Machine Learning for Rainfall-Driven Flood Prediction: A Comparative Study of Six Classifiers for Data-Driven Early Warning

Jayesh Deshmukh[1], Akash Bhosale[2], Shivam Dubey[3], Yash Kharat[4], Dr. Sunil Mahajan[5]
[1,2,3,4]*M.Sc. Computer Science Students, Department of Computer Science, MAEER's MIT Arts, Commerce and Science College, Alandi (D), Pune, Maharashtra, India*
[5]*Associate Professor, Department of Electronics, MAEER's MIT Arts, Commerce and Science College, Alandi (D), Pune, Maharashtra, India*

***Abstract*—Flood early-warning systems increasingly use data-driven methods to complement conventional hydrological forecasting. This paper presents a comparative study of six supervised machine-learning classifiers—K-Nearest Neighbours (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM) and Naïve Bayes (NB)—for rainfall-based flood classification. Monthly rainfall records for the Pune meteorological subdivision of Maharashtra, India, were used for model development, with the year 2022 reserved as an out-of-time check. The workflow covers data cleaning, feature engineering, scaling, hyperparameter tuning and evaluation using accuracy, precision, recall and F1-score. DT, RF and SVM achieved a test accuracy of 1.000, LR and KNN achieved 0.991, and NB achieved 0.947. Because near-perfect scores in a natural-hazard task warrant scrutiny, the paper also examines how label construction, temporal leakage, outlier handling and class imbalance can inflate performance, and outlines the validation steps required before such models can support operational early warning.**

***Index Terms*—Flood prediction, machine learning, rainfall analysis, Random Forest, early-warning systems, temporal validation.**

I. INTRODUCTION

Floods cause fatalities, displacement, crop losses, infrastructure damage and prolonged economic disruption, and their frequency and intensity are expected to change with a warming climate [1], [2]. In monsoon-affected regions such as Maharashtra, rainfall is a major driver of flooding; however, the relationship between precipitation and flood occurrence is shaped by catchment conditions, river levels, land use, drainage, soil moisture and reservoir operations. Reliable early warning therefore depends on timely observations and on the careful interpretation of predictive models.

Physically based hydrological models represent runoff and catchment processes explicitly, but they require detailed inputs, calibration and considerable computational resources. Statistical and machine-learning (ML) methods offer a complementary approach by learning patterns directly from historical observations [3]. Their value, however, depends on the quality of the labels, the representativeness of the training data and evaluation procedures that reflect real operational use.

This paper compares six widely used classifiers within a rainfall-based flood prediction workflow. The contributions are: (i) a uniform preprocessing and tuning pipeline applied to all six models; (ii) a comparison of their validation and test performance; and (iii) a critical discussion of the methodological factors that may explain unusually high scores, together with the validation steps needed before deployment in a warning service.

II. RELATED WORK

Reviews of flood-prediction research describe a growing role for ML alongside hydrological modelling. Mosavi et al. [3] survey a wide range of data-driven flood-prediction methods and highlight the potential of ensemble and hybrid approaches. Kratzert et al. [5] demonstrated Long Short-Term

Memory networks for rainfall–runoff modelling, while Nevo et al. [4] described ML within an operational flood-forecasting framework.

The six classifiers considered here embody different assumptions, summarised in Table I. Decision Trees learn threshold-based rules [6]; Random Forests aggregate many decorrelated trees to reduce the instability of a single tree [7]; SVMs construct maximum-margin boundaries using a selected kernel [8]; Logistic Regression models class probability through a linear predictor [9]; KNN relies on proximity between observations [10]; and Naïve Bayes applies a conditional-independence assumption [11]. Their relative performance depends on feature design, sample size, class balance and the evaluation split [12].

Table I Characteristics of the Compared Classifiers

Model	Decision principle	Interpretability
DT	Axis-aligned threshold rules	High
RF	Bagged ensemble of trees	Moderate
SVM	Maximum-margin kernel boundary	Low
LR	Linear log-odds model	High
KNN	Majority of nearest neighbours	Moderate
NB	Conditional independence, Bayes rule	Moderate

A recurring challenge in environmental prediction is validation under temporal and spatial dependence. Randomly splitting correlated observations can make test performance appear stronger than performance on genuinely future data [17]. Data leakage can also occur when feature selection, preprocessing or label construction uses information unavailable at prediction time [16]. The comparison presented here is interpreted in light of these risks.

III. MATERIALS AND METHODS

The overall workflow is shown in Fig. 1.

A. Study Data

Monthly rainfall records for the Pune meteorological subdivision of Maharashtra, India, were obtained from India Meteorological Department (IMD) gridded datasets.

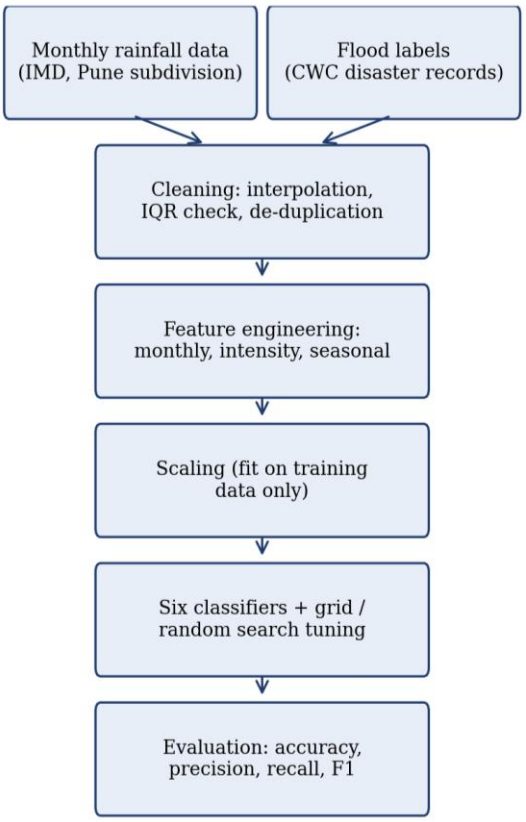


Fig. 1. Workflow of the proposed flood classification study.

Binary flood labels were derived from district-level disaster reporting archives maintained by the Central Water Commission (CWC). A year was labelled as a flood event when annual precipitation exceeded the 95th historical percentile and a basin overflow was reported.

B. Study Periods and Data Partitioning

The data were divided into three blocks (Table II): a historical block (1901–2015), a recent block (2016–2021) and a latest-year block (2022) reserved as an out-of-time hold-out. Within the development data, a 70%/15%/15% split was used for training, validation and testing.

Table II Data Periods and Their Roles

Data block	Period	Role
Historical record	1901–2015	Model development
Recent record	2016–2021	Model development
Latest year	2022	Out-of-time hold-out

C. Data Cleaning and Preprocessing

Missing monthly values were filled by linear interpolation along the time axis. Outliers were identified using the interquartile-range (IQR) rule and replaced with the feature median, and duplicate records were removed.

Min–max scaling and standardisation were evaluated as alternative scaling schemes; in every case the scaler was fitted on the training data only and then applied to the validation and test data to prevent information leakage.

D. Feature Engineering

Three feature groups were constructed: monthly rainfall totals, rainfall-intensity measures, and seasonal aggregates for the monsoon (June–September), post-monsoon and dry periods. Tree-based feature-importance scores were used to retain the most informative features.

E. Classifiers and Hyperparameter Tuning

All six classifiers were implemented in Python using scikit-learn [13], NumPy, pandas and Matplotlib in a Jupyter Notebook environment.

Grid search and random search [14] were used for tuning. The search covered tree depth and minimum samples per leaf for DT and RF, number of estimators for RF, kernel type and regularisation parameter for SVM, regularisation strength for LR, and number of neighbours for KNN.

F. Evaluation Metrics

Performance was assessed using accuracy, precision, recall and F1-score, defined in (1)– (4), where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives respectively.

$$Accuracy = (TP + TN) / (TP + TN + FP + FN) \quad (1)$$

$$Precision = TP / (TP + FP) \quad (2)$$

$$Recall = TP / (TP + FN) \quad (3)$$

$$F1 = 2 \times Precision \times Recall / (Precision + Recall) \quad (4)$$

For flood warning, recall is particularly important because false negatives represent missed flood events, while precision matters because repeated false alarms erode public trust.

When flood years are rare, accuracy alone can be misleading, and precision–recall analysis is more informative [15].

IV. RESULTS AND DISCUSSION

A. Comparative Performance

Table III and Fig. 2 present the validation and test accuracy of the six classifiers. DT, RF and SVM achieved perfect accuracy (1.000) on both partitions. LR and KNN followed closely at 0.991 test accuracy, while NB recorded the lowest values (0.893 validation, 0.947 test), consistent with its conditional-independence assumption being violated by strongly correlated monthly and seasonal rainfall features.

Table III Validation and Test Accuracy of the Classifiers

Classifier	Validation	Test
Decision Tree (DT)	1.000	1.000
Random Forest (RF)	1.000	1.000
Support Vector Machine (SVM)	1.000	1.000
Logistic Regression (LR)	0.992	0.991
K-Nearest Neighbours (KNN)	0.990	0.991
Naïve Bayes (NB)	0.893	0.947

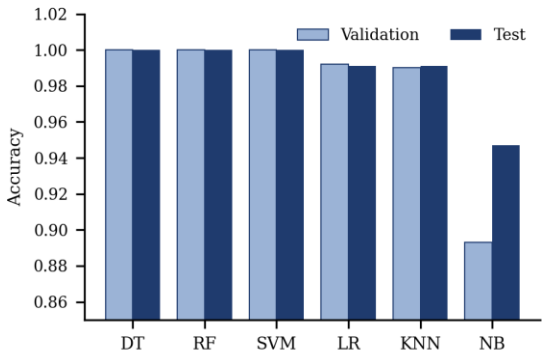


Fig. 2. Validation and test accuracy of the six classifiers.

These results suggest that threshold-based (DT, RF) and margin-based (SVM) decision boundaries fit the constructed classification task well. On their own, however, they do not establish that the models can predict real flood events in future seasons or unseen locations, for the reasons discussed below.

B. Label Construction and Leakage

Because the flood label is partly defined by annual rainfall exceeding the 95th percentile, and rainfall is also the main input, the models may partly recover the

rule used to create the label. This form of target leakage [16] can explain near-perfect scores, particularly for threshold-based learners such as DT and RF. Defining labels from independent event records, restricting predictors to information available at the intended warning time and using temporally blocked validation [17] are therefore necessary for an unbiased estimate of predictive skill.

C. Outlier Treatment and Class Balance

Replacing extreme rainfall values with the median may suppress precisely the events most relevant to flood warning. Extreme values should be retained unless confirmed as sensor or processing errors; robust scaling or an explicit outlier indicator is preferable. In addition, flood years are expected to be a minority class, so class-specific precision, recall, specificity, false-alarm rate and confusion matrices should accompany accuracy for every model.

D. Candidate Model for Further Testing

Random Forest is a reasonable candidate for further testing because ensembles are less sensitive than a single tree to small changes in training data [7], while a Decision Tree offers a compact and readable rule set for decision makers. This is a candidate-selection observation rather than a deployment recommendation.

V. DIGITAL EARLY-WARNING FRAMEWORK

A validated rainfall-based classifier could form one component of a broader digital early-warning service organised in four layers. A data layer ingests rain-gauge observations, weather feeds and archived records with automated quality checks. An analytics layer runs the validated model to produce area-specific risk estimates. A decision-support layer presents these estimates to officials together with uncertainty and relevant hydrological information. A communication layer distributes approved alerts through SMS, mobile applications and public announcements in local languages.

Explainability, monitoring and accountability are essential in such a system. Tree-based models provide readable decision rules, and tools such as SHAP [18] can explain individual predictions of more complex models. Model outputs should complement, not replace, hydrological forecasts and expert judgement, and any deployment requires documented alert thresholds, procedures for handling false alarms and missed events, and periodic reassessment as rainfall patterns change.

VI. LIMITATIONS AND FUTURE WORK

The study uses rainfall-only inputs, which cannot represent all processes responsible for flooding. Monthly aggregates can conceal short-duration intense rainfall, and flood occurrence also depends on river stage, soil moisture, reservoir operations, terrain, land cover and drainage. The overlap between the label definition and the input features is a further limitation. Future work will publish a data dictionary, label definitions, preprocessing code, final hyperparameters, random seeds and confusion matrices; adopt rolling-origin or time-blocked validation and, where relevant, spatially blocked evaluation; define the target from independent flood-event records; preserve extreme rainfall unless it is a documented error; and benchmark the models against simple threshold baselines and established hydrological forecasts. Daily or hourly rainfall and river-gauge observations will also be explored for short-lead-time warnings.

VII. CONCLUSION

This paper compared six machine-learning classifiers for rainfall-driven flood classification in the Pune meteorological subdivision. Decision Tree, Random Forest and Support Vector Machine achieved the highest accuracy, followed by Logistic Regression and K-Nearest Neighbours, with Naïve Bayes performing lowest. The central conclusion is, however, methodological: near-perfect scores in a natural-hazard prediction task demand transparent label construction, leakage-resistant temporal validation, consistent sample accounting and full class-specific evaluation. Once these requirements are met, lightweight ML models can be assessed as components of explainable, low-cost decision-support systems for disaster management.

ACKNOWLEDGMENT

The authors thank the Department of Computer Science and the Department of Electronics, MAEER's

MIT Arts, Commerce and Science College, Alandi (D), Pune, for their support and facilities.

REFERENCES

[1] H. Tabari, "Climate change impact on flood and extreme precipitation increases with water availability," *Sci. Rep.*, vol. 10, Art. no. 13768, 2020.

[2] V. Mishra and H. L. Shah, "Hydroclimatological perspective of the Kerala flood of 2018," J. Geol. Soc. India, vol. 92, no. 5, pp. 645–650, 2018.

[3] A. Mosavi, P. Ozturk, and K. W. Chau, "Flood prediction using machine learning models: Literature review," Water, vol. 10, no. 11, Art. no. 1536, 2018.

[4] S. Nevo et al., "Flood forecasting with machine learning models in an operational framework," Hydrol. Earth Syst. Sci., vol. 26, no. 15, pp. 4013–4032, 2022.

[5] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger, "Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks," Hydrol. Earth Syst. Sci., vol. 22, no. 11, pp. 6005–6022, 2018.

[6] J. R. Quinlan, "Induction of decision trees," Mach. Learn., vol. 1, pp. 81–106, 1986.

[7] L. Breiman, "Random forests," Mach. Learn., vol. 45, pp. 5–32, 2001.

[8] C. Cortes and V. Vapnik, "Support-vector networks," Mach. Learn., vol. 20, no. 3, pp. 273–297, 1995.

[9] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.

[10] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, 1967.

[11] H. Zhang, "The optimality of naive Bayes," in Proc. 17th Int. Florida Artif. Intell. Res. Soc. Conf. (FLAIRS), 2004, pp. 562–567.

[12] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

[13] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.

[14] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res., vol. 13, pp. 281–305, 2012.

[15] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets," PLoS ONE, vol. 10, no. 3, Art. no. e0118432, 2015.

[16] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," ACM Trans. Knowl. Discov. Data, vol. 6, no. 4, Art. no. 15, 2012.

[17] D. R. Roberts et al., "Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure," Ecography, vol. 40, pp. 913–929, 2017.

[18] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. 30 (NeurIPS)*, 2017, pp. 4765–4774.