

Explainable AI for Rainfall and Flood Prediction

Riya Jadhav, Wynona Toby, Sojith Sunny, Prisha Bhosale, Dr. Lata Ragma

Department of Computer Engineering

Fr. Conceicao Rodrigues Institute of Technology

Email: 1022148@comp.fcrit.ac.in, 1022265@comp.fcrit.ac.in, 1022259@comp.fcrit.ac.in,

1022118@comp.fcrit.ac.in, lata.ragma@fcrit.ac.in

Abstract—Accurate prediction of rainfall and flood events is essential for effective disaster preparedness and risk mitigation. Machine learning techniques have remarkably improved forecasting accuracy by processing complex environmental data and identifying subtle patterns that traditional methods might overlook. However, the widespread adoption of these advanced models is often hindered by their lack of transparency, limiting trust among meteorologists and decision-makers. This research systematically evaluates a range of predictive models, including both conventional and state-of-the-art machine learning approaches, alongside leading explainability techniques. By doing so, the study addresses the critical challenge of balancing predictive accuracy with interpretability, ensuring that meteorological forecasts are not only precise but also understandable and actionable. The integration of explainable AI methods enhance model transparency and foster greater confidence in automated predictions, ultimately supporting more informed decision-making in weather-related risk management. This work aims to contribute to the development of trustworthy AI systems that empower communities to better prepare for and respond to extreme weather events.

Index Terms—Rainfall prediction, Flood prediction, Explainability, Transparent AI

I. INTRODUCTION

Predicting rainfall and flood events has long been a critical area of meteorology and environmental science research. With increasing climate variability and extreme weather conditions, effective prediction systems have become indispensable to mitigate risks to infrastructure, agriculture, and human safety. Traditional hydrological models have played an important role in forecasting, but often face limitations when handling the dynamic and non-linear behavior of environmental factors. This challenge has led to adopting data-driven approaches such as machine learning (ML), which can effectively identify complex patterns within weather data. ML models have shown notable improvements in prediction accuracy. However, these models often lack transparency, making it difficult for decision-makers to understand why specific predictions are made. This opacity limits the trust and adoption of artificial intelligence (AI) systems, particularly in critical domains such as disaster management. Consequently, the need for Explainable AI (XAI) techniques has grown, allowing stakeholders to interpret the model's behavior and understand the contributing factors behind the predictions. Previous research has explored various ML models to predict rainfall and floods, including traditional and advanced

algorithms. Although some models excel in capturing temporal patterns in meteorological data, others provide stable performance in various environmental conditions. The choice of model architecture, data attributes, and evaluation methods significantly influences the prediction results. Explainability techniques further complement these models by highlighting key variables responsible for prediction trends, ensuring improved transparency and accountability. This study investigates multiple ML models and interpretability methods to identify an optimal combination that balances accuracy and explainability. By analyzing the model's performance across diverse environmental data and evaluating the efficiency of different XAI techniques, this research aims to address the trade-off between prediction precision and interpretability. The findings contribute to improving the reliability of AI-driven meteorological systems, enabling more informed decision-making processes in managing flood and rainfall risk.

II. LITERATURE SURVEY

A critical review of recent literature reveals a diverse range of ML and AI techniques applied to rainfall and flood prediction. Table I summarizes representative studies, highlighting their methodologies, key findings, and notable limitations.

The study in [1] improves flood prediction using a deep neural network and machine learning models (SVM, KNN, Naïve Bayes) based on weather data. It generates real-time alerts for evacuation and leverages historical data (1990-2002), with potential enhancements through topographical factors.

The solution in [2] uses Gaussian Process Regression (GPR) to predict heavy rainfall days from long-term data, aiding emergency preparedness. Despite its computational intensity, GPR balances accuracy with data-driven decision making and could be optimized for real-time use.

The paper [3] reviews rainfall prediction techniques, highlighting hybrid models combining time series, regression, neural networks, and decision trees. It emphasizes better data preprocessing, model scalability, and data sufficiency to improve accuracy.

The solution proposed in [4] revolves around the use of Explainable AI (XAI) to improve the transparency and interpretability of flood prediction models. By applying XAI

TABLE I
SUMMARY OF KEY STUDIES ON RAINFALL AND FLOOD PREDICTION

Ref	Method(s) Used	Key Findings	Limitations
[1]	DNN, SVM, KNN, Naïve Bayes	Real-time flood alerts using historical weather data; potential for topographical enhancements	Limited by historical data scope; lacks real-time topographical integration
[2]	Gaussian Process Regression (GPR)	Accurate heavy rainfall prediction; supports emergency preparedness	Computationally intensive; real-time optimization needed
[3]	Hybrid Models (Time Series, Regression, Neural Nets, Decision Trees)	Emphasizes data preprocessing and scalability for improved accuracy	Data sufficiency and model scalability remain challenges
[4]	Explainable AI (XAI) with ML	Improves interpretability of flood prediction models; enhances user trust	Black-box nature of some AI models persists; field validation lacking
[5]	Logistic Regression, Decision Trees, SVM, KNN, ANN	Region-specific flood risk assessment; actionable insights for disaster management	Limited geographic scope; adaptation to other regions untested
[6]	Hybrid ML (Neural Nets + Decision Trees)	Robust rainfall prediction across regions; scalable models	Integration complexity; generalizability to new climates not fully explored

techniques, the authors ensure that users can understand the relevance of various features in flood predictions, addressing the challenge of black-box AI models. Machine learning models, enhanced with XAI, allow stakeholders to analyze key parameters such as rainfall intensity, topography, and temperature. This helps build trust in the AI predictions, making them more actionable for emergency planning and risk assessment. The proposed solution thus bridges the gap between accuracy and interpretability in flood prediction models.

The research in [5] presents an AI / ML-based solution for prediction of rainfall and assessment of flood risk using models such as logistic regression, decision trees, SVM, KNN and ANN. Focused on Kolhapur, it assists disaster management with actionable insights. Though limited to Maharashtra, it can be adapted for broader regional and national flood preparedness.

The solution proposed in [6] focuses on the application of hybrid machine learning models for improving rainfall prediction accuracy. The integration of neural networks with decision trees and other machine learning techniques is highlighted as an effective way to overcome the limitations of traditional models. The authors demonstrate that hybrid models, which combine the strengths of multiple algorithms,

are more robust in predicting complex weather patterns. The solution emphasizes the scalability and adaptability of these models, making them applicable across diverse geographic regions and climate conditions, thus offering a comprehensive approach to rainfall forecasting.

III. METHODOLOGY

The methodology outlines the study, covering data collection, preprocessing, model training, evaluation, and interpretability analysis.

A. Data Collection and Preprocessing

The study incorporated the use of historical and real-time meteorological data.

- The historical data recorded was collected from sources such as IMD(Indian Meteorological Department) and Kaggle.
- For real-time weather data, the Python libraries Open-Meteo, Meteostat, and OpenWeatherMap were used. By providing location, start and end date, we can obtain weather data of that time period at the specified location.
- Missing values were handled using the Standard Imputer. Here, missing values are replaced with the mean of available values.
- To normalize the feature distribution across data, the Standard Scaler was used which scales the data to have zero mean and unit variance.

B. Data Processing

- Independent factors- precipitation_sum, rain_sum, precipitation_hours were used to predict the rain.
- These precipitation values and other factors like river discharge, soil moisture, etc. were then used to predict if a flood might occur or not.
- The data was divided into 80-20 for training and testing the models.

C. Model Evaluation

To evaluate the performance of the models, the Accuracy, MAE, RMSE and R^2 scores were measured and compared.

- Accuracy: Measures the percentage of correctly predicted instances over the total instances. Used for classification models like SVM.
- MAE (Mean Absolute Error): Calculates the average absolute difference between actual and predicted values. Used for regression models like Random-Forest.
- RMSE (Root Mean Square Error): Emphasizes larger errors by squaring differences, ideal for time-series models like LSTM.
- R^2 Score (Coefficient of Determination): Evaluates model fit by comparing predicted and mean values. Used for regression tasks.

D. Model Training

- 1) **Random-Forest:** Random Forest is an ensemble method that combines the results of multiple individual trees to improve accuracy and reduce overfitting. Each tree is trained on a random subset of features, ensuring diversity. The prediction is obtained through majority voting for classification or averaging for regression.
 - Use RandomizedSearchCV for tuning key parameters efficiently.
 - Increasing n-estimators generally improves accuracy but may slow down training.
 - Tune max_depth to control overfitting.
 - Use min_samples_split to balance model complexity.
- 2) **XGBoost:** Extreme Gradient Boosting (XGBoost) enhances traditional boosting algorithms by incorporating regularization techniques for better performance. It computes residual errors for each sample and fits a decision tree on these residuals. It minimizes the objective loss function.
 - Use early_stopping_rounds during training to stop once validation loss stabilizes.
 - Reduce learning_rate for smoother convergence.
 - Tune max_depth and subsample for optimal model generalization.
- 3) **SVM:** Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression. The algorithm maps input data to a higher-dimensional space using kernel functions, improving its ability to handle non-linear data. SVM's robustness makes it effective for rainfall and flood prediction by distinguishing between complex patterns in environmental data. Its margin-based approach enhances generalization, particularly when the dataset has clear boundaries or minimal noise.
 - For large datasets, consider using LinearSVC for better scalability.
 - Select the RBF kernel for non-linear data.
 - Adjust C for margin and misclassification control.
 - Tune gamma to manage boundary flexibility.
- 4) **SVC:** (Support Vector Classifier) SVC is a variant of SVM optimized for classification tasks. It leverages kernel functions like RBF, polynomial, and sigmoid for improved decision boundaries.
 - Use GridSearchCV to determine optimal C and gamma.
 - For unbalanced data, tune the class_weight parameter.
- 5) **LSTM:** Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that efficiently models sequential data by maintaining long-term dependencies. LSTM achieves this through memory cells and gated mechanisms that control the flow of information. Forget

gate controls what information to forget, input gate decides which values to update and output gate regulates the output information. Its architecture prevents vanishing gradient issues, ensuring stable training over long data sequences.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (1)$$

where f_t is the forget gate, i_t is the input gate and \tilde{C}_t is the candidate cell state.

- Increase hidden units to expand model capacity.
- Use dropout layers to prevent overfitting.
- Tune timesteps to capture appropriate sequence lengths.

- 6) **Hyperparameter Tuning:** Hyperparameter tuning optimizes model performance by adjusting key settings. Methods used for tuning are Grid search, Random search, Bayesian optimization. Tuning parameters like learning rate, tree depth, or kernel type directly influences model accuracy. In rainfall and flood prediction, hyperparameter tuning ensures optimal model performance by balancing bias-variance trade-offs.

E. Explainability

The study employed the Local Interpretable Model-agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) technique to enhance model transparency.

LIME works by generating a simplified, interpretable model that approximates the original complex model's behavior within a local region around a given prediction.

- LIME creates multiple synthetic data points by slightly altering the input features.
- The original model predicts outcomes for these perturbed instances.
- Each synthetic point is assigned a weight based on its proximity to the original data point.
- A simple interpretable model (e.g., linear regression) is trained using the weighted dataset.
- Features with positive weights are considered to affect the predictions whereas features with negative weights does not significantly affect the predictions.

SHAP works by generating a simplified, interpretable model that approximates the original complex model's behavior within a local region around a given prediction. It assigns each feature an importance value by calculating its marginal contribution across all possible feature combinations. This ensures a fair distribution of feature influence on the prediction.

- SHAP calculates the contribution of each feature by analyzing combinations of features.
- It creates multiple perturbed instances by selectively including or excluding features.

- The model predicts outcomes for these perturbed instances.
- The contribution of each feature is calculated by averaging its marginal impact across all combinations.
- Positive SHAP values indicate features that increase the prediction, while negative values denote features that reduce it.

IV. IMPLEMENTATION DETAILS

A. Data Collection and Preprocessing

The implemented system leverages deep learning and explainable AI (XAI) techniques to predict rainfall and potential flooding in a given location. The goal is not just to forecast weather conditions but also to provide clear, interpretable insights into the factors influencing these predictions.

As shown in Fig. 1, the system begins with user input to identify a location of interest. It then gathers real-time weather data for the past 60 days to make predictions. The model itself has been trained on 15 years of historical weather data, ensuring it has learned long-term patterns and trends before being applied to new data.

Before training, the historical dataset undergoes preprocessing, including handling missing values with a standard imputer and applying logarithmic transformations to normalize the data. The features are then standardized using a scaling technique to maintain consistency. Once the data is prepared, a Long Short-Term Memory (LSTM) model is trained to recognize patterns in weather conditions, particularly those that indicate rainfall and flooding.

B. Prediction and Decision-Making

When a new query is received, the prediction engine analyzes the past 15 days of weather data and determines whether rainfall is expected. If no rainfall is predicted, the system simply informs the user that conditions are normal. However, if rainfall is detected, the system further assesses whether it poses a flood risk. If a flood is likely, explainability techniques such as Local Interpretable Model-agnostic Explanations (LIME) or SHapley Additive exPlanations (SHAP) come into play, breaking down the decision-making process into understandable insights. This allows users to see which weather factors contributed most to the prediction.

C. Implementation

The system is developed using Python, with key libraries including Jupyter Notebook, TensorFlow/Keras for deep learning, Pandas and NumPy for data handling, and Scikit-learn for preprocessing. Weather data is sourced from APIs or meteorological databases and stored in a structured format for analysis. The LSTM model is implemented using TensorFlow/Keras, given its strength in handling sequential data. It has been trained on 15 years of historical weather data, allowing it to recognize long-term weather trends. When

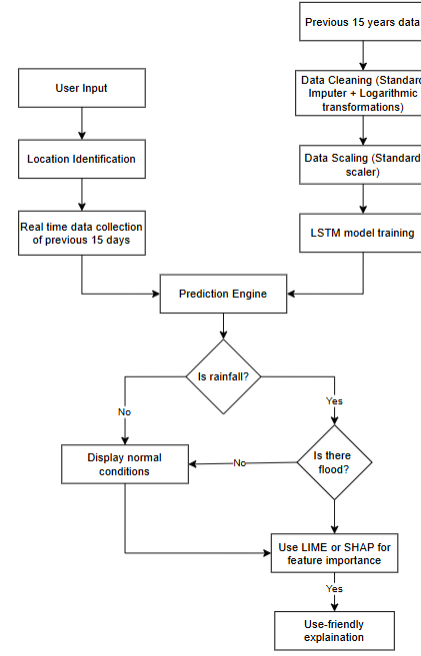


Fig. 1. Flowchart of the Implemented System

making real-time predictions, the model takes the most recent 15 days of weather data as input.

To enhance transparency, the system integrates SHAP and LIME for explainability. These tools help highlight which meteorological parameters—such as temperature, humidity, or pressure—most influenced the model’s prediction. The results are visualized using Matplotlib to make them more accessible to users. For deployment, the system is implemented as a mobile application using React Native. The app runs locally on the user’s device, retrieving real-time weather data from APIs and processing predictions without reliance on cloud-based services. The trained model is embedded within the app, ensuring offline functionality and faster response times. Users can input their location and receive predictions along with explanations directly on their mobile devices.

By combining accurate predictions with interpretable insights, this system not only forecasts rainfall and flooding but also empowers users to understand the reasoning behind each prediction, making weather forecasting more transparent and actionable.

V. RESULTS

Accurate and interpretable rainfall and flood prediction is essential for effective disaster preparedness, agricultural planning, and infrastructure management. Predictive models must balance precision and recall to ensure reliability while minimizing false alarms. Moreover, flood forecasting requires a comprehensive assessment of multiple environmental factors, ensuring timely warnings that can be understood and acted upon by stakeholders. Given the high stakes involved,

integrating Explainable AI (XAI) is critical to improving transparency, trust, and decision-making in weather prediction systems.

This study leverages machine learning models—including Random Forest, Long Short-Term Memory (LSTM), Support Vector Machines (SVM), and XGBoost—to predict rainfall intensity and flood risk. To enhance interpretability, SHAP (Shapley Additive Explanations) is employed to analyze the global impact of different meteorological factors, while LIME (Local Interpretable Model-Agnostic Explanations) provides localized insights into specific predictions. These explainability techniques help identify key drivers of extreme weather events, making AI-driven forecasts more actionable.

A. Evaluation of the models

To evaluate the effectiveness of different machine learning approaches, this study compared Random Forest, Long Short-Term Memory (LSTM), Support Vector Machine (SVM), and XGBoost, assessing their performance across key metrics, as shown in Table II. Each model demonstrated varying levels of accuracy, recall, and error rates, impacting their suitability for rainfall and flood forecasting.

TABLE II
PERFORMANCE EVALUATION OF MODELS

Metric	Random Forest	LSTM	SVM	XGBoost
Accuracy (%)	92.34	94.21	86.57	93.10
Precision (%)	90.12	92.45	85.33	91.72
Recall (%)	91.89	93.78	84.29	92.50
F1-Score (%)	91.00	93.10	84.80	92.10
R ² Score	0.85	0.88	0.79	0.87
MSE	12.4	9.7	18.9	10.3
RMSE	3.52	3.12	4.35	3.21

LSTM emerged as the best-performing model, achieving an accuracy of 94.21%, the highest among all models. Its ability to learn from sequential data allows it to capture long-term dependencies in rainfall patterns, making it particularly effective for time-series forecasting. It also recorded the lowest Mean Squared Error (MSE) of 9.7 and a Root Mean Squared Error (RMSE) of 3.12, indicating lower prediction errors. However, LSTM requires high computational resources and longer training times, which may limit its practicality for real-time applications.

XGBoost followed closely with an accuracy of 93.10%, offering a strong balance between predictive power and computational efficiency. It achieved a recall of 92.50% and an F1-score of 92.10%, ensuring reliable identification of extreme weather conditions while maintaining a low false positive rate. Additionally, its MSE of 10.3 and RMSE of 3.21 were slightly higher than LSTM's, but it remained competitive in terms of performance. Unlike LSTM, XGBoost requires careful hyperparameter tuning but is computationally more efficient, making it a solid choice for large-scale weather prediction.

Random Forest performed well, with an accuracy of 92.34%

and recall of 91.89%, making it particularly effective for flood risk assessment. Its MSE of 12.4 and RMSE of 3.52** were slightly higher than those of XGBoost and LSTM, indicating moderate prediction errors. While Random Forest provides high interpretability and is less prone to overfitting, its lack of sequential learning limits its ability to capture long-term rainfall trends, making it less suitable for forecasting prolonged weather patterns. SVM had the lowest accuracy at 86.57%, along with the highest MSE (18.9) and RMSE (4.35), suggesting it struggled with the complexity of meteorological data. While it performed reasonably well in structured classification tasks, it lacked the flexibility needed to handle nonlinear relationships in weather patterns. Given its lower accuracy and higher error rates, SVM is not recommended for large-scale rainfall and flood prediction. Overall, LSTM and XGBoost were the most effective models, with LSTM excelling in capturing time-dependent weather trends and XGBoost providing a strong balance between accuracy and efficiency. Random Forest remains a reliable option for flood risk prediction due to its high recall, while SVM's lower performance makes it less suitable for real-world meteorological forecasting.

B. Explainability using SHAP and LIME

Ensuring the transparency and interpretability of machine learning models is crucial, particularly in high-stakes applications such as rain and flood prediction. While predictive accuracy is vital, understanding the reasoning behind a model's decisions enhances trust and facilitates informed decision-making by meteorologists, policymakers, and disaster response teams. To achieve this, we employ SHapley Additive exPlanations (SHAP) and Local Interpretable Model-Agnostic Explanations (LIME), which provide insights into the contributions of individual features toward model predictions.

1) *SHAP-Based Feature Attribution*: SHAP values quantify the marginal contribution of each feature to the model's predictions, offering both global and local interpretability. The SHAP summary plot shown in Fig. 2 illustrates the impact of meteorological parameters on flood risk estimation, highlighting precipitation hours, maximum temperature, and wind speed as the most influential factors. The color gradient represents feature values, with high values in red and low values in blue. Notably, an increase in precipitation hours and maximum temperature is strongly correlated with a heightened flood risk, whereas lower wind speeds are associated with reduced flood likelihood.

Furthermore, SHAP values reveal intricate feature interactions. Temperature variations (temperature_2m_max and temperature_2m_min) exhibit nonlinear relationships with flood risk, indicating the presence of threshold effects or interdependencies that complex models such as LSTM and XGBoost are well-equipped to capture. Negative SHAP values for certain features suggest counterbalancing influences, such

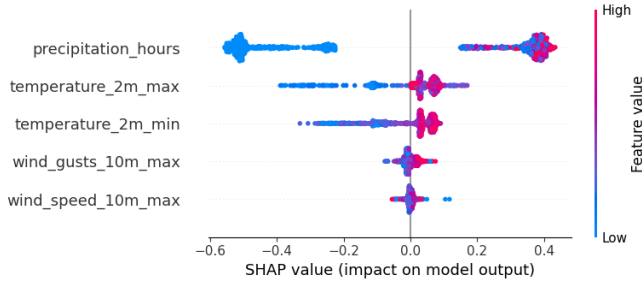


Fig. 2. SHAP evaluation of feature importance for historical data

as wind gusts mitigating the impact of heavy precipitation, thereby refining model predictions.

2) *LIME-Based Local Explanations*: LIME provides instance-specific explanations by approximating the model's decision boundary using interpretable linear models. The LIME visualization in Fig. 3 highlights the contribution of different features to flood risk predictions for specific data instances. Seasonal variations play a significant role, with months such as MAM (March-April-May) and APR (April) exerting a strong positive influence, aligning with monsoonal precipitation patterns. Conversely, months such as JJAS (June-July-August-September) and JUN (June) exhibit negative contributions, suggesting seasonal dependencies in flood risk assessment.

A similar analysis on real-time data, as shown in Fig. 4, reinforces the model's sensitivity to environmental conditions. Feature 2 emerges as the most significant positive predictor, whereas Feature 4 exhibits a dampening effect on flood risk predictions, demonstrating the variability of meteorological influences across different time frames and geographic conditions. The weighted contributions identified in LIME align closely with the feature importance rankings derived from SHAP, underscoring the robustness of the model's interpretability framework.

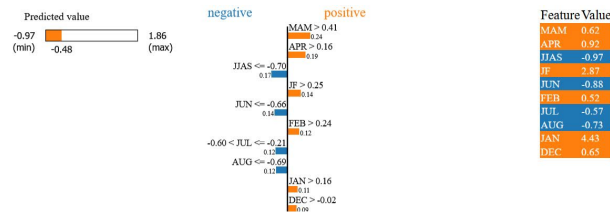


Fig. 3. LIME evaluation of feature importance for historical data

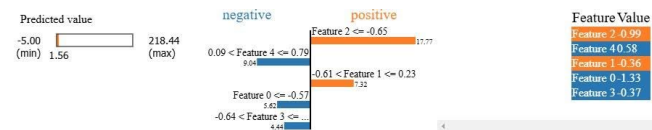


Fig. 4. LIME evaluation of feature importance for real-time data

VI. CONCLUSION

The study trained multiple models on 15 years of historical meteorological data, and LSTM achieved the highest accuracy, making it the most suitable for time-series forecasting. Preprocessing included logarithmic transformation and Standard Scaler to normalize data and improve convergence. For predicting the upcoming week's weather, a 15-day rolling window was used, and the model was trained for 5 epochs to optimize performance. Explainability techniques were applied to identify key influencing factors, ensuring transparency.

REFERENCES

- [1] S. Sankaranarayanan, M. Prabhakar, S. Satish, P. Jain, A. Ramprasad, and A. Krishnan, "Flood prediction based on weather parameters using deep learning," *Journal of Water and Climate Change*, vol. 11, no. 4. IWA Publishing, pp. 1766–1783, Nov. 14, 2019. doi: 10.2166/wcc.2019.321.
- [2] Subrahmanyam, Kandula & Ramsenthil, C & Imran, Girach & Chakravorty, Aniket & Sreedhar, R & Ezhilraj, E & D., Bala Subrahmanyam & Ramachandran, Radhika & kishore kumar, Karanam & M, Rajasekhar & Jha, Chandra. (2021). "Prediction of heavy rainfall days over a peninsular Indian station using the machine learning algorithms." *Journal of Earth System Science*. 130. 10.1007/s12040-021-01725-9.
- [3] Gomathy, C K, Annapareddy Bala Narasimha Reddy, Aravapalli Pavan Kumar, Aile Lokesh (2021), "A Study On Rainfall Prediction Techniques", *International Journal of Scientific Research in Engineering and Management (IJSREM)*, Volume: 05 Issue: 10, Oct - 2021.
- [4] Kadiyala, Sai & Woo, Wai Lok. (2022). "Flood Prediction and Analysis on the Relevance of Features using Explainable Artificial Intelligence." 10.48550/arXiv.2201.05046.
- [5] A. Nagure, S. Chandekar, P. Mahadik, V. Yeolekar, M. Syed and J. Chakankar, "Rainfall Prediction for Flood Risk Assessment of Kolhapur Region Using Ai/ML Models," 2023 1st International Conference on Cognitive Computing and Engineering Education (ICCEE), Pune, India, 2023, pp. 1-8, doi: 10.1109/ICCEE55951.2023.10424675.
- [6] Dotse, Sam-Quarcoo & Larbi, Isaac & Limantol, Andrew & De Silva, Liyanage. (2023). A review of the application of hybrid machine learning models to improve rainfall prediction. *Modeling Earth Systems and Environment*. 10.1007/s40808-023-01835-x.