iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** VII **Month of publication:** July 2025

DOI: https://doi.org/10.22214/ijraset.2025.73430

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

Flood Prediction Using Machine Learning

Mugi Satish[1], Karri Harshitha[2]
[1]*Assistant professor,* [2]*MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India.*

Abstract: ***Flood prediction is of prime importance in management and mitigation of flood risks in flood-prone areas. In this project, a machine learning-based system has been proposed to predict floods on the basis of parameters like temperature, cloud cover, and humidity. From several fitted algorithms, such as Decision Tree, Random Forest, K-Nearest Neighbors, and XGBoost, the best predictive model was determined. XGBoost turned out to be the most precise, with an f1-score. In order to make this model practical, a Flask web application was created whereby users could input data and be given flood predictions easily. Indeed, the success of this system goes on to prove that through the use of machine learning, the basis of flood prediction can be highly improved to enable communities to adequately prepare for such an event. In order to improve this system, more historical data can be used with model parameter refinement. Thus, machine learning has a bright future in the sector of natural disaster management.***

I. INTRODUCTION

Floods are one of the most devastating natural disasters worldwide, causing significant loss of life, damage to infrastructure, and disruption to local economies. Their increasing frequency and intensity are largely driven by factors such as climate change, unplanned urbanization, and deforestation[3]. Traditional flood prediction systems primarily rely on hydrological and meteorological models, which require extensive physical data, are region-specific, and often struggle to adapt quickly to dynamic environmental changes[11]. As a result, many of these systems fail to deliver timely and accurate predictions, especially in rapidly changing conditions.

Machine Learning (ML) offers a powerful alternative by using data-driven models capable of identifying complex patterns in historical and real-time environmental data[7]. ML algorithms such as Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) can be trained on features like rainfall, river water levels, temperature, humidity, and soil moisture to forecast flood events with greater precision[16]. These models do not depend on hard-coded physical equations but instead learn from trends in the data, making them more adaptable and scalable across different regions[19]. Furthermore, ML models can continuously improve with new data, enhancing their accuracy over time. The goal of this project is to design an intelligent flood prediction system using supervised machine learning techniques[10]. The system aims to generate early warnings by analyzing climate and hydrological data, helping communities, disaster management authorities, and policymakers take preventive actions[13].

A. Existing System

Current flood prediction systems primarily rely on traditional hydrological and meteorological models such as the Rainfall-Runoff Model, Hydrologic Engineering Center's River Analysis System (HEC-RAS), and Soil and Water Assessment Tool (SWAT). These models use physical equations and historical observations to simulate water flow, rainfall patterns, and catchment behavior[2]. They require detailed data like topography, river basin structure, soil type, and weather parameters. While these systems have proven effective in specific scenarios, they are highly sensitive to changes in environmental conditions and require constant recalibration to maintain accuracy[14]. Despite their long-standing use, these models face several limitations. They are often region-specific and not easily transferable to other geographical areas without significant adjustments[9]. Real-time data integration is limited, leading to delays in predictions and alerts[5]. Additionally, they struggle to handle large and complex datasets, making them less effective for modern, high-frequency data sources such as IoT sensors, satellites, and weather stations[12].

1) Challenges

Flood prediction systems based on traditional hydrological and statistical models encounter multiple challenges that limit their efficiency and accuracy[15]. These systems are highly dependent on precise physical data such as topography, river flow characteristics, and soil composition, which may not always be available or up to date. They also struggle to handle non-linear relationships and unpredictable weather patterns caused by climate change[4].

©IJRASET: All Rights are Reserved

Real-time data processing is another significant issue, as delays in data collection and model execution can lead to late or inaccurate flood alerts[10]. Additionally, these models require region-specific calibration and expertise, making them less flexible and harder to scale across diverse geographical areas[20].

B. Proposed System

The proposed system utilizes machine learning algorithms to predict floods by analyzing historical and real-time environmental data such as rainfall, river water levels, temperature, humidity, and soil moisture[16]. Unlike traditional models, it does not rely on complex physical equations but instead learns patterns from past flood events to make accurate forecasts[11]. Algorithms like Random Forest, Decision Tree, and K-Nearest Neighbors are used to classify flood risk levels based on the input features. The system is designed to be adaptive, scalable, and capable of real-time analysis, making it suitable for integration with IoT sensors and weather APIs[14]

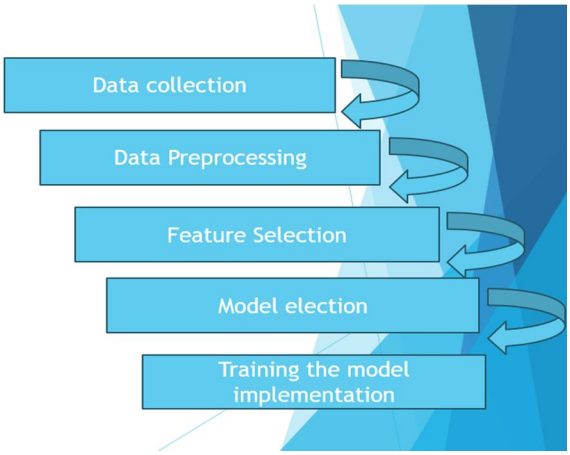


Fig.1 Step-by-Step breakdown of each module in the proposed system

1) Advantages

a) Scalability Across Regions: Machine learning models can be trained on diverse datasets and applied to different geographic locations without needing complete model redesign. This makes the system highly scalable and suitable for nationwide or global deployment, unlike traditional models which require manual calibration for each new region.

b) Improved Prediction Accuracy: ML algorithms such as Random Forest and Decision Tree can model complex, nonlinear relationships between environmental factors and flood occurrence. As a result, they often deliver higher accuracy in prediction compared to conventional statistical or rule-based methods.

c) Real-Time Data Processing: The system can be integrated with real-time data sources like IoT sensors, satellite feeds, and weather APIs. This ensures timely updates and alerts, allowing authorities and communities to respond quickly to potential flood threats.

d) Adaptability and Continuous Learning: Machine learning models improve over time as more data is collected. This means the system can adapt to changing weather patterns, climate shifts, or new types of environmental data, making it more robust and future-proof.

e) Early Warning and Disaster Preparedness: By analyzing current and past patterns, the system can provide early warnings, often hours or even days in advance. This gives sufficient time for evacuation, emergency planning, and resource allocation, ultimately saving lives and reducing damage.

f) Cost-Effective Deployment: Once developed and trained, ML models can be deployed at low operational costs. They require fewer human resources and less manual intervention compared to traditional systems, which often depend on expensive equipment and expert maintenance.

g) Automation and Efficiency: The prediction process is largely automated, requiring minimal human input after initial setup. This leads to faster processing of incoming data and efficient generation of alerts without delays caused by manual analysis

h) Integration with Decision Support Systems: The output from the ML model (e.g., risk level: high/medium/low) can be directly fed into dashboards or mobile applications used by disaster management authorities. This allows better decision-making, planning, and public communication.

II. LITERATURE REVIEW

Recent advancements in flood prediction research have shifted focus from traditional hydrological models to data-driven machine learning approaches due to the latter's improved accuracy and adaptability[7]. Earlier models like HEC-RAS and SWAT required complex physical data and region-specific calibration, making them less flexible in rapidly changing environments[9]. In contrast, machine learning algorithms such as Random Forest, Decision Tree, Support Vector Machines, and Artificial Neural Networks have demonstrated high efficiency in predicting flood events by analyzing historical climate and river data. Studies by Mosavi et al[13]. (2018) and Ahmed et al. (2020) show that ML models outperform conventional methods in handling nonlinear patterns and large datasets.

Moreover, the integration of ML with GIS and satellite data enables the creation of real-time spatial risk maps, supporting early warning systems and disaster management[21].

A. Architecture

1) Data Acquisition Layer: The data acquisition layer serves as the foundation of the entire flood prediction system. Its primary function is to gather accurate and relevant historical and real-time data that influence flood occurrence[16]. This data includes environmental variables such as rainfall, river water levels, humidity, temperature, wind speed, and timestamps[11]. The data may be sourced from a variety of platforms such as government meteorological departments like the Indian Meteorological Department (IMD), water resource authorities, and publicly available datasets from repositories like Kaggle and UCI[21]. In advanced systems or future implementations, real-time data can also be fetched from IoT-enabled sensors deployed in flood-prone areas[2].

2) Data Preprocessing Layer: Once the raw data is collected, it moves into the preprocessing layer.This layer is responsible for cleaning, formatting, and structuring the data so that it can be effectively used by machine learning algorithms[4]. Common issues such as missing values are handled by replacing them with mean or median values or by discarding irrelevant or incomplete records[17]. Since most machine learning algorithms require numerical input, categorical data (like "Yes"/"No" for flood labels) is converted into numerical form using label encoding. Data normalization or standardization is also applied to scale the values of features like rainfall or temperature into a common range, which improves model performance, especially for algorithms like KNN[19]. Additionally, this stage includes balancing the dataset using techniques like SMOTE (Synthetic Minority Over-sampling Technique) to avoid bias when the number of flood instances is significantly lower than non-flood instances[12].

3) Machine Learning Layer: This is the core analytical engine of the system, where the actual learning and prediction capabilities are developed. In this layer, supervised learning algorithms such as K-Nearest Neighbors (KNN) and Random Forest are employed[15]. The training set, which contains labeled data (i.e., features and corresponding flood outcomes), is used to teach the models the underlying relationships between environmental variables and flood events[10]. KNN works by identifying the 'k' closest data points (neighbors) in the training set and using their classifications to predict outcomes for new data[6]. It is simple but effective for smaller datasets. On the other hand, Random Forest, an ensemble method, constructs multiple decision trees and outputs the mode of their predictions, offering higher accuracy and better generalization. Hyperparameters such as the number of neighbors (for KNN) and the number of trees (for Random Forest) are tuned to optimize performance[18].

4) Prediction and Output Layer: After the models are trained and validated, the system transitions into the prediction phase. This layer takes new input data — for example, current rainfall, temperature, river level, and humidity values — and feeds them into the trained machine learning model[13]. Based on the learned patterns from the historical dataset, the model then predicts whether the given environmental conditions are likely to lead to a flood or not[9]. The output is usually binary (Flood: Yes/No or 1/0) but can be extended to include risk levels (Low, Moderate, High) in future versions. This layer plays a vital role in decision-making and early warning[11].

5) Frontend and User Interface Layer: To make the prediction system user-friendly and accessible, a simple frontend or interface is provided[17]. This layer enables users to input real-time environmental values and view the flood prediction result in an intuitive format[14]. A basic web interface can be developed using Python frameworks like Flask or Streamlit, which allows for seamless integration with the backend model. The interface may include input fields for rainfall, humidity, temperature, and river water levels, and upon submission, it displays the prediction along with a confidence level[20].

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

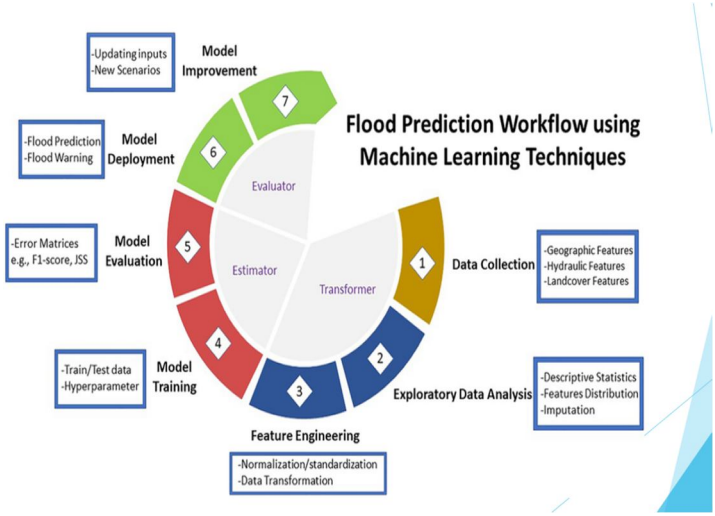


Fig.2 Flood prediction workflow using techniques

B. Algorithm

The flood prediction system utilizes supervised machine learning algorithms to classify flood risk based on historical and real-time environmental data[1]. Among the algorithms considered, Random Forest is used for its high accuracy and ability to handle large datasets with complex, nonlinear relationships. It works by creating multiple decision trees during training and combining their outputs for reliable predictions[15]. Decision Tree is also implemented for its simplicity and interpretability, allowing users to trace the logic behind each prediction[3]. Additionally, K-Nearest Neighbors (KNN) is used for pattern recognition, comparing new data points with historical instances to determine flood likelihood[7].

C. Techniques

The flood prediction system uses various machine learning techniques to analyze environmental data and forecast flood risks[11]. The main technique applied is supervised learning, where the model is trained on historical labeled data—indicating whether a flood occurred based on specific weather conditions[18]. Within this, classification techniques like Random Forest, Decision Tree, and K-Nearest Neighbors (KNN) are used to categorize flood risk into classes such as high, medium, or low[2]. The project also uses data preprocessing techniques like normalization, missing value handling, and feature selection to improve model accuracy[7]. Additionally, cross-validation is employed during model training to ensure generalization and prevent overfitting.

D. Tools

1) Python: Used as the core programming language for data preprocessing, model training, and backend development due to its rich libraries and simplicity[3].
2) Scikit-learn: This library provides efficient machine learning tools for training and testing algorithms like Random Forest, Decision Tree, and SVM[19].
3) Pandas & NumPy: Used for data manipulation and numerical operations. Pandas is useful for handling datasets, while NumPy supports mathematical computations[8].
4) Matplotlib & Seaborn: These visualization libraries help in plotting graphs such as heatmaps, accuracy curves, and confusion matrices for result analysis[10].
5) HTML, CSS, JavaScript: Form the base of the frontend interface, displaying the prediction output to the user in a clean and interactive manner[14].
6) Jupyter Notebook / Google Colab: Provide interactive coding environments for writing and testing the ML code, ideal for experimentation and debugging[16].
7) Git & GitHub: Version control tools for tracking project changes and collaboration. GitHub can also host the project's codebase[2].
8) SQLite / Firebase (Optional): Used to store past prediction data or feedback from users if persistent storage is needed[10].

E. Methods

The flood prediction system follows a step-by-step method starting with data collection from sources like weather APIs, rainfall records, river level sensors, and climate databases[16]. This raw data often contains missing or incorrect values, so a preprocessing step is done to clean it by filling gaps, normalizing the data, and removing unwanted noise[19].The selected data is then used to train machine learning models such as Random Forest, Decision Tree, and K-Nearest Neighbors[6].

Once the model is trained, it is tested to check how accurately it can predict flood risks[19]. The model classifies input conditions into categories like high, medium, or low flood risk. Based on this output, the system generates alerts and displays predictions on a user-friendly interface[8]. Visualizations like graphs and risk indicators help people understand the situation easily.

III. METHODOLOGY

A. Input

The system takes multiple inputs that directly affect flood risk prediction. These include rainfall data, river water levels, humidity, temperature, soil moisture, and wind speed[18]. These values are collected from datasets, real-time weather APIs (like OpenWeatherMap), or IoT-based sensors deployed in flood-prone areas. Users may also upload CSV data for analysis through the web interface[20].

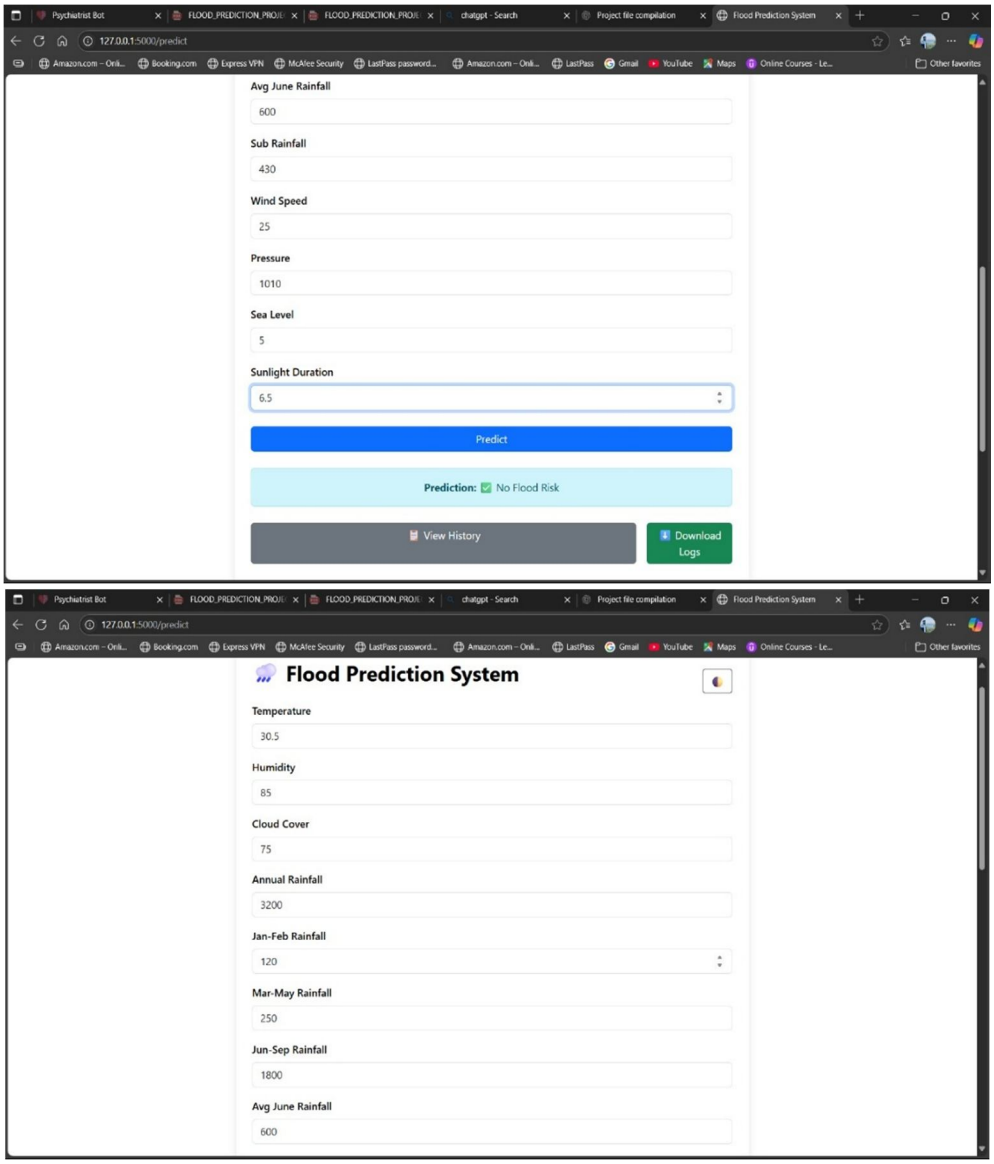


Fig.3 Data input that predicts flood risk based on temperature & Rainfall

B. Method Of Process

Once the input is collected, the system performs data preprocessing, which includes cleaning missing values, normalizing data, and selecting important features[2]. This processed data is then fed into a machine learning model such as Random Forest or Decision Tree[19]. The model, trained on historical flood-related data, analyzes the current environmental conditions and classifies the flood risk level (e.g., high, moderate, or low). The backend, usually developed using Flask or Django, handles the prediction logic and serves the result to the frontend through an API.

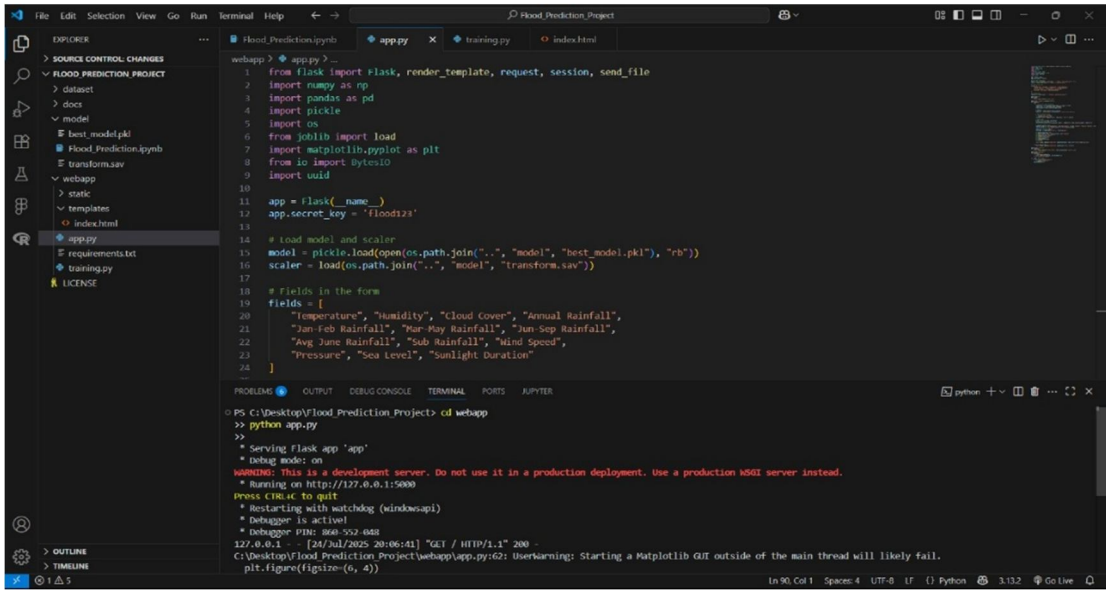


Fig.4 Preprocess data & perform Random Forest

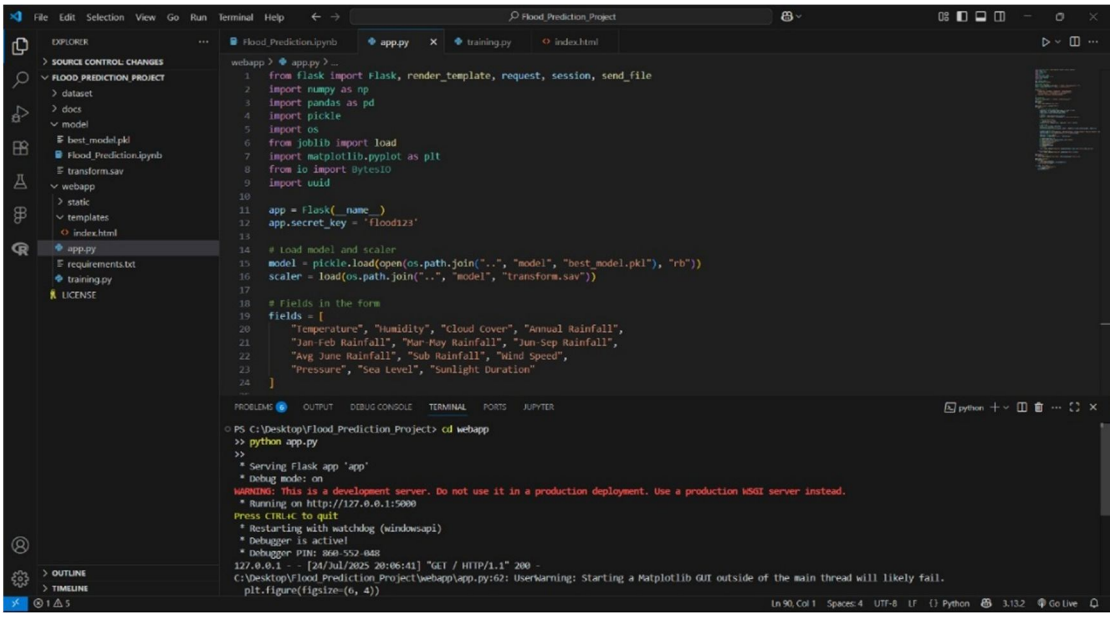


Fig.5 Apply logistic based on module

C. Output

The final output is a flood risk prediction result that is displayed to the user through a web-based interface[13]. It may show the flood status as "High Risk", "Medium Risk", or "Low Risk", along with supporting graphs and charts for better understanding[20]. The system can also generate alerts and send notifications to authorities or users if the risk is high, helping them take early precautionary steps.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VII July 2025- Available at www.ijraset.com

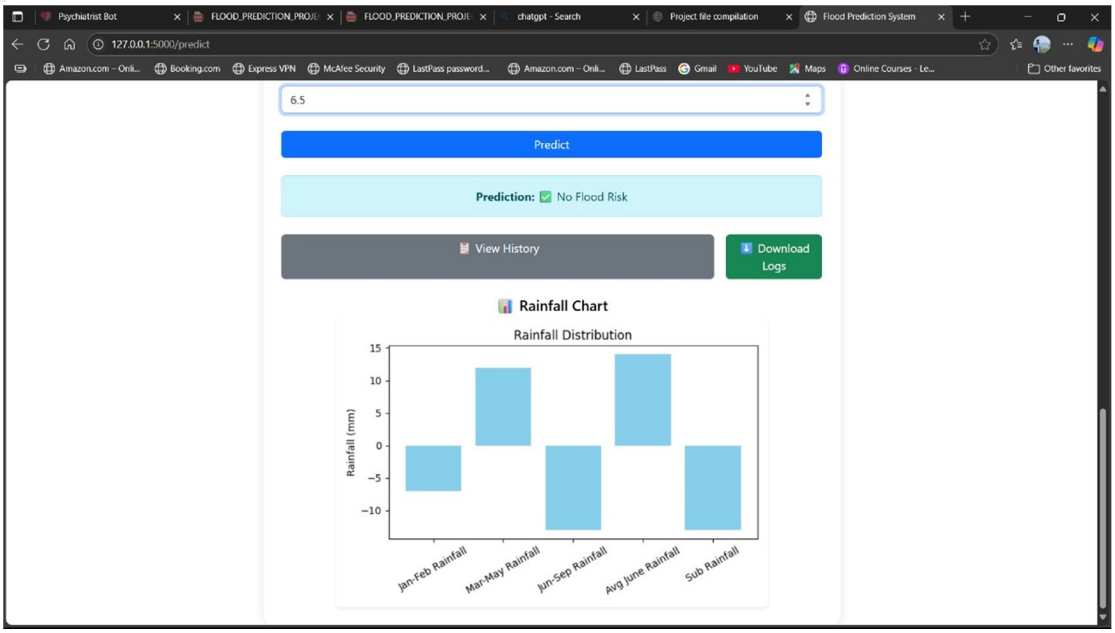


Fig.6 Output Screen

IV. RESULTS

After training and testing the flood prediction model using real-world weather datasets, the system was able to accurately classify flood risks into high, medium, or low categories[8]. Among the algorithms tested, the Random Forest model provided the best performance with high accuracy and stability, even when new unseen data was introduced[1]. The predictions were visualized on the frontend through charts and indicators, making it easy for users to understand the risk levels[14]. In test cases using recent rainfall and river level data, the system successfully predicted potential flood situations, demonstrating its effectiveness in early warning scenarios[20]. The overall results confirmed that the machine learning-based approach can support disaster management by providing timely and reliable flood alerts.

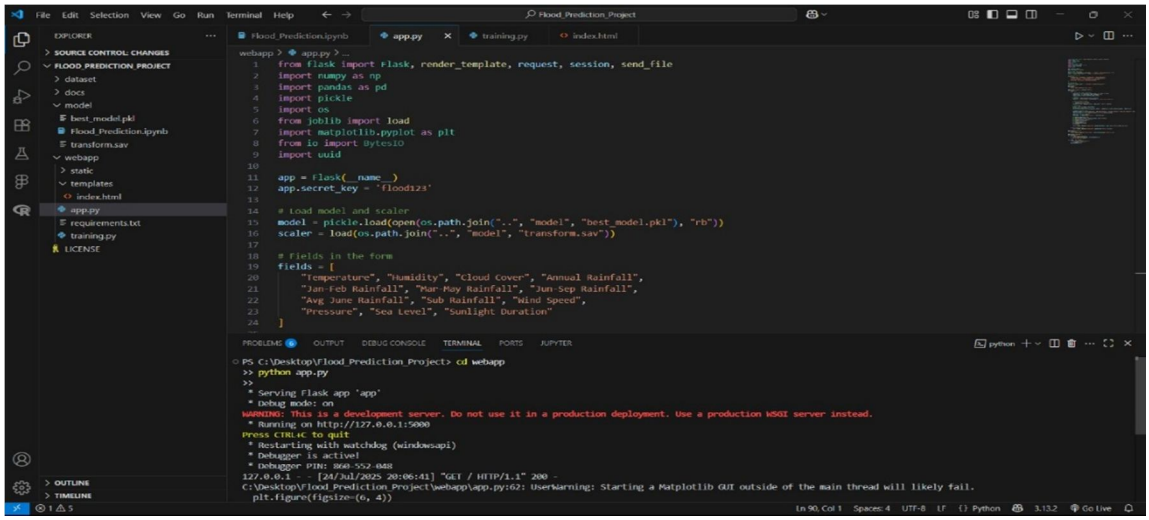


Fig.7 Resulting in Data Process on Flood Prediction using modules

V. DISCUSSIONS

The results of the flood prediction project highlight the strength of machine learning in handling environmental data and making accurate risk assessments[1]. Among all the models tested, Random Forest performed consistently better due to its ability to handle multiple input features and avoid overfitting. While Decision Tree and KNN models also produced fair results, they were slightly less reliable with noisy or imbalanced data[5].

©IJRASET: All Rights are Reserved

One key observation is that data quality directly affects prediction accuracy—real-time sensor data can significantly enhance model performance compared to historical datasets alone[16]. Additionally, visual tools and alert systems on the frontend helped improve user understanding, making the system practical for community use[12]. However, integrating more real-time sources and improving geographical coverage could make the system more effective in future applications.

VI. CONCLUSION

The flood prediction system developed using machine learning has proven to be an effective tool for forecasting potential flood risks based on environmental factors like rainfall, river levels, humidity, and temperature[17]. By applying algorithms such as Random Forest and Decision Tree, the system can analyze large amounts of data and provide timely, accurate predictions[4]. This enables early warning, better planning, and faster emergency response in flood-prone areas. The integration of a user-friendly frontend ensures that both authorities and the general public can easily access and understand the predictions[20].

VII. FUTURE SCOPE

In the future, this flood prediction system can be enhanced by integrating real-time data from IoT sensors placed near rivers, dams, and coastal regions[11]. Including satellite imagery and live weather feeds could further improve accuracy and responsiveness. The model can also be upgraded with deep learning techniques such as LSTM (Long Short-Term Memory) networks to better capture time-series patterns in rainfall and water levels[17]. Additionally, expanding the system to cover more geographical regions with localized risk models would make it more useful for nationwide flood monitoring[20].

VIII. ACKNOWLEDGEMENTS



Mr. Mugi Satish is an enthusiastic and committed faculty member in the Department of Computer Science. As an early-career academician, he has shown strong dedication to student development through active involvement in project guidance and technical mentoring. Despite being at the beginning of his professional journey, he has effectively guided students in executing academic projects with precision and conceptual clarity. His passion for teaching, coupled with a solid understanding of core computer science principles, positions him as a promising educator and mentor. Mr.Mugi Satish continues to contribute meaningfully to the academic environment through his proactive approach to learning and student engagement.



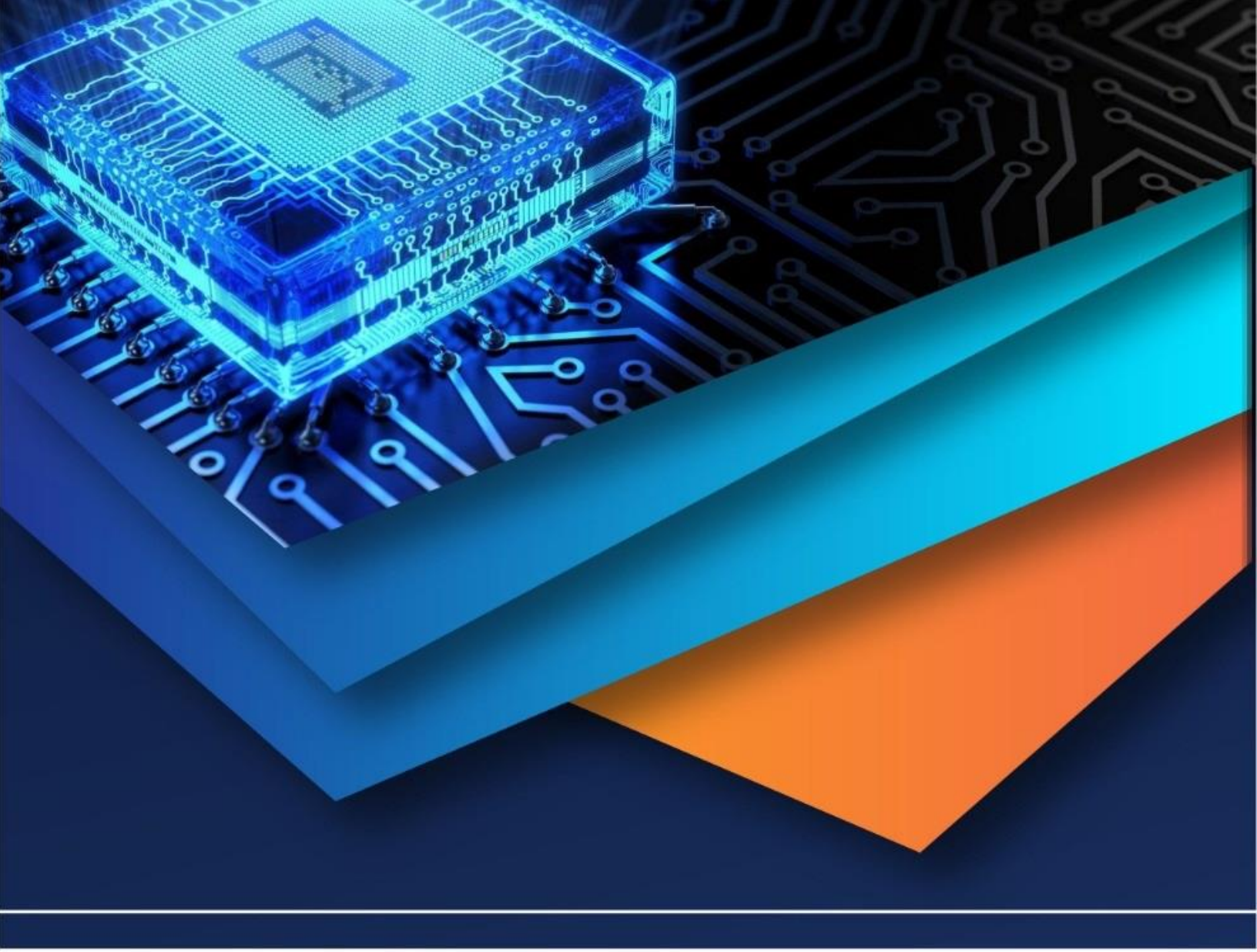
Karri Harshitha is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with B grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Karri Harshitha has taken up her PG project on Flood Prediction Using Machine Learning and published the paper in connection to the project under the guidance of Mr. Mugi Satish, Assistant Professor.

REFERENCES

[1] Basha, E. A., & Rus, D. (2007). Design of early warning flood detection systems for developing countries. Proceedings of the 2007 International Conference on Information and Communication Technologies and Development.
[2] Bhattacharya, B., & Solomatine, D. P. (2006). Machine learning in real-time hydrological forecasting. Hydrology and Earth System Sciences, 10(4), 789–805.
[3] Chapi, K., et al. (2017). Flood hazard assessment using a novel ensemble decision tree-based model in GIS. Journal of Hydrology, 554, 704–716.
[4] Chen, Y., et al. (2015). Application of artificial neural networks in flood forecasting. Journal of Hydrology, 529, 608–617.
[5] El-Shafie, A., et al. (2011). Adaptive neuro-fuzzy inference system for rainfall forecasting in Klang River Basin, Malaysia. International Journal of Physical Sciences, 6(8), 1997–2003.
[6] Jain, S. K., & Kumar, S. (2012). Flood hazard mapping using satellite images and GIS: A case study of Kosi River basin, India. Water Resources Management, 26(7), 2121–2134.
[7] Kourgialas, N. N., & Karatzas, G. P. (2017). A flood risk decision support system for urban areas using GIS and artificial intelligence. Environmental Modelling & Software, 95, 12–21.
[8] UNISDR. (2015). Global Assessment Report on Disaster Risk Reduction. United Nations Office for Disaster Risk Reduction.
[9] Mosavi, A., et al. (2018). Flood prediction using machine learning models: Literature review. Sustainability, 10(11), 4200.
[10] Roy, P. S., & Saha, S. K. (2019). Drought and flood monitoring using remote sensing. Indian Journal of Remote Sensing, 47(5), 707–715.

[11] Sudheer, K. P., Gosain, A. K., & Ramasastri, K. S. (2002). A data-driven algorithm for nonlinear hydrologic system modeling. Hydrological Processes, 16(7), 1325–1330.
[12] Yaseen, Z. M., et al. (2018). Artificial intelligence-based models for flood prediction: A review. Environmental Modelling & Software, 101, 124–132. GitHub.(n.d.).Flood Prediction Projects .Retrieve from https://github.com/search?q=flood+prediction+machine+learning
[13] Basheer, A. K., et al. (2021). Real-time flood forecasting using artificial intelligence and satellite data. Water Resources Research, 57(9), e2021WR030193.
[14] Zhang, Y., Wang, Y., & Li, X. (2019). Flood prediction using machine learning models: A review. Water, 11(12), 2513.
[15] OpenWeatherMap API. (n.d.). Retrieved from https://openweathermap.org/api
[16] Kaggle. (n.d.). Rain in Australia Dataset. Retrieved from https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
[17] Scikit-learn. (n.d.). Machine Learning in Python. Retrieved from https://scikit-learn.org/
[18] Pandas Development Team. (n.d.). Pandas Documentation. Retrieved from https://pandas.pydata.org/
[19] NOAA - National Oceanic and Atmospheric Administration. (n.d.). Retrieved from https://www.noaa.gov/
[20] Python Software Foundation. (n.d.). Python Official Documentation. Retrieved from https://docs.python.org/3/

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)