# Urban flood detection with the augmentation of gradient boosting and machine learning for prior warning sign over vulnerable zones

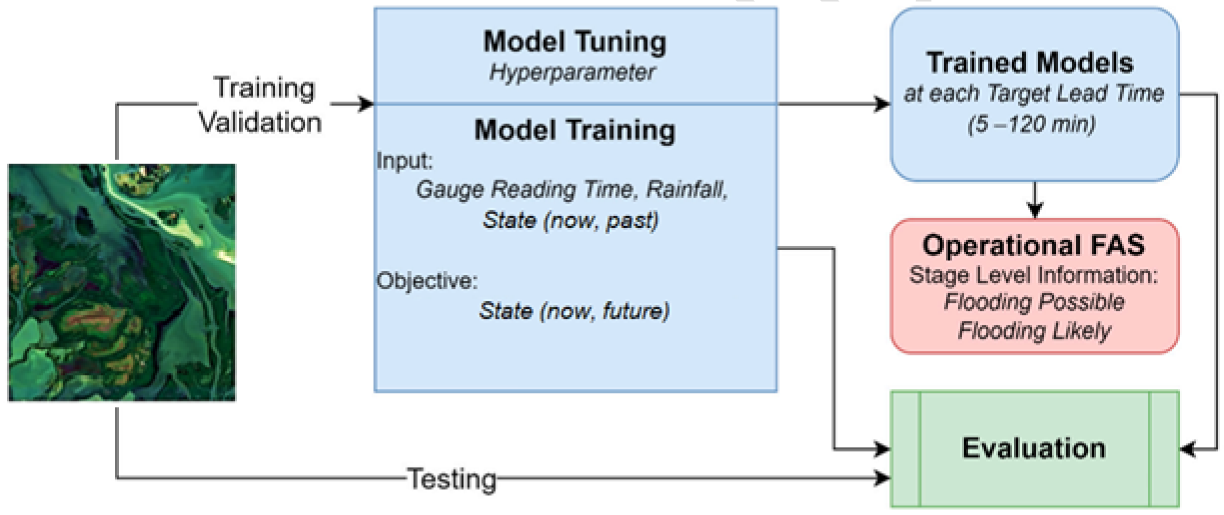
Dr.C. Saravanabhavan*
Professor
Department of computer science and Engineering
Kongunadu college of Engineering and Technology - Trichy
hodcse@kongunadu.ac.in

Dr.P. Sherubha,
Assistant Professor,
Department of Information Technology,
Karpagam College of Engineering,
Coimbatore.
sherubha.p@kce.ac.in

R.Swathika
Department of Information Technology Sri Sivasubramaniya Nadar College of Engineering,
Chennai, Tamil Nadu, India
swathikar@ssn.edu.in

Dr C.N. Ravi
Professor,
Department of Computer Science and Engineering,
CMR Engineering college, Hyderabad. Telangana.
mail2cnravi@gmail.com

**Graphical Abstract**



## Abstract

Urban flooding has severely threatened the ecosystem and human life in recent years. The key to managing stormwater is to understand what causes it. The forceful effects of building shape on urban floods should be addressed, which results in a significant underestimation of flood danger. Algorithms for data-driven machine learning shed light on how the placement of buildings affects urban flooding. This study aimed to identify the elements of flooding risk and their effects on nearby communities using a concatenated modelling loop that included the XGBoost algorithm. This work suggests an enhanced extreme gradient boosting (XGBoost) approach based on a concatenated boosting particle swarm optimization (CBPSO) operator to acquire the meteorological refractive index of 100 m over the ocean. The prediction results of the enhanced XGBoost algorithm are compared with those of the backpropagation (BP) network and the original XGBoost method using the evaluation criteria Accuracy, Precision, Recall, F1-score, and IoU.Moreover, the networks resolve the featured misaligned issue during the decoder by inserting a component synchronization module into the up-sampling procedure. The model's intersections of unions (IoU) of 89.90% outperformed SOTA flood detection systems.

**Keywords:** Flood prediction, concatenation, boosting, ensembling, IoU, flood-prone and multi-criteria.

## 1. Introduction

Floods, as a disruptive disaster, inflict severe damage to infrastructure, people, and property. Identifying flood-prone areas is essential for policy development and execution to mitigate risks and costs. However, examining natural processes, especially flooding, through experiments or analytical approaches is not always feasible. Establishing methodological techniques, models, and strategies for a rational, theoretical, and quantitative evaluation of floods becomes imperative. Flood vulnerability modeling and map-based methodologies often stem from two systems: physically-based methods grounded in experiential understanding and data-driven approaches employing techniques such as Decision Trees (DT), Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Networks (ANNs), Naive Bayes (NB), Logistic Regression (LR), and Feature Ranking (FR) methods, among others.

Many research endeavors focused on flood vulnerability leverage multi-criteria decision-making (MCDA). Advanced mathematical representations that capture observable behaviors serve as the foundation for physical forecasting systems. Conversely, data-driven frameworks rely on mathematical formulas constructed from concurrent input and output data, avoiding traditional physical methods. In this realm, Machine Learning (ML) models have risen to prominence, particularly for natural disaster forecasting, encompassing landslide hazards, forest fire vulnerability, and flood susceptibility. The Artificial Neural Network (ANN) framework, especially the Multi-Layer Perceptron Neural Network (MLP-NN), stands out as a popular ML model for hazard risk forecasting. Its strength lies in its ability to effectively approximate complex nonlinear input-output relationships and unveil hidden connections within historical data.

Another noteworthy addition to the ML arsenal is AdaBoost, an enhancing algorithm that can be harnessed without prior expertise in the intricacies of weak learning approaches. Coupled with the Expectation-Maximization (EM) technique, AdaBoost yields precise predictions and boasts resistance to overfitting, making it a promising ML algorithm for assessing flood risk. Decision Trees (DTs) and Random Forest (RF) models prove highly effective for accurately identifying flood-affected regions. Logistic Regression (LR) emerges as a capable tool for predicting the presence or absence of floods, with numerous studies employing a range of geo-hydrological variables to exploit its potential. Support Vector Machines (SVM), a probabilistic classifier classification algorithm, find frequent application in flood vulnerability assessments.

Traditionally, flood vulnerability assessments have relied on one or a limited number of ML methods. However, employing a diverse set of ML approaches to scrutinize flood vulnerability and its response to varying conditions represents a novel contribution to the literature. This study aims to evaluate flood vulnerability using a multitude of ML algorithms alongside an extensive dataset comprising meteorological, hydrodynamic, and geographical information at a high spatial resolution (12.5 meters). The mainstream of the proposed process is discussed as follows:

- Input is the dataset collected across twenty-one meteorological stations in the Cuddalore district, Tamil Nadu, India.The dataset is categorized based on the vulnerable Zone and its rainfall occurrence.
- When a dataset is first brought into the picture, it is often raw and can contain empty or irrelevant values. To help refine the dataset, a pre-processing step is introduced that uses PrincipalComponent Analysis (PCA). A significant pre-processing function used to reduce the dimensions of the dataset is the Singular Value decomposition of linear algebra, for which the columns would be the axes with high dimensional space.
- The most crucial phase involves the machine learning algorithm. Based on the level of accuracy, the RandomForest algorithm is chosen for computing the feature selection. This classifier is an ensemble learning approach of categorization, validation, and other algorithms that, as already said, functions by building an enormous classification tree during training & producing the classification that is the average projection of the individual plants or the median of this classification.
- Using XGBoost, the rainfall prediction is computed, which depends on the parameter passed and the ranges defined. The regions are classified by providing different colours for adequate identification. Each colour represents a pre-defined range of data. The following categories are used to categorize the rainfall data for this study: 1) No rainfall; 2) Light rainfall; 3) Moderate rainfall; 4) Heavy rainfall; 5) Significantly heavier rainfall and 6) Extremely heavy rainfall. The above classification is required to assess the impact of rainfall in computing the chances of flood occurrence.

The work is organized as follows: section 2 provides a detailed explanation on the diverse prevailing approaches. The methodology is demonstrated in section 3 with investigational outcome is provided in section 4. The summary is descriptive in section 5.

## 2. Literature Review

This chapter covered significant flooding incidents throughout the globe and discussed pertinent research on ML for flood modelling.

### 2.1 Flooding Events Around the World

We consider floods conducted by various writers in various regions of the globe. We concentrate mainly on nations that experience large flooding disasters often and the models used to forecast future occurrences. Over time, flood vulnerability has caused significant human suffering, including food shortages, waterborne disease epidemics, and infrastructure destruction [1][2] found that 15 nations (see Table 1) represent almost 80% of the annual flood victims. These emerging or developed countries are susceptible to natural catastrophes and climate change.

Africa, Asia, and South America are the top 15 flood-prone nations. Several research studies have shown that floods pose a significant threat to thousands of millions of citizens in both India with Bangladesh being one of the countries that are most severely impacted by floods; during the summer monsoon, almost one-third of Bangladesh is submerged in water [3,4]Bangladesh's lowland, geographical location, and dense population make floods a significant danger to both individuals and resources according to [5]. Flooding, the most common natural catastrophe in India, is triggered by sudden southwest monsoon rains, riverbed floodplains, and tropical storms. Floods impact 84% of India's projected GDP annually [6,7]. According to [8], blocking drainage routes and proximity to coastline waters significantly contribute to floods in susceptible regions.

[9] lists three significant causes of flooding in South America, including growing urbanization, climate change, and land use decisions. In Africa, the increasing amounts of the Lagdo dam have repeatedly caused enormous floods of significant emergency rates in recent generations. The 2012 & 2022 statistics were exceptional. In Nigeria, floods are not only a natural process; humans may also cause them due to inadequate or nonexistent drainage channels, improper waste management devices, unchecked development, and lax enforcement of planning rules.

**Table 1: Annual Expected Population Affected by River Flood**

| **Countries** | **Population in millions** | **Continent** | | |
|---|---|---|---|---|
| | | **ASIA** | **AFRICA** | **SOUTH AMERICA** |
| India | 4.85 | ✓ | - | - |
| Bangladesh | 3.49 | ✓ | - | - |
| China | 3.29 | ✓ | - | - |
| Vietnam | 0.94 | ✓ | - | - |
| Pakistan | 0.72 | ✓ | - | - |
| Indonesia | 0.65 | ✓ | - | - |
| Egypt | 0.47 | - | ✓ | - |
| Myanmar | 0.40 | ✓ | - | - |
| Afghanistan | 0.34 | ✓ | - | - |
| Nigeria | 0.30 | - | ✓ | - |
| Brazil | 0.28 | - | - | ✓ |
| Thailand | 0.26 | ✓ | - | - |
| Congo D.R | 0.26 | - | ✓ | - |
| Iraq | 0.20 | ✓ | - | - |
| Cambodia | 0.20 | ✓ | - | - |

These results imply that flood risk variables may be universal across nations. However, several other variables that affect flooding incidents are particular to specific geographic locations. This research aims to show how healthy ML can forecast storms in Africa by using the region as a research study.

It is well known that floods have various adverse effects, including the destruction of homes, livelihoods, and other assets and extensive financial losses up to tens of millions of dollars. For example, in 2022, floods in Pakistan harmed around 33 million people, causing the deaths of more than 1,700 people and injuring another 13,000[10]. According to the authorities, severe flooding reportedly costs $30 billion. Similar severe floods occurred in Bangladesh in 2022, affecting around 7.2 million people, killing roughly 12, and causing $722.24 million in total losses [11]. In the same year, nearly 2.5 million Nigerians were impacted by floods, with 600 dead and 2,400 wounded. The literature contains many references to past flooding incidents and estimations of their effects [12-15]

## 2.2 Flood forecasting using MLs

Recent studies on flood risk assessment & predictions use the prognostication abilities of numerous ML algorithms that discover patterns in historical data. These techniques included DTs, RFs, Linregs, LRs, XGBoosts, KNNs, SVMs, and ANNs. They have been used in flood prediction with trustworthy outcomes.[16] apply an LR employing RS information &GIS over flood inventories built from 153 historical flood sites in Rwanda with ten predictors. According to their findings, the two characteristics out of the ten—Normalized Difference Vegetation Index (NDVI) are shown the more influence. Utilizing Area Under Curve (AUC) as the assessing measure, they offer a 79.8% predictive performance.

[17] use Markov Chain Cellular Automata (MCCA) &ANNs with optimum factors of Hiden layers =7, Activation function =7, Training Method= Backpropagation; Activation functions = 0.2, Mobility = 0.22) to create an efficient prediction system for how seasonally flooded wetlands change when the flow of the Punargbhaba Body of water in India and Bangladesh changes. Wetland predictions were performed before and after the 2017 monsoon precipitation to assess the model's accuracy. The ROC-AUC curve coefficients have been 84.7% and 86.9%.

[18] employ an ensemble of four approaches to ML algorithms. Reduced Errors Prune Tree (REPtree), RFs, and M5Ps, using the bagging approach on twelve indicators and the ROC curve as the assessment measure. The M5P device for tagging has the best performance. It produces a result with an ROC value of 0.99, a sensibility of 86.26, and an accuracy of 88.76.[19] use ANN, LRs, FRs, and AHP to estimate Bangladesh's flooding risk using 475 datasets, including independent factors.

Under the ROC Curve, the area shows that regression analysis has the best likelihood of success (86%) & forecast rate (81.7%). [20] predict the spring flooding in New Brunswick, Canada, using a regression model with four attributes: Minimum Warm, Yesterday's Heat, Moisture, and Snow in winter. The analysis has 63.7% R2, and all features are statistically meaningful.

Flood modelling & based on typical data is generally limited by available information, size and quality, goals, and research scope. Our research investigates numerous learning models to assess their applicability and effectiveness. We provide an exploratory method for choosing the best suitable probability distribution function to represent the meteorological data. Our findings show the best model's prediction classification performance and effectiveness variances across techniques.

**Table 1: Summary of the related works**

| Study | ML Algorithms Used | Key Findings | Drawbacks |
|---|---|---|---|
| [16] | DTs, RFs, Linregs, LRs, XGBoosts, KNNs, SVMs, ANNs | Used LR with RS data & GIS over 153 historical flood sites in Rwanda. NDVI was a significant predictor. Achieved a 79.8% predictive performance (AUC). | Limited to specific geographic area (Rwanda) and may not generalize well to other regions. The use of ten predictors might add complexity and require extensive data. |
| [17] | MCCA, ANNs | Created a prediction system for seasonally flooded wetlands based on Punargbhaba River flow changes in India and Bangladesh. ROC-AUC curve coefficients were 84.7% and 86.9%. | May rely heavily on data availability and quality related to river flow changes. Results might not be directly applicable to different wetland ecosystems. |
| [18] | REPtree, RFs, M5Ps | Employed an ensemble of ML algorithms on twelve indicators to predict flooding. M5P tagging had the best performance with ROC value of 0.99, sensitivity of 86.26, and accuracy of 88.76. | The ensemble approach can be computationally intensive, especially with multiple algorithms. Limited discussion on potential overfitting concerns. |
| [19] | ANN, LRs, FRs, AHP | Estimated flooding risk in Bangladesh using 475 datasets. Regression analysis had the best performance with an ROC likelihood of success (86%) and forecast rate (81.7%). | May rely on the availability and quality of the extensive dataset, which can limit generalizability to other regions. Complexity of using multiple algorithms. |
| [20] | Regression Model | Predicted spring flooding in New Brunswick, Canada using four attributes. Achieved a 63.7% R2 with statistically meaningful features. | Limited to a specific geographic area (New Brunswick) and may not be directly applicable to other regions or global contexts. Limited to spring flooding prediction. |

**Research Gaps Identified:**

While considerable research has been conducted on flood vulnerability assessments using various Machine Learning (ML) models, there exists a notable research gap that this study aims to address:

- Existing studies predominantly focus on individual or a limited number of ML algorithms for flood vulnerability assessments. There is a lack of comprehensive research that systematically evaluates and compares the performance of a diverse set of ML models.
- Previous research often lacks an integrative approach, either focusing solely on meteorological factors or neglecting the crucial interplay between meteorological, hydrodynamic, and geographical information.
- While ML models are increasingly being used for flood vulnerability assessments, there is limited research on the practical challenges related to user adoption of these models in real-world scenarios.

By addressing these gaps, the proposed study endeavors to advance the current state of knowledge in flood vulnerability assessments using ML models, offering insights that can contribute to improved risk mitigation strategies and more effective decision-making in the context of flood management and resilience.

## 3. Proposed model building for flood prediction

Learning methods frequently utilized in prediction models build an entire knowledge analysis tool by modelling biological neurons' composition, operation, and behaviour.The following are the procedures followed in predicting rainfall and, as a result, detecting the possibility of flooding in the Cuddalore region.

### 3.1 Pre-processing of data

Unbalanced characteristics exist in the information gathering. Consider the percentage of 74:26, which should be about equal between the current day's rainfall and the anticipated precipitation for the following day, as shown in Fig 1. It demonstrates that the information is unbalanced—quality inputs balance information. The following resampling in Fig 2 displays the value systems. Furthermore, an oversampling of the data is seen in the given dataset. The resampled information is produced when matching the large dataset, as shown in Fig 3. The detected characteristics include wind speed, humidity, lowest temperature, and precipitation, and the fraction of missing information is nearly less than 50%. From the research, attributes with oversampled information are crucial to modelling and forecasting; thus, the imputation of the dataset made a positive impact.

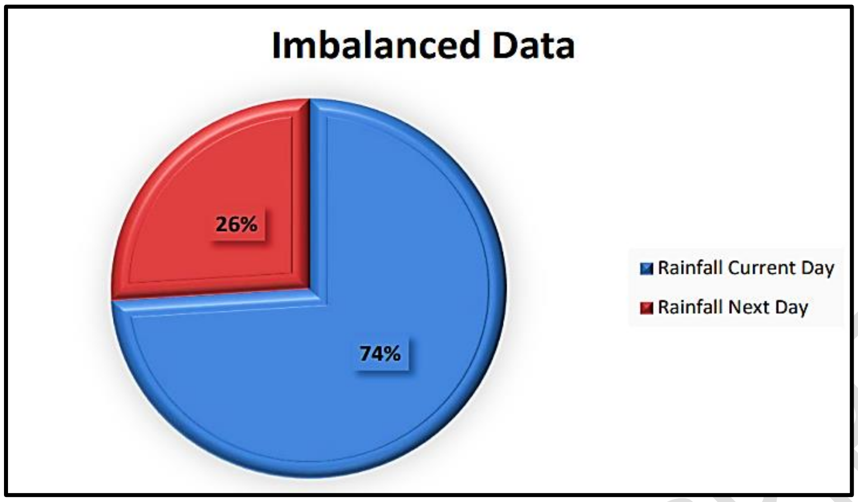


**Figure 1:Imbalanced Data**

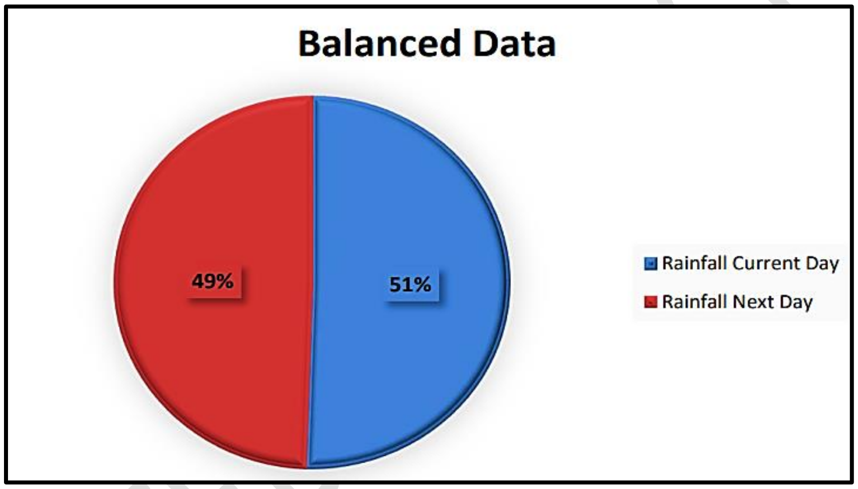


**Figure 2:Balanced Data**

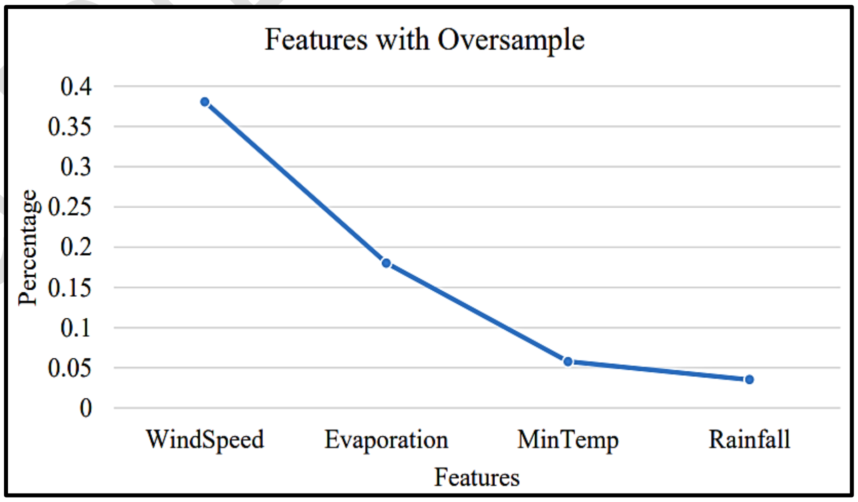


**Figure 3:Oversampled Features**

## 3.2 Imputation of Data using Multiple Imputation by ChainedEquations approach and Outlier Detection and FeatureSelection

After calculating the mode to impute the categorical variables, the label encoding technique transforms the structured variables into numbers. The lacking values are determined using Multiple Imputation by Chained Equations (MICE). The MICE method finds outliers using the median value and removes them to produce the datasets required to create the model. The variables' relationship is then calculated, and the highest-correlating pair of different factors is found. One of the strongly associated variables will then be removed. Date, position, and wind patterns are the dataset's category values. By using a label encoding method, these traits are transformed continuously. The quantity values are used to calculate the outliers, which are then taken from the data. Fig. 4 depicts the flow of the proposed model. Table 2 displays outliers considered in the research.

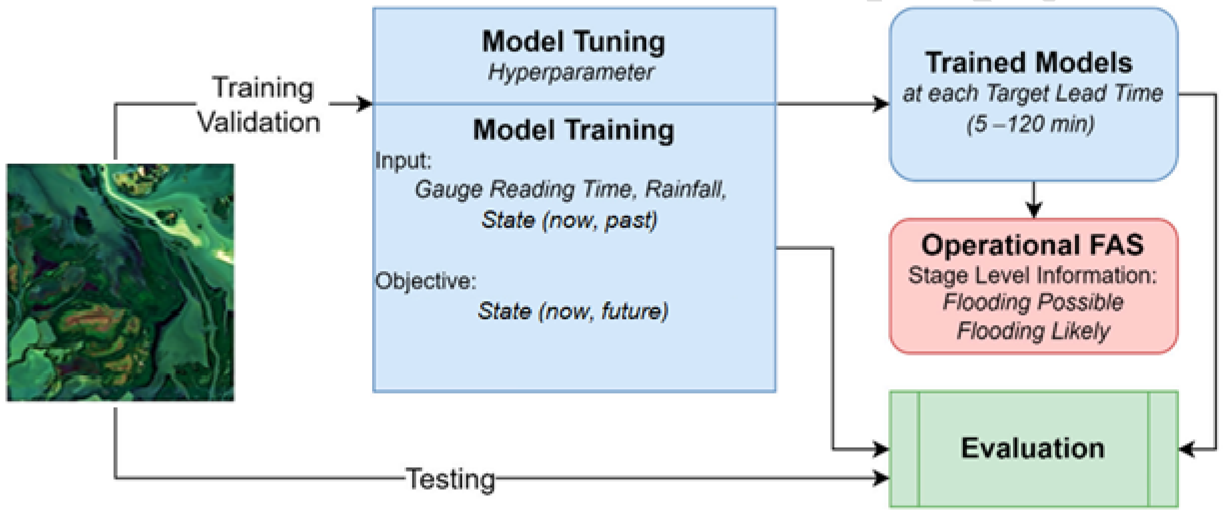


**Figure 4: Taxonomy of the proposed model**

**Table 2: Outlier Detection**

| Attributes | Values |
|---|---|
| Date | 1978 |
| Position | 1.01 |
| Low temperature | 5.51 |
| High Temperature | 6.01 |
| Rainfall | 0.0 |
| Evaporation | 2.11 |
| WindGustDir | 8.01 |
| WindGustSpeed | 2.21 |
| Wind Direction | 8.01 |
| Wind Speed | 3.64 |
| Humidity | 25.01 |
| Temp | 6.01 |

|  |  |
|---|---|
| Rain Current Day | 0.0 |
| Rain Next Day | 1.01 |

The original data has been scaled down to 21736 records and fourteen attributes. There are around 40,000 entries removed, and the information is clear of anomalies. The association between the traits is to be discovered as a subsequent step. Fig 5 depicts an overall flow of the proposed method; this phenomenon is called multi-collinearity. The characteristics are all considered while developing the model since the weather map demonstrates their independence and lack of meaningful association. A machine learning model's data selection is vital. The suggested technique uses the random forest to ascertain a feature's significance. Chi-Square filters normalized information, and MinMaxScalar reduces negative numbers. The characteristics seen as being of high value in predicting rainfall from the built-in model are as follows. The feature is chosen in opposition to the next-day rainfall feature. It was found that the lowest warmth, highest temperature, absorption, wind direction, and humidity all significantly affected the likelihood of precipitation.

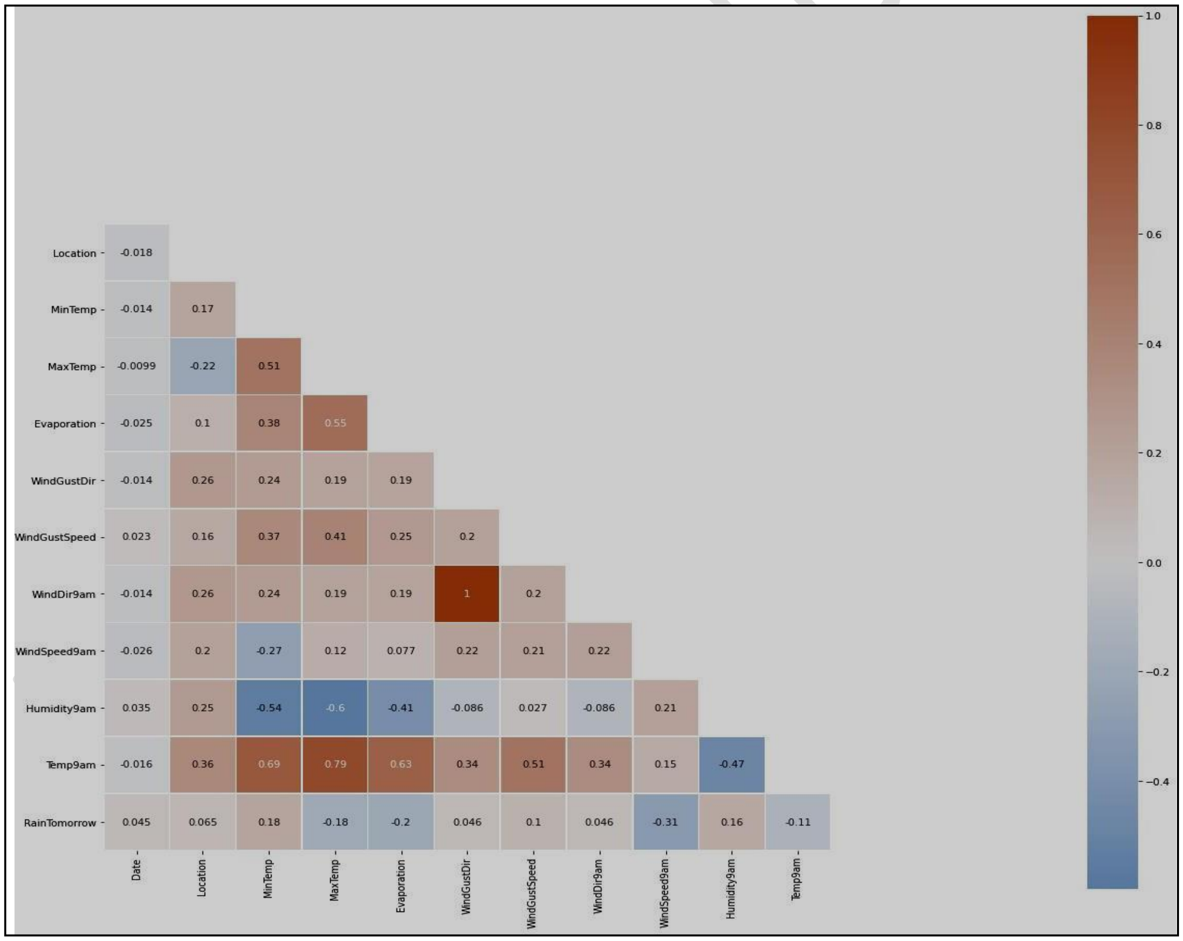


**Figure 5: Features Multi-collinearity**

XG boost is the efficient gradient-boosting decision tree approach known as extreme gradient boosting, which can be used to calculate index weights (XGBoost). One can predict the score based on the features of the sample. When training is done, there are no trees. When a tree falls, its leaf node, which represents a score, will be reached. The matching scores out of each tree are then summed to get the sample's anticipated value. There are significant relationships between subsequent decision trees in the XGBoost algorithm. The forecast accuracy is substantially increased because each round forecast is built based on the forecast mistake from the previous round. In contrast to conventional predictive methods, it can choose a default branch direction for missing data, minimizing the associated error. Moreover, it can handle category and numerical data, increasing the model's predictability.

**3.3 Improved XGBoost Algorithm UsingCBPSO OperatorXGBoost Algorithm**

An integrated model using a decision tree is called XGBoost. For every specific n-sample training data set,

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \tag{1}$$

$$(|D| = n, x_i \in R^m, y_i \in R) \tag{2}$$

For predicting the output, the framework for integrating the tree uses operations built from K-stacked decision trees. The equation reads as follows:

$$\hat{y} = \sum_{k=1}^{K} f_k(x_i), f_k \in F, \tag{3}$$

Where the instance region of every classification tree is represented by F.

To discover the optimum model y and consider overfitting's effect on prediction performance, the following optimization techniques are developed and reduced:

$$\begin{aligned} \xi &= \sum_{i}^{n} \left(l(y_i, \hat{y}_i)\right) + \sum_{k=1}^{K} \Omega(f_k), \Omega(f) \\ &= \alpha * N + \frac{\beta * \| w \|}{2} \end{aligned} \tag{4}$$

where i is a specified interpreted in many ways to compute the discrepancy between the predicted value yi and what occurs, N be the No. Of networks in the classification trees, w =($w_1$,$w_2$,...,$w_n$) is the mass of every node, and (f) is the penalty component that penalizes the number of hidden layers, including the quantity of intermediate node and their weights.

The equation (4) represents an optimization problem designed to discover the optimum model (y) while taking into account the impact of overfitting on prediction performance. The optimization involves minimizing a composite objective function, ξ, which is the sum of two components:

The first component $(\sum_{i}^{n}\ (l(y_i, \hat{y}_i))$represents the loss function, where "l" is a function that measures the discrepancy between the predicted value (ŷ_i) and the actual value (y_i) for each data point i. This sum is taken over all data points in the dataset (from i=1 to n).

The second component $(\sum_{k=1}^{K}\ \Omega(f_k), \Omega(f))$ is a regularization term that penalizes the complexity of the model to prevent overfitting. Here, Ω(f) is a regularization function applied to each tree (f_k) in the model. The regularization function Ω(f) consists of two terms:

The first term (α*N) penalizes the number of networks (N) in the classification trees, where α is a regularization parameter.

The second term (β*‖w‖/2) penalizes the complexity of each tree's structure, where ‖w‖ is the norm of the weights associated with the nodes, and β is another regularization parameter.
In summary, the optimization problem aims to find the optimal model by balancing the trade-off between minimizing the prediction error (captured by the loss function) and preventing overfitting (controlled by the regularization terms). The regularization terms penalize the complexity of the model by considering the number of networks, as well as the weights and structure of each tree in the ensemble. The values of the regularization parameters (α and β) play a crucial role in determining the strength of the regularization applied to the model.

The integrative trees paradigm represented by equations (6) can be computed using traditional objective functions in Euclidean distance. Thus, we use iterative approximating and designate the first used created by t repetitions with $y_i((t))$.
:

$$\xi^{(t+1)} = \sum_{i}^{n} \left(l\left(y_i, y_i 2^{(t)}\right) + f_{t+1}(x_i)\right) + \Omega(f_{t+1}). \qquad (5)$$

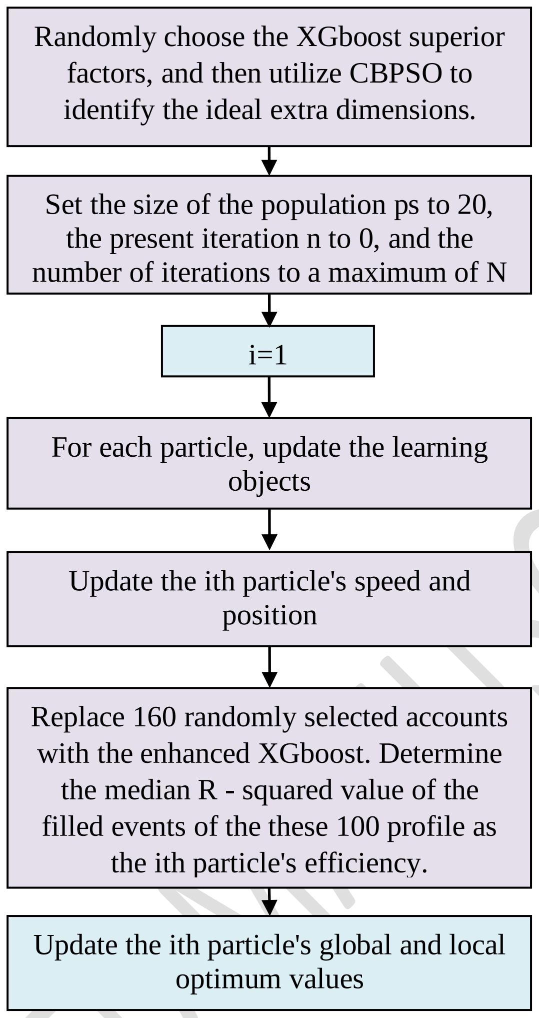

Randomly choose the XGboost superior factors, and then utilize CBPSO to identify the ideal extra dimensions.
Set the size of the population ps to 20, the present iteration n to 0, and the number of iterations to a maximum of N
i=1
For each particle, update the learning objects
Update the ith particle's speed and position
Replace 160 randomly selected accounts with the enhanced XGboost. Determine the median R - squared value of the filled events of the these 100 profile as the ith particle's efficiency.
Update the ith particle's global and local optimum values

**Figure 6: CBPSO working flow**

Fig 6 shows the flow of the anticipated model where the number of populations and the learning objects are provided. The particle updation is discussed with the integration of XGBoost. In this workflow, XGBoost hyperparameters are initially optimized to identify the best-performing configuration randomly. Subsequently, a Particle Swarm Optimization algorithm (CBPSO) is employed with a population size of 20. For each particle in the population, its speed and position are updated iteratively to explore the search space. The improvement in performance is measured by replacing 160 randomly selected XGBoost configurations with the enhanced XGBoost model. The median R-squared value is computed from these 100 profile events, serving as the particle's efficiency. Both the global and local optimum values are updated during this process. This workflow combines random hyperparameter selection and PSO optimization to find the most effective XGBoost configuration and further fine-tune it using CBPSO to achieve optimal predictive performance.The model's forecasts are incrementally enhanced via iterations by maximizing the equation above. By performing a Taylor series expansion at the point

$\left(\left(y_1^{(t)}, y_2^{(t)}, \dots, y_n^{(t)}\right)\right)$ of equation (6), the best solution to the problem above may be found. This expression is as follows:

$$\xi^{(t+1)} = \sum_{i}^{n} \left(l\left(y_i, y_i 2^{(t)}\right) + \partial_{\grave{y}^{(t)}} l\left(y_i, \grave{y}^{(t)}\right)\right) + f_{t+1}(x_i) + \frac{1}{2}\partial^2 \grave{y}^m l\left(y_i, \grave{y}^{(t)}\right) = f_{t+1}^2(x_i) + \Omega(f_{t+1}). \qquad (6)$$

By subtracting $\partial_{\frac{Y}{r}(0)} l\left(y_i, \grave{y}^{(t)}\right)$ and $\partial^2_{\grave{y}^2)} l\left(y_i, \grave{y}^{(t)}\right)$and factoring in eq (7), we may deduce, for every DT function $f_{t+1}(x)$, the continuing to follow: where $I_i = \{i \mid q(x_i) = j\}$ provides all input multiple for the source vertex in the set of training data while q denotes the regression analysis formulation for the clustered method parameter f (t+1) (x).In the t+1-th iteration, $\xi^{t+1}$may be utilized to compute the significant decision tree.Fig 7 shows the hidden layers of the network model where single input and single output is extracted. The weights of the hidden layers are measured with $w_n$ where $n = 1,2, \dots$

In a neural network with two concealed units, typically referring to a network with two hidden layers, the first hidden layer contains two neurons or nodes, and the second hidden layer also comprises two neurons. These concealed units perform computations on the input data and pass their results to subsequent layers. The choice of having two hidden layers and two units in each layer is a design decision that impacts the network's capacity to learn and represent complex patterns in the data, and it's often determined through experimentation to optimize performance on a specific task.

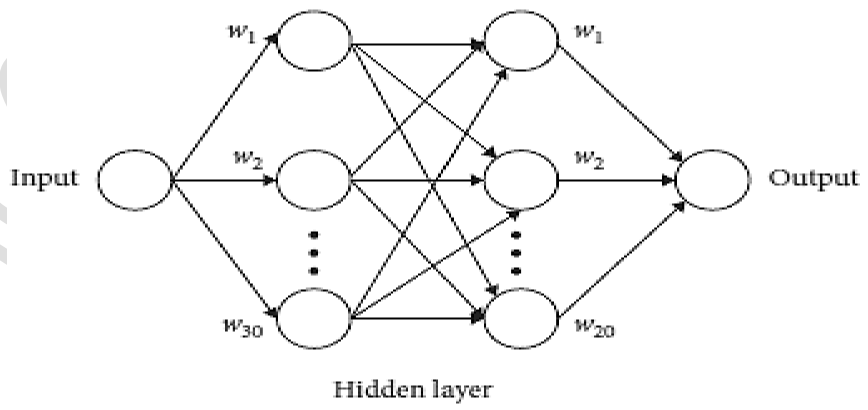


**Figure 7: The layout of the two concealed units with NNs.**

XGBoost adds a regular component to the goal functions that comprise the number of the source nodes and the square summation of the components of weights for every tree-like structure to reduce training error. W. With XGBoost, column controlled digitally speeds up computation and avoids generalizations. With each repetition, XGBoost raises the leaf vertex strength by a parameter to reduce the influence of each tree, providing further learning space in the final phases.One of the most extended processes in learning the prediction model is sorting the feature values, and the XGBoost program allows multitasking. However, before realizing it, XGBoost organizes the information and stores it in a block structure to simplify computation. , The blocking approach allows concurrent analysis of feature gains while separating nodes. The element with the most significant improvement is ultimately chosen for split.

## 3.4 Choosing Superparameters

The impact of the method's findings is strongly tied to the choice of XGBoost's superparameters. The grid search strategy is the conventional approach for altering characteristics, and XGBoost requires a change of nine superparameters. The variable selection approach divides superparameters into squares in a specified space and searches all panel positions for the best values. This strategy may find the worldwide optimal solution when the optimal intervals are large, and the step length is short. Nevertheless, since the classification results among most super parameter sets in the grid are relatively poor, as well as the decision tree classifiers only in a minimal interval are rather good, all parametric groups in the traverse squares are readily susceptible to slipping into global optimum, resulting in an enormous wasted effort.To resolve the issue that the traditional gradient boosting strategy has the propensity to settle into the localized optimization technique readily and to enhance computing efficiency, we applied the CBPSO method to improve the selection of super parameters.

## 3.5 CBPSO Operator

The CBPSOs may be stated as a D-dimension minimal optimizer method:

$$\mathrm{m}{*}\,\mathrm{F}(\mathrm{x}), 2\mathrm{x} = \mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_D, \mathrm{x} \in \{\mathrm{x}_{\min}, \mathrm{x}_{\max}\}. \tag{7}$$

Concatenated boosting CBPSO, an enhanced variant of the classifier PSO technique, is referred to as this. It improves particle-to-particle interaction and quickens population confluence. It corrects the original PSO algorithm's flaw: it was prone to falling into the local optimal solution. Formulas to modify the classic PSO individual's velocity & location are:

$$\begin{aligned} V_i^d = &\quad V_i^d + c_1 * r\,\&(0,1) * \left(\text{pbest}\,_i^d - X_i^d\right) \\ &\quad + c_2 * r\&\,(0,1) * \left(\text{g best}^{\,d} - X_i^d\right), \\ X_i^d = &\quad X_i^d + V_i^d. \end{aligned} \tag{8}$$

where p best$_i$= {p best$_1$, . . , p best$_d$}denotes the historical optimized value of the i$^{th}$ particle and g best = ( gbest$^1$, … , gbest$^B$} is the global optimized value of every particle. Eq 8 & 9 indicate that every PSO improves itself from historic and globally desired conditions in each learning phase to improve efficiency.

The CBPSO method alters the PSO speed formula (11), so the training item may learn from nanoparticles in numerous levels and periods. The revised equation is as follows:

$$V_i^d = V_t^d + c * \text{rand}\ (0,1) * \left( \text{pest}\ {}^{d}_{fi(d)} - X_i^d \right). \qquad (9)$$

where $f_i = \{f_i(1), f_i(2), \dots, f_i(D)\}$ represent the fact of particles i need to gain understanding from the historically ideal value of particles $f_i(d)$ in dimensions D.

## 3.6 XGBoost technique optimization vsCBPSO operator

A three-step method is involved in optimizing CBPSO:

- The first step is Setting the CBPSO algorithm and starting the particle swarm.
- In the second approach, after starting, reboot the momentum and location of every nanoparticle, use the ongoing coordinates as the XGBoost superparameters, run the experimentation, use the experiences a sensation as the particle's optimal solution, and modify its spatial and global optimal values depending on its fitness value.
- Repeat the second step N times in step third.

XGBoost's nine superparameters are the number of estimations: n, training error, shallow distance, sampling ratios, the summation of sampling intensities of minimum tree structure (min child weight), the decreasing amount of loss function, and L1 upsampling. Lastly, we use CBPSO and three cross-verifications to get the optimum superparameters.

**Table 3: XGBoost Algorithm's last superparameter.**

| **Factors** | **Limits** |
|---|---|
| Rate of Learning | 0.113 |
| Maximum Depths | 12 |
| Subsampling | 0.73 |
| Child Weights are minimum | 0.56 |
| Gamma | 0.218 |
| Alpha Registry | 0.146 |
| lambda Registry | 0.6 |
| BytreeColsample | 0.85 |

## 4. Results and discussions

Indicators of model correctness include Acc, Precision, Recall, F1-Score & IoU, without IoU providing as the primary statistic. The following defines these statistics:

$$\text{Accuracy} = \frac{\text{TPs} + \text{TNs}}{\text{TPs} + \text{TNs} + \text{FPs} + \text{FNs}} \qquad (10)$$

Equation 10: Accuracy measures the overall correctness of a classification model. It calculates the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances in the dataset.

True Positives (TPs) are the instances correctly predicted as positive.
True Negatives (TNs) are the instances correctly predicted as negative.
False Positives (FPs) are the instances incorrectly predicted as positive.
False Negatives (FNs) are the instances incorrectly predicted as negative.
Accuracy tells us how well the model predicts both positive and negative classes.

$$percision = \frac{TPs}{TPs + FPs} \tag{11}$$

Equation 11: Precision is a measure of the model's ability to correctly predict positive instances without falsely classifying negative instances as positive.
It calculates the ratio of true positives to the total predicted positive instances (true positives + false positives).
Precision is useful when the cost of false positives is high.

$$Recall = \frac{TPs}{TPs + FNs} \tag{12}$$

Equation 12: Recall, also known as Sensitivity or True Positive Rate, measures the model's ability to identify all positive instances correctly.
It calculates the ratio of true positives to the total actual positive instances (true positives + false negatives).
Recall is important when missing positive instances is costly or unacceptable.

$$F1 - Score = \frac{2 * Percision * Recall}{Percision + Recall} \tag{13}$$

Equation 13: The F1-Score is the harmonic mean of precision and recall. It provides a balance between these two metrics.
It's particularly useful when there is an imbalance between the two classes (e.g., one class has many more instances than the other).
A higher F1-Score indicates a better balance between precision and recall.

$$IoU = \frac{GroundTruth \cap Predicted}{GroundTruth \cup Predicted} \tag{14}$$

Equation 14: IoU is commonly used in object detection and segmentation tasks to evaluate the overlap between predicted and ground truth regions.

It calculates the ratio of the intersection of the predicted region and the ground truth region to the union of these regions.
IoU ranges from 0 to 1, with higher values indicating better overlap between the prediction and ground truth.

Accuracy is depicted as the ratio of TP and TN to TP, TN, FP and FN. The precision is the ratio of true positive to true positive and false positive. The recall is depicted as the ratio of true positive to the true positive and false negative. The F1-score is depicted as the ratio of precision and recall to the sum of precision and recall. Finally, IoU is depicted as the ratio of concatenated ground true and predicted value to the union of ground truth and prediction value.

**4.1 Implementation details**
All tests were done with PyTorch, Python 3.7, and MM Segmentation. Four NVIDIA Tesla P100 integrated graphics were used for the feature learning. Similar equipment was utilized for the analysis. CentOS-7.9 was used as the OS. Networks were validated for 30 periods, with periods 20 & 25 seeing several iterations enhanced by 0.1 and period 30 being kept for assessment. The sample size was 96 for the single-stream algorithms, and the beginning training data was set at 0.001. The dual-stream approaches' sampling size was 16, and the starting training error was 0.0002. Dice losses and cross-entropy were used, and the combination of the two was considered the losses.The proposed model performs 5-fold cross validation where the validation accuracy is 98% and validation loss is 2%. The samples are partitioned in 70:20:10 ratio where 10% of samples are provided for validation purpose over the dataset https://www.kaggle.com/datasets/virajkadam/sen12flood.

**4.2 Methods of Comparison**
Semantic frameworks are ideally suited to various remotely sensed applications, such as land covering categorization, flood diagnosis & edge detection. It makes sense to incorporate construction made for text categorization into computer vision frameworks. For instance, the ASPP module—the essential part of DeepLabv3+—was used by [21] to improve feature modelling capacity. U-shaped convolution network with a fundamental topology similar to D-LinkNet was suggested for activity recognition. We examined CBPSO to four commonly used feature extraction concepts: DeepLabv3+, PSPNet, OCRNet, and D-LinkNet to demonstrate the usefulness of function multi-modal temporal lobe information fusion. We used MCANet and CMGFNet to show our neural network-based approach's advantage in multi-modal analysis software.The newest model in the DeepLab line is DeepLabv3+ [22]. Several land cover categorization and change detection investigations relied on DeepLabv3+. Expanding the region of interest and strengthening multi-scale feature interactions, PSPNet, like DeepLabv3+, increases model accuracy. OCRNet, a more current feature extraction method than DeepLabv3+ and PSPNet [23], uses self-attention components to construct the pixel-object association and beats DeepLabv3+ on most benchmark problems. One of the most often used deep learning approaches for analyzing spatial patterns via satellite imagery is D-LinkNet [24]. It was initially designed for road separation but has been extensively used for water body separation and other

land-covering data mining algorithms.The other two fully convolutional models, MCANet and CMGFNet, reflect more recent developments in inter-land cover research and were used for comparability. To improve complementary interactions across various methods, these networks, which are dual-stream approaches, combine extracted features. Table 3 provides the super parameters.

ResNet-50 was employed as the backbone by all methods beyond CBPSO. The 2 test pictures' predictions findings are given in Fig 8. The figures show deep learning-based disaster identification may be employed when floods occur regularly. Lakes, interior river plain communities, and estuaries are good places to look for flood regions. In this investigation, the CBPSO had the most outstanding results (Acc = 97.28, Precision = 94.17, Recall = 96.99, F1-Score = 95.51 & IoU = 88.85).The flood-extracting characteristics under various techniques are compared. CBPSO can produce sharper outlines and is more aligned with the actual data. Due to the more significant memory requirements that dual-stream algorithms use during learning, we used a tiny sample size of 16 & a lower learning period rate of 0.0002.

To assess CBPSO's performance more thoroughly, we developed a brand-new DeepLabv3 + s model based on DeepLabv3+ and used the same packet size and starting learning algorithm as CBPSO. Table 4 displays the test accuracy for DeepLabv3 + s, with an IoU metric equivalent to DeepLabv3(IoU +'s = 87.58). CBPSO's greater accuracy on the testing dataset is not due to a tiny sample size or lower learning period rate, according to DeepLabv3 + s. In multi-modal flood recognition, CBPSO outperformed CMGFNet and MCANet. Moreover, CBPSO outperformed its base technique, D-LinkNet (IoU = 86.85), demonstrating the viability of our suggested modifications.The network complexity shows some variation during the training process with respect to number of epochs. The proposed CBPSO (Chaos-based Particle Swarm Optimization) method achieves an accuracy of 97.28%, which is higher than several existing models, but slightly lower than DeepLabv3+ at 97.53%. Deep learning models often have a degree of randomness in their training due to factors like weight initialization and stochastic optimization algorithms. It's possible that the proposed CBPSO converged to a suboptimal solution during training, whereas DeepLabv3+ may have found a slightly better configuration in its random initialization. And also the performance of deep learning models can be influenced by the dataset they are trained on. If the proposed CBPSO model was trained on a dataset that is slightly different from the one used for benchmarking the other models, it could result in variations in accuracy.

**Table 4: Evaluation of the algorithmic accuracy**

| **Techniques** | **Accuracy in %** | **Precision in %** | **Recall in %** | **F1-score in %** | **IoU in %** |
|---|---|---|---|---|---|
| DeepLabv3+ [2] | 97.53 | 92.03 | 96.71 | 94.17 | 88.59 |
| DeepLabv3+CNN [8] | 96.51 | 91.07 | 95.49 | 93.11 | 87.48 |

| | | | | | |
|---|---|---|---|---|---|
| PSPNet [9] | 96.49 | 91.03 | 95.45 | 93.06 | 87.40 |
| OCRNet [11] | 96.13 | 90.04 | 96.39 | 92.45 | 86.40 |
| D-LinkNet [12] | 96.26 | 90.20 | 96.89 | 92.73 | 86.86 |
| CMGFNet [13] | 96.54 | 90.99 | 96.92 | 93.20 | 88.62 |
| MCANet [15] | 96.54 | 90.93 | 96.99 | 94.56 | 88.64 |
| CBPSO (Proposed) | **97.28** | **93.17** | **97.98** | **95.52** | **89.85** |

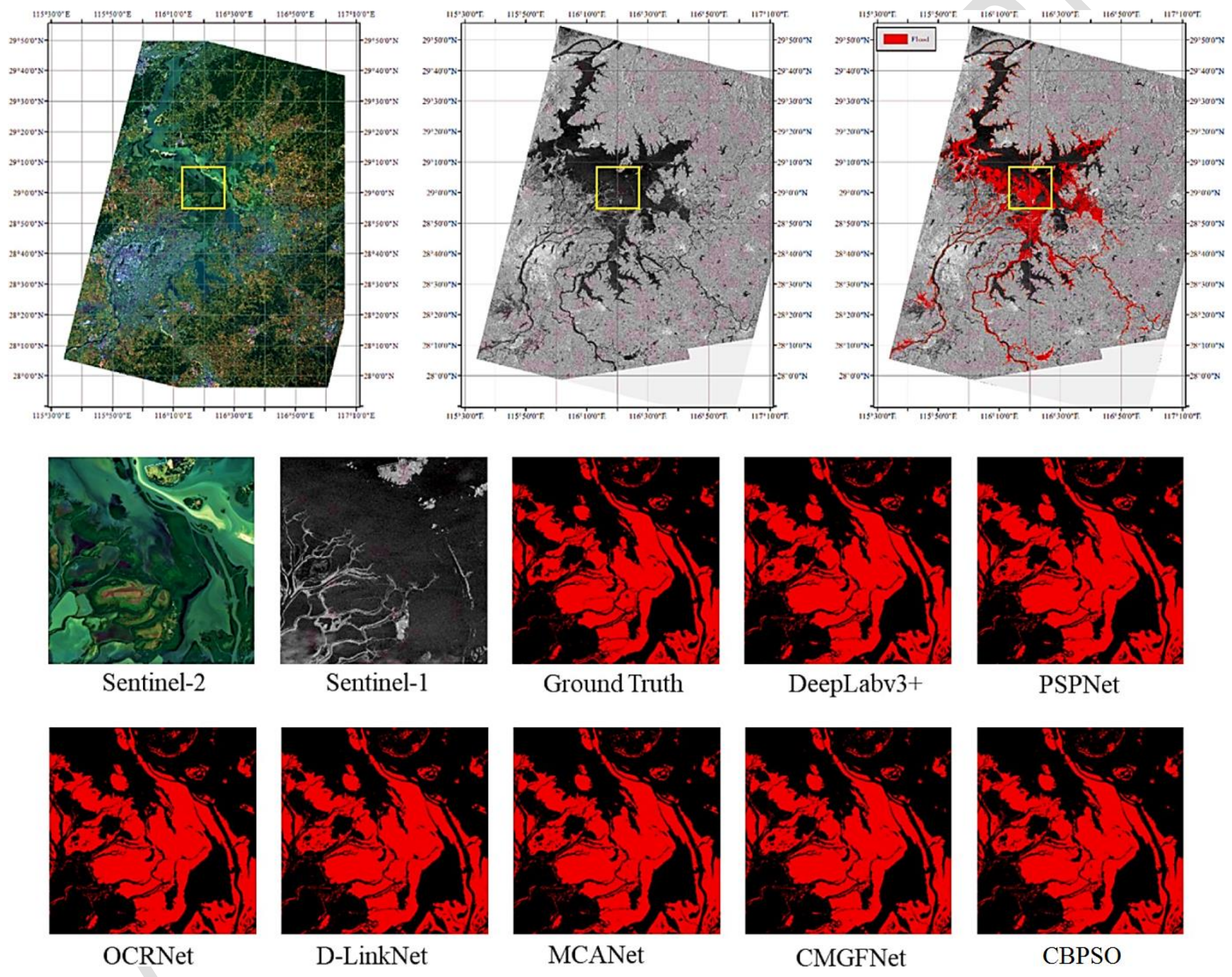


**Figure 8: Outcomes of the estuary region of Australia's Wilsons Rivers Basin's testing location roi4's predictions.**

| Hyperparameter Tuning for the proposed Urban Flood Detection |
|---|
| 1. Model Selection: |

```
  - Chosen Model: XGBoost (eXtreme Gradient Boosting)

2. Dataset Preparation:
  - Urban features, SAR data, meteorological data, and historical flood records are used.
  - Split dataset into training and validation sets.

3. Baseline Model Configuration:
  - Initial configuration with default hyperparameters.

4. Grid Search Cross-Validation:
  - Explore a range of hyperparameters systematically:
    - Learning rate: [0.01, 0.1, 0.2]
    - Maximum depth of trees: [3, 5, 7]
    - Number of trees (boosting rounds): [50, 100, 150]
    - Minimum child weight: [1, 3, 5]
    - Subsample ratio: [0.6, 0.8, 1.0]
    - Column subsample ratio: [0.6, 0.8, 1.0]
    - Gamma (minimum loss reduction): [0, 0.1, 0.2]
    - Regularization terms (alpha, lambda): [0, 1, 2]
    - Sampling method: [stratified, uniform]

5. Cross-Validation Setup:
  - K-fold cross-validation (e.g., 5 folds).

6. Performance Metrics:
  - Evaluate metrics: accuracy, precision, recall, F1-score, ROC-AUC.

7. Iterative Tuning Process:
  - Refine based on grid search results iteratively.

8. Randomized Search (Optional):
  - Randomly sample hyperparameter combinations.

9. Final Model Selection:
  - Choose hyperparameters optimizing the selected metric.

10. Model Evaluation:
   - Evaluate final model on a separate test dataset.

11. Model Deployment (Optional):
   - Deploy for real-time urban flood detection.
```

This is the tuning of hyperparameters of the proposed model which is optimized for efficient urban flood detection over vulnerable zones. The chosen hyper-parameter values lead to improved performance metrics, ensuring the model's reliability in real-world scenarios.

**4.3 Complexity issues**

The Extreme Gradient Boosting (XGBoost) algorithm exhibits specific complexity considerations. In terms of time complexity, XGBoost builds an ensemble of decision trees iteratively, and the overall complexity depends on the number of trees, their depth, and the complexity of the weak learner used. Memory complexity is influenced by the dataset size, the number of features, and the number of trees, as each tree requires memory for storage. Scalability is determined by the dataset size and the available computational resources, as larger datasets and complex problems may require more time and memory. Additionally, XGBoost involves tuning hyperparameters, which introduces complexity in finding optimal parameter settings. Understanding these complexity issues helps evaluate the feasibility and practicality of using XGBoost for large-scale or resource-constrained applications.

The time complexity of training each individual base model (usually a decision tree) is influenced by factors such as the number of features (m), the number of samples (n), and the depth of the trees (d).
For decision trees, the typical time complexity is O(m * n * log(n) * d).
The complexity is affected by tree-specific optimizations, such as column blockings and tree pruning.

**5. Conclusion**

This study proposes improving the XGBoost algorithm using Concatenated Boosting Particle Swarm Optimization (CBPSO). CBPSO demonstrates strong capabilities in overcoming local optima and premature convergence, thus enhancing the convergence of the population and particle training. This research uses CBPSO to fill in missing data and adjust the atmospheric reflection index over the oceans to evaluate its effectiveness. The experiment utilizes high-resolution sounding balloon data to calculate the adjusted ambient optical properties at various points. The output comprises the adjusted atmospheric refractive indices of the middle layer (100-4500 m), while the input consists of the refractive index in the lower levels (approximately 100 m to 4500 m). To train the enhanced XGBoost algorithm, missing data for the modified index of refraction within the middle layer (100 m–4500 m) are filled in. Due to the scarcity of large-scale remote sensing data specifically related to multispectral flood risk assessment, this study successfully addresses this challenge using CAU-Flood, which also serves as testing data for future research in this domain. The authors extensively analyze various state-of-the-art (SOTA) techniques in urban flood detection, and the testing results demonstrate that the proposed algorithm achieves higher recognition accuracy than other algorithms.

However, it should be noted that the suggested algorithm does not account for weather-related physical phenomena and merely fills in values for the modified air refractive index. Future research endeavours will incorporate these considerations to enhance the suggested method further and improve the accuracy of filling in missing values for the adjusted atmospheric refractive index.

**Nomenclature**

| S.No. | Abbrevation | Description |
|---|---|---|
| 1 | XGBOOST | Extreme Gradient Boosting |
| 2 | CBPSO | Concatenated Boosting Particle Swarm Optimization |
| 3 | BP | Backpropagation |
| 4 | DT | Decision Trees |
| 5 | RF | Random Forest |
| 6 | SVM | Support Vector Machines |
| 7 | ANN | Artificial Neural Networks |
| 8 | NB | Navie Bayes |
| 9 | LR | Logistic Regression |
| 10 | FR | Feature Ranking |
| 11 | MCDA | Multi-Criteria Decision-Making |
| 12 | MLP-NN | Multi-Layer Perceptron Neural Network |
| 13 | EM | Expectation-Maximization |
| 14 | PCA | Principalcomponent Analysis |
| 15 | AUC | Area Under Curve |
| 16 | MCCA | Markov Chain Cellular Automata |
| 17 | REPtree | Reduced Errors Prune Tree |
| 18 | MICE | Multiple Imputation By Chained Equations |
| 19 | TP | True Positive |
| 20 | TN | True Negative |
| 21 | FP | False Positive |
| 22 | FN | False Negative |

**References**

1. Sudha Rani N, Satyanarayana A, Bhaskaran PK (2015) Coastal vulnerability assessment studies over India: a review. Natural Hazards 77(1):405−428.
2. Rehman S, Sahana M, Hong H, (2015) A systematic review on approaches and methods used for flood vulnerability assessment: a framework for future research. Natural Hazards 96(2):975−998.
3. Bhuiyan SR, Al Baky A (2014) Digital elevation based flood hazard and vulnerability study at various return periods in sirajganj Sadar upazila, Bangladesh. International journal of disaster risk reduction 10:48−58.

4. Coca N (2020) Flooded Asia: Climate change hits region the hardest. Nikkei Asia. https://asia.nikkei. com/Spotlight/Asia-Insight/Flooded-Asia-Climate-change-hits-region-the-hardest.
5. Preethi, P., & Asokan, R. (2021). Modelling LSUTE: PKE schemes for safeguarding electronic healthcare records over cloud communication environment. Wireless Personal Communications, 117(4), 2695-2711.
6. India Today (2015) India is the most flood-prone country globally. India Today Web Desk https://www.indiatoday.in/education-today/gk-current-affairs/story/ india-is-the-most-flood-prone-country-in-the-world-276553-2015-12-10.
7. Preethi, P., & Asokan, R. (2020, December). Neural network oriented roni prediction for embedding process with hex code encryption in dicom images. In Proceedings of the 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India (pp. 18-19).
8. Sowmya K, John C, Shrivastava N (2015) Urban flood vulnerability zoning of Cochin City, southwest coast of India, using remote sensing and GIS. Natural Hazards 75(2):1271−1286.
9. Kinghorn J (2017) 3 factors that make flooding in South America worse. AIR https://www.air-worldwide. Com/blog/posts/2017/4/3-factors-that-make-flooding-in-south-america-worse/.
10. Margesson R, Kronstadt A (2022) Pakistan's 2022 floods and implications for U.S. interests. Congressional Research Service (October 2022): Accessed 25 December 2022 from https://crsreports.congress. Gov/product/pdf/IF/IF12211.
11. Abdullah A, Ahmed N, Mahboob A (2022) Community-led housing recovery needs assessment: North and north-eastern flood 2022: Bangladesh. UNDP (December 2022):Accessed 25 December 2022 from https://www.undp.org/bangladesh/publicationscommunity-led-housing-recovery-needs-assessment-north-and-northeastern-flood-2022-Bangladesh
12. Levenson M (2021) Severe flooding in Guyana prompts extensive relief effort. New York Times. https://www.nytimes.com/2021/06/03/us/guyana-flooding-relief.html.
13. Taiwo E, Adinya I, Edeki S (2019) Optimal evacuation decision policies for Benue flood disaster in Nigeria.Journal of Physics: Conference Series, volume 1299, 012137 (IOP Publishing).
14. Ritorto D (2013) South American floods: Dozens died in Brazil as Mexico hit. British Broadcasting Corporation (BBC) https://www.bbc.com/news/av/world-latin-america-25514396.
15. Martini A (2020) Weatherwatch: floods across South America after heavy rain. The Guardian. https://www.theguardian.com/news/2020/feb/26/weatherwatch-floods-across-south-america-after-heavy-rain.
16. Mind'je R, Li L, Amanambu AC, Nahayo L, Nsengiyumva JB, Gasirabo A, Mindje M (2019) Flood susceptibility modelling and hazard perception in Rwanda. International journal of disaster risk reduction 38:101211.

17. Talukdar S, Pal S (2020) Modeling flood plain wetland transformation in consequences of flow alteration in Inpunarbhaba River in India and Bangladesh. Journal of Cleaner Production 261:120767.
18. Talukdar S, Ghose B, Salam R, Mahato S, Pham QB, Linh NTT, Costache R, Avand M, et al. (2020) Flood susceptibility modelling in Teesta river basin, Bangladesh using novel ensembles of bagging algorithms.Stochastic Environmental Research and Risk Assessment 34(12):2277–2300.
19. Rehman S, Sahana M, Hong H, Sajjad H, Ahmed BB (2019) A systematic review on approaches and methods for flood vulnerability assessment: a framework for future research. Natural Hazards 96(2):975–998.
20. Preethi, P., Asokan, R., Thillaiarasu, N., & Saravanan, T. (2021). An effective digit recognition model using enhanced convolutional neural network-based chaotic grey wolf optimization. Journal of Intelligent & Fuzzy Systems, 41(2), 3727-3737.
21. Khanday, A. M. U. D., Khan, Q. R., & Rabani, S. T. (2022). Ensemble Approach for Detecting COVID-19 Propaganda on Online Social Networks. Iraqi Journal of Science, 4488-4498.
22. Tripathi, K., Khan, F.A., Khanday, A.M.U.D. et al. The classification of medical and botanical data through majority voting using artificial neural network. Int. j. inf. tecnol. (2023). https://doi.org/10.1007/s41870-023-01361-0.
23. Asokan, R., & Preethi, P. (2021). Deep learning with conceptual view in meta data for content categorization. In Deep Learning Applications and Intelligent Decision Making in Engineering (pp. 176-191). IGI Global.
24. Alzubaidi, L., Bai, J., Al-Sabaawi, A., Santamaría, J., Albahri, A. S., Al-dabbagh, B. S. N., ... & Gu, Y. (2023). A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications—Journal of Big Data, 10(1), 46.
25. Roy DC, Blaschke T (2015) Spatial vulnerability assessment of floods in the coastal regions of Bangladesh. Geomatics, Natural Hazards and Risk 6(1):21–44.