Journal of Theoretical and Applied Information Technology
15th October 2021. Vol.99. No 19
© 2021 Little Lion Scientific

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NEW METHODOLOGY FOR EFFICIENT FLOOD RISK MAPPING USING MACHINE LEARNING TECHNIQUES

[1]SWATI SHARMA, [2]VINEET SHARMA

[1]Assistant Professor, Department of Computer Science and Engineering, KIET Group of Institutions, 13 km milestone Delhi-Meerut Road, Moradabad, Ghaziabad, UP, (India) 201 002

[2]Professor, Department of Computer Science and Engineering, KIET Group of Institutions, 13 km milestone Delhi-Meerut Road, Moradabad, Ghaziabad, UP, (India) 201 002

E-mail: [1]Swati.sharma@kiet.edu, [2]Vineet.sharma@kiet.edu

ABSTRACT

The flood is the cause of destruction for many places in the world. Flood prediction is a complex method due to its nature. The flood arrives with vast destruction in the society. The flood assessment is an essential task for the government bodies to take measures at the right time. In this study, a new procedure is designed to choose the best machine learning model for efficient flood mapping using machine learning techniques. An automated algorithm is created using a Decision Tree, Random Forest, Gradient Boosting Classifier, and KNN techniques. The Random Forest Classifier gave the highest accuracy with 88.58% with this (the considered) dataset. For the evaluation of the model, confusion matrix, learning curve, and classification reports are used. In this study, Open Flood Risk by postcode dataset is used.

Keywords: *Flood Risk Mapping, Machine Learning, Decision Tree, Random Forest, KNN*

1. INTRODUCTION

Our planet has experienced a drastic increase in flooding and extensive rainfall of more than 50%
over the decade. Now, these calamities have become four times more frequent than they used to be in the 80s. Moreover, the flood is known to the world as the most common natural disaster which affects human lives and creates heavy economic
damage. The frequency of this calamity is expected to soar due to unplanned development and urbanization, increased deforestation, and continued precipitation [1]. Due to the devastating nature of the flood, it has to be analyzed and predicted to prevent damage, affecting the susceptible region's economic and mental levels and its inhabitants. Prediction of the flood beforehand can provide inhabitants enough time to evacuate the sensitive area [2].

Flood hazard risk assessment plays a vital role in ensuring the healthy and sustainable development of human society. This risk assessment is calculated by the probability of flood occurrence [3]. Therefore, flood hazard risk assessment will be an effective solution for predicting the flood. Furthermore, it is a qualitative or semi-quantitative method that considers combining the influence of disaster-inducing factors and hazard-inducing environments [4]. Thus, it can be very beneficial in flood insurance, floodplain management, disaster warning systems,

and evacuation plans, providing an effective decision-making technique [5]. Applying Machine Learning Algorithms has significantly increased in the last decade.

Researchers are using ML algorithms to solve different kinds of real-life problems because these algorithms can drastically improve computation and saving precious time and effort. Moreover, these algorithms can better solve non-linear problems, which is a significant advantage for flood risk management systems as we have to work with a multi-variable and non-linear relationship between indices and risk levels. Eventually, ML algorithms can be leveraged in applying Flood hazard risk assessment, saving precious time and effort and beneficial while working on non-linear relationships
.

An automated machine learning model created using Decision Trees, Random Forest, Gradient Boosting classifiers, and KNN is proposed. The idea is to give data to the proposed model, and it will further provide data for its different algorithms for training. After the training process, it will save all the accuracies and give the best accuracy model using the validation data.

The postal code flood risk dataset is taken from the Environment Agency's Risk of Flooding from Rivers and Sea, which allocated a risk level to England, UK. Conclusively, this study's prominent plan and goal is to create an automated machine learning model using the best machine learning algorithms and successfully classify flood risk in the susceptible area.

This paper justifies with following important contributions:

1. This paper will work as a source of reference for research scholars who wanted to contribute to advanced computer-based techniques of flood detection.
2. This paper is contributing an important procedure to accommodate all machine learning algorithms and give the best classifier model to predict the flood risk.
3. This paper is reducing the trouble of coding each classifier individually and then deciding the best classifier among them.
4. This procedure is scalable and many models can be changed as per requirement analysis.

The rest of the paper is structured into five different sections. In section two, related works are mentioned, which is giving various pieces that are already done. In section 3 proposed methodology of the work is discussed in detail. Section 4 is about the results and analysis which are done during the flood risk mapping. Finally, the conclusion of this paper is given in section 5. In the end, references are cited.

2. Literature Survey

With the motto to find out some significant works that have been done earlier; crucial studies that have been made regarding flood prediction using machine learning techniques are discussed below:

Ebenezer Danso-Amoako et al. (2012) publish
-ed a paper that introduces a rapid expert-based assessment method supported by an artificial neural network (ANN) model for dam failure of SFRB (Sustainable Flood Retention Basins). They used Dam Height, Dam length, Mean Annual Rainfall, flood water surface area, etc., as input features. As a result, the ANN model got a cross-validation (R^2) value of 0.70, implying that the tool is likely to predict variables for new datasets [6].

Sungwon Kim and Vijay P. Singh (2013) published a paper that uses neural network models and class segregation of the former to forecast floods. They used MLP-NNM (Multilayer perceptron-neural networks model), GRNNM (Generalized Regression Neural Networks Model), and KSOFM-NNM (Kohonen Self-Organizing Feature Maps Neural Network Models). KSOFM-NNM turns out to be more accurate than MLP-NNM and GRNNM for the testing data of Methods I and II for single conventional application and class segregation implementation [7].

Milad Jajarmizadeh et al. (2014) published a paper on comparison between the performance of Support Vector Machine (SVM) and Soil and Water Assessment Tool (SWAT) models. Data was collected over 19 years (1990-2008) on a particular region in Iran. After training and testing, the training accuracy of SVM is 98%, and the testing accuracy is 84%, and the SWAT model gets training accuracy is 92%, and the testing accuracy is 83% [8].

Sanjeet Kumar et al. (2015) published a paper to develop an ensemble modeling approach for reservoir inflow forecasting. In this paper performance of bootstrap-based wavelet artificial neural network (BWANN) is compared with wavelet-based ANN (WANN), wavelet-based MLR (WMLR), bootstrap and wavelet analysis based multiple linear regression models (BWMLR), standard ANN, and standard multiple linear regression (MLR) models for inflow forecasting. After using all methodologies, the accuracy of MLR is 76%, WMLR is 80%, BWMLR is 80%, WANN is 93%, and BWANN is 86% [9].

K.S. Kasiviswanathan et al. (2016) published a paper in which WNN (wavelet-based neural network) is leveraged through BB (block bootstrap sampling) to forecast streamflow. A comparison is also shown between WNN and ANN (artificial neural network), combining ensemble methods using BB. The BB is used to sample data to create an ensemble of data. The result is in the form of 7-day lead-time forecasting. In every evaluation metric, ANN-BB was significantly outperformed by WNN-BB, particularly when forecasting high flows in the long lead-time forecasting. For example, WNN-BB got RMSE of 49.21 in Flood I, 19.31 in Flood II, and 12.70 in Flood III; on the other hand, ANN-BB got 95.47 in Flood I, 41.92 in Flood II, and 27.54 in Flood III [10].

Shasha Han and Paulin Coulibalya (2017) published a paper that laid down a comprehensive study on all Bayesian forecasting methods applied for forecasting floods from 1999 to 2016. Conclusively, they found out that the Bayesian flood forecasting approach is an effective and advanced way for flood estimation as it takes into account all the sources of uncertainties and generates a predictive distribution of the river stage; river discharge, therefore, produces more reliable and accurate flood forecasts [11].

Amir Mosavi et al. (2018) reviewed ANN (Artificial Neural Network), MLP (Multilayer perceptron), ANFIS (Adaptive neuro-fuzzy inference system), WNN (Wavelet Neural Network), SVM (Support vector machine), Decision tree, and EPS (Ensemble prediction systems) algorithms. They compared them by training then flood resource variables. The basic methodology they applied was dividing flood prediction into short and long term and then further dividing that into single and hybrid methods. They found out that ANN is the most popular algorithm used in research, but hybrid algorithms gain more popularity than single algorithms [12].

JeeranaNoymanee and Thanaruk (2019) published a paper aiming to improve the existing system, uses hydrological modeling amplified through machine learning algorithms. They used five algorithms, i.e., Linear Regression, Neural Network Regression, Bayesian Linear regression, and Boosted Decision Tree Regression. As the hydrological model (MIKE 11- NAM model) could not be improved because of inaccurate rainfall data, machine learning algorithms are used as an error forecasting model. The best algorithm was Bayesian linear regression which reduced the error from MIKE 11 22.02% and 44.40% for one day in advance [13].

Ho Jun Keum et al. (2020) published a paper that created a real-time flood map using the classification-based real-time flood prediction model. Geo-ANFIS (Geo-adaptive network-based fuzzy inference system), Linear, and Non-linear Regression were used to train the input data. To achieve this model, they combined EPA-SWMM (a hydraulic urban runoff analysis) with machine learning algorithms mentioned earlier to make this classification model. They used cumulative rainfall, representative cumulative, and LiDAR (Light Detection and Ranging) data as an input for their classification model. The Gamma test minimized the uncertainty of these data. The output was received as a 2D map which can predict rainfall-induced inundation potentially. After comparing it with a verified 2D flood model, this model attained a goodness-of-fit of 85% [14].

Based on previous research studies, it is observed that most of the experiments are conducted with various conventional techniques of flood detection and few machine learning techniques have been found to detect the flood on an individual basis. This creates a gap in research to design a process to select the fittest algorithm of machine learning. This will reduce the time to analyze the individual algorithms and then decide on the best algorithm. The procedure designed to solve these issues is scalable and valid to a variety of problems.

3. PROPOSED WORK:

3.1 Dataset Description

This dataset is collected by the Risk of Flooding from Rivers and Sea and Open Postcode Geo by the United Kingdom

Government and named Open Flood Risk by postcode dataset [15]. This data contains ten fields: postcode, FID, PROB_4BAND, SUITABILITY, PUB_DATE, RISK_FOR_INSURANCE_SOP, easting, northing, latitude, and longitude. Initially, the dataset was containing 10,48,575 instances with 41.4% missing values and metadata attributes. This dataset is improved after removing the missing values and remove the extra unwanted features. The dataset now has 71,762 instances with no missing values with six segments.

3.2 New Procedure for selecting best classifier

In the machine learning system, the biggest problem is the selection of algorithms. It is found that many times the data scientists are facing problems in selecting the best algorithm. A system can automatically select the best algorithm for flood mapping based on the geographical coordinates in this procedure. The architecture of the new process is given in Fig. 1.

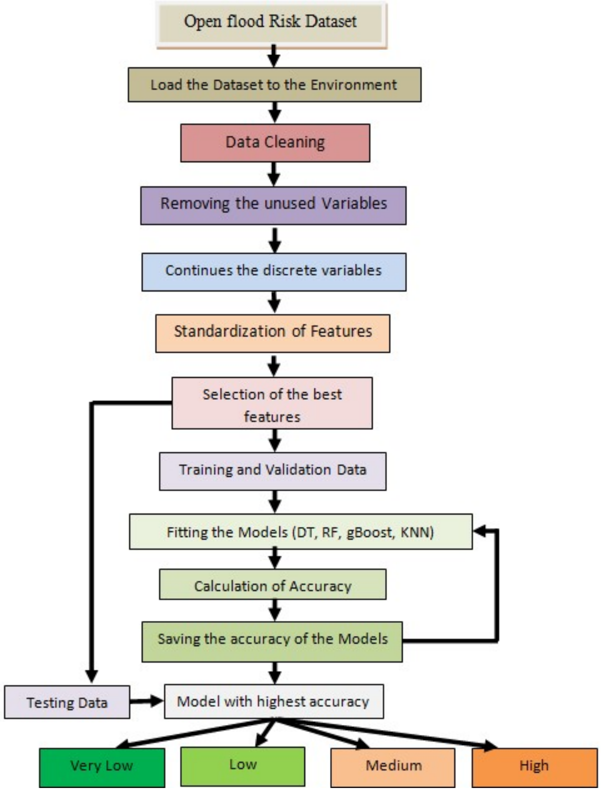

Open flood Risk Dataset
Load the Dataset to the Environment
Data Cleaning
Removing the unused Variables
Continues the discrete variables
Standardization of Features
Selection of the best features
Training and Validation Data
Fitting the Models (DT, RF, gBoost, KNN)
Calculation of Accuracy
Saving the accuracy of the Models
Testing Data
Model with highest accuracy
Very Low
Low
Medium
High

Fig. 1. New Methodology for the Flood Risk Mapping

During the creation of the model following are the main steps that are involved:

3.2.1. Data Cleaning

Initially, data was containing missing values and unwanted characters. During the data cleaning, all the instances with missing values have been removed [16]. In addition, some of the cases were filled by taking the 'mode' of a particular column.

3.2.2. Removing unused variables

The variables which were indicating the name, ids, etc., are removed from the feature. So, for example, in this dataset, postcode, FID, and PUB_DATE are drawn.

3.2.3. Continues Discrete Variables

The internal calculations inside an algorithm take place with numerical data. Hence using the one-hot encoder, discrete columns are continuous for obtaining one feature per value.

3.2.4. Standardization

A dataset is standardized when its 'mean' is zero, and the standard deviation is one. The standardization of features is done to scale the features in a single range.

3.2.4. Standardization

After the standardization, the number of features was increased. In this analysis, all the features are selected.

3.2.5. Splitting the data into training and testing

The dataset is split into the training and testing after the standardization to train and test the model. In this analysis, 70% of data is used in training, and 30% is to test the model.

3.2.6. Fitting the model

After dividing the data into training and testing, the models are fitted to calculate the accuracy. The different models used in the calculation are:

3.2.6.1 Decision Tree:

A Decision Tree is an efficient algorithm in handling a large amount of data. It is used for both classification and regression. The decision tree uses a top-down approach in the classification [17]. The structure of the decision tree contains two types of nodes: decision node and leaf node. The first decision node is called the root node. Different learning algorithms in the decision trees like ID3, CART, C4.5, etc. [18]. In this work, the ID3 algorithm is used. The two well-known methods used to build the decision tree are the Information Gain and Gini Index methods [19].

The formula for the calculation of the information gain is given in equation (1):

$$Entropy\,(S) = -p_i \log_2 p_i$$

$$G(S, A) = Entropy\,(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} * Entropy(S_v) \qquad (1)$$

Where S is the entropy of the sample, A is an attribute, p is the probability.

The formula for the calculation of the Gini index is given in equation (2):

$$Gini\,(N) = 1 - \sum_{t \in targetvalues} (1 - p(t))^2$$

$$Gini\,(A) = \sum_{f \in feature\ values} \frac{|S_f|}{|S|} * Gini(N_f) \qquad (2)$$

In this work, the Information Gain method is used.

3.2.6.2 Random Forest:

Random Forest is an ensemble learning classifier. Ensemble learning means an orchestra of algorithms is acting together to solve a problem. In addition, random Forest is a bagging classifier. Bagging is a combination of bootstrap and aggregation [20].

Random Forest uses an orchestra of decision trees to solve a problem. In this orchestra, decision trees are designed in a parallel manner. First, the data for each decision tree is given by row sampling and column sampling with replacement [21]. Second, the feeding of data to the base classifier is called bootstrapping, and once the outputs are obtained from the base classifiers using the majority voting method, the output is obtained. An illustration of Random Forest is given in Fig. 2.

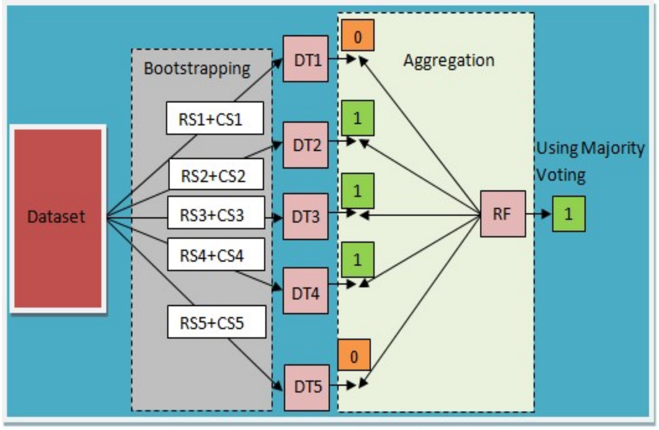


Fig. 2. Random Forest Classification

Suppose there is a binary classification of the data. Each of the decision trees is producing the output in terms of zero and one. If there is a maximum number of one from the classifier, then based on their majority, the prediction of the Random Forest classifier will be one.

The decision trees classify the data as having low bias and high variance at its maximum depth. Random Forest uses row sampling and column sampling in different decision trees [22].

3.2.6.3 Gradient Boosting Classifier:

Gradient boosting is an ensemble classifier. Ensemble techniques are of two types bagging and boosting. Boosting is an iterative procedure. In this method, the first uniform distribution of the probabilities is done on the given training instances and then adaptively changes the probability distribution of the training data [23]. Hence, each training instance has equal weight and then iteratively weight changes. There are different types of boosting algorithms like Adaboost, Gradient Boosting, Xtreme Gradient Boosting. In this work, a Gradient boosting classifier is used.

In Gradient boosting, base learners have generated sequentially so that the present base learner is always more effective than the previous one. In the GradientBoosting, the increment of the weights of misclassified items is not performed, but the loss function of the previous leaner is optimized by adding a new adaptive model that adds a weak learner to reduce the loss function. It has three main components:

1. Loss function
2. Weak Learner
3. Additive Learner

Let us now understand the GradientBoosting algorithm. Three inputs that are needed in this are data, loss function, and the number of trees.

Step 1) Initialize the model with constant values using the function in equation (3)

$$F_0(x) = argmin_\gamma\left(\sum L(y,\gamma)\right) \quad (3)$$

where L is a loss function.

Step 2) Compute the pseudo residual.

Step 3) Fit a base learner.

Step 4) Calculate the error of the base learner and minimize the loss.

Step 5) Update the model

3.2.6.4 K-Nearest Neighbors:

K-Nearest Neighbor is a supervised learning algorithm. K-Nearest Neighbor works on storing the data and then wait for the query point to find out the distances of all the points. Then, the majority is considered for the new query point [24].

If two of several nearest neighbors are the same in the majority voting method, then reduce the neighbors by 1. If two neighbors have the same distance, then that neighbor is selected first in the training instances.

KNN is also called the lazy learning algorithm as it waits for the query point for the predictions. Finally, the KNN is called the instance-based learner as it stores the instances for its predictions [25].

The calculation of the distances from the query point to the training instances is calculated by different formulae:

1. Euclidean distance:

$$\sqrt{\sum_{k=1}^{n}\left(x_k - y_k\right)^2} \quad (4)$$

2. Manhattan distance:

$$\sum_{k=1}^{n}\left|x_k - y_k\right| \quad (5)$$

3. Minkowski distance:

$$\left(\sum_{k=1}^{n}\left|x_i - y_i\right|^p\right)^{1/p} \quad (6)$$

In this study, equation 4 is used.

3.2.7. Calculation of Accuracy

The accuracy of each of the models is calculated with a confusion matrix. In addition, learning curves and classification reports are given to support the accuracy of the algorithms.

3.2.8. Saving the Accuracy

After calculating the accuracy of each of the models, accuracies are saved into a list to find the best accuracy model.

3.2.9. A model with the highest accuracy

The model with the highest accuracy is obtained, and a test dataset is given to this model to predict the output.

3.2.10. Prediction of Flood Risk

The flood risk predictions are obtained from the best accuracy model, and each of the class levels is very low, Low, Medium, and high.

3.2.11. Pseudocode for new procedure

```
Step 1) Read the data file
        df = read(datafilepath)
Step 2) Remove the instances with null values
        df.dropna(subset(coloumn_names))
Step 3) Remove the unused variables
        df.drop(postcode)
        df.drop(FID)
        df.drop(PUB_DATE)
Step 4) continues the discrete variables
        cate=[column_names]
        enc=Onehotencoder()
        enc_obj=enc.fit_transform(df[cate])
        df.cate=df.dataframe(enc_obj.toarray())
        df=df.join(df.cate)
Step 5) Selection of features
        x=[features]
        y=[target]
Step 6) Split into training and testing
        x_train,y_train,x_test,y_test=train_test_split(x,y,test_ssize=0.3)
Step 7) selects the model
        clf = modelname()
Step8) Fit the model
        clf.fit(x_train,y_train)
Step 9) Saving the accuracy
         list=[model_accuracies]
        list_models=[Model_names]
Step 10) Model with highest accuracy
        Index_max_model=index(max(list))
        Max_model=list_models[index_max_model]
Step 11) prediction of results
        Max_model.predict(x_test)
```

4. Results and Analysis

To find out the maximum accuracy algorithm, algorithms have been analyzed in terms of testing and cross-validation accuracy, confusion matrix, classification report, roc accuracy, learning curve, scalability curve of the model, and performance curve of the model. The accuracy is calculated from the confusion matrix and is given as equation (7):

$$\text{Accuracy} = \frac{(TP+TN)}{(TP+FN)+(FP+TN)} \qquad (7)$$

Where TP = True Positive

TN = True Negative

FP = False Positive

FN = False Negative

In the classification report, precision, recall, and f1 score are mentioned. The formula used in these calculations is given as equations (8), (9), (10):

$$\text{Precision} = \frac{TP}{TP+FP} \qquad (8)$$

$$\text{Recall} = \frac{TP}{TP+FN} \qquad (9)$$

$$\text{F1 score} = \frac{2*(Precision*Recall)}{(Precision+Recall)}$$

(10)

As already discussed in this work, four algorithms have been used for finding the best accuracy model. Let us discuss the result of each of the algorithms one by one:

4.1 Decision Tree Classifier

The confusion matrix of the decision tree classifier is given in Fig. 3.

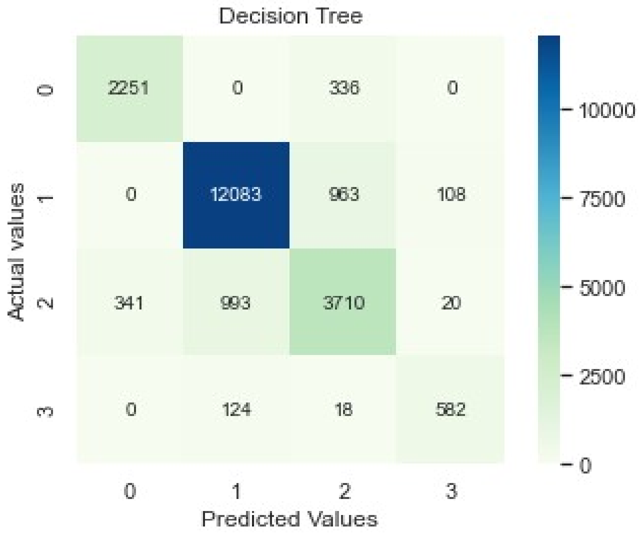


Fig. 3. Confusion Matrix of decision Tree

The accuracy is calculated using equation (7)

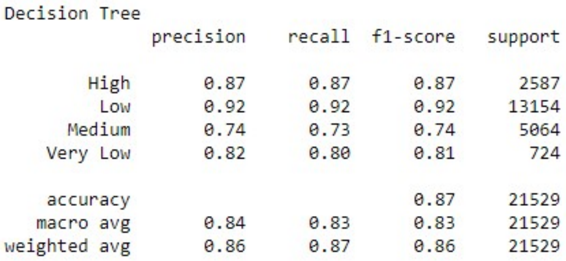
$$\text{Accuracy} = \frac{(2251+12083+3710+582)}{(2587)+(13154)+(5064)+(724)}$$

$$\text{Accuracy} = \frac{18626}{21529} = 0.8651$$

This shows that the testing accuracy achieved is 86.51%. However, on ten-fold cross-validation, accuracy is 85.99%.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The classification report of the decision tree is given in Fig. 4.



```
Decision Tree
              precision    recall  f1-score   support

        High       0.87      0.87      0.87      2587
         Low       0.92      0.92      0.92     13154
      Medium       0.74      0.73      0.74      5064
    Very Low       0.82      0.80      0.81       724

    accuracy                           0.87     21529
   macro avg       0.84      0.83      0.83     21529
weighted avg       0.86      0.87      0.86     21529
```

Fig. 4. Classification Report of Decision tree

The roc accuracy score (macro average) obtained in the decision tree is 88.12%. The learning curve, scalability of the model, and performance of the model are plotted in Fig. 5, 6, 7, respectively.

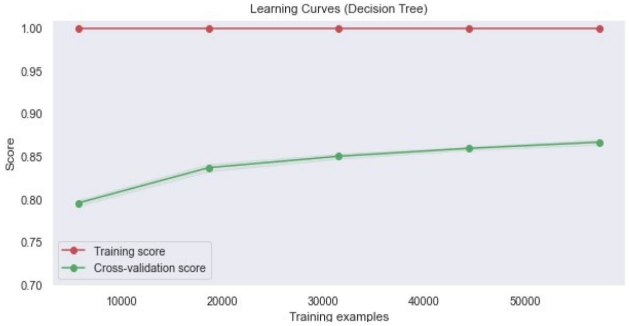


Fig. 5. The learning curve of Decision Tree

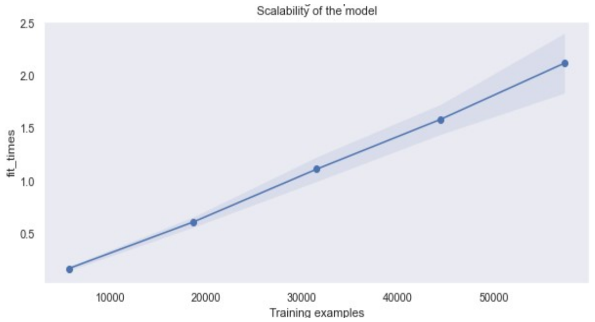


Fig. 6. Scalability of decision tree Model

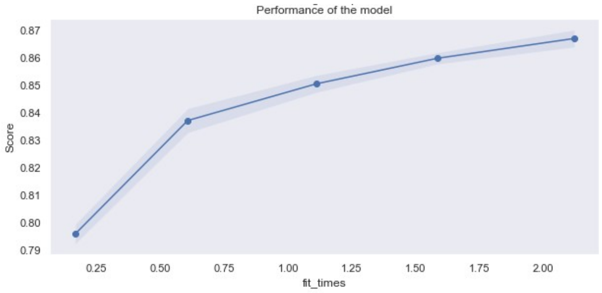
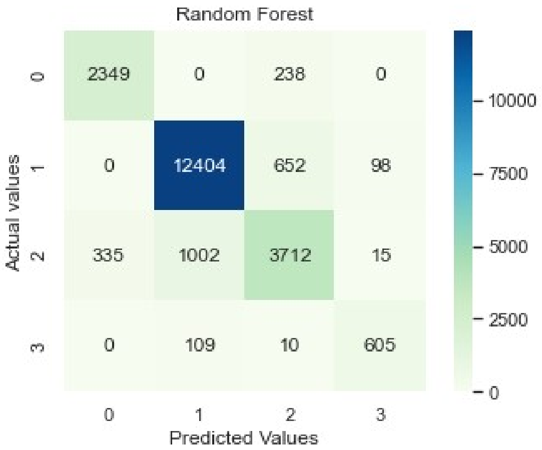


Fig. 7. Performance of decision tree Model

4.2 Random Forest Classifier

The confusion matrix of the random forest classifier is given in Fig. 8.



Random Forest

Actual values	0	1	2	3
0	2349	0	238	0
1	0	12404	652	98
2	335	1002	3712	15
3	0	109	10	605

Predicted Values

Fig. 8. Confusion Matrix of Random Forest Classifier

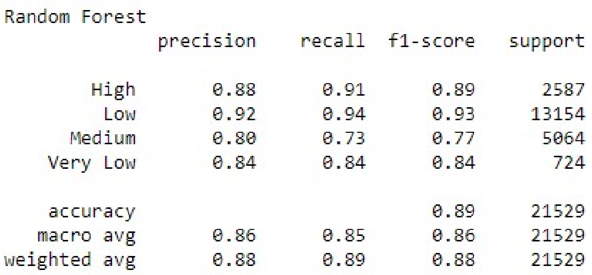
The accuracy is calculated using equation (7)

$$\text{Accuracy} = \frac{(2349 + 12404 + 3712 + 605)}{(2587) + (13154) + (5064) + (724)}$$

$$\text{Accuracy} = \frac{19070}{21529} = 0.8857$$

This shows that testing accuracy achieved is 88.57%. However, on ten-fold cross-validation, accuracy is 88.20%.

The classification report of the random forest is given in Fig. 9.



```
Random Forest
              precision    recall  f1-score   support

        High       0.88      0.91      0.89      2587
         Low       0.92      0.94      0.93     13154
      Medium       0.80      0.73      0.77      5064
    Very Low       0.84      0.84      0.84       724

    accuracy                           0.89     21529
   macro avg       0.86      0.85      0.86     21529
weighted avg       0.88      0.89      0.88     21529
```

Fig. 9. Classification Report of Random Forest

The roc accuracy score (macro average) obtained in the random forest is 96.30%. The learning curve, scalability of the model, and performance of the model are plotted in Fig. 10, 11, 12, respectively.

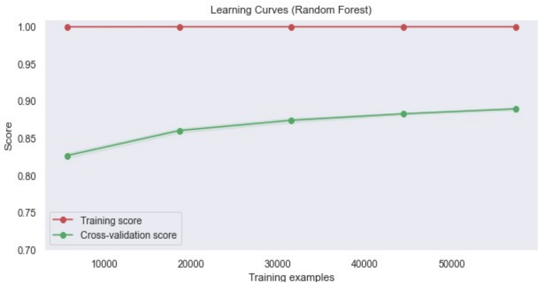


Fig. 10. The learning curve of Random Forest

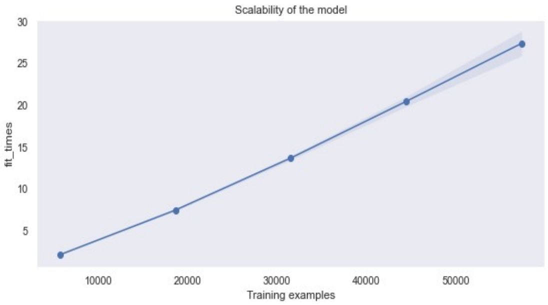


Fig. 11. Scalability of random forest model

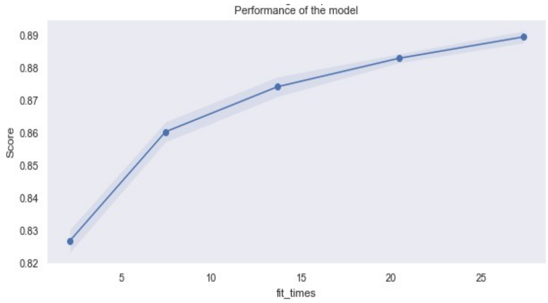


Fig. 12. Performance of random forest model

4.3 Gradient Boosting Classifier

The confusion matrix of the Gradient boosting forest classifier is given in Fig. 13.

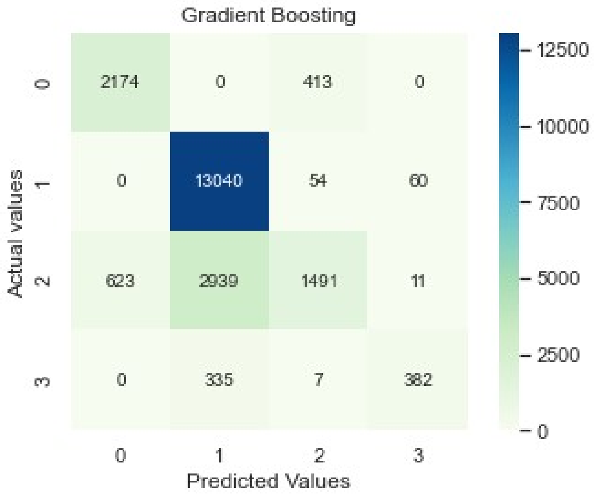


Fig. 13. Confusion Matrix of Gradient Boosting Classifier

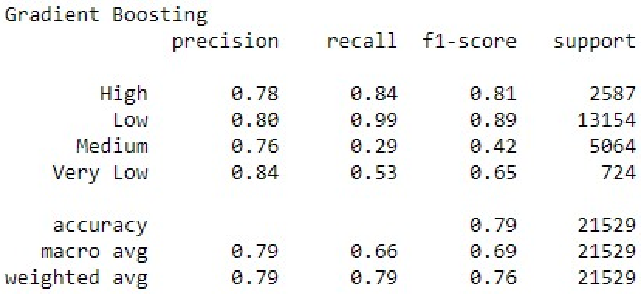
The accuracy is calculated using equation (7)

$$\text{Accuracy} = \frac{(2174 + 13040 + 1491 + 382)}{(2587) + (13154) + (5064) + (724)}$$

$$\text{Accuracy} = \frac{17087}{21529} = 0.79367$$

This shows that testing accuracy achieved is 79.37%. However, on ten-fold cross-validation, accuracy is 79.32%.

The classification report of the GradientBoosting is given in Fig. 14.



```
Gradient Boosting
              precision    recall  f1-score   support

        High       0.78      0.84      0.81      2587
         Low       0.80      0.99      0.89     13154
      Medium       0.76      0.29      0.42      5064
    Very Low       0.84      0.53      0.65       724

    accuracy                           0.79     21529
   macro avg       0.79      0.66      0.69     21529
weighted avg       0.79      0.79      0.76     21529
```

Fig. 14. Classification Report of Gradient Boosting

The roc accuracy score (macro average) obtained in the GradientBoosting is 89.84%. The learning curve, scalability of the model, and performance of the model are plotted in Fig. 15, 16, 17, respectively.

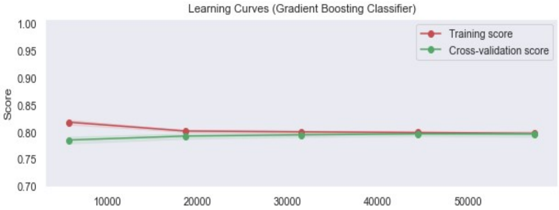


Fig. 15. The learning curve of Gradient Boosting

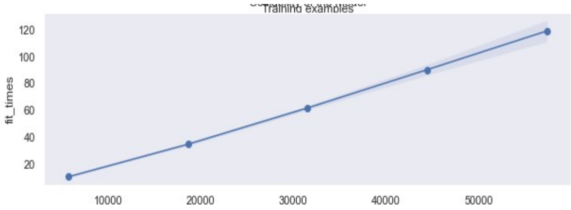


Fig. 16. Scalability of Gradient boosting model

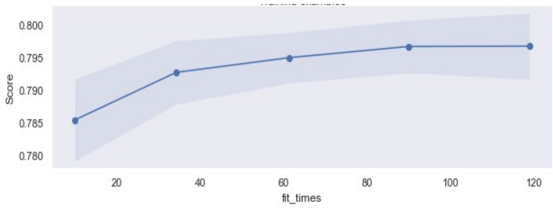


Fig. 17. Performance of Gradient boosting model

1. 4.4 K-Nearest Neighbor (KNN)

The confusion matrix of the KNN classifier is given in Fig. 18.

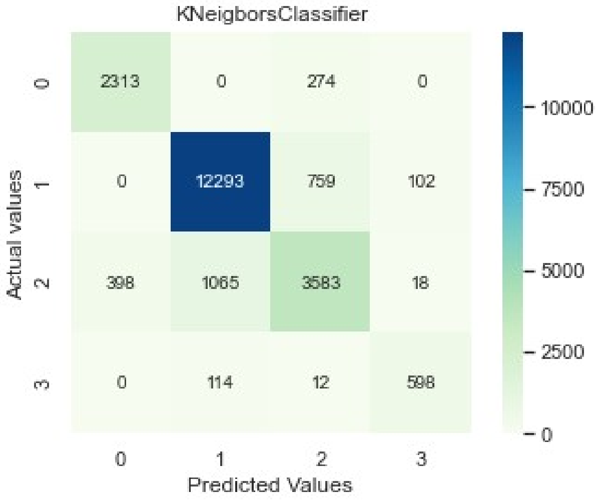


Fig. 18. Confusion Matrix of KNN Classifier

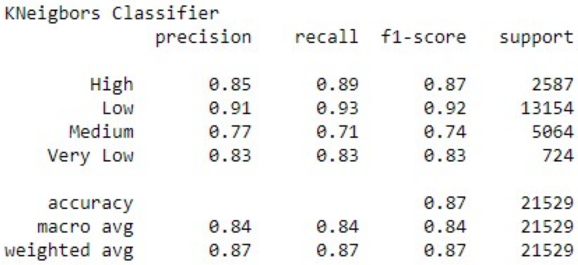
The accuracy is calculated using equation (7)

$$\text{Accuracy} = \frac{(2313 + 12293 + 3583 + 598)}{(2587) + (13154) + (5064) + (724)}$$

$$\text{Accuracy} = \frac{18787}{21529} = 0.8726$$

This shows that the testing accuracy achieved is 87.26%. However, on ten-fold cross-validation, accuracy is 86.52%.

The classification report of the KNN is given in Fig. 19.



```
KNeigbors Classifier
              precision    recall  f1-score   support

        High       0.85      0.89      0.87      2587
         Low       0.91      0.93      0.92     13154
      Medium       0.77      0.71      0.74      5064
    Very Low       0.83      0.83      0.83       724

    accuracy                           0.87     21529
   macro avg       0.84      0.84      0.84     21529
weighted avg       0.87      0.87      0.87     21529
```

Fig. 19. Classification Report of KNN

The roc accuracy score (macro average) obtained in the KNN is 94.49%. The learning curve, scalability of the model, and performance of the model are plotted in Fig. 20, 21, 22, respectively.

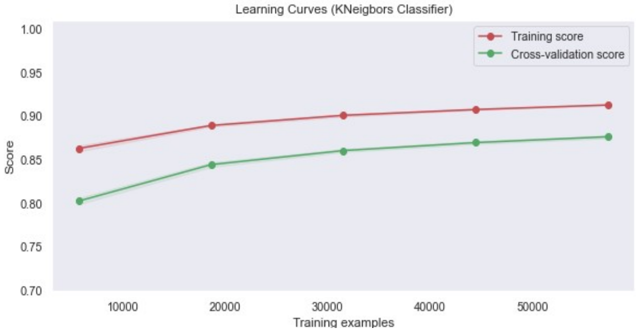


Fig. 20. The learning curve of the KNN Model

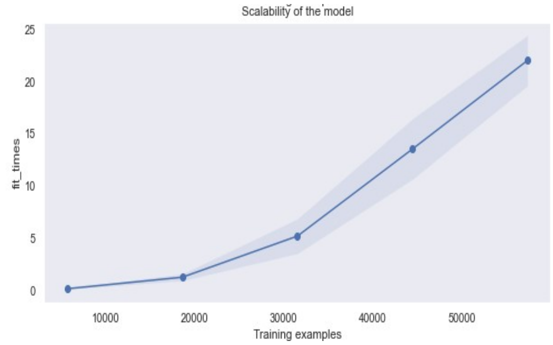


Fig. 21. Scalability of KNN Model

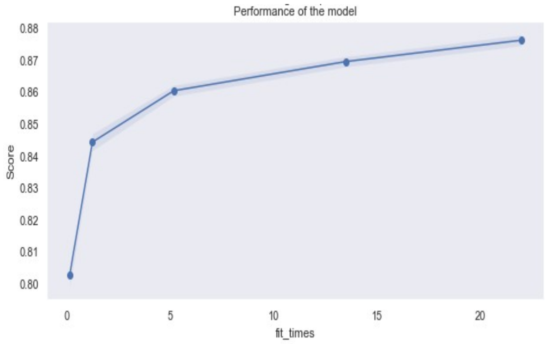


Fig. 22. Performance of KNN Model

A detailed analysis in the tabular form of the algorithms is given in Table 1.

Models	Cross-Validation	Testing	ROC accuracy
Decision tree	85.99%	86.52%	88.12%
Random forest	**88.20%**	**88.58%**	**96.30%**
Gradient Boosting	79.32%	79.37%	89.84%
K- Nearest neighbor	86.52%	87.26%	94.94%

Table 1. Analysis of Results

With this analysis on this dataset, it is found that the Random Forest Classifier is showing its highest accuracy. Hence it is selected as the best classifier to map the flood risk.

4.5 Comparison with previous works:

According to the free lunch theorem, it is said that none of the algorithms is the best fit on all the datasets. Hence, it is essential to note that two works can be satisfactorily compared when they have the same environment of experiments in terms of data, platform, etc. Although, a comparison of recent year work is given in Table 2.

Author	Model name and accuracy	Dataset
Ho Jun Keun et al. [14]	Geo-ANFIS (85%)	LiDAR
Proposed Work	Random Forest (88.58%)	Open Flood Risk by postcode

Table 2. Comparison of works

5. CONCLUSION AND FUTURE SCOPE

This work has been done with the hypotheses of building a new procedure for selecting the best algorithm by itself to reduce the analysis time of the algorithms individually. In this present work, from the analysis of each algorithm done individually, it is found that Random forest has the highest accuracy given by the procedure. The different models' Decision tree, Random Forest, Gradient Boosting, and KNN gave the 86.52%, 88.58%, 79.37%, and 87.26% accuracy. Therefore, the procedure designed was working correctly and provided an accuracy of 88.58% with Random Forest Classifier. The other algorithms also show better results and sometimes it is not mandatory that based on the highest accuracy we can select the best algorithm. In many cases, it is seen that f1 score, precision, and recall also play an important role. But the result obtained in this work is satisfactory as it is satisfying all the requirements of the best classifier.

In the advancement of this work, it is suggested to improve the model's accuracy as it has about 12% of misclassification. This may be achieved by using hybrid algorithms, any advanced ensemble technique with tree model, deep learning models, and can also be done by changing the dataset or increasing the data.

This process has some limitations like it will take some time in finding the best algorithm. Hence, it may have some performance issues. It is a single process being designed to deal with multiple algorithms hence, algorithms that are sensitive to outliers and missing values need to be taken care of earlier. Flood risk is a major problem in India and its features of prediction vary with geographical conditions hence, while collecting the data for flood risk prediction it should be taken care of before fitting the algorithm.

6. ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Dr. Vineet Sharma, who guided me throughout this research work. I would also like to thank my friends and family who supported me and offered deep insight into the study.

REFERENCES

[1] Mahyat Shafapour Tehrany et al. (2014) "Flood susceptibility assessment using support vector machine model with different kernel types," Catena, Vol. 125, Pages 91-101.

[2] Mohammed Khalaf et al (2015). "Advance Flood Susceptibility Detection and Notification System based on Sensor Technology and Machine Learning Algorithm," International Conference on Systems, Signals and Image Processing (IWSSIP), Page 105-109.

[3] Zhaoli Wang et al (2015). "Flood hazard risk assessment model based on random forest," Journal of Hydrology, Volume 527, Pages 1130-1141.

[4] Stephane Hallegatte et al (2013). "Future Flood losses in major coastal cities," Nature Climate Change, Vol. 3, Issue 9, Pages 802-806.

[5] Qiang Zou et al (2013). "Comprehensive flood risk assessment based on set pair analysis-variable fuzzy sets model and fuzzy AHP. Stoch Environ Res Risk Assess, Vol. 27, Pages 525-546.

[6] Ebenezer Danso-Amoako et al (2012). "Predicting dam failure risk for sustainable flood retention basins: A generic case study for the wider Greater Manchester area," Computers Environment and Urban Systems, Vol. 36, Issue 5, Page No: 423-433.

[7] Sungwon Kim et al (2013). "Flood Forecasting Using Neural Computing Techniques and Conceptual Class Segregation," Journal of the American Water Resources Association (JAWRA), Vol. 49, Issue 6, Pages 1421-1435.

[8] Milad Jajarmizadeh et al (2014). "Application of SVM and SWAT models for Monthly Streamflow Prediction," KSCE Journal of Civil Engineering, 19, 345-357.

[9] Sanjeet Kumar et al (2015). "Reservoir Inflow Forecasting Using Ensemble Models Based on Neural Networks, Wavelet Analysis, and Bootstrap Method," Water Resources Management, Vol. 29, Pages 4863-4883.

[10] K.S. Kasiviswanathan et al (2016). "Potential application of wavelet neural network ensemble to forecast streamflow for flood management," Journal of Hydrology, Vol. 536, Pages 161-173.

[11] Shasha Han, Paulin Coulibaly (2017) "Bayesian Flood Forecasting Methods: A Review, Journal of Hydrology, Vol. 551, Pages 340-351.

[12] Amir Mosavi et al (2018). "Flood Prediction Using Machine Learning Models: Literature Review," Water, Vol. 10, Issue 11, Page 1536.

[13] Jeerana Noymanee, Thanaruk Theeramunkong (2019) "Flood Forecasting with Machine Learning Technique on Hydrological Modeling," Procedia Computer Science, Vol. 156, Pages 377-386.

[14] Ho Jun Keum et al (2020). "Real-Time Flood Disaster Prediction System by Applying Machine Learning Technique," KSCE J Civ Eng, Vol. 24, Pages 2835-2848.

[15] https://www.getthedata.com/open-flood-risk-by-postcode

[16] Malley B., Ramazzotti D., Wu J.T. (2016) Data Pre-processing. In: Secondary Analysis of Electronic Health Records. Springer, Cham.

[17] G. Chaubey, Dhanajay Bisen, Siddhartha Arjaria, and Vibhash Yadav (2020), "Thyroid Disease Prediction Using Machine Learning Approaches," National Academy Science Letters.

[18] Podgorelec V., Zorman M. (2012) Decision Trees. In: Meyers R. (eds) Computational Complexity. Springer, New York, NY.

[19] Panhalkar, A.R., Doye, D.D (2020). A novel approach to build accurate and diverse decision tree forests. *Evol. Intel*.

[20] Stefanowski J. (2007) Combining Answers of Sub-classifiers in the Bagging-Feature Ensembles. In: Kryszkiewicz M., Peters J.F., Rybinski H., Skowron A. (eds)(2007) Rough Sets and Intelligent Systems Paradigms. RSEISP. Lecture Notes in Computer Science, vol 4585. Springer, Berlin, Heidelberg.

[21] Rebala G., Ravi A., Churiwala S. (2019) Random Forests. In: An Introduction to Machine Learning. Springer, Cham.

[22] Dhivyaa, C.R., Sangeetha, K., Balamurugan, M. *et al (2020)*. Skin lesion classification using decision trees and random forest algorithms. *J Ambient Intell Human Comput*.

[23] Bahad P., Saxena P. (2020) Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics. In: Singh Tomar G., Chaudhari N., Barbosa J., Aghwariya M. (eds) (2019) International Conference on Intelligent Computing and Smart Communication. Algorithms for Intelligent Systems. Springer, Singapore.

[24] Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine.*

[25] Adeniyi, D., A., Wei, Z., Yongquan, Y. (2016). Automated web usage data mining and Recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics.*