Flood Prediction with Random Forest in the U.S.

Daniel Liang
Department of Computer Science
New York University, NY, USA
dzl217@nyu.edu

Sakawkarn Piyawitwanich
Department of Computer Science
New York University, NY, USA
sp4887@nyu.edu

Satyajeet Maharana
Department of Computer Science
New York University, NY, USA
sm8235@nyu.edu

***Abstract*—Floods can be categorized into both slow-developing flowing water and flash flowing water. Floods are temporary but have an exceptional impact on the area while flooding. We built a Random Forest prediction model based on and visualize the probability of flooding among different areas in Tableau. We also built a small email service to alert counties about the possibility of future flooding. We hope that our application can help with future flood detection and prevent within the United States.**

Keywords—Spark, analytics, machine learning, flood prediction, random forest

I. Introduction

Floods are a natural disaster that has affected many areas of the world and have shaped the priorities of disaster relief groups and governments alike. Generally a flood gets predicted in some short period prior and hasty measures are taken to warn the public about the disaster. Afterwards much reparation is needed in the area impacted and treatment must be given to whoever was hurt in the process. With Big Data, there are more opportunities to predict when and where floods will happen based on previously acquired data to gain an estimate further in advance. This will allow more permanent and safer measures to be taken to alleviate or even avert the negative impact of these types of disasters.

We found several published papers on machine learning that talk about using Neural Networks and Random Forests to help with storms and floods forecasting. Other papers were focused on Big Data, where previous flood prediction work was done using MapReduce in correlation with Machine Learning. In the Related Work section, we presented the papers that we found correlated to our work in the project. The range of the papers we found talk about a variety of topics ranging from data mining techniques we can use for the flood prediction to improving a random forest model. These papers influenced and inspired how our application was designed and modeled.

We collected the datasets from three different sources provided by the United States National Climatic Data Center (NCDC) and the National Oceanic and Atmospheric Administration (NOAA). Then, we analyzed and extracted important features such as the amount of precipitation and wind in certain areas out of the datasets. We built the random forest prediction model using MLlib in Apache Spark to predict the outcomes of future floods. We then built a Tableau workbook to visualize the results of flood prediction and a small application for emailing county officials about upcoming floods in their area.

II. Motivation

Floods are incredibly destructive and can happen due to many different factors including snow, wind, and even as a result of other natural disasters like hurricanes. Since floods cause a lot of damage to human life, animals and property, we want to find the way to help with this natural disaster. Having a more accurate prediction will allow the damage to be mitigated so counties can even save money for disaster relief in those areas. Theoretically, anyone could use this application but in particular, county officials as well as disaster and prevention relief groups would use the application to predict flooding around their regions of interest and can warn people in a position to take measures against floods. Interested parties of this application include politicians, city planners, disaster relief groups, and government officials dealing with weather-related issues. The application is ideal to generate more insight alongside with current flood detection and prevention methods to protect areas of the United States against unavoidable flooding.

III. Related Work

S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti and B. Chakraborty wrote a review on Application of Data Mining Techniques to Combat Natural Disasters. [11] There are three main tasks that they have identified in order to leverage the data mining techniques in combatting natural disasters:

1. Prediction
2. Detection
3. Disaster Management Techniques

In our project, we specifically focused on developing models to solve the problem of "Prediction" of "Floods",. Our project scope is currently limited to prediction of areas which are susceptible to floods and the possible timing of the floods. The data mining techniques will be used to predict the magnitude of floods and track the flow of water to various regions.

Their work identified various different techniques that can be used for flood prediction. Below is a quick summary of what they built and did:

- A decision tree model trained based on the selected parameters/features to assess the damage from floods. The various data sources used are Hydrological data, Remote sensing data and GIS data. The studies related to this particular topic were done in Germany.

- A logistic regression and frequency ratio model based on the spatial data to find the susceptible flood areas. The various data sources used are Meteorological data (digital elevation model), river and rainfall data. The studies related to this were done in Malaysia.

- An Artificial Neural network model based on monsoon and hydrological data to predict the monsoon floods 1 day ahead. The ANN model actually performed much better than the previous regressive models. The studies related to this were done in India.

- A probabilistic and ensemble model built for predicting flood for a medium-large scale African River basin. Again, the data sources for this model were the hydrological datasets.

- A flood routing model based on past flood data. The data mining techniques used were Muskingum flood routing model and Cuckoo Search. The data sources for this model were Hydrological and Hydraulic datasets.

- A disaster management study was done which studied the usage of tweets during floods to identify the key players (for eg: the involvement of local police authorities in rescue operations).

- A classification model to predict landslides and floods. The various factors considered were the rainfall, land use, soil type, slope etc. The models used Support vector machines and Naive Bayes techniques for mining the data.

The above are the specific studies related to the prediction of floods. However, the paper analyzes various other techniques used for prediction of similar natural disasters like Storms, Earthquakes, Tsunamis etc. Although not specific to our project, we believe that the techniques used in prediction of storms and landslides would be relevant in the prediction of floods as well. The paper talks about a particular project undertaken by the National Hurricane Center (NHC) of USA who took the current storm data and compared it with the historical storm data to understand the effects based on "Storm Similarity Index" (SSI).

Apart from these "Prediction" techniques, an effort has been made to utilize Twitter data for real time "Detection" of various natural disasters which is helpful for efficient disaster management and minimal loss of lives. The specific tasks for the "Prediction" of floods are

1. Identification of the flood susceptible areas

2. Build flood routing models to predict the course of floods

3. A predictive model to assess damage to property/ infrastructure due to floods

In their research paper, they have identified some useful models which can act as inspiration for our model. The models often used are Support Vector Machines, Artificial Neural Networks, Decision Trees. They also give a high-level review of the types of data and features within the data which would be useful for analysis and prediction of natural disasters. Most of the data could be identified as "Big Data" : Volume (GIS Data, Meteorological Data, Social Media Data), Variety (Text, Time Series, Spatial Data, GIS Images), Velocity (because of the rate in which data is generated as well as because of the speed in which a decision needs to be taken).

K. Jitkajornwanich, U. Gupta, R. Elmasri, L. Fegaras and J. McEnery studied MapReduce on how to speed up the storm identification from big rainfall data [1]. Their paper describes a MapReduce algorithm that converts raw rainfall data into relevant storm characteristics. The authors initially explain their previous work with the data's format, the types of storms that they categorized with their previous algorithm, and the process for storm identification system. They used a method to identify local storms and hourly storms and then had users to identify overall storms. Using MapReduce allowed the architecture for the identification system to change and be optimized; because the raw data could be used with MapReduce, there was no need for an event system to identify local storms when it could be derived with a MapReduce job on its own. It grouped the precipitation values by site and ordered it by time (map/shuffle and sorting phases) while finding the local storms for a site by the group values (reduce phase). Another MapReduce job is created without a reduce method to output hourly storms data in text file format. Using MapReduce had resulted in a speedup of roughly 19x for local storm identification and 13x for hourly storm identification, which makes it far closer to real-time analysis than the previous method.

Their research accomplishes two things for our application: it gives more insight in what traits of the rainfall data are relevant for predictions, which is vital to flood prediction, and provides a parallelized way of thinking for developing an algorithm to efficiently predict for floods. Although MapReduce is an older method for analyzing Big Data, the methods that the authors used in this paper can almost be directly converted to Spark for even faster processing of data, and show how we could potentially parse the types of data we will also use. This application benefits from our further understanding of how weather datasets are used specifically for storms and how Big Data had previously impacted flood prediction applications.

Changhyun Choi, Jeonghwan Kim, Jongsung Kim, Donghyun Kim, Younghye Bae, and Hung Soo Kim created prediction models for heavy rain damage occurrence to mitigate or prevent disasters from occurring within Korea [16]. Data was gathered from the Korea Meteorological Administration from 1994 to 2015, and contained information of 27 variables including temperatures, precipitation, humidity, etc. These variables were used as explanatory variables in the prediction model. A binary representation was used for heavy rainfall damage data obtained from the Annual Disaster Report in Korea, and this acted as the response variable in the model. The authors introduced the concepts behind random forests such as decision trees, bagging, and boosting methods and had made four different models based on each of the concepts. Two algorithms were used alongside the models: predicting heavy rainfall damage on a given day with same-day weather observation data, and predicting heavy rainfall damage on a given day with past weather observation data. The authors later explained how to handle the different types of variables and their method of fine-tuning the models' parameters. They had concluded the model using boosting methods while also predicting with past weather observation data yielded the best producing results.

Their work is very relevant to the model we want to create using big data. While the prediction model for this project uses random forests, it is still beneficial to see more work has done on a very similar and related topic using the same type of data. Furthermore, the paper explains their logic in classifying the data into different types of variables, as well as how it was split into a training and test data sets to train each model. The background and intuition gathered from this paper will be very beneficial for a prediction model based on random forests.

A. Mosavi, P. Ozturk, and K. Chau wrote a literature review of Flood Prediction Using Machine Learning Models [6]. They compared machine learning algorithms we can use for flood prediction and evaluate the best algorithm that has the best performance for the prediction. Because our project is to implement a model for flood prediction, we want to find what would be the best possible algorithm that fits our project. The authors have tried analyzing the prediction model with Artificial Neural Networks (ANNs), Multilayer Perceptron (MLP), Adaptive Neuro-Fuzzy Inference System (ANFIS), Wavelet Neural Network (WNN), Support Vector Machine (SVM), Decision Tree (DT), and Ensemble Prediction Systems (EPSs). From 2008 to 2017, ANN had been the most used algorithm followed by SVM, SVR and DT. Additionally, the authors suggests that using hybrid algorithm is more popular than using a single algorithm. When doing performance analysis, ANN, SVM and SVR algorithms gave the best result for a single method. While decomposition and ensemble methods give better result in hybrid. ANN gives high accuracy, has fault tolerance and can do parallel computing but it has some issues with generalization. The paper also suggests that WNN produces more consistent results than ANN. For our project, we can try researching more about NNs and SVMs for ease of use and better prediction results.

M. A. Sit and I. Demir wrote a paper on Decentralized Flood Forecasting Using Deep Neural Networks [14]. This paper explores the particular application of Artificial Neural networks in the prediction of floods at particular areas in a future time period. In the study, a benchmark dataset is presented for flood forecast to generate and test artificial neural networks performance for flood prediction. The key ideas are divided into the following modules:

- Preliminary information about the task and raw data
- Data set creation and preprocessing steps
- The architecture of the Neural Network
- Outcomes of the performance metrics

The paper covers significant related work where back-propagation network-centric approaches were used in flood prediction. The key idea behind using Artificial neural networks instead of the standard applied machine learning methods is that Artificial neural networks have been widely utilized for similar tasks in hydrology (e.g. reservoir inflow prediction, precipitation prediction, runoff analysis). Also, we are introduced to the concept of exploiting time-series data in all the flood datasets which could be crucial in increasing the accuracy of our models. One more application is mentioned, which is similar to flood prediction – Traffic speed prediction. They are similar in the fact that both of them rely on the changes in connected points in the network. This paper proposes a benchmark dataset for future applications of flood prediction. In their approach, they take into consideration their historical stage data of points on selected river basins and rainfall data. Utilizing some of the ideas in using time-series data can ideally also increase the accuracy in the model we build.

C. Tang, D. Garrea and U. von Luxburg attempts to answer the question "When do random forests fail?" [8] Firstly, the random forest algorithm predicts the results by averaging the predictions of individual trees in the ensemble of the trees. The paper focuses on the errors that can occur to the random forest algorithm, especially the tuning parameters e.g. rate of data subsampling, numbers of trees, and number of leaf nodes in a tree. The suggestion of most studies is that the rate of subsampling is 1 with a large number of trees and a small number of leaf nodes. The paper states that there are three beliefs about the parameter setups in the random forest algorithm. The first belief is that adding more trees to the ensemble reduces the generalization error. The second belief is that it is good to use deep trees where there is only one leaf node in each tree. This can lead to overfitting since the variance is high although the bias is low. The third belief is that randomizing the construction of the trees helps with lowering the variance. The experiment in this paper is done on unsupervised learning. They have tried working with random forest with no subsampling and too severe subsampling and found that the algorithm will be inconsistent. They also found that randomized trees, which believe to solve overfitting, can lead to overfitting forests. The conclusion is that the parameter that needs to be tuned carefully is data subsampling. In terms of our random forest model, this information was carefully considered when tuning our model to increase the accuracy and precision of the predictions.

IV. APPLICATION DESIGN

Our design diagrams are indicated in Fig. 1-2. We first create the training inputs of Random Forest model, which are derived from a merged set of data from three datasets: the NCDC Storm Events dataset, NOAA Daily Summaries dataset and NOAA Precipitation Reconstruction dataset that are stored in HDFS in Dumbo. Then we train our model and produce prediction results, which is ultimately displayed and visualized in Tableau. The samples of final visualization of the application is shown in fig. 3-5 below.

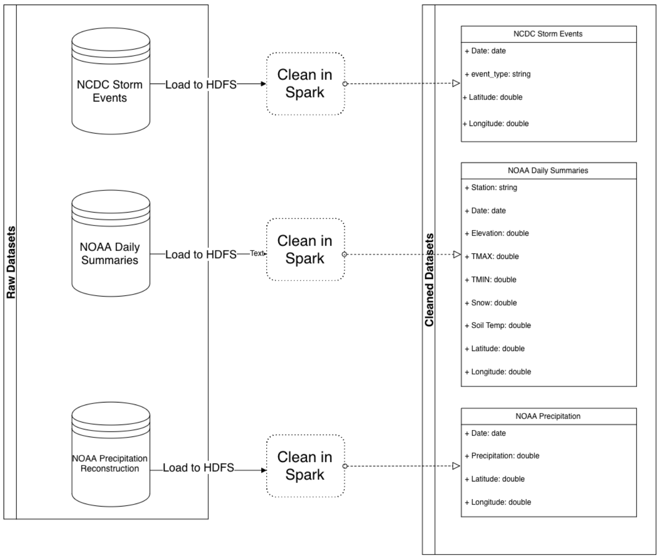

Fig. 1: Design diagram

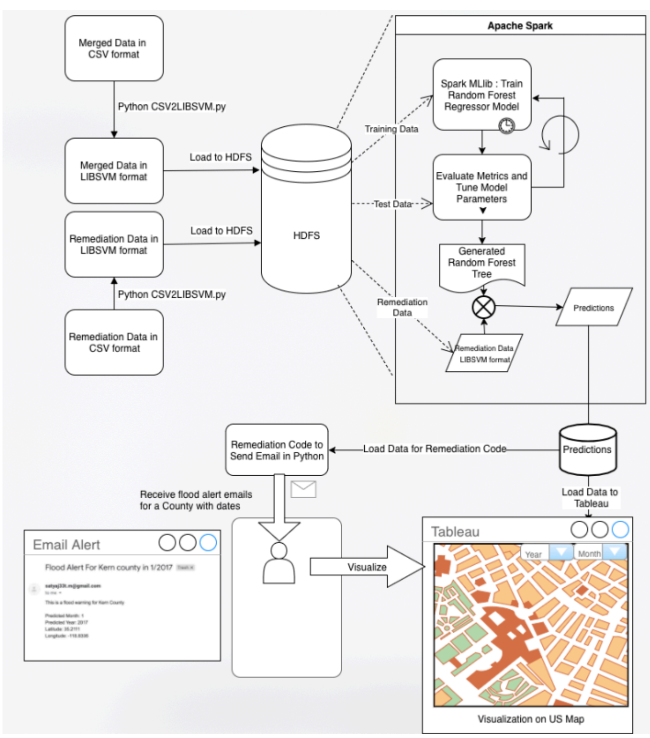

Fig. 2: Design diagram

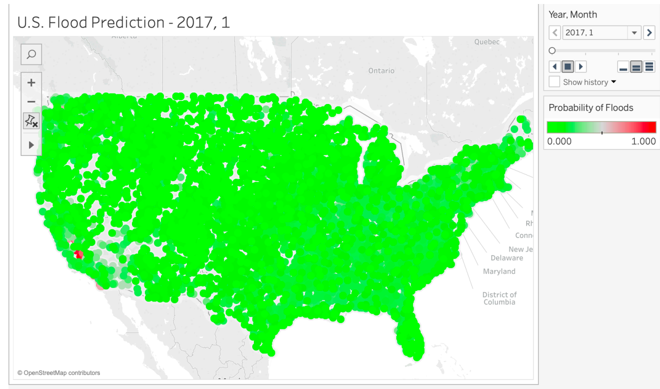

Fig. 3: Application UI design

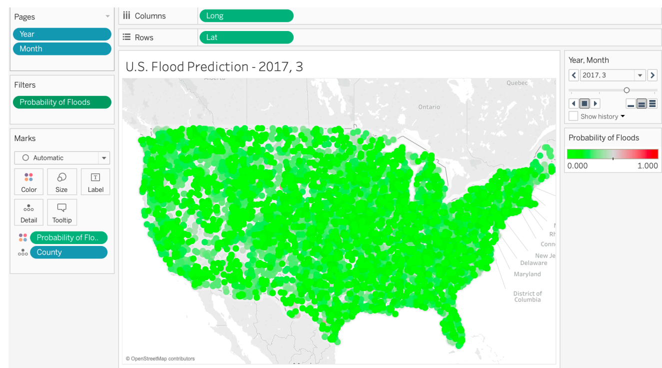

Fig. 4: Application UI design

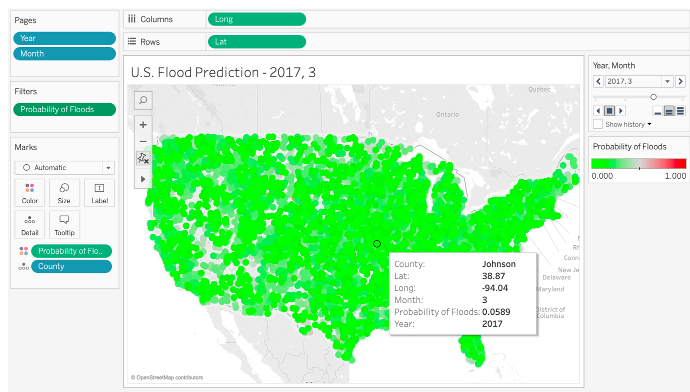

Fig. 5: Application UI design

The UI design of the final application is an interactive map of the United States shown in different colors according to a coordinate's probability of flood at a particular time. When the user clicks or hovers on an area of interest, the probability of flood happening in that area will appear in a box near the mouse pointer. In addition, a user can filter coordinates based on their probability values. For example, a user can filter out data points where there is clearly no flooding and can clearly see areas where the probability of flooding is above 0.9, or any other threshold the user would like to see the data from. Finally, a scrollbar can be used to view data from different times separated by month and year.

V. Datasets

The first dataset considered is the NCDC Storm Events dataset. The dataset covers the personal injuries and damage estimates from storms that affected areas in the U.S. The data will be filtered for flood specific events by the timing of each flood, followed by its location. This will give an idea of where and when floods in the U.S. have occurred in the past. The schema of the dataset is in Table 1 below.

Column Name	**Data Type**
year	Int
month	Int
begin_day	Int
begin_time	Int
episode_id	Int
event_id	Int
state	String
event_type	String
flood_cause	String
begin_location	String
begin_lat	Double
begin_lon	Double
end_lat	Double
end_lon	Double
episode_narrative	String
event_narrative	String

Table 1: Schema of NCDC Storm Events dataset

The second dataset used is the NOAA Daily Summaries. The dataset contains daily weather data within the U.S. since 1763. This can be a source to look at many key metrics relating to flooding at specific times in the United States like a region's rainfall, temperature, wind speeds. These metrics are used alongside the other datasets to confirm and compare against when flooding occurred and how much rainfall occurred at the time. The schema of the dataset is in Table 2 below.

Column Name	**Data Type**
station	String
date	Date
type	String
measurement_flag_attribute	String
quality_flag_attribute	String
source_flag_attribute	String
time_of_observation	String
observation	Datetime

Table 2: Schema of NOAA Daily Summaries dataset

The last dataset used is the NOAA Precipitation Reconstruction. This dataset is a monthly precipitation analysis on a 2.5 degree latitude by 2.5 degree longitude global grid since 1948. The data is collected globally from gauge observations at 17,000 stations for the land portion and historical gauge observations over islands and land areas for the oceanic portion. This dataset is vital for comparing how much precipitation occurred when floods did happen, which is derived from the NCDC Storm Events dataset. This dataset is updated quasi real-time at NOAA/CPC. The schema of the dataset is in Table 3 below.

Column Name	**Data Type**
year	Int
month	Int
latitude	Double
longitude	Double
precipitation	Double

Table 3: Schema of NOAA Precipitation Reconstruction dataset

VI. Remediation

After we acquire the results from the prediction model, we save them as an Excel worksheet to connect to Tableau so that we can visualize the results accordingly. Directly using the UI is an ideal use-case for people within their region to look at potential flooding patterns. For other users that want to help other counties, an application was created to email officials in counties where the probability of flooding is above a certain configurable threshold. Based on our observations, we have set the threshold to a 90% probability. The application will email county officials in that region to notify them about future flooding based on our prediction model.

VII. Experiments and Results

Geospatial Join

We have three different data sources, and our key idea was to combine the three datasets to form a merged dataset which contains features from all the datasets which describe the weather and climate details for a particular region for a particular day and a labelled column which denotes if there was a flood in a particular region for a particular day.

All the three data sources have a date attribute and a region attribute, described by latitude and longitude coordinates, both of which can be used to join the datasets together. Combining the three datasets by date was very straightforward; the real challenge however was combining the datasets based on the region. It is not straightforward to join the datasets based on the latitude and longitude coordinates directly, since it is unlikely for two rows from different datasets to have the same latitude and longitude values.

Solution

To join all three datasets, we needed to reduce the granularity of the region data from Latitude and Longitude coordinates to the city and state level. We utilized the below tools and datasets to accomplish this:

- GeoPandas: An open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types.
- United States Cities GeoJSON dataset: GeoJSON is a format for encoding a variety of geographic data structures. The cities are represented by the MultiPolygon data structure. Each city in a state is represented by a Geometrical MultiPolygon attribute in the JSON file. A simple example of how the cities are represented is given below:

{"type":"Feature","properties":{"NAME":"New York","INTPTLAT":"+40.6642738","INTPTLON":"-073.9385004"},"geometry":{"type":"MultiPolygon","coordinates":[[[[-74.040312,40.700539],[-74.040177,40.700605],[-74.040064,40.700635],[-74.039932,40.700653],[-74.039795,40.700627],[-74.039726,40.70059],[-74.039608,40.700538]]]]}}

When a dataset with Latitude and Longitude coordinates is spatially joined with this GeoJSON file, we can find the location of this particular Point in the space defined by the MultiPolygon attribute from the GeoJSON file.

Next, we followed the below procedure:

- For each of the three datasets, first convert the Latitude and Longitude coordinates to a Geometric data structure known as a Point. This represents a particular point in the geographic space of coordinates.
- Spatially join all the three datasets with the United States City GeoJSON datasets by finding the location of each Point in the MultiPolygon attribute represented in the GeoJSON file, to add additional City name and State name attributes to the three original datasets.
- Merge all the three datasets into a single file in Spark using the City name and State name as the common attributes

Labelled Dataset

After combining the three datasets, we have a new attribute field from the NCSC Storm dataset : event_type. This column contains three different values: "Flood", "Flash Flood" or ""(blank). Based on the values in this event_type column, we create a new column named "HasFlooded?" which contains the below values:

1 : if event_type field is "Flood" or "Flash Flood"
0 : if event_type field is "" (blank)

Next we dropped the column "event_type" as it is no longer needed and we transformed the information to another column "HasFlooded?". This "HasFlooded?" column now represents if a particular region has flooded on a particular day. This will be used in training of our model to be able to predict the probability of flood.

Data Preparation

For the purposes of using the Spark MLlib, we would first need to convert our merged dataset from CSV format to a different format which is used natively with Spark MLlib, known as the LIBSVM format. LIBSVM is a text format in which each line represents a labelled sparse feature vector using the following format: label index1:value1 index2:value2.

In order to convert our merged dataset from CSV to LIBSVM format, we wrote a small Python program for this task.

Model

Since, we have labelled training data and a continuous value to predict, we would be using a supervised regression learning model. Based on the results of multiple sources of papers, we have identified that we would be using a Random Forest Regression model for our use case.

We used the Apache Spark MLlib library, which has a rich set of APIs for various Machine Learning models on the NYU HPC Cluster, to call functions that help us create the random forest model. We loaded our merged dataset in LIBSVM format into HDFS, then loaded the dataset in Apache Spark into a Spark dataframe and trained our model with the dataframe.

Evaluation Metrics

We use a 70:30 split of training data to test data so we can successfully define the metrics of our model. Once our model is trained, we use the model to predict the "Probabilities" of each row present in the test dataset. We achieved an accuracy of 80%, a precision of 85%, a recall of 80% and a F1 score of 82%. Among various predictions from the model, one notable prediction was that of the heavy flooding that occurred in California throughout the early months of 2017. This prediction matches past reported news and observations posted about the flood [17], which also serve as a measure of validation that the model has the capability to predict accurately given the weather data used for our application.

VIII. Conclusion

We created a flood prediction application that uses various sources of publicly available weather data with a random forest model. After cleaning and extracting features from the data, we utilize various tools like Spark and MLlib along with

GeoPandas and the United States GeoJSON dataset to train a model that has an accuracy of 80% and a precision of 85%. The prediction results of the model are displayed in a Tableau workbook which has seen success in predicting floods that have occurred in the real world. We believe that this application can be used as another tool alongside existing technologies for flood detection and prevention. Our application will be able to support and validate findings from existing tools, which will result in a more thorough check for floods. This can hopefully prompt more preemptive action from various counties across the United States to mitigate damages from flooding.

IX. Future Work

In future work, we would like to further investigate how to integrate a stream of real-time weather data provided by the NCDC and the NOAA with the application. This allows us to generate predictions of future flooding in a particular region in a more efficient manner. Further improvements on the model can also be made with better feature engineering and fine-tuning parameters. Finally, we'd like to use the existing data to provide more information about the intensity of floods that may occur alongside the probability of flooding.

Acknowledgment

We would like to thank NYU HPC Cluster for providing a cluster to store the data and run the codes. We also thank the United States National Climatic Data Center (NCDC) and the National Oceanic and Atmospheric Administration (NOAA) for providing datasets for us to use in this project.

References

1. K. Jitkajornwanich, U. Gupta, R. Elmasri, L. Fegaras, and J. McEnery. Using MapReduce to Speed Up Storm Identification from Big Raw Rainfall Data. CLOUD COMPUTING 2013 : The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization. IARIA, 2013.
2. T. K. Apostolopoulos, and K. P. Georgakakos. Parallel computation for streamflow prediction with distributed hydrologic models. Journal of Hydrology Volume 197, Issues 1-4, 1, Pages 1-24, October 1997.
3. G. Arduino, P. Reggiani, and E. Todini. Recent advances in flood forecasting and flood risk assessment. Hydrology and Earth System Sciences, 9(4), 280-284, 2005.
4. S. H. Elsafi. Artificial Neural Networks (ANNs) for flood forecasting at Dongola Station in the River Nile, Sudan. Alexandria Engineering Journal Volume 53, Issue 3, Pages 655-662, September 2014.
5. K. P. Georgakakos. REAL-TIME PREDICTION FOR FLOOD WARNING AND MANAGEMENT. U.S.- Italy Research Workshop on the Hydrometeorology, Impacts, and Management of Extreme Floods Perugia (Italy), November 1995.
6. A. Mosavi, P. Ozturk, and K. Chau. Flood Prediction Using Machine Learning Models: Literature Review. Water 10(11):1536, October 2018.
7. A. Paul, and P. Das. Flood Prediction Model using Artificial Neural Network. International Journal of Computer Applications Technology and Research Volume 3, Issue 7, 473-478, 2014.
8. C. Tang, D. Garrea, and U. von Luxburg. When do random forests fail? 32nd Conference on Neural Information Processing Systems, Montreal, Canada, NeurIPS 2018.
9. R. Xu. Improvements to random forest methodology. Graduate Theses and Dissertations. 13052, 2013.
10. R. Trifonov, R. Yoshinov, G. Pavlova, and G. Tsochev. Artificial neural network intelligent method for prediction. AIP Conference Proceedings 1872, 020021, 2017.
11. S. Goswami, S. Chakraborty, S. Ghosh, A. Chakrabarti, and B. Chakraborty. A Review on Application of Data Mining Techniques to Combat Natural Disasters. Ain Shams Engineering Journal Volume 9, Issue 3, Pages 365-378, September 2018.
12. D. F. Lekkas, C. Onof, M. J. Lee, and E. A. Baltas. Application of Artificial Neural Networks for flood forecasting. Global Nest: the Int. J. Vol 6, No 3, pp 205-211, 2004.
13. V. Thiemig, B. Bisselink, F. Pappenberger, and J. Thielen. A pan-African Flood Forecasting System. Hydrology and Earth System Sciences Discussions 11(5):5559-5597, May 2014.
14. M. A. Sit, and I. Demir. Decentralized Flood Forecasting Using Deep Neural Networks. arXiv:1902.02308v1, February 2019.
15. E. Michael, and O. Patience. Flood Prediction in Nigeria Using Artificial Neural Network. American Journal of Engineering Research (AJER) Volume-7, Issue-9, pp-15-21, 2018.
16. C. Choi, J. Kim, J. Kim, D. Kim, Y. Bae, and H. Kim. Development of Heavy Rain Damage Prediction Model Using Machine Learning Based on Big Data. Hindawi, Advances in Meteorology Volume 2018, Article ID 5024930, 11 pages.
17. 2017 California Floods. https://en.wikipedia.org/wiki/2017_California_floods