Flood Probability Classification Utilizing Ensemble Learning to Account for Uncertainty

By: Vatsal Sivaratri and Connor Friedman

1 Abstract

Floods remain a present and worsening problem within our modern society. As the temperature of the earth slowly climbs, the magnitude of natural disasters has slowly worsened and worsened throughout the years. In order to prevent these storms from impacting our communities, machine learning researchers have started to implement flood and storm predictive models in their work to ensure that communities are able to avoid the disaster entirely and without doing so for unnecessary reasons (such as a false alarm). Due to the sheer importance placed upon these models being correct, many such classification algorithms have been used to ensure that their predictive readings are indeed accurate, with ensemble learning being by far the most utilized. However, despite the effectiveness of ensemble models, current solutions often fail to account for the nuanced interplay of environmental factors such as infrastructure deterioration, deforestation, and political mismanagement, which can drastically influence flood susceptibility.

This project proposes the development of a novel hybrid machine learning algorithm that combines environmental feature engineering with weighted ensemble learning to predict flood risks more accurately. By leveraging key attributes like climate change scores, urbanization rates, and watershed management data, the model seeks to minimize false positives while enhancing predictive accuracy for at-risk areas. Through rigorous experimentation and comparison against traditional models, this research aims to provide actionable insights for community planning and disaster prevention, contributing to a more resilient approach to climate-related threats.

2 Introduction

For years predictions regarding potential weather disasters have been plagued by inaccuracy due to overfitting and other methods of uncertainty. This is especially the case for areas around my own house, places notorious for getting incorrect weather forecasts falsely predicting flooding when none occurs and failing to predict flooding that occurs regardless. This can be attributed to two main issues found when using artificial intelligence models to predict natural disasters. Firstly, weather data can almost never be found for specific areas and is generally designed to cater towards a larger and more diverse environment. While this does work when predicting total weather outcomes, its large generalizations are almost never useful when analyzing your own environment as it fails to account for other factors such as urbanization, rate of flooding, deforestation of surrounding areas, and more, making “flood predictions” practically useless.

Secondly, as the chaos of our planet's ecosystem increases, so does the sheer amount of uncertainty found within the systems used to predict its ever changing outcomes. Because of this, it's important for any model built to predict such an event to be able to adapt to its uncertain nature, a feat that most classification models tend to struggle with. While this is accounted for in modern weather prediction models, it's oftentimes not found to be working in tandem with the first issue our model works to address (making the "uncertainty" found within such situations essentially useless), especially when it comes to working with ensemble models of utilizing more than one classification models in its bagging process.

3 Related Work

There has been quite a lot of research done in regards to how different methods of classification perform when utilized within an ensemble learning model, each with its own strengths and weaknesses acknowledged by our very own model. [1] utilizes deep learning in conjunction with ensemble learning to produce a highly efficient alternative to Bayesian NN's. While this is effective in some instances, it fails to fully negate any uncertainty within the data, uncertainty present throughout storm data, making it difficult to exclusively rely on in a large majority of instances, an issue our method resolves by introducing multiple methods of classification, something our group can conclude by looking at examples that specifically utilize such methods for predicting the rate of flooding any specific area contains, a detail found in [2].

Other flood related ensemble models have been implemented with varying levels of efficiency. The primary inspiration behind this paper came from [4], a study that uses ensemble learning to predict flood related statistics, which I thought was fascinating upon my first read. After playing around with their proposed algorithm, while the theory behind it was fantastic, the classification models used in each bag are notorious for being rigorous and highly susceptible to uncertainty. After reading [5], a study which analyzes the impact and importance of considering uncertainty in weather prediction models, we came up with the idea to apply this line of thinking when combining the results of each output within the total ensemble. In order to solve this issue, we not only switched to deep learning but we also did so while implementing a weighting system that considers how uncertain each approach was and deliberately making it less influential depending on that result.

While uncertainty estimation is well studied separately in DNNs (via MC-dropout) and GBMs (via quantile regression or bootstrapping), our model's novelty comes from its ability to compute dynamic weights inversely proportional to uncertainty. By utilizing a dynamic method of weight calculation, our program ensures that bags with less certainty are considered less in the overall classification, a method of calculation which almost never finds itself being used within an ensemble setting. This method was proposed and experimented with in [3], but it is yet to be used in both ensemble and weather settings.

4 Description of Dataset

The data was found on Kaggle from a competition hosted by the website themselves. The dataset contains ~1.12m instances and 21 attributes (Dimension – 22). Some features include:

MonsoonIntensity: The intensity of the storm
TopographyDrainage: How good natural drainage of the area is (how well it drains water)
River_Management: How well maintained the surrounding rivers are (prior to storm)
Deforestation: How badly deforested the area is
Urbanization: How urbanized the area is
ClimateChange: A numerical value that calculates how at risk the area is to climate change
DamsQuality: Quality of dams that prevent flooding
Siltation: Particle pollution
AgriculturalPractices: Quality of agriculture in the area
Encroachments: Increase in infrastructure that makes the area more adverse to flooding
DrainageSystems: How good the draining is in the area (how well it drains water)
Watersheds: How many watersheds surround the location
DeterioratingInfrastructure: Quality and maintenance of area's infrastructure
PopulationScore: How populated the area is
PoliticalFactors: Other factors that make the area more susceptible to flooding not previously mentioned
FloodProbability: Class

5 Methods

5.1 Deep Neural Network (DNN)

The deep neural network (DNN) used in this study consists of three fully connected layers: an input layer matching the number of features, two hidden layers with 64 and 32 neurons respectively, and a single output layer with a single neuron using a linear activation function. The ReLU activation function is applied in the hidden layers, and dropout regularization (30%) is incorporated after each layer to prevent overfitting. The model is trained using the Adam optimizer and minimizes the mean squared error (MSE) loss, making it suitable for continuous probability predictions. DNNs are an excellent choice for this dataset, as they can capture complex, non-linear relationships between environmental, geographic, and human impact features.

The innovation of this approach is uncertainty quantification using Monte Carlo (MC) Dropout. During training, dropout helps prevent overfitting by randomly setting a fraction of neuron activations to zero. But standard NNs disable dropout during inference, and this means that a single deterministic prediction is produced for each input. To estimate uncertainty, we can use MC Dropout, where multiple stochastic forward passes are performed at test time while maintaining dropout activation. This results in different predictions each time, which forms a distribution from which uncertainty can be computed. Specifically, the mean of the predictions serves as the final flood probability estimate, while the variance quantifies model uncertainty:

$$Mean\ (\hat{y}) = \frac{1}{T}\sum_{t=1}^{T} \hat{y}_t$$

$$Variance\ (\sigma^2) = \frac{1}{T}\sum_{t=1}^{T} (\hat{y}_t - \hat{y})^2$$

, where *T* is the number of stochastic passes. A high variance indicates greater uncertainty, which means less confidence in the model's prediction.

5.2 Gradient Boosting Machines (GBMs)

Gradient Boosting Machines (GBMs) are an ensemble-based approach that sequentially improves weak learners (usually decision trees) by minimizing a specified loss function. We used LightGBM, an optimized gradient boosting framework that is good with tabular data with missing values. The model is trained with an RMSE loss function and includes early stopping based on validation performance to prevent overfitting. GBMs have been established to work really well with tabular data, making it a good choice for this task.

Unlike standard implementations, which provide a single deterministic prediction, we use bootstrap resampling to estimate uncertainty. Instead of training a single GBM, we trained multiple GBMs on different randomized subsets of the training data (Yes we ensemble'd the ensemble). Each model produces a slightly different prediction on the test set, leading to a distribution of outputs. The final prediction is the mean of all individual model outputs, and uncertainty is the variance across the ensemble.

$$Mean\ (\hat{y}_{GBM}) = \frac{1}{N}\sum_{i=1}^{N} \hat{y}^{(i)}_{GBM}$$

$$Variance\ (\sigma^2_{GBM}) = \frac{1}{N}\sum_{i=1}^{N} (\hat{y}^{(i)}_{GBM} - \hat{y}_{GBM})^2$$

Where *N* is the number of bootstrapped models. IF the models disagree significantly, the variance is high, indicating low confidence.

5.3 Support Vector Regression (SVR)

Support Vector Regression (SVR) is a kernel-based method that maps input features into a higher-dimensional space where linear relationships can be more easily modeled. It constructs a function that fits the data within a margin of tolerance ε, prioritizing support vectors that define the model while ignoring smaller deviations. SVRs are excellent in handling high-dimensional continuous data while capturing non-linear relationships.

To estimate uncertainty in SVR predictions, we trained multiple SVRs on randomized bootstrap samples of the training set. The final mean prediction and uncertainty are computed as follows:

$$Mean\ (\widehat{y}_{SVR}) = \frac{1}{N}\sum_{i=1}^{N} \widehat{y}_{SVR}^{(i)}$$

$$Variance\ (\sigma_{SVR}^{2}) = \frac{1}{N}\sum_{i=1}^{N} (\widehat{y}_{SVR}^{(i)} - \widehat{y}_{SVR})^{2}$$

where *N* represents the number of resampled SVR models. This is the same approach we used for the GBM, effectively creating analogs of random forests to decision trees.

To mitigate overfitting, we used a couple of techniques. Early stopping was used in both the GBM and DNN to prevent unnecessary iterations beyond optimal performance. Dropout regularization was applied to the DNN, ensuring that the model didn't become overly reliant on specific neurons. Also, bootstrap sampling was great in reducing variance by ensuring no single subset of data overly influenced predictions.

6 Results & Discussion

	MSE	MAE	R2 Score
DNN	0.00040	0.01595	0.84504
GBM	0.00059	0.01987	0.77274
SVR	0.00085	0.02409	0.67283
BayesRidge	0.00040	0.01580	0.84434
DecisionTree	0.00250	0.03979	0.03739
LinearReg	0.00040	0.01580	0.84434
RandomForest	0.00094	0.02503	0.63808
Ensemble	0.00054	0.01877	0.79343

Table 1. Model Results

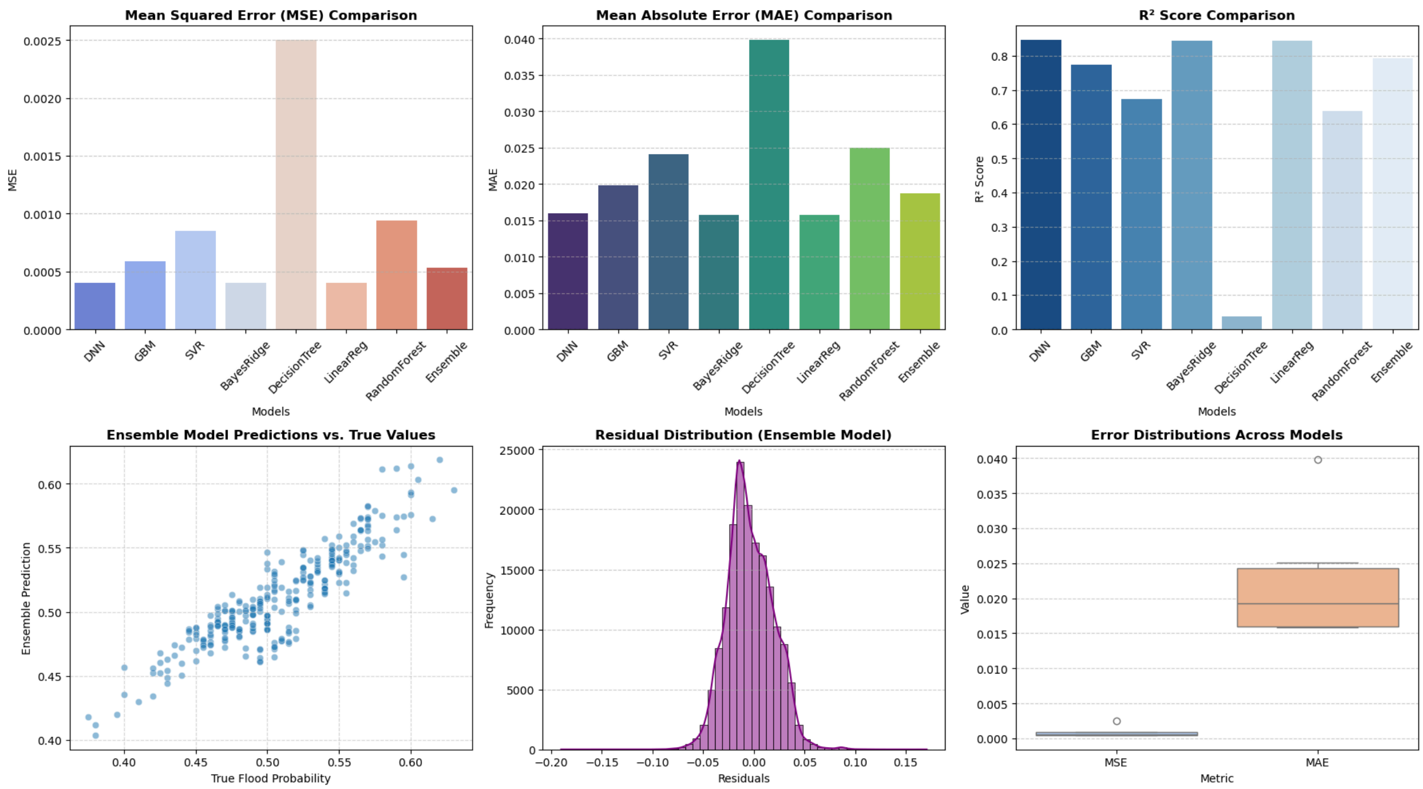

Figure 1. Table of Graphs

The evaluation of model performance was done by 3 main metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), and the R^2 Score (Coefficient of Determination). MSE measures the average squared difference between predicted and actual values, making it sensitive to larger errors. MAE captures absolute deviations and provides an interpretable measurement in the same units as the target variable. The R^2 score indicates how well the model explains the variance in the target variable, with higher values suggesting stronger predictive capability. The metrics were chosen because the problem at hand involves predicting continuous probabilities, and error quantification is important to understand reliability.

From the comparative performance in the table and graph, it's clear, much to our disappointment, that the LinReg algorithm did the best, with the lowest MSE and MAE scores and the highest R^2 score, followed shortly by the DNN. Decision Trees were by far the worst, exhibiting the highest MSE and MAE values. This aligns with expectations,a s decision trees are prone to overfitting unless paired with ensembling methods like boosting or bagging. SVR performed reasonably well, but it was definitely outperformed by the other two.

While the ensemble didn't score the best in the metrics, it did still perform quite well, and, more importantly, was confident in its predictions. The residual distribution plot shows that the ensemble model's errors approximate a normal distribution, which indicate stability in it's predictions. The scatter plot further confirms that this model closely tracks true flood probabilities, with minimal deviation. These show that explicitly incorporating uncertainty estimation into the prediction process enhances robustness.

7 Conclusion

The result of this study highlights the effectiveness of ensemble learning and uncertainty-aware modeling in prediction flood probability. Although individual models like DNN and LinReg had the lowest error rates and highest R^2 scores, the ensemble shined in its prediction confidence. By aggregating predictions in a weighted manner based on variance, the ensemble approach not only reaches high accuracies but also provides more reliable results.

For future work, additional improvements could be explored with more computational resources and time. One thing that could be done is integrated real-time weather and geospatial data to make the task more relevant and consistently improve the predictive power. Also, incorporating additional environmental factors like soil saturation, past rainfall trends,a nd urban infrastructure data will improve model reliability. Another promising avenue would be using deep ensemble methods for even better uncertainty estimation, or attention-based models to improve the model's ability to capture spatial and temporal dependencies in flooding patterns.

8 Contributions

*These roles are not exclusive to one person. This is just focusing on the MAIN things each person focused on

Vatsal Sivaratri - Wrote code for GBM + DNN + Combining 3 models into ensemble, Wrote methods, results, conclusion; Made presentation.
Connor Friedman - Did a lot of background research; Wrote preprocessing code and SVR model; Wrote abstract, introduction, related work, and description of dataset; Proofread entire paper.

9 Citations

[1] - https://arxiv.org/pdf/2104.02395
[2] - https://www.mdpi.com/2073-4441/13/21/4239 (this link isn't working currently)
[3] - https://www.sciencedirect.com/science/article/abs/pii/S1566253506000819
[4] - https://www.tandfonline.com/doi/full/10.1080/19475705.2023.2203798#abstract
[5] - https://www.sciencedirect.com/science/article/pii/S0022169423013847