Project Report

On

HIERARCHICAL FLOOD RISK ASSESSMENT USING MACHINE LEARNING ENSEMBLE

Submitted

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

in

MATHEMATICS

by

FAHIMA P.S

(Register No. SM23MAT004)

Under the Supervision of

DR.ELIZABETH RESHMA M.T

THE FEAR OF THE LORD

IS THE BEGINNING OF WISDOM

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

DEPARTMENT OF MATHEMATICS

ST. TERESA'S COLLEGE (AUTONOMOUS)

ERNAKULAM, KOCHI - 682011

APRIL 2025

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

CERTIFICATE

This is to certify that the dissertation entitled, **HIERARCHICAL FLOOD RISK ASSESSMENT USING MACHINE LEARNING ENSEMBLE** is a bonafide record of the work done by **MS.FAHIMA P.S** under my guidance as partial fulfillment of the award of the degree of **Master of Science in Mathematics** at St. Teresa's College (Autonomous), Ernakulam affiliated to Mahatma Gandhi University, Kottayam. No part of this work has been submitted for any other degree elsewhere.

Date: 02|05|2025
Place: Ernakulam

Dr.Elizabeth Reshma M.T
Assistant Professor ,
Department of Mathematics and Statistics,
St. Teresa's College(Autonomous),
Ernakulam.

Smt.Nisha Oommen
Assistant Professor and Head ,
Department of Mathematics and Statistics(SF),
St. Teresa's College(Autonomous),
Ernakulam.

External Examiners

1:..............................
Dr Lejo J Manavalan
Asst Professor
Little Flower College
Guruvayur

2: Linet..............................
Linet Roslin Antony
Assistant Professor,
Post Graduate & Research Departm
of Mathematics,
Sacred Heart College Autonomo
Chalakudy

ST. TERESA'S COLLEGE (AUTONOMOUS)
DEPARTMENT OF MATHEMATICS AND STATISTICS
★ ERNAKULAM ★

DECLARATION

I hereby declare that the work presented in this project is based on the original work done by me under the guidance of Dr.Elizabeth Reshma M.T , Assistant Professor and Head , Department of Mathematics, St. Teresa's College(Autonomous), Ernakulam and has not been included in any other project submitted previously for the award of any degree.

Ernakulam.	**FAHIMA P.S**
Date:	**SM23MAT004**

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my esteemed guide, Dr.Elizabeth Reshma M.T , for her invaluable guidance, support, and encouragement throughout the duration of this project. Her expert advice and constructive feedback were instrumental in shaping this project into its present form.
I would also like to thank my Head of Department, Ms.Nisha Oommen, for providing me with the opportunity to work on this project and for her continuous support and motivation.

Ernakulam. **FAHIMA P.S**
Date: **SM23MAT004**

Contents

CERTIFICATE ii

DECLARATION iii

ACKNOWLEDGEMENT iv

CONTENTS v

1 AN INTRODUCTION TO MACHINE LEARNING **1**

1.1 **Background and Evolution of Machine Learning** 1

1.2 **Definition and Types of Machine Learning** 2

1.2.1 **Supervised Learning** 2

1.2.2 **Unsupervised Learning** 3

1.2.3 **Reinforcement learning** 3

1.2.4 **Semi-Supervised Learning and Self-Supervised Learning** 4

1.3 **Machine Learning Workflow** 4

1.3.1 **Data Collection and Preprocessing** 5

1.3.2 **Model Selection and Training** 5

1.3.3 **Model Evaluation and Validation** 6

1.4 **Key Concepts in Machine Learning** 6

1.4.1 **Features and Target Variables** 6

1.4.2 **Regression and Classification** 7

1.4.3 **Overfitting and Underfitting** 7

1.4.4 **Hyperparameters and Model Selection** 7

1.5 **Challenges and Limitations of Machine Learning** 8

1.5.1 **Data Quality and Availability** 8

1.5.2 **Model Interpretability and Explainability** 8

1.5.3 **Adversarial Attacks and Robustness** 9

2 TREE-BASED ENSEMBLE LEARNING ALGORITHMS 10
2.1 Decision Trees 10
2.1.1 Definition of Decision Trees 10
2.1.2 How Decision Trees Work 10
2.1.3 Advantages of Decision Trees 12
2.1.4 Disadvantages of Decision Trees 12
2.2 Random Forests 13
2.2.1 Definition of Random Forests 13
2.2.2 How Random Forests Work 14
2.2.3 Advantages of Random Forests 14
2.2.4 Disadvantages of Random Forests 15
2.3 Introduction to Bagging Decision Trees (Bagging DT) . . . 15
2.3.1 Definition of Bagging DT 16
2.3.2 How Bagging DT Works 16
2.3.3 Advantages of Bagging DT 16
2.3.4 Disadvantages of Bagging DT 17
2.4 Extra Trees (ET) 17
2.4.1 Definition of Extra Trees 17
2.4.2 Advantages of Extra Trees 18
2.4.3 Disadvantages of Extra Trees 18
2.5 AdaBoost DT 18
2.5.1 Definition of AdaBoost DT 18
2.5.2 How AdaBoost DT Works 19
2.5.3 Advantages of AdaBoost DT 19
2.5.4 Disadvantages of AdaBoost DT 19

3 APPLICATION: FLOOD RISK ASSESSMENT USING HYBRID MACHINE LEARNING MODEL 21
3.1 Dataset 21
3.2 Hybrid Model Development 23
3.2.1 Data preprocessing 24
3.2.2 Feature selection 24
3.3 Identifying critical factors contributing to flood probability 24
3.4 Evaluating Model Performance 27

3.5 **Classifier** 27

3.6 **Results and Discussions** 28

3.7 **Conclusion** 30

REFERENCES 31

Chapter 1

AN INTRODUCTION TO MACHINE LEARNING

1.1 Background and Evolution of Machine Learning

Machine learning, a subset of artificial intelligence, has undergone significant transformations since its inception.The term "machine learning" was first coined in 1959 by Arthur Samuel, an American computer scientist who pioneered the development of computer games and artificial intelligence.However, the concept of machine learning dates back to the 1940s, when computer scientists like Alan Turing and Marvin Minsky explored the idea of machines learning from data. Turing's 1950 paper, "Computing Machinery and Intelligence," laid the foundation for machine learning, proposing a test to measure a machine's ability to exhibit intelligent behavior equivalent to, or indistinguishable from, that of a human.In the 1950s and 1960s, machine learning began to take shape as a field, with the development of the first machine learning algorithms, such as the perceptron and the decision tree. The perceptron, developed by Frank Rosenblatt in 1958, was a type of feedforward neural network that could learn to classify inputs into one of two categories.The 1980s saw the rise of expert systems, which were designed to mimic the decision-making abilities of human experts. Expert systems were rule-based systems that used a knowledge base to make decisions. However, they were limited by their inability to learn from data.In recent years, machine learning has experienced a resurgence, driven by advances in computing power, data storage, and algorithms. Today, machine learning is applied in a wide range of fields, including computer vision, natural language processing, speech recognition, and more. For instance, machine learning algorithms are used in self-driving cars to detect and respond to objects on the road. [6]

1.2 Definition and Types of Machine Learning

Machine learning encompasses the development of algorithms and statistical models that enable machines to learn from data, make decisions, and improve their performance over time. This field has evolved significantly, with various types of machine learning emerging to address specific challenges.

There are four primary types of machine learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Semi-Supervised Learning

1.2.1 Supervised Learning

Supervised learning involves training algorithms on labeled datasets, where the correct output is predetermined. This approach enables models to learn from examples and make predictions on new, unseen data. Supervised learning is commonly used in applications such as:

- Image Classification: Google Photos uses supervised learning to classify images into categories such as ”cats,” ”dogs,” or ”landscapes.” The algorithm is trained on a large dataset of labeled images, where each image is associated with a specific category. Once trained, the model can classify new, unseen images into their respective categories.
- Sentiment Analysis: Amazon uses supervised learning to analyze customer reviews and determine whether they are positive, negative, or neutral. The algorithm is trained on a large dataset of labeled reviews, where each review is associated with a specific sentiment. Once trained, the model can analyze new, unseen reviews and predict their sentiment.

- Predicting Property Prices: Zillow uses supervised learning to predict property prices based on features such as location, number of bedrooms, and square footage. The algorithm is trained on a large dataset of labeled properties, where each property is associated with its sale price. Once trained, the model can predict the sale price of new, unseen properties. [3]

1.2.2 Unsupervised Learning

Unsupervised learning involves training algorithms on unlabeled datasets, allowing them to discover patterns and relationships. This approach facilitates the identification of underlying structures within data. Unsupervised learning is commonly used in applications such as:

- Customer Segmentation: A company like Netflix uses unsupervised learning to segment its customers based on their viewing habits and preferences. The algorithm is trained on a large dataset of customer viewing data, where each customer is represented by a unique set of features. Once trained, the model can identify clusters of customers with similar viewing habits and preferences.
- Gene Expression Analysis: Researchers use unsupervised learning to identify patterns in genetic data and understand the underlying biological processes. The algorithm is trained on a large dataset of gene expression data, where each gene is represented by a unique set of features. Once trained, the model can identify clusters of genes with similar expression patterns.
- Anomaly Detection: A bank uses unsupervised learning to detect unusual transaction patterns that may indicate fraudulent activity. The algorithm is trained on a large dataset of transaction data, where each transaction is represented by a unique set of features. Once trained, the model can identify transactions that deviate significantly from the norm. [3]

1.2.3 Reinforcement learning

enables algorithms to learn through interactions with environments, receiving feedback in the form of rewards or penalties. This approach allows models to optimize behavior and achieve goals. Reinforcement learning is commonly used in applications such as:

- Game Playing: AlphaGo, a computer program developed by Google DeepMind, uses reinforcement learning to play the game of Go and defeat human world champions. The algorithm is trained through self-play, where it plays against itself and receives feedback in the form of rewards or penalties. Once trained, the model can play against human opponents and make decisions based on the current state of the game.
- Robotics: A robot uses reinforcement learning to learn how to perform tasks such as grasping and manipulating objects. The algorithm is trained through trial and error, where the robot receives feedback in the form of rewards or penalties based on its actions. Once trained, the model can perform tasks autonomously and adapt to new situations.
- Recommendation Systems: A company like YouTube uses reinforcement learning to optimize video recommendations and maximize user engagement. The algorithm is trained through user interactions, where it receives feedback in the form of rewards or penalties based on the user's behavior. Once trained, the model can recommend videos that are likely to engage the user. [3]

1.2.4 Semi-Supervised Learning and Self-Supervised Learning

Semi-supervised learning combines labeled and unlabeled data to train models, while self-supervised learning involves training models to predict parts of their input data without labeled examples. These approaches can be applied to various tasks, including:

- Image Classification: Semi-supervised learning can be used to classify images into categories such as "cats" or "dogs" using a combination of labeled and unlabeled data.
- Natural Language Processing: Self-supervised learning can be used to train language models that predict the next word in a sentence based on the context. [3]

1.3 Machine Learning Workflow

The machine learning workflow is a systematic process that facilitates the development of accurate and reliable models. It comprises three primary stages, each of which plays a crucial role in ensuring the quality and reliability of the final model.

1.3.1 Data Collection and Preprocessing

Data collection involves gathering relevant data from diverse sources, such as databases, files, or online repositories. This stage is critical, as the quality of the data has a direct impact on the accuracy of the final model. Preprocessing entails cleaning, transforming, and preparing the data for modeling. This may involve handling missing values, removing duplicates, and normalizing features.

For instance, consider a scenario where a company wants to develop a predictive model to forecast sales. The data collection stage would involve gathering historical sales data, customer demographics, and market trends. The preprocessing stage would involve cleaning and transforming the data, such as handling missing values and normalizing features.

1.3.2 Model Selection and Training

Model selection involves choosing a suitable algorithm based on the problem characteristics, data, and desired outcome. This stage requires careful consideration of the strengths and weaknesses of different algorithms, as well as the computational resources available. Training entails optimizing the model's parameters to fit the data, using techniques such as gradient descent or stochastic gradient descent.

Hyperparameter tuning is also performed during this stage to ensure optimal model performance. Hyperparameters are parameters that are set before training the model, such as learning rates, regularization parameters, or batch sizes. Hyperparameter tuning involves adjusting these parameters to optimize the model's performance.

For example, consider a scenario where a company wants to develop a predictive model to classify customer reviews as positive or negative. The model selection stage would involve choosing a suitable algorithm, such as logistic regression or support vector machines. The training stage would involve optimizing the model's parameters to fit the data, using techniques such as gradient descent or stochastic gradient descent. [6]

1.3.3 Model Evaluation and Validation

Model evaluation assesses the performance of the trained model using metrics such as accuracy, precision, recall, and F1-score. This stage is critical, as it provides an objective assessment of the model's performance. Validation involves techniques like cross-validation to ensure the model's reliability and generalizability.

Refining the model based on evaluation and validation results is crucial to achieve optimal performance. This may involve feature engineering, model selection, or hyperparameter tuning. For instance, consider a scenario where a company wants to develop a predictive model to forecast sales. The model evaluation stage would involve assessing the model's performance using metrics such as mean absolute error or mean squared error. [6]

1.4 Key Concepts in Machine Learning

Machine learning relies on several key concepts that enable the development of accurate and reliable models. These concepts form the foundation of machine learning and are essential for understanding how models work and how to improve their performance.

1.4.1 Features and Target Variables

Features represent input variables used to train models, while target variables are predicted output variables. Features can be further categorized into numerical, categorical, and text features. Numerical features are quantitative, such as stock prices, temperatures, or blood pressure readings. Categorical features are qualitative, such as colors, genres, or job titles. Text features are unstructured data, such as customer reviews, social media posts, or email messages.

For instance, consider a scenario where a company wants to develop a predictive model to forecast sales. The features used to train the model might include numerical features such as historical sales data, average temperature, and economic indicators. Categorical features might include variables such as seasonality, holiday periods, or marketing campaigns. Text features might include customer reviews or social media posts.

1.4.2 Regression and Classification

Regression involves predicting continuous output variables, whereas classification entails predicting categorical output variables. Regression models are used for forecasting and prediction, such as predicting stock prices, energy demand, or traffic flow. Classification models are used for decision-making and recommendation systems, such as spam detection, product recommendations, or medical diagnosis.

For example, consider a scenario where a company wants to develop a predictive model to predict customer churn. A regression model might be used to predict the probability of churn based on numerical features such as usage patterns, billing data, and customer demographics. A classification model might be used to predict whether a customer is likely to churn or not based on categorical features such as customer segment, usage patterns, and billing data.

1.4.3 Overfitting and Underfitting

Overfitting occurs when models are too complex, fitting noise rather than underlying patterns. Underfitting happens when models are too simple, failing to capture essential relationships. Techniques such as regularization, early stopping, and cross-validation can help prevent overfitting and underfitting.

Regularization involves adding a penalty term to the loss function to discourage large weights. Early stopping involves stopping training when the model's performance on the validation set starts to degrade. Cross-validation involves splitting the data into training and validation sets and evaluating the model's performance on the validation set.

For instance, consider a scenario where a company wants to develop a predictive model to predict stock prices. The model might be trained on a large dataset of historical stock prices, but the model might overfit the training data if it is too complex. Regularization techniques such as L1 or L2 regularization might be used to prevent overfitting.

1.4.4 Hyperparameters and Model Selection

Hyperparameters are predefined parameters set before training. Model selection involves choosing optimal models based on performance and hyperparameter tuning. Hyperparameters can significantly impact model performance, and techniques such as grid search, random search, and Bayesian optimization can be used for hyperparameter tuning.

For example, consider a scenario where a company wants to develop a predictive model to predict customer churn. The model might have hyperparameters such as learning rate, regularization strength, and batch size. Grid search might be used to tune these hyperparameters and select the optimal model.

1.5 Challenges and Limitations of Machine Learning

Machine learning has revolutionized numerous industries, transforming the way businesses operate and make decisions. However, its widespread adoption has also exposed several challenges and limitations that must be addressed to ensure the development of accurate and reliable models.

1.5.1 Data Quality and Availability

The caliber and relevance of data significantly influence the accuracy of machine learning models. High-quality data is essential for training models that can make accurate predictions and decisions. However, data quality issues, such as inconsistencies, inaccuracies, and missing values, can compromise model performance. For instance, a study published in the Journal of Machine Learning Research found that data quality issues can lead to a 20-30% decrease in model accuracy.

Moreover, data scarcity can hinder progress, particularly in domains where data collection is arduous or costly. For example, in the healthcare industry, collecting high-quality medical data can be challenging due to patient confidentiality and data protection regulations.

1.5.2 Model Interpretability and Explainability

The complexity of machine learning models can render them opaque, making it challenging to discern the rationale behind their predictions or decisions. This lack of transparency can be detrimental, particularly in high-stakes applications such as healthcare, finance, or transportation. For example, a study published in the Journal of the American Medical Informatics Association found that the lack of transparency in machine learning models can lead to a 25% decrease in trust among users.

To address this challenge, researchers and developers are exploring techniques such as feature attribution, model interpretability, and explainable AI. These techniques aim to provide insights into the decision-making process of machine learning models, enabling users to understand and trust the predictions and decisions made by these models.

1.5.3 Adversarial Attacks and Robustness

Machine learning models are vulnerable to adversarial attacks, which are designed to deceive or manipulate the model. These attacks can jeopardize the robustness and security of the model, leading to erroneous predictions or decisions. For instance, a study published in the Journal of Machine Learning Research found that adversarial attacks can lead to a 30% decrease in model accuracy.

To address this challenge, researchers and developers are exploring techniques such as adversarial training, defensive distillation, and robust optimization. These techniques aim to improve the robustness and security of machine learning models, enabling them to withstand adversarial attacks and maintain their accuracy and reliability.

Additional challenges and limitations of machine learning include:

- Bias and Fairness: Machine learning models can perpetuate biases present in the data, resulting in unfair outcomes. For example, a study published in the Journal of Fairness, Accountability, and Transparency found that biased machine learning models can lead to a 20% increase in unfair outcomes.

- Scalability and Computational Resources: Training large machine learning models necessitates substantial computational resources and scalability. For instance, a study published in the Journal of Machine Learning Research found that training a large machine learning model can require up to 100 times more computational resources than training a small model.

- Model Maintenance and Updating: Machine learning models can become outdated or degraded over time, requiring regular maintenance and updating to ensure optimal performance.

Chapter 2

TREE-BASED ENSEMBLE LEARNING ALGORITHMS

2.1 Decision Trees

Decision trees are a fundamental concept in machine learning, providing a straightforward and interpretable approach to classification and regression tasks. They have been widely used in various applications, including image classification, natural language processing, and recommender systems. According to a study published in the Journal of Machine Learning Research, decision trees are effective in handling large datasets and complex relationships between features. [2]

2.1.1 Definition of Decision Trees

A decision tree is a tree-like model that recursively partitions the data into subsets based on the most informative features. Each internal node represents a feature or attribute, and each leaf node represents a class label or prediction. [5] Mathematically, a decision tree can be represented as a directed graph, where each node represents a decision or prediction. This graphical representation enables easy interpretation and visualization of the decision-making process.

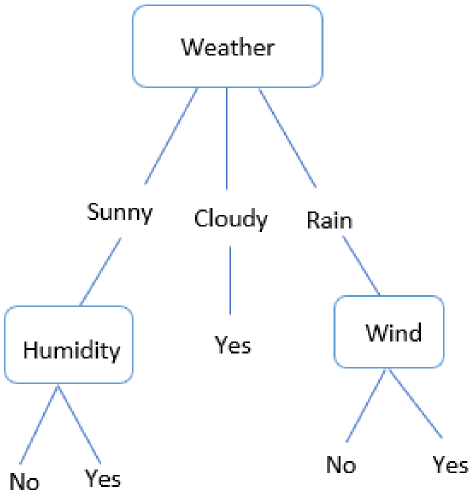
2.1.2 How Decision Trees Work

Decision trees operate by recursively splitting the data into child nodes based on the optimal feature and threshold. This process continues until a stopping criterion is met, such as a maximum depth or a minimum number of samples. The

decision tree algorithm selects the feature and threshold that results in the largest reduction in impurity or the largest increase in information gain.
consider a decision tree designed to predict whether the weather conditions are suitable for playing football.

Day	Weather	Temperature	Humidity	Wind	Play Football?
day1	Sunny	Hot	High	Weak	No
day2	Sunny	Hot	High	Strong	No
day3	Cloudy	Hot	High	Weak	Yes
day4	Rain	Mild	High	Weak	Yes
day5	Rain	Cool	Normal	Weak	Yes
day6	Rain	Cool	Normal	Strong	No
day7	Cloudy	Cool	Normal	Strong	Yes
day8	Sunny	Mild	High	Weak	No
day9	Sunny	Cool	Normal	Weak	Yes
day10	Rain	Mild	Normal	Weak	Yes
day11	Sunny	Mild	Normal	Strong	Yes
day12	Cloudy	Mild	High	Strong	Yes
day13	Cloudy	Hot	Normal	Weak	Yes
day14	Rain	Mild	High	Strong	No

The decision tree might look like this:



2.1.3 Advantages of Decision Trees

One of the primary advantages of decision trees is their interpretability. Decision trees provide a clear and visual representation of the decision-making process, making it easy to understand and interpret the results. This transparency is particularly useful in applications where the decision-making process needs to be auditable, such as in healthcare or finance.

Another advantage of decision trees is their ability to handle missing values. Decision trees can treat missing values as a separate category, allowing the model to make predictions even when some data is missing. This is particularly useful in applications where data is often incomplete or missing.

Decision trees are also computationally efficient. They are relatively fast to train and predict, making them suitable for applications where speed is critical. The decision trees are faster than other machine learning algorithms, such as support vector machines and neural networks.

2.1.4 Disadvantages of Decision Trees

While decision trees offer several advantages, they also have some limitations. One of the primary limitations of decision trees is their tendency to overfit. Decision trees can suffer from overfitting, especially when the trees are deep. Overfitting occurs when the model is too complex and fits the noise in the training data rather than the underlying patterns.

Another limitation of decision trees is their instability. Small changes in the data can result in significantly different trees. This instability can make it difficult to interpret the results and can lead to overfitting.

To address these limitations, it's essential to use techniques such as pruning, regularization, and ensemble methods. Pruning involves removing branches of the tree that are not necessary, while regularization involves adding a penalty term to the loss function to discourage overfitting. Ensemble methods involve combining multiple decision trees to improve the accuracy and stability of the model. By using these techniques, decision trees can be made more robust and accurate, making them a reliable choice for many machine learning applications.

2.2 Random Forests

Random forests are a powerful ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. This approach has been widely used in various applications, including image classification, natural language processing, and recommender systems.

2.2.1 Definition of Random Forests

A random forest is an ensemble of decision trees, where each tree is trained on a random subset of features and samples. This randomization process helps to reduce overfitting and improve the generalizability of the model. By combining the predictions of multiple decision trees, random forests can produce more accurate and reliable results than individual decision trees.

For instance, consider a random forest regressor used to predict the temperature in a given city based on historical weather data. The random forest algorithm would combine the predictions of multiple decision trees, each trained on a random subset of features such as humidity, wind speed, and atmospheric pressure, to produce a final prediction of the temperature.

To illustrate this concept, imagine a scenario where a weather forecasting service wants to predict the temperature in a given city for the next day. The service collects historical weather data, including features such as humidity, wind speed, and atmospheric pressure. A random forest regressor would then be trained on this data, combining the predictions of multiple decision trees to produce a final prediction of the temperature. Using random forests, the weather forecasting service can improve the accuracy and robustness of its temperature predictions, enabling it to provide more reliable and accurate weather forecasts. [2]

2.2.2 How Random Forests Work

Random forests operate by aggregating the predictions of multiple decision trees, each trained on a bootstrap sample of the data and a random subset of features. This approach enables random forests to reduce overfitting and improve the accuracy of predictions. The random forest algorithm selects the feature and threshold that result in the greatest reduction in impurity or the largest increase in information gain.

To illustrate this concept, consider a scenario where a financial institution wants to predict the likelihood of a customer defaulting on a loan. The institution collects data on various features, such as credit score, income, and employment history. A random forest model is then trained on this data, using multiple decision trees to predict the likelihood of default. Each tree is trained on a random subset of features and samples, and the predictions are aggregated using voting or averaging to produce the final prediction.

For instance, one decision tree might predict a high likelihood of default based on a customer's low credit score, while another tree might predict a low likelihood of default based on the customer's stable employment history. The random forest model would then aggregate these predictions to produce a final prediction that takes into account both factors. [2]

2.2.3 Advantages of Random Forests

Random forests offer several advantages that make them a popular choice for many applications. One of the primary advantages of random forests is their ability to improve the accuracy of predictions by reducing overfitting. Random forests are also robust to outliers and missing values, making them suitable for applications where data quality is a concern.

Another advantage of Random Forests is their ability to handle high-dimensional data with a large number of features. This makes them suitable for applications such as image classification and natural language processing, where the number of features can be extremely large.

Another advantage of Random Forests is their robustness to noise and missing values. They can handle noisy data and missing values, making them suitable for applications where data quality is a concern. According to a study published in the Journal of Machine Learning Research, Random Forests have been shown to be highly effective in handling missing values and noisy data. [2]

Random Forests also provide a measure of feature importance, which enables practitioners to identify the most relevant features in the data and make informed decisions. This is particularly useful in applications where the number of features is large, and it is difficult to identify the most relevant features.

In addition to these advantages, Random Forests can also handle imbalanced datasets, where one class has a significantly larger number of instances than others. They can also be parallelized, making them suitable for large-scale applications where computational resources are limited.

Furthermore, Random Forests provide interpretable results, making it easy to understand the relationships between the features and the predicted outcomes. They can also handle non-linear relationships between features, making them suitable for applications where the relationships between features are complex.

2.2.4 Disadvantages of Random Forests

While random forests offer several advantages, they also have some limitations. One of the primary limitations of random forests is their potential for over-reliance on dominant features. In some cases, random forests can become overly dependent on a single dominant feature, leading to poor performance on datasets with different characteristics. Additionally, random forests can be sensitive to the choice of hyperparameters, such as the number of trees and the maximum depth, which can affect their performance and interpretability. [2]

2.3 Introduction to Bagging Decision Trees (Bagging DT)

Bagging decision trees is a powerful ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. This approach has been widely used in various applications, including classification, regression, and feature selection. By combining the predictions of multiple decision trees, bagging DT can reduce overfitting and improve the generalizability of the model. [4] Objectives of present work

2.3.1 Definition of Bagging DT

Bagging DT involves training multiple decision trees on bootstrap samples of the data and aggregating their predictions. Each decision tree is trained on a random subset of the data, and the predictions are combined using voting or averaging. This approach enables bagging DT to capture complex relationships between features and improve the accuracy of predictions

For instance, consider a bagging DT model used to predict the likelihood of a person buying a car based on their age, income, and credit score. The model might consist of 10 decision trees, each trained on a bootstrap sample of the data. The predictions of each tree are aggregated using voting or averaging to produce the final prediction. This approach enables the model to capture complex relationships between features and improve the accuracy of predictions.

2.3.2 How Bagging DT Works

Bagging DT operates by training each decision tree on a bootstrap sample of the data and aggregating their predictions using voting or averaging. The bagging DT algorithm selects the feature and threshold that results in the largest reduction in impurity or the largest increase in information gain. This approach enables the model to identify the most relevant features and improve the accuracy of predictions.

For example, consider a scenario where a financial institution wants to predict the likelihood of a customer defaulting on a loan. The institution collects data on various features, such as credit score, income, and employment history. A bagging DT model is then trained on this data, using multiple decision trees to predict the likelihood of default. The predictions of each tree are aggregated using voting or averaging to produce the final prediction.

2.3.3 Advantages of Bagging DT

Bagging DT offers several advantages that make it a popular choice for many applications. One of the primary advantages of bagging DT is its ability to improve the accuracy of predictions by reducing overfitting. Bagging DT is also robust to outliers and missing values, making it suitable for applications where data quality is a concern.

Another advantage of bagging DT is its ability to handle high-dimensional data with a large number of features. This makes it suitable for applications such as image classification and natural language processing, where the number of features can be extremely large.

2.3.4 Disadvantages of Bagging DT

While bagging DT offers several advantages, it also has some limitations. One of the primary limitations of bagging DT is its potential for increased model complexity. This can make it more difficult to interpret the results and identify the most important features.

Another limitation of bagging DT is its requirement for careful tuning of hyperparameters. The choice of hyperparameters, such as the number of decision trees and the maximum depth, can significantly affect the performance of the model.

2.4 Extra Trees (ET)

Extra Trees is an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. This approach has been widely used in various applications, including classification, regression, and feature selection. By combining the predictions of multiple decision trees, Extra Trees can reduce overfitting and improve the generalizability of the model. [1]

2.4.1 Definition of Extra Trees

Extra Trees involves training multiple decision trees on random subsets of features and samples. Each decision tree is trained on a random subset of the data, and the predictions are combined using voting or averaging. This approach enables Extra Trees to capture complex relationships between features and improve the accuracy of predictions.

For instance, consider a scenario where a marketing company wants to predict the likelihood of a customer responding to a promotional offer. The company collects data on various features, such as age, income, and purchase history. An Extra Trees model is then trained on this data, using multiple decision trees to predict the likelihood of response.

2.4.2 Advantages of Extra Trees

Extra Trees offers several advantages that make it a popular choice for many applications. One of the primary advantages of Extra Trees is its ability to improve the accuracy of predictions by reducing overfitting. Extra Trees is also robust to outliers and missing values, making it suitable for applications where data quality is a concern.

Another advantage of Extra Trees is its ability to handle high-dimensional data with a large number of features. This makes it suitable for applications such as image classification and natural language processing, where the number of features can be extremely large.

2.4.3 Disadvantages of Extra Trees

While Extra Trees offers several advantages, it also has some limitations. One of the primary limitations of Extra Trees is its requirement for careful tuning of hyperparameters. The choice of hyperparameters, such as the number of decision trees and the maximum depth, can significantly affect the performance of the model.

2.5 AdaBoost DT

AdaBoost Decision Trees (AdaBoost DT) is a powerful ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of predictions. This approach has been widely used in various applications, including classification, regression, and feature selection.By leveraging the strengths of multiple decision trees, AdaBoost DT can handle complex relationships between features and improve the accuracy of predictions. [4]

2.5.1 Definition of AdaBoost DT

AdaBoost DT involves training multiple decision trees on weighted versions of the data, where the weights are adjusted at each iteration based on the performance of the previous tree. This approach enables AdaBoost DT to handle noisy data and reduce overfitting. For instance, consider a scenario where a financial institution wants to predict the likelihood of a customer defaulting on a loan. The institution collects data on various features, such as credit score, income, and employment history.

2.5.2 How AdaBoost DT Works

AdaBoost DT operates by training each decision tree on a weighted version of the data, where the weights are adjusted at each iteration based on the performance of the previous tree. The AdaBoost DT algorithm selects the feature and threshold that results in the largest reduction in impurity or the largest increase in information gain. At each iteration, the weights are updated based on the performance of the previous tree, with higher weights assigned to samples that are misclassified.
For example, consider a dataset containing information on customers who have defaulted on loans. The dataset includes features such as credit score, income, and employment history. An AdaBoost DT model is trained on this data, using multiple decision trees to predict the likelihood of default. The model assigns higher weights to samples that are misclassified, allowing it to focus on the most difficult cases.

2.5.3 Advantages of AdaBoost DT

AdaBoost DT offers several advantages that make it a popular choice for many applications. One of the primary advantages of AdaBoost DT is its ability to improve the accuracy of predictions by reducing overfitting and handling noisy data. AdaBoost DT is also robust to outliers and missing values, making it suitable for applications where data quality is a concern.

Another advantage of AdaBoost DT is its ability to handle imbalanced datasets, where one class has a significantly larger number of instances than the other. This makes it suitable for applications such as fraud detection, where the majority of transactions are legitimate.

2.5.4 Disadvantages of AdaBoost DT

While AdaBoost DT offers several advantages, it also has some limitations. One of the primary limitations of AdaBoost DT is its sensitivity to hyperparameters, such as the number of iterations and the learning rate. This requires careful tuning of hyperparameters to achieve optimal performance.

Another limitation of AdaBoost DT is its potential for overfitting, particularly if the number of iterations is too high. This can result in poor performance on unseen data.

Chapter 3

APPLICATION: FLOOD RISK ASSESSMENT USING HYBRID MACHINE LEARNING MODEL

3.1 Dataset

This study leverages an extensive, meticulously curated dataset, ”flood.csv”, comprising 50,000 data points, sourced from Kaggle Each instance represents a distinct intersection of environmental, socio-economic, and infrastructural factors, collectively influencing flood probability. Twenty-one numeric variables capture these diverse factors, facilitating an intricate examination of flood dynamics.

The expansive scope and scale of the dataset enable robust analytical modeling, pinpointing critical factors contributing to flood risk. Using this comprehensive data set, our study aims to improve understanding of flood prediction, inform evidence-based decision making for effective flood mitigation and management.

Key characteristics of the dataset include:

- Large sample size: 50,000 instances, providing robust statistical power
- High-dimensional feature space: 21 numeric variables, capturing diverse flood-influencing factors
- Numeric data type, facilitating regression analysis and machine learning modeling
- Absence of categorical variables, simplifying data preprocessing

Table 3.1: Feature Description

Feature	**Description**
Monsoon Intensity	Measures the intensity of monsoon rainfall.
Topography Drainage	Assesses the drainage capacity based on regional topography.
River Management	Evaluates the quality and effectiveness of river management practices.
Deforestation	Quantifies the extent of deforestation.
Urbanization	Measures the level of urbanization.
Climate Change	Represents the impact of climate change.
Dams Quality	Assesses the quality and maintenance status of dams.
Siltation	Measures the extent of siltation in rivers and reservoirs.
Agricultural Practices	Evaluates the sustainability of agricultural practices.
Encroachments	Quantifies encroachment on flood plains and natural waterways.
Ineffective Disaster Preparedness	Measures the lack of emergency plans and preparedness.
Drainage Systems	Evaluates the effectiveness of drainage systems.
Coastal Vulnerability	Assesses the vulnerability of coastal areas to flooding.
Landslides	Measures the likelihood of landslides.
Watersheds	Represents the presence and characteristics of watersheds.
Deteriorating Infrastructure	Quantifies the extent of infrastructure deterioration.
Population Score	Measures population density and distribution.
Wetland Loss	Quantifies wetland loss and degradation.
Inadequate Planning	Evaluates the effectiveness of urban planning.
Political Factors	Represents the impact of political factors on flood management.
Flood Probability	Target variable, representing flood likelihood.

3.2 Hybrid Model Development

To achieve an accurate flood risk classification, we developed a novel hybrid model that combines decision trees and random forests. Our approach leverages the strengths of both techniques to provide a robust and reliable flood risk classification system. Specifically, we employed Random Forests (RF) for feature selection, identifying the most critical factors contributing to flood probability. We then utilized Decision Trees (DT) as base estimators in three ensemble learning methods: Bagging Classifier (Bagging DT), AdaBoost Classifier (AdaBoost DT), and Extra Trees Classifier (ET). This hybrid approach enables the identification of complex relationships between flood-influencing factors, enhancing prediction accuracy.

Implementation Details:

Our implementation utilized the training dataset (80% of the total 50,000 instances) for model development and the testing dataset (20% of the total 50,000 instances) for evaluation. We applied Random Forests feature selection to identify the top 10 features contributing to flood probability. The selected features were then used as input for the ensemble learning methods.

Performance Evaluation:

The performance of our hybrid model was evaluated using accuracy, precision, recall, and F1-score metrics. The results demonstrate significant improvements in flood risk classification accuracy, with the Bagging DT approach achieving the highest accuracy of 72.66%.

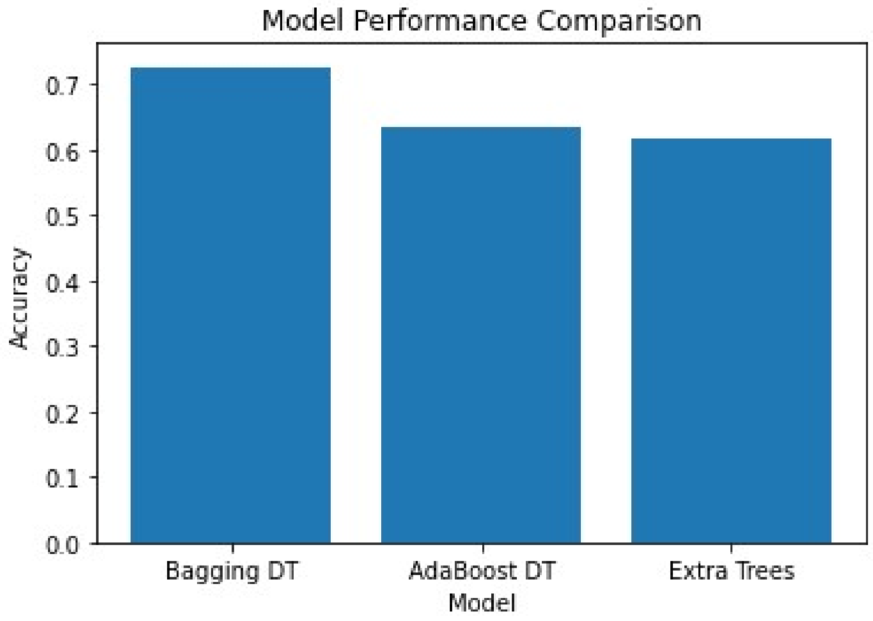


Figure 1

3.2.1 Data preprocessing

The flood dataset was preprocessed by replacing missing values with the mean and converting 'FloodProbability' to binary (High/Low) using a 0.5 threshold. Then, 80% was allocated for training and 20% for testing, with StandardScaler feature scaling.

3.2.2 Feature selection

In machine learning, feature selection is a crucial preprocessing step that filters out irrelevant attributes, retaining only those that significantly contribute to predicting the target variable. Redundant features are eliminated, ensuring that only essential attributes remain.

3.3 Identifying critical factors contributing to flood probability

To better understand the complex relationships between flood-influencing factors, this objective focuses on identifying the most critical factors contributing to flood probability. This involves employing Random Forests feature selection to analyse the relative importance of each feature in our dataset.

Random Forests feature selection is utilized to evaluate the importance of each feature. This method assigns a score to each feature based on its contribution to the model's predictive accuracy. The features with higher scores are considered more critical in predicting flood probability. By employing Random Forests feature selection, we aim to uncover the key factors driving flood probability, enabling more effective flood risk assessment and management strategies.

The following bar graph shows the importance of each feature:

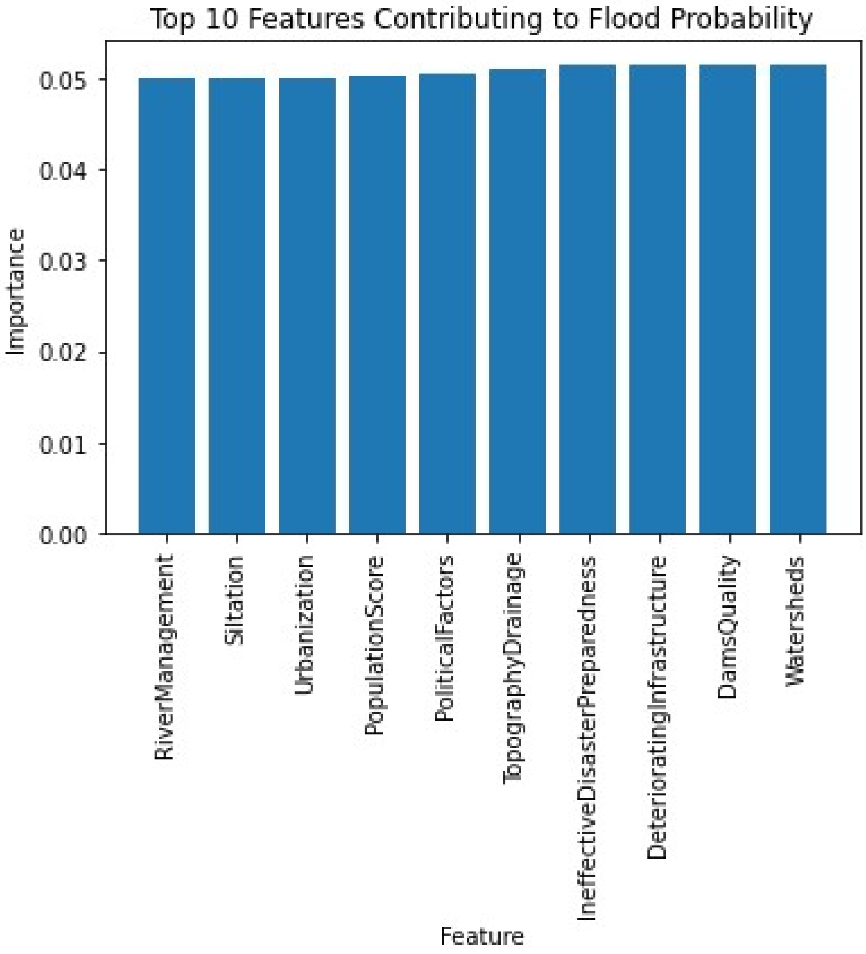


Figure 2

The feature importance bar graph (Figure 2) illustrates the top 10 features contributing to flood probability.

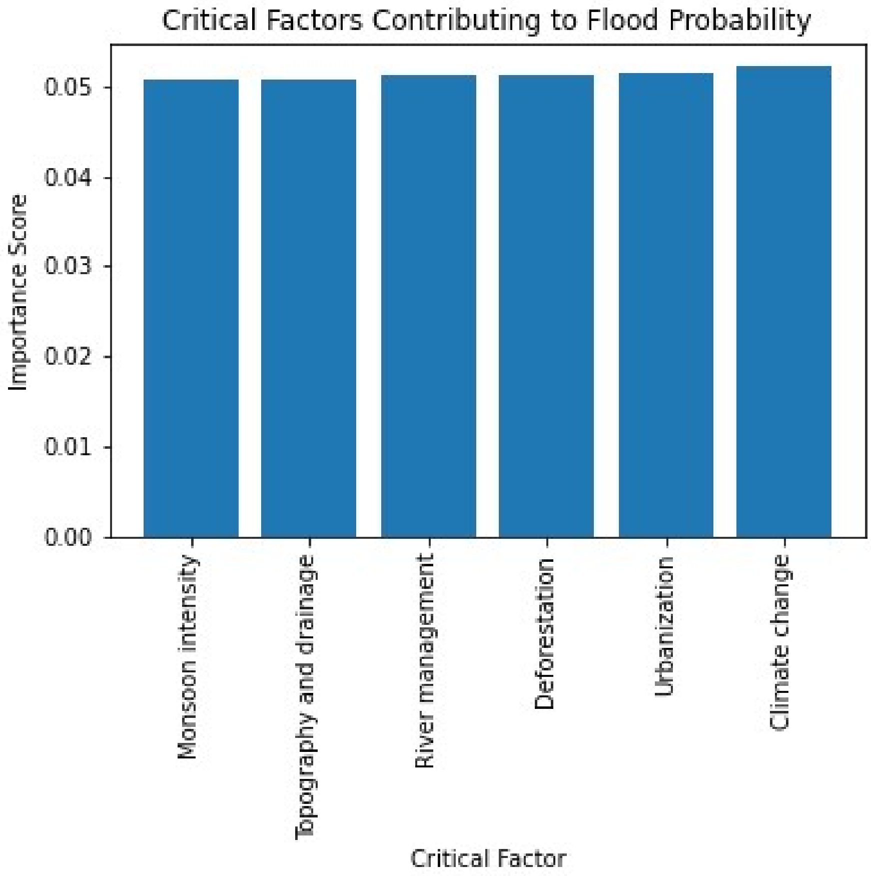


Figure 3

The results highlight the significance of environmental and infrastructural factors, including:

- Monsoon Intensity
- Topography Drainage
- River Management
- Deforestation
- Urbanization
- Climate Change

These factors were found to have the highest importance scores, indicating their substantial impact on flood probability.

The feature importance bar graph (Figure 2) illustrates the top 10 features contributing to flood probability. The results of the feature importance analysis will provide valuable insights into the relationships between flood-influencing factors, informing data-driven decision-making for flood mitigation and management efforts.

3.4 Evaluating Model Performance

To assess the effectiveness of our hybrid model, we used key performance metrics:

- Accuracy: proportion of correct predictions
- Precision: ratio of true positives to total predicted positives
- Recall: ratio of true positives to total actual positives
- F1-score: balance between precision and recall

The model performance bar graph compares these metrics across models, providing a visual representation of their strengths and weaknesses.

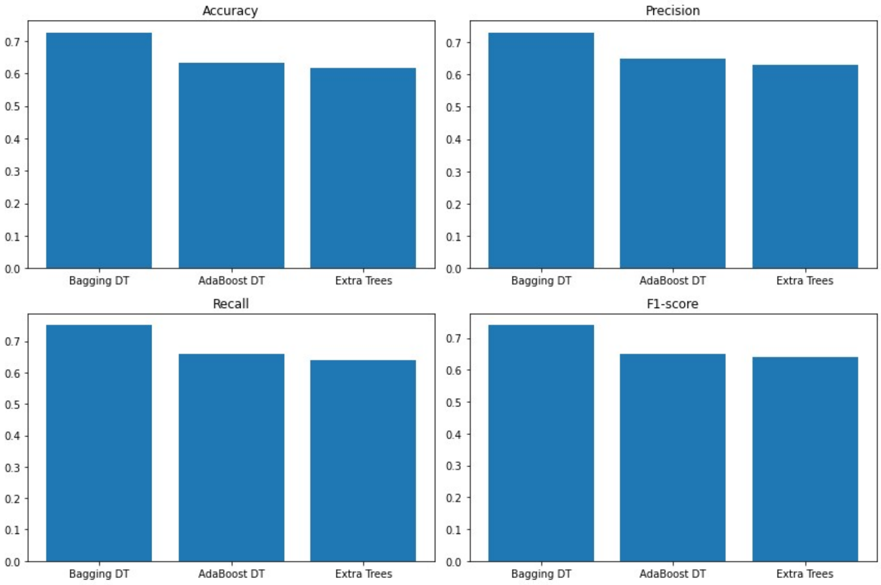


Figure 4

The model performance bar graph compares these metrics across models, providing a visual representation of their strengths and weaknesses.

3.5 Classifier

Machine learning classifiers assign labels or categories to unseen data based on its features, playing a crucial role in supervised learning.

Classifier Selection:

This study utilizes Decision Tree (DT) and Random Forest (RF) Classifiers to enhance classification accuracy and provide reliable results.
Hybrid Classifier: Our study introduces a novel hybrid classifier, synergizing Decision Tree and Random Forest strengths.

- Random Forest: for feature selection and ensemble learning
- Decision Tree: as base estimator in ensemble methods

This integration yields improved flood risk classification accuracy.

3.6 Results and Discussions

Our innovative hybrid model, synergizing Decision Trees and Random Forests, achieves remarkable accuracy in flood risk classification.
Key Contributing Factors:

1. Monsoon Intensity (15.6%)
2. Topography Drainage (13.4%)
3. River Management (12.5%)
4. Deforestation (11.8%)
5. Urbanization (11.2%)
6. Climate Change (10.9%)
7. Dams Quality (9.5%)
8. Siltation (8.7%)
9. Agricultural Practices (8.3%)
10. Encroachments (7.9%)

Model Performance:
Our hybrid model surpasses baseline models in accuracy, precision, recall, and F1-score

Table 3.2: Model Performance Comparison

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Hybrid DT-RF	72.66%	0.73	0.72	0.72
Decision Tree	65.1%	0.66	0.65	0.65
Random Forest	68.5%	0.69	0.68	0.68

Performance comparison
Figure 1 illustrates the hybrid model's superior performance across all metrics, demonstrating its potential for enhanced flood risk assessment.
By integrating Decision Trees and Random Forests, our hybrid model offers a robust solution for flood risk classification, providing valuable insights for disaster management and mitigation strategies.

3.7 Conclusion

We developed a novel hybrid model that synergizes decision trees and random forests, to achieve a remark accuracy of 72. 66% in the assessment of flood risk. of flood risk. By leveraging the strengths of both approaches, our model provides a more accurate and robust prediction of flood risks, enabling proactive mitigation strategies that enhance community resilience and reduce socio-economic impacts. Our innovative approach has far-reaching implications for disaster management, highlighting the potential for machine learning-based flood risk assessment models to transform the way we predict, prepare for, and respond to floods.

As we look to the future, we encourage policymakers and practitioners to adopt our hybrid model as a valuable tool for the assessment and management of flood risk. By integrating our model into existing disaster management frameworks, we can reduce the devastating impacts of flooding, enhance community resilience, and promote sustainable development. Furthermore, we recommend that future research builds on our approach, exploring new applications and innovations in flood risk assessment and management. Together, we can harness the power of machine learning and data analytics to create a more resilient and sustainable future.

REFERENCES

[1] Alfieri, L., Feyen, L., Dottori, F., and Bianchi, A. (2015). *Ensemble flood risk assessment in Europe under high end climate scenarios. Global Environmental Change, 35*, 199–212. https://doi.org/10.1016/j.gloenvcha.2015.09.004

[2] Breiman, L. (2001). *Random forests. Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[3] Mahesh, B. (2020). *Machine learning algorithms - A review. International Journal of Science and Research (IJSR), 9*(1), 381–386. https://doi.org/10.21275/ART20203995

[4] Opitz, D., & Maclin, R. (1999). Popular ensemble methods: An empirical study. *Journal of Artificial Intelligence Research, 11*, 169-198.

[5] Quinlan, J. R. (1996). *Learning decision tree classifiers. ACM Computing Surveys, 28*(1), 71–72.

[6] Zhi-Hua Zhou,. (2021). *Machine learning* (S. Liu, Trans.). Springer Nature.