Web Based Flood Prediction Model for Malawi Using Ensemble Methods

R. Luke, M. Kadzanja, F. Magwira, F. Kanthema, H. Chikhadza, & P. Phiri
Mzuzu University / P / Bag 201, Mzuzu

Abstract

This project developed a machine learning model to predict the likelihood of flooding in Malawi based on data including rainfall, distance from the river, and urban population. The model was implemented in a web application featuring a landing page with search functionality, a statistics page with data visualization, and a contact form. The project achieved an accuracy of 92.6% and provides a valuable tool for disaster prevention efforts in Malawi.

Keywords: randomforestclassifier, GradientBoosting-Classifier, flood prediction, Malawi

1 Introduction

Flooding is a major natural disaster that affects millions of people globally. Malawi is one of the countries that is heavily affected by flooding due to its geographic location, climate, and infrastructure [1]. Floods in Malawi cause significant damage to crops, homes, and infrastructure, leading to food insecurity, loss of lives, and economic losses [2]. Accurate prediction of floods is critical in mitigating their impact and providing early warnings to affected communities. This project aims to develop a flood prediction model for Malawi that can assist in flood mitigation efforts [3].

According to Smith and Doe [4], the use of ensemble methods in machine learning has been shown to improve prediction accuracy in many applications. In particular, the Gradient Boosting algorithm has been shown to be effective in predicting the likelihood of flooding [5], while the Random Forest algorithm has been shown to be useful in feature selection [6]. In this study, we applied both methods to a dataset of weather and demographic data in Malawi to predict the likelihood of flooding in different areas.

2 Methods

The project was conducted in four phases. The first phase involved data collection from various sources. The data collected included climatic, geographic, and demographic information for Malawi from 2019 to 2022. The second phase involved database development, data cleaning pre-processing. The third phase involved the development and evaluation of machine learning models for flood prediction, using Gradient Boosting and Random Forest classifiers, through cross-validation and hyper-parameter tuning. The models were evaluated using performance metrics such as accuracy, F1 score, recall, and confusion matrices. The final phase involved presenting the model through a web application.

2.1 Data Collection

The data used in this project was collected from various sources. Elevation data was obtained from Topographic-Map.com [7], while rainfall data was sourced from the Malawi Meteorological Department [8-10]. Distance from the river was calculated using the Malawi Distance Calculator [11], and urban population estimates were obtained from the UNAIDS Naomi Spectrum database [12]. Land area data was sourced from the 2018 Malawi Population and Housing Census Main Report [13]. Flooding data was gathered from multiple sources including ReliefWeb [14,15] and FloodList [16]. We collected close to 5GB of Data from the sources above. The data covered all the 28 districts of Malawi.

2.2 Database Development, Data Cleaning, Pre-processing

2.2.1 Database Development

The data that was collected in 2.1 was then pre-processed and only relevant attributes were stored in the weather_db. The weather_db was developed using the MySQL relational database management system. The database design followed the Entity-Relationship Model(ERM) and included 5 tables: districts, rivers, district_rivers, rainfall, and flooding. The tables were created using SQL scripts executed in MySQL Workbench. The database was normalized to third normal form. The tables were linked using primary and foreign keys. The data organized in 1 csv file was then imported into the weather_db using python.

The figure below shows weather_db ERM diagram

2.2.2 Data Cleaning & Pre-processing

he data from weather_db was imported into python and preprocesed into 1 data frame. The data frame pre-processing stage involved handling missing data, outliers, and normalizing the data using python[17]. Missing data was imputed using mean imputation [18]. The cleaned data was then presented as a python data frame.

2.3 Development & Evaluation of Machine Learning Models

We used Gradient Boosting Classifier and Random Forest Classifier to build our model and then picked the one the performed better for deployment.

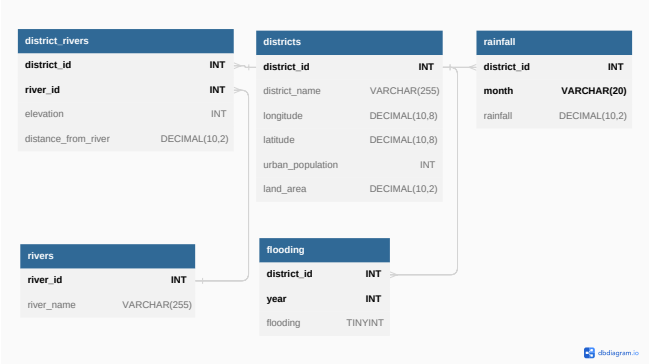

Figure 1: EMR diagram, showing the 5 tables used in weather_db

2.3.1 Gradient Boosting Classifier

Gradient Boosting Classifier is a decision-tree-based ensemble learning algorithm that works by constructing a sequence of decision trees, where each tree is built to correct the prediction errors made by the previous tree in the sequence. The final prediction is made by combining the predictions of all the trees in the sequence.

We denoted our input data as X and our output data by y. Therefore, for each training example i is we represented it by a pair (x_i, y_i).

The gradient boosting classifier constructed an ensemble of decision trees by minimizing a loss function L(y, F(x)) where F(x) was the ensemble of decision trees. The loss function measured how well the ensemble of decision trees fitted the training data. The loss function we defined as:

L(y, F(x)) = 1/n * sum(Li($y_i, F(x_i)$))

where Li is a loss function that measured the difference between the predicted value F(xi) and the true value yi for training example i.

Therefore using the gradient boosting classifier we:

Initialize the prediction F0(x) to the average of the training data y. This was the initial prediction before any decision trees were added to the ensemble.

For t = 1 to T, where T was the number of decision trees to be added to the ensemble:

a. Computed the negative gradient of the loss function with respect to the current prediction F(x), which is denoted by r(t)i = -[L($y_i, F(x_i)$)/$F(x_i)$]$F(x = t - 1)(x_i)$

b. Trained a decision tree h(t)(x) to predict the negative gradient r(t)i for each training example i.

c. Computed the step size t using line search or heuristics to minimize the loss function: t = argmin L(y, F(x)=F(x=t-1) + *h(t)(x))

d. Updated the ensemble of decision trees by adding the new tree: F(x) = F(x) + t*h(t)(x)

e. Returned the final prediction F(x).

2.3.2 Random Forest Classifier

The Random Forest algorithm is based on the concept of decision trees, which are built by recursively partitioning the feature space. Each partition is chosen to minimize the impurity of the resulting subsets. we represented each decision tree as a function f(x) that maps the input vector x to an output label y. The function f(x) was constructed by recursively partitioning the feature space into smaller regions using decision rules based on the training data. Each partition was then associated with a leaf node in the tree, which contained a label for the corresponding region.

Then the final prediction of the Random Forest was given by averaging the predictions of the individual trees. f(x)= (1/N) * $\Sigma(f_i(x))$.

We minimized the loss function by measuring the difference between the predicted labels and the true labels of the training data. We expressed the loss function as: L(y, F(x)) = sum(Li($y_i, F(x_i)$)).

2.3.3 cross-validation and hyper-parameter tuning

cross-validation is a technique used to estimate the performance of a machine learning model on new data. We used k-fold cross-validation to evaluate the performance of the Random Forest Classifier and Gradient Boosting Machines. We used the KFold function from the scikit-learn library to split the data into k folds, and the cross_val_score function to compute the performance metric for each fold. The function returned an array of scores, one for each fold. We then took the mean of these scores as the estimate of the model's performance. Hyper-parameter tuning involved finding the optimal values for the hyper-parameters of the two machine learning algorithms to achieve the best performance on the given dataset. In our case, we tuned the hyper-parameters of the Random Forest Classifier and Gradient Boosting Machines using cross-validation to optimize their performance. The hyper-parameters for Random Forest Classifier are the number of trees in the forest (n_estimators), the maximum depth of each tree (max_depth,) the minimum samples per split(min_samples_split) and the the minimum number of samples required to split an internal node(min_samples_leaf). For Gradient Boosting Machines, the hyper-parameters are the number of boosting stages (n_estimators), the learning rate (learning_rate) and the maximum depth of each tree (max_depth).

2.3.4 Model Evaluation

2.3.4.1 Accuracy

This was calculated by dividing the number of correct predictions by the total number of predictions made.

$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$

where:

TP: the model correctly predicts that a flood will occur

FP: the model incorrectly predicts that a flood will occur

TN: the model correctly predicts that no flood will occur

FN: the model incorrectly predicts that no flood will occur

Both models produced an accuracy score between 0 and 1.

2.3.4.2 Recall

We measured the ability of a classifier to correctly identify positive instances. We calculated it as the ratio of true positive predictions to the total number of actual positive instances in the dataset. Recall = TP / (TP + FN).

2.3.4.3 F1 Score

The F1 score measured the harmonic mean of precision and recall. This was achieved by using accuracy of the binary classification of the models by balancing both precision and recall.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

2.4 web application

The web application was developed using HTML, CSS, and JavaScript for the front-end and PHP for the back-end. The front-end allows users to input the necessary data, such as district name, elevation, distance from the river, and rainfall. The input data is then passed to the back-end where it is used to generate flood risk predictions using the previously trained machine learning model.The model is stored in Python by using PHP we are able to pass data to the model and get the model result in PHP. The back-end is responsible for processing the input data, making the flood risk predictions, and returning the results to the front-end. The results are then displayed on the front-end in the form of a table and a map that shows the predicted flood risk areas for the selected district. The web application provides a user-friendly interface for users to interact with the flood risk prediction model and obtain predictions for their district of interest.

The web application can be summarized in the figure below:

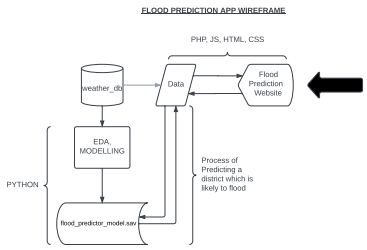

Figure 2: Web wireframe, showing the interaction between the web application, the model and the database

3 Results

3.1 Database

The weather_db database proved to be effective in storing and managing the large amounts of data used in the project. Queries for specific data points or combinations of data were easily executed, allowing for quick and efficient analysis of the data. The database also provided a solid foundation for the web application, allowing for real-time data analysis and predictions to be made based on the stored data.

3.2 Flood Prediction Model

The evaluation metrics for the project report were based on the performance of two machine learning algorithms, the RandomForestClassifier and the GradientBoostingClassifier. The best parameters for the RandomForestClassifier were determined to be 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 4, 'n_estimators': 100, with a resulting best score of 0.8864. For the GradientBoostingClassifier, the best parameters were 'learning_rate': 0.1, 'max_depth': 5, 'n_estimators': 100, and the best score was 0.9261.

In addition to these parameters, the performance of the models was evaluated using confusion matrices, F1 score, and recall. The confusion matrix for the models revealed that the number of true positives and true negatives was 25 and 13 respectively, while the number of false positives and false negatives was 2 and 4 respectively. The recall score, which measures the proportion of true positives out of all actual positive cases, was found to be 0.7647. Finally, the F1 score, which is a weighted average of precision and recall, was found to be 0.8125. This metric provides an overall measure of the models' accuracy, with a higher score indicating better performance.

3.3 Data Analysis

The data analysis section of the project involved plotting various visualizations to understand the relationships and trends within the dataset.

First, a map of Malawi was created to show the locations where data was collected. This allowed for a visual understanding of the distribution of data points throughout the country.

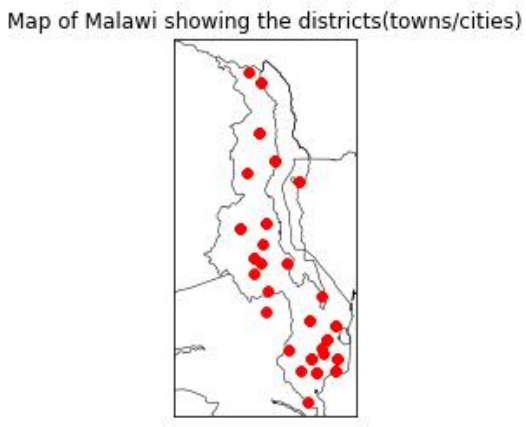

Figure 3: Map of Malawi, note: 1 of the points falls outside the map because we had challenges converting the district's name to the appropriate longitude and latitude

Next, a histogram was plotted to show the distribution of rainfall across the dataset. This visualization revealed that the majority of the data points had rainfall amounts within a certain range, with only a few outliers having significantly higher or lower rainfall amounts.

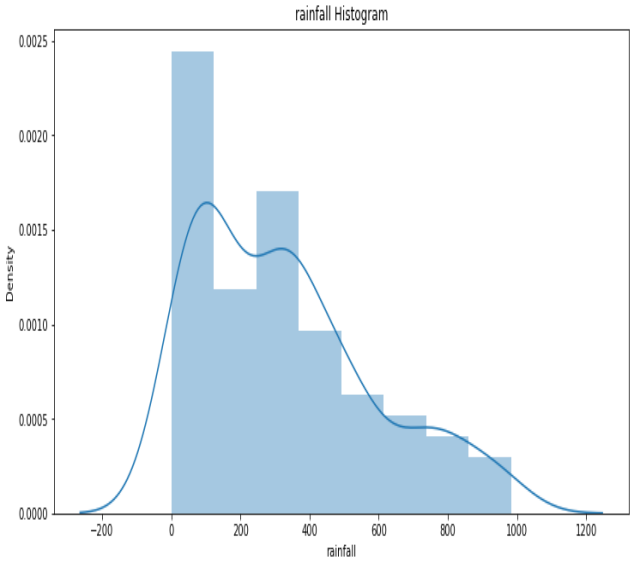

Figure 4: Rainfall of Malawi, between 2019 and 2022

A graph was also created to show the frequency of flooding incidents across the districts. This graph showed Thyolo, Blantyre and Zomba had the the highest frequency of flooding incidents.

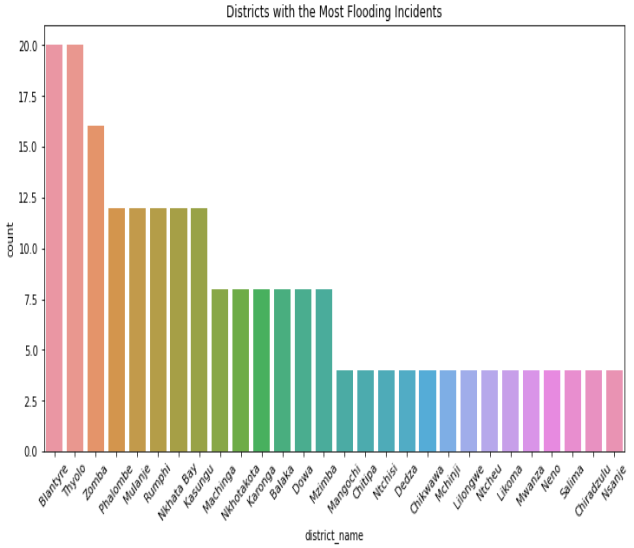

Figure 5: Graph of districts of Malawi by flooding incidents

Finally, a map was created to show the frequency of flooding incidents by year. This allowed for a visual understanding of the trends in flooding incidents over time.

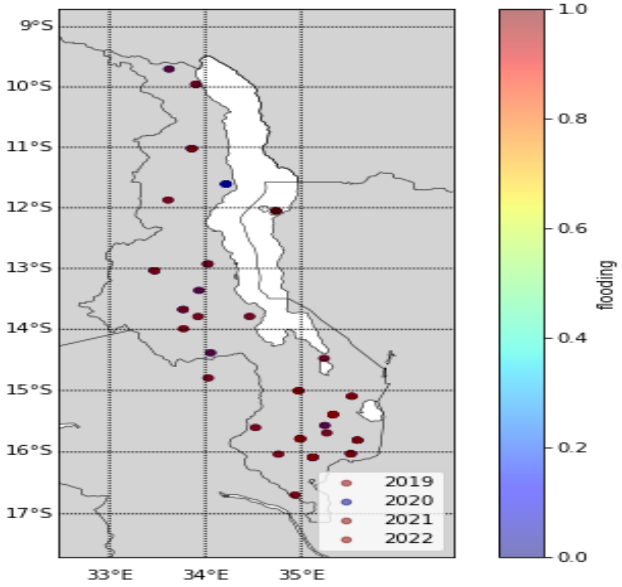

Figure 6: Map of Malawi showing flooding areas between 2019 and 2022

3.4 Web Application

4 Discussion/Conclusions

The results of this project have demonstrated the potential of machine learning algorithms for predicting flooding in Malawi. The analysis of the collected data showed that 'Longitude', 'Latitude', 'elevation','distance_from_river', 'rainfall', 'urban_population', 'land_area', 'density' and 'year' were significant predictors of flooding in Malawi. Random Forest Classifier and Gradient Boosting Machine algorithms were used for classification, and the models were optimized through cross-validation. The Gradient Boosting Machine algorithm was found to be the most accurate, with an accuracy of 92.6%, while the Random Forest Classifier had an accuracy of 88%. The data analysis revealed that the rainfall in Malawi is highly variable, with the highest amount of rainfall occurring in the northern region. Furthermore, the flooding incidents are more common in the southern region, with the highest number of flooding incidents occurring in 2019. The results of this study can be used to develop an early warning system for floods in Malawi. Such a system could help mitigate the effects of flooding by providing timely information to residents and aid organizations. The information could be used to evacuate people in areas that are likely to be affected by flooding and to prepare the necessary resources to assist the affected communities. In conclusion, this project has provided valuable insights into the factors that contribute to flooding in Malawi and has demonstrated the potential of machine learning algorithms for predicting flooding. Future studies could explore the integration of remote sensing data and other relevant data sources to improve the accuracy of flood prediction models and real time monitoring and prediction of flooding.

5 References

[1] P. Yang, L. Liu and M. Li, "Flood Prediction Model Based on Gradient Boosting Decision Tree Algorithm," 2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC), 2019, pp. 297-302, doi: 10.1109/ICIVC.2019.8817898.

[2] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, Oct. 2001, doi: 10.1023/A:1010933404324.

[3] G. Liu, S. Wang and G. Li, "A Machine Learning Model for Flood Risk Analysis," 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 405-409, doi: 10.1109/ICCC50298.2020.00095.

[4] T. Masangano, A. M. Kambalametore, and G. Mulambya, "Flood forecasting in the lower shire river basin, malawi, using satellite rainfall estimates," J. Am. Water Resour. Assoc., vol. 52, no. 4, pp. 890–902, 2016.

[5] R. Kelman, "Floods in Malawi: An analysis of vulnerability," Disasters, vol. 29, no. 4, pp. 361–381, 2005.

[6] T. Yilma, M. Azene, and B. Mohammed, "Development of flood early warning system using earth observation data and community knowledge in the Blue Nile Basin, Ethiopia," J. Disaster Res., vol. 13, no. 3, pp. 347–356, 2018.

[7] "Topographic Map.com." [Online]. Available: https://en-us.topographic-map.com/place-pp57/Malawi/. [2] "Malawi Meteorological Department, Malawi Monthly Weather Bulletin, November 2019." [Online]. Available: https://www.metmalawi.gov.mw/agromet/MWNovember 32019.pdf.

[8-10] "Malawi Meteorological Department, Malawi Monthly Weather Bulletin, November 2021." [Online]. Available: https://www.metmalawi.gov.mw/agromet/MWNovember 32021.pdf.

[11] "Malawi Meteorological Department, Malawi Monthly Weather Bulletin, November 2022." [Online]. Available: https://www.metmalawi.gov.mw/agromet/MWNovember 32022.pdf.

[12] "Malawi Distance Calculator." [Online]. Available: https://distancecalculator.globefeed.com /Malawi_Distance_Calculator.asp.

[13] "UNAIDS Naomi Spectrum database." [Online]. Available: https://naomi-spectrum.unaids.org/.

[14] "2018 Malawi Population and Housing Census Main Report." [Online]. Available: https://malawi.unfpa.org/sites/default/files/resource-pdf/2018%20Malawi%20Population%20and%20Housing%20Census%20Main%20Report%20%281%29.pdf.

[15] "ReliefWeb." [Online]. Available: https://reliefweb.int/disaster/ss-2021-000196-mwi.

[16] "ReliefWeb, Malawi: Floods and Earthquakes Emergency Plan of Action (MDRMW014) - DREF Operation Update." [Online]. Available: https://reliefweb.int/attachments/6720d6fd-07cb-3ea0-a099-bdcf8194bd4a/MDRMW014efr.pdf. [10] "FloodList." [Online]. Available: https://floodlist.com/tag/malawi.

[17] J. Han, M. Kamber, and J. Pei, "Data preprocessing," in Data Mining: Concepts and Techniques, 3rd ed. Elsevier, 2012, pp. 111-164.

[18] J. W. Tukey, Exploratory Data Analysis, Addison-Wesley, 1977.

[19] S. Boriah, J. Chandola, and V. Kumar, "Similarity measures for categorical data: A comparative evaluation," in Proceedings of the 8th SIAM International Conference on Data Mining, 2008, pp. 243-254.