Flood Prediction Using ML Models

Aditya Kumar Singh
IIITD
mail

Harsh Vishwakarma
IIITD
mail

Nutan Kumari
IIITD
mail

Pandillapelly Harshvardhini
IIITD
mail

Abstract

Flood prediction is essential for disaster management, helping minimize loss of life and property. This report presents an AI-based flood prediction system using machine learning (ML) models for regression analysis, focusing on flood risk levels. The dataset includes 1,048,576 entries with features such as Monsoon Intensity, Topography, Deforestation, Urbanization, Dams Quality, Infrastructure, and Climate Change.

The study uses diverse data, including seasonal variation and climate models, to enhance the accuracy of flood forecasts. By evaluating various ML models, the research identifies the most effective approach for predicting flood risk, aiding communities and disaster management agencies in making timely, informed decisions to mitigate flood impacts. GitHub- Project

1. Motivation

Floods are among the most destructive natural disasters, causing significant damage to property, loss of life, and environmental degradation. Inaccurate or delayed flood detection can lead to displacement and economic losses. Our project aims to develop an adaptive flood detection system using machine learning to improve prediction accuracy, reliability, and early warnings. Predicting flood risk based on environmental and meteorological features is both promising and impactful, motivating us to develop a model to enhance disaster management and response.

2. Problem Statement

Flood prediction is a complex challenge influenced by factors like rainfall, soil moisture, river flow, and climate conditions. Traditional hydrological models often struggle to capture the non-linear relationships between these variables and handle the large volumes of data from modern monitoring systems. Machine learning models, on the other hand, can analyze vast, multidimensional datasets and identify intricate patterns for more accurate predictions. However, these models face challenges like noisy data, feature selection, and tuning to avoid overfitting. This project aims to develop a machine learning-based flood prediction model that addresses these issues, providing a reliable tool for early warnings and disaster preparedness, ultimately reducing the impacts of floods on vulnerable communities.

3. Literature Survey

Traditional flood prediction methods have relied on statistical models and rule-based algorithms using data like rainfall, river levels, and weather patterns. While effective in many cases, these methods struggle with complex environmental factors and dynamic weather changes. To address these challenges, AI and machine learning models are increasingly being adopted for more accurate flood predictions. Our survey highlights the widespread use of machine learning, deep learning, and ensemble methods in improving prediction accuracy. Below, we discuss key studies from our literature review.

3.1. Flood Forecasting Using Dynamic Artificial Neural Networks

The reference to this research paper is provided at [**?**] in the reference section of this report.

In this study, dynamic Artificial Neural Networks (ANNs), including the Nonlinear Autoregressive Network with Exogenous Inputs (NARX) and Elman Neural Networks, were utilized to predict floodwater levels at the Yu-Cheng Pumping Station in Taipei, Taiwan. The study aimed to develop an accurate real-time forecasting model for urban flood control using multi-step-ahead predictions.

The research demonstrated that NARX networks outperformed Elman neural networks in real-time forecasting due to their recurrent connections from the output layer, which significantly enhanced prediction accuracy. The NARX model was tested in two scenarios: Scenario I used both rainfall and water level data as inputs, while Scenario II used only rainfall data. The results showed high coefficients of efficiency ranging from 0.9 to 0.7 (Scenario I) and 0.7 to 0.5 (Scenario II) for 10-60-minute-ahead forecasts.

This study highlights the value of NARX networks in accurately predicting water levels for flood control systems and suggests that such models can be highly beneficial for government authorities in urban flood management.

However, the study also highlighted that LSTMs require large datasets to train effectively and can be computationally expensive. Despite these challenges, the model achieved high prediction accuracy and successfully identified flooding events in advance, allowing for better preparedness.

3.2. Machine Learning Models for Flood Prediction

The reference to this research paper is provided at [**?**] in the reference section of this report. The field of flood prediction has evolved significantly with the integration of machine learning techniques, particularly in the use of supervised learning models such as Random Forests, Support Vector Machines (SVM), and Artificial Neural Networks (ANNs). Traditionally, flood prediction relied on hydrological models that often struggled with accuracy and generalization due to their reliance on physical parameters. However, the emergence of machine learning has provided a more robust alternative by leveraging large datasets to identify patterns in flood events. Among these methods, machine learning algorithms have shown considerable promise in predicting flood occurrences and assessing flood risk. Random Forests, for instance, have achieved impressive performance metrics, including Matthews Correlation Coefficients (MCC) and accuracy rates that exceed 90 percent. Furthermore, the incorporation of Geographic Information Systems (GIS) techniques alongside machine learning has enhanced spatial analysis capabilities, allowing for more effective identification of high-risk areas. This review highlights the transition from traditional hydrological models to data-driven machine learning approaches, underscoring the transformative potential of these methodologies in flood prediction and risk management.

4. Dataset

The dataset used in this project contains over 1.8 million records, each corresponding to a specific set of environmental and geographical conditions. The dataset's features span various metrics crucial to flood prediction, including meteorological variables, topographic indicators, and potential hydrological patterns.

4.1. Dataset structure

The dataset is organized in a tabular style, with columns denoting various features affecting flood probability and rows denoting individual observations. The columns in the dataset are as follows:

These columns represent the environmental factors that are considered potential predictors of floods. They include:

1. **Monsoon Intensity:** The level of rainfall during the monsoon season.
2. **Topography Drainage:** Efficiency of the terrain in draining excess water.
3. **River Management:** Measures taken to control river flow and prevent flooding.
4. **Deforestation:** The loss of trees that can lead to reduced soil absorption of water.
5. **Urbanization:** The expansion of cities, increasing impervious surfaces that prevent water absorption.
6. **Climate Change:** Long-term changes in weather patterns affecting rainfall and flood risks.
7. **Dams Quality:** The structural condition of dams and their ability to regulate water.
8. **Siltation:** The accumulation of sediments in rivers, reducing water flow capacity.
9. **Agricultural Practices:** Farming techniques that may affect water retention or runoff.
10. **Encroachments:** Human settlements or constructions on floodplains increasing flood risk.
11. **Ineffective Disaster Preparedness:** Inadequacy of systems designed to mitigate flood impacts.
12. **Drainage Systems:** The effectiveness of infrastructure to manage excess water.
13. **Coastal Vulnerability:** The susceptibility of coastal areas to flooding due to sea-level rise or storms.
14. **Landslides:** Earth movements triggered by heavy rain, which can worsen flood situations.
15. **Watersheds:** Areas where water collects and drains into rivers, affecting water flow.
16. **Deteriorating Infrastructure:** Aging or poorly maintained structures that may fail during floods.
17. **Population Score:** The population density, which can increase the impact of flooding in affected areas.
18. **Wetland Loss:** Reduction of wetlands that naturally absorb floodwaters.
19. **Inadequate Planning:** Poor urban or regional planning that fails to account for flood risks.
20. **Flood Probability:** The predicted likelihood of flood occurrence.

4.2. Data Analysis and Preprocessing

Prior to model training, the dataset was thoroughly analyzed using Python libraries such as pandas, numpy, and visualization tools like seaborn and matplotlib. The dataset underwent necessary preprocessing steps, handling missing values (if any), and feature scaling to prepare it for modeling.

At first, the dataset was imported from a csv file and converted to a dataframe. Then Null values, dimensions of the dataframe etc. was analyzed. Upon inspection, the dataframe consisted of (1863262 rows x 22 columns). 745305 null values were present in the dataset in the target column i.e. FloodProbability. These null values has been cleaned by removing them.

We conducted a comprehensive analysis of the distribution of each feature in the dataset. These graphs help in identifying patterns, and the overall spread of data, which is essential for understanding the underlying structure and ensuring effective model development.

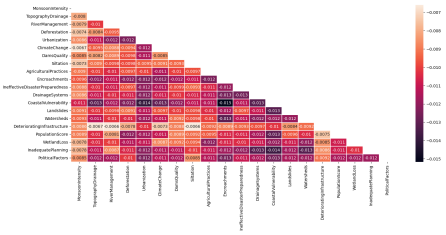

Figure 1. Correlation analysis using heatmap.

A Heatmap (Figure 1) was created to analyze feature correlations.Upon calculating correlation coefficients from the heat map we got the important features in decending order. IneffectiveDisasterPreparedness, Encroachments, and DrainageSystems showed perfect correlation, indicating redundancy. To reduce multicollinearity, IneffectiveDisaster-Preparedness and DrainageSystems were initially removed, retaining Encroachments.
But after training and hyperparameter tuning(In further steps),we observed that removing these features did not improve performance. After reintroducing the features, including IneffectiveDisasterPreparedness and DrainageSystems, the model's accuracy increased. This indicates that while the features are correlated, they provide additional predictive power, justifying their inclusion in the final model.

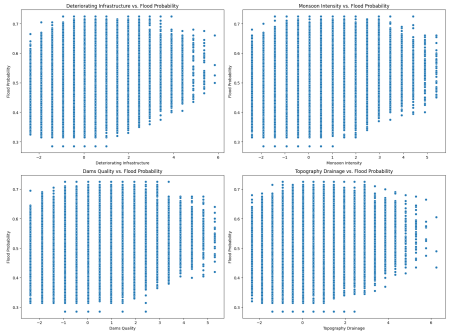

Figure 2. Scatter plots between important features.

The figure 2 shows scatter plots of some important features, each examining the relationship between flood probability and key factors. And Figure 3 shows the box plots for some of the features.

Insights gathered from the above plot:

1. Deteriorating Infrastructure: As infrastructure worsens, flood probability increases.
2. Monsoon Intensity: Higher monsoon intensity leads to a higher likelihood of flooding.
3. Topography Drainage: Poor drainage increases flood risk, but after a certain point, the probability stabilizes.
4. Dams Quality: Better-quality dams reduce flood risk, while lower-quality dams are associated with higher flood probability.

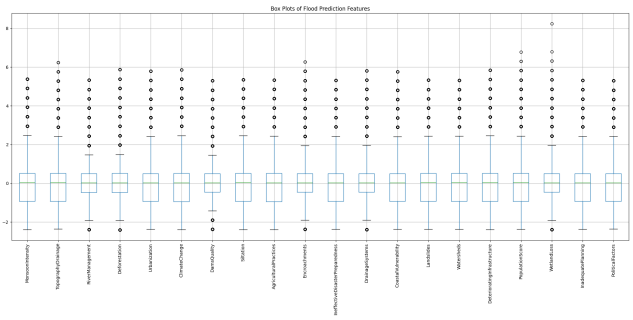

Figure 3. Boxplots of features.

5. Methodology and model details

We have removed some features as mentioned in the preprocessing section. This is a regression problem where we aim to predict the Flood Probability using various environmental factors.

5.1. Model selection

For our project where we are predicting flood, the following models were selected for training and testing:

1. Linear Regression
2. Support Vector Machine(SVM)
3. Decision Tree
4. Randon Forest
5. Multilayer Perceptron

5.2. Model training and testing

The dataset used for training contained several independent features relevant to flood prediction. The target variable, FloodProbability, and its binned counterpart, FloodProbabilityBin, were removed from the feature matrix to retain only the predictor features. We split the dataset into training and testing sets using an 80-20 ratio, ensuring the models were trained on 80 of the data and tested on the remaining 20 to assess generalization. Since the feature values ranged between -1 and +1, scaling was deemed unnecessary, except for SVR, which required scaled inputs for effective training.
Various models were trained and evaluated to predict FloodProbability, including Linear Regression, Decision Tree Regressor, Random Forest, Support Vector Regressor (SVR), and a Multilayer Perceptron (MLP). Linear Regression served as a baseline model due to its simplicity in predicting continuous variables. The Decision Tree Regressor captured non-linear relationships, while Random Forest, as an ensemble method, combined multiple decision trees to enhance accuracy and reduce overfitting. SVR aimed to find a hyperplane that best fit the data in a high-dimensional space, necessitating feature scaling before training. The Multilayer Perceptron (MLP), a fully connected feedforward neural network with input, hidden, and output layers, allowed for comparison between traditional models and a deep learning-based approach.
The models were evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R^2), and Root Mean Squared Error (RMSE). MSE measured the average squared difference between actual and predicted values, where a lower value indicated better performance. MAE, focusing on the magnitude of errors, provided insight without considering direction. R^2 reflected the proportion of variance explained by the model, with higher values indicating better fit. RMSE, as the square root of MSE, expressed errors in the same units as the target variable, making interpretation more intuitive.

Model	MSE	MAE	R2 Score	RMSE
Linear Regression	0.00067	0.0207905	0.740890	0.025951
Decision Tree	0.0025653	0.040291	0.013045	0.050649
Random Forest	0.0010128	0.02591	0.610338	0.03182
SVM	0.0009	0.0248	0.6470	0.0303
MLP	0.000660	0.020590	0.745892	0.025700

Table 1. Error Metrics on Test Set

Model	MSE	MAE	R2 Score	RMSE
Linear Regression	0.00040	0.01579	0.84487	0.02008
Decision Tree	0.001813	0.034189	0.302367	0.042583
Random Forest	0.0311	0.0255	0.6270	0.0311
SVM	0.0004	0.0164	0.8395	0.0204
MLP	0.00036	0.01492	0.861429	0.01897

Table 2. After Hyperparameter tuning and feature map

6. Results

6.1. Before Hyperparameter Tuning

Before tuning hyperparameters and adding new features, the models displayed varying levels of performance on the test set. Linear Regression performed well, explaining 74.1% of the variance, with an MSE of 0.000673, MAE of 0.02079, and RMSE of 0.02595, making it a strong baseline model. The Decision Tree Regressor, however, struggled, explaining only 1.3% of the variance with an MSE of 0.00257, MAE of 0.04029, and a high RMSE of 0.05065, indicating poor generalization and significant overfitting.

Random Forest Regressor achieved better results, explaining 61% of the variance with an MSE of 0.00101, MAE of 0.02592, and RMSE of 0.03183, but still lagged behind the baseline. The Support Vector Regressor (SVR) showed moderate performance, while the Multi-Layer Perceptron (MLP) Regressor emerged as the best model, explaining 74.6% of the variance, with an MSE of 0.000660, MAE of 0.02059, and RMSE of 0.02570.

Overall, the MLP provided the highest accuracy, while Linear Regression offered a simpler, computationally efficient alternative. Random Forest provided a balance between complexity and performance, but the Decision Tree was the weakest model due to high error rates and poor generalization.

6.2. After Hyperparameter Tuning

After incorporating two additional features and tuning hyperparameters (where applicable), all models were re-evaluated, yielding notable improvements in performance.

For Linear Regression, which has no tunable hyperparameters, the inclusion of new features significantly boosted performance. The model now explains 84.5% of the variance (up from 74.1%), with an MSE of 0.000403, MAE of 0.01579, and RMSE of 0.02008 on the test set, confirming its status as a reliable baseline.

The Decision Tree Regressor showed substantial improvement after hyperparameter tuning, with the best parameters being `max_depth=15`, `min_samples_leaf=6`, `min_samples_split=15`, and `max_features='sqrt'`. Despite improvements, it still underperformed compared to other models, explaining only 30.2% of the variance for training and 47% for training, with an MSE of 0.001813, MAE of 0.03419, and RMSE of 0.04258. However, overfitting was significantly reduced, addressing a critical issue observed earlier.

The Random Forest Regressor demonstrated marked improvements with tuned parameters: `n_estimators=200`, `max_depth=20`, `min_samples_split=5`, `min_samples_leaf=2`, and `max_features='sqrt'`. It now explains 62.7% of the variance, with an MSE of 0.00101, MAE of 0.0255, and RMSE of 0.0311. And train r2 score is 85%, which reduced overfitting. The model shows better generalization and effectively balances bias and variance.

The Support Vector Regressor (SVR), optimized with `kernel='linear'` and `C=1`, achieved strong performance, maintaining nearly identical results across training and testing sets. It explained 83.9% of the variance for both training and testing, with an MSE of 0.0004, MAE of 0.0164, and RMSE of 0.0204, indicating excellent generalization without overfitting.

Finally, the Multi-Layer Perceptron (MLP) Regressor, though not further tuned, benefitted from feature addition. It now explains 86% of the variance for both training and testing, with an MSE of 0.000360, MAE of 0.01492, and RMSE of 0.01898, making it the best-performing model. The MLP achieved superior accuracy without overfitting, maintaining its edge over other models.

6.3. GUI Implementation

The Flask application features a user-friendly GUI that enables users to input data on various environmental factors and predict flood likelihood using a pre-trained machine learning model. The features are, first, The GUI includes an HTML form where users can input values for 20 environmental factors as in data. Second, Upon form submission, the backend processes the inputs and utilizes a pre-trained model (loaded from a .pkl file) to generate a prediction, displaying results like "Flood Expected" or "No Flood" on the webpage. This intuitive interface simplifies user interaction with the model, allowing real-time predictions based on environmental data..

7. Conclusion

In this project, redundant features were initially removed to reduce multicollinearity, but reintroducing features like IneffectiveDisasterPreparedness and DrainageSystems improved model performance. Models including Linear Regression, SVM, Decision Trees, Random Forests, and MLP were evaluated. MLP performed best with an R^2 of 0.861, outperforming Linear Regression. Decision Trees and Random Forests showed overfitting, but after hyperparameter tuning, their performance improved, with Random Forest achieving an R^2 of 0.627 on the test set. Hyperparameter tuning and feature adjustments led to better generalization. MLP remains the top choice, with Linear Regression offering a strong, simpler alternative.

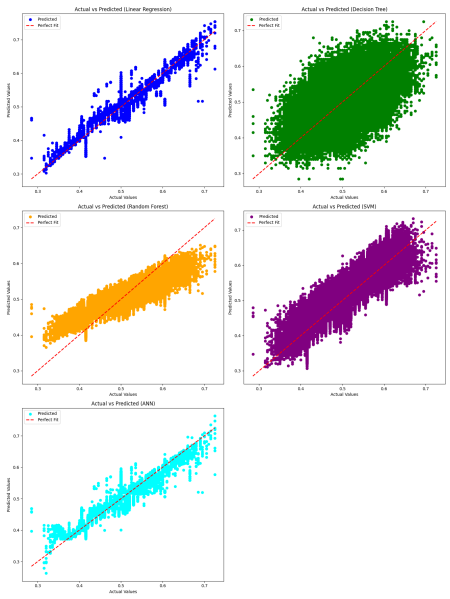

Figure 4. Scatter plots between important features.

7.1. Contribution

1. Dataset and Litreture survey: Nutan kumari, Harsh Vishwakarma
2. Dataset cleaning and preprocessing: Aditya Kumar Singh, Pandillapelly Harshvardhini
3. Methodology, Model training and testing, Hyperparameter Tuning: Pandillapelly Harshvardhini, Aditya Kumar Singh
4. GUI : Nutan Kumari
5. Report: Harsh Vishwakarma, Pandillapelly Harshvardhini, Nutan Kumari, Aditya Kumar Singh

References

[1] Numpy API's Documentation
[2] Pandas API's Documentation
[3] scikit-learn classes and various functions documentation
[4] Matplotlib API's Documentation
[5] Machine Learning Models for Flood Prediction
[6]Flood Forecasting Using Dynamic Artificial Neural Networks