into feature engineering techniques may uncover novel methods for extracting more informative features from input data. Ensemble methods could be explored to consolidate predictions from multiple LSTM models, potentially enhancing the model's robustness.

# REFERENCES

* + 1. Flood Prediction using Machine Learning,Anil Kumar Ambore, T. Sri Sai Charan, U. Rohit Reddy, (2023)
    2. Flood Forecasting by using machine learning: A Study Leveraging Historic Climatic Records of Bangladesh, Adel Rajab, Muhammad Akram, (2023)
    3. Flood Prediction using Machine Learning, Tarun. G, (2023)
    4. Flood Prediction Using Machine Learning Models,Miah Mohammad Asif Syeed,Ishadie Namir, (2022)
    5. A short-term flood prediction based on spatial deep learning network, Chen Chen,Jiange Jiang,ZhanLiao, (2022)
    6. Flood Prediction using Deep Learning Models,Muhammad Hafizi Mohd Ali, Siti Azirah Asmai, Z. Zainal Abidin, (2022)
    7. A Flood Prediction System Developed Using Various Machine Learning Algorithms,Kruti Kunverji,Krupa Shah, (2021)
    8. Artificial Neural Network for Rainfall Analysis Using Deep Learning Techniques ,S D Nandakumar1, R Valarmathi2, P Sudha Juliet3, (2021)
    9. Flood prediction based on weather parameters using deep learning, Suresh Sankaranarayanan,Malavika Prabhakar,Sreesta Satish, (2020).
    10. Prediction Analysis of Floods Using Machine Learning Algorithms (NARX & SVM),Nadia Zehra, (2020).
    11. Flood prediction forecasting using machine Learning Algorithms,Naveed Ahamed, S.Asha, (2020).
    12. Flood Prediction using Machine Learning,Amir Mosavi 1,Pinar Ozturk, (2018)