Machine Learning Wildfire Prediction based on Climate Data

Group 75

Yujian Xiong Jie Wu Zizhan Chen

***Abstract*—We made a complete analyze over 1.8 million US wildfire cases from 1992 to 2015, extracting climate data of the fires' occurrence. Then we implemented multiple machine learning methods (focus on Gradient Boosting Decision Tree and Deep Neural Networks) to predict wildfire size based on climate features. Our accuracy is limited around 32%. We suspect the reasons to include unpredictable human activities, low correlation from monthly-average climate data, and lack of geographical features including elevation, slope, and soil type.**

***Index Terms*—Wildfire, Prediction, Climate Hazards, Greenhouse Gas**

I. Introduction

Wildfires, are a popular and imperative topic in weather research and hazard detection, not only because of its uncertainty and difficulty to predict, but also because of its sever damage to our environment and huge contribution to Greenhouse gas and global warming, specially in high forest carbon stock areas like California. Our test shows the same results 1, number of wildfires is increasing over recent years, and the annual total burned area is kind of correlated to some Greenhouse gas emissions.

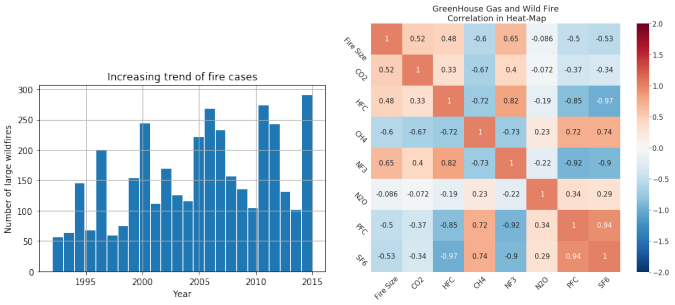

Fig. 1. Increasing trend of wildfire (left) and its correlation heat-map between Greenhouse gas

Though wildfire prediction is not an on-heat topic in ML area, there are still a lot of researchers who kept working on this topic in recent decades.

Predict wildfire cases based on climate data is a basic topic among all prediction methods. We are interested in this special topic because its relation with physics and the good feature structure of wildfire case data and climate data. In this project, we will work on a US wildfire dataset from kaggle, and find various climate datasets from different official climate administration, and extract features from all of them. Then we will apply different ML algorithms on these data try to make predictions on wildfire class, and compare their performance.

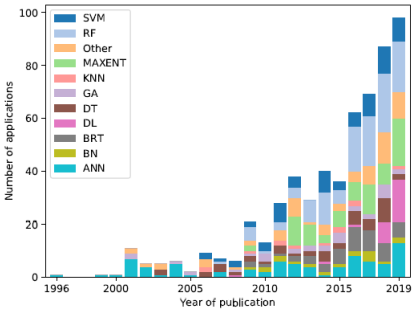

Fig. 2. Number of ML applications on wildfire science by category and by year

II. Related Work

ML methods for fire occurrence and severity prediction have been used extensively in early studies. The most commonly used ML method in studies predicting fire occurrence were ANNs before 2010s. In 1996, Vega-Garcia, C., Lee el al. [1] has started to apply neural network technology to wildfire occurrence prediction in Canada, and achieves prediction accuracy of 85% for no-fire observations and 78% for fire observations. In the early 2000s, Alonso-Betanzos et al. [2] also used ANN to predict daily fire occurrence risk using humidity, rainfall, temperature and fire history in the Galicia region of Spain.

In recent years, random forest and ensemble algorithms have become the more popular methods [3] (Figure 2). Stojanova, Kobler, Ogrinc, Zenko, and Dzeroski [4] evaluated multiple machine learning methods for predicting wildfire occurrences using meteorological data in Slovenia, including ensemble methods (RF, Decision Trees with bagging and AdaBoost) and single classier methods (i.e., KNN, Naive Bayes, SVM and Decision Trees). The experiment shows that RF and decision trees with bagging displayed the best performance on prediction precision. For the ensemble methods, Dutta et al. [5] applied a two-stage ML approach, with the first stage DBNet generating features from climate data and then fed the extracted features to ten ML classifiers in the second stage. The author found that conventional KNN and bagging trees were the two classifiers with the best performance of 91.8% and 94.5% accuracy.

Other methods such as the maximum entropy (MaxEnt) has also been applied for wildfire risk forecasting. For instance, De Angelis et al. [6] used MaxEnt to evaluate predictive

power of different climate variables and fire indices and found that the combinations of different meteorological variables and fire indices are able to improve predictive robustness for identifying fire weather conditions.

III. Datasets and Features

The base data is the wildfire of US dataset from kaggle [12], it contains 1.8 million wildfire cases from 1992 to 2015. Each fire case contain the Latitude and Longitude of its location, the date and time when it happened, and variable fire-size indicating the total area it burned down.

An interesting part is that compared to do regression on fire-size, it is more convenient and accurate to make prediction on fire-class. According to an official glossary from National Wildfire Coordinating Group, we can divide different fire cases into 1 to 7 (or 'A' to 'G') fire-class based on their size [17]. They contain a total of 12 climate features, such as different component of wind-speed at different height, precipitation, different measurement of temperature, and vegetation coverage. Besides these, we also added Longitude and Latitude as 2 geometry features.

For our prediction task, we need to find climate data for each single fire case, so datasets which cover same or larger area and time period than the wildfire data is preferred. We finally managed to collect 14 climate features from different climate datasets [14]–[16].

The next step is extracting climate data on the special time and location of the fire case to indicate the weather condition when the wildfire occurs. Due to the restriction of our datasets, only monthly-average climate features are available. So, several query functions were used to extract climate data based on Latitude, Longitude, and month of occurrence. With this, we can extract climate condition from multiple datasets corresponding to only 1 wildfire event, we overlapped the fire case of May to July in 2004, Alaska, to give a visual understanding of the relationship between fire size and features 3. A correlation heat-map between fire size and features are also shown as an numeric example.

Our device is not capable for working on all 1.8 million fire cases, and the time cost may also be redundant. So we did random sampling from all fire case, to generate a smaller data frame for model training. In the sampling process, we keep identical case number of each fire class.

IV. Methodology

Let $F(x)$ denoting the machine learning model which generates classification prediction with climate data as its input.Out method is to use decision trees output f as an approximation for $F(x)$ by the weighted sum of all classification results from decision trees. [11]

$$F \approx \sum_{i}^{T} r_i f_i = \sum_{i}^{T} \sum_{k}^{L_i} r_i b_{ik} \omega_{ik}$$

where T is the number of decision trees, L_i is the number of leaves in ith tree, ω_{ik} is the parameters deviding each region, and b_{ik} is the value predicted in the leaf.

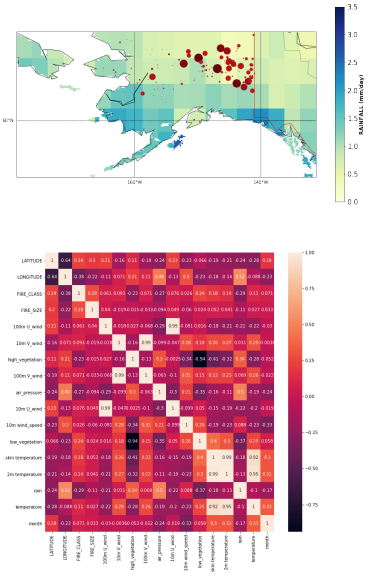

Fig. 3. The overlapping of wildfire case and precipitation in AK (top) and the correlation heat-map between fire size/class and climate features (bottom)

Our aim is to find a model F, such that minimizes loss function $L(F, y)$. The method to update F is by finding a steepest descent gradient.

$$r_t = -\frac{\partial L(y_t, F(x_t)}{\partial F(x_t)}$$

Update the decision tree

$$\omega_{it} = \text{argmin}_{\omega_{it}} L(f_{i,t}(x_t)), i = 1, \ldots, T$$

Finally, we update F with a regulation factor ν called learning rate.

$$F_{t+1} = F_t + \nu r_t \sum_{i}^{T} \mathbf{b_{i,t}} \omega_{\mathbf{i,t}}$$

A. Deep neural networks

Neural network is composed of connecting nodes divided into input layers, hidden layers and output layers. The data are input through input layer, combined at hidden layer, and finally we use the output of output layers as our prediction.

Only by linear combination the model cannot deal with non-linear and complex problems, so there are activation functions between each layer.

$$a(x) = b + \mathbf{w^T x}$$

$$h(x) = g(a(x))$$

As for out model, we use 4 hidden layers, and ReLu as activation functions acting on hidden layers. The activation function for output layers is Sigmoid, a commonly used function for classification. The approximated number of node of hidden layers is around 20 to 300 hundred, referring to empirical formulas $\sqrt{N_x N_y}$,$\sqrt{2 * (N_y + 1) * N_s}$, where

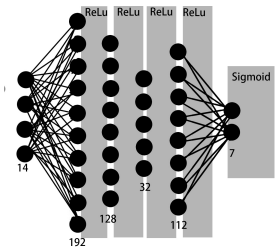

Fig. 4. Neural Network Structure

$N_x = 14, N_y = 7$ is the number of input and output respectively, $N_s = 5600$ is the number of samples [10].

In practice, we find node number and layer number is by Tuner, a hyper parameter adjusting tool on Keras. Giving several possible values to choose from, Tuner will select values randomly. Finally, after trying a sufficient number of times, it will output a combination of hyper parameters that produces the best accuracy. Figure 4 shows the structure of our model.

V. Experiments and Results

We implemented a variety of machine learning methods to find the optimal choice of model on wildfire occurrence prediction based on climate data. In this section, we will describe our experiments in detail and analyze their performance results. Our emphasis will be on Gradient Boosting Decision Tree and Deep Neural Network methods. For all of our experiments, we use UCSD Datahub as our computing platform to enable maximum model training efficiency.

A. GBDT

Our implementation of Gradient Boosting Decision Tree is mainly based on sklearn library GradientBoostingClassifer [7]. The overall hyper-parameter of the GBDT framework can be divided into two categories: Tree-specific parameters and boosting parameters. Tree-specific parameters affect each individual tree in the GBDT model while boosting parameters only affect the boosting operation in the model. By varying the parameters under these two categories, our experiment is designed to find the GBDT model with optimal performance for wildfire prediction.

In our experiment, the hyperparameters we focused on include boosting parameters n_estimators, learning_rate, and subsample, as well as tree parameters max_depth, min_samples_split, min_samples_leaf and max_features. "n_estimators" directly controls the number of sequential trees to be modeled in the framework. "learning_rate" controls the magnitude of estimated changes using the output of each tree, and it determines the impact of each tree on the final outcome. Although gradient boosting is fairly robust to overfitting, considering the trade-off between n_estimators and learning_rate, these two parameters are tuned together. "subsample" decides the fraction of observations to be selected for each decision tree through random sampling. A slightly smaller value tends to make the model more robust by reducing the variance. "max_depth" determines the maximum depth of the individual estimators and thus limits the number of nodes in the tree. It is often used to control overfitting as higher depth will let the model to learn specific relations to a particular sample. "min_samples_split" defines the minimum number of samples required to split an internal node, and "min_samples_leaf" controls the minimum number of samples required to be at a leaf node. By optimizing both of these two parameters in a relatively large dataset can effectively help reduce overfitting. "max_features" refers to the number of features to consider when searching for the best split. Reducing max_features to be less than the number of total features would lead to a reduction of variance but an increase in bias.

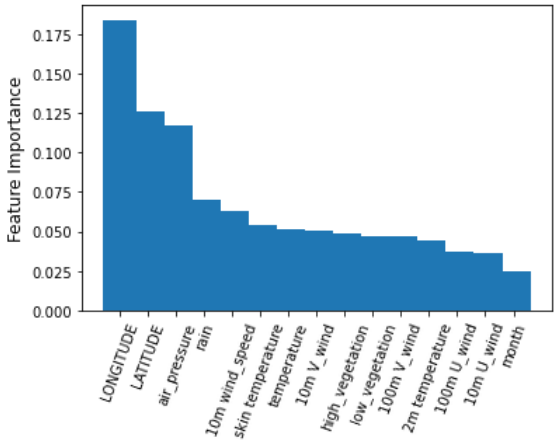

Fig. 5. GBDT feature importance

We started with default parameter settings of GradientBoostingClassifer to create a baseline model for evaluating model performance. The evaluation metric is ROC-AUC with "onv vs. rest" method which computes the AUC of each class against the rest. By applying exhaustive search in combination with cross validation, we tuned the GBDT parameters stage by stage considering the correlations between some of the parameters as described above. At last, the optimized model achieves a training score of 0.6577 and test accuracy of 0.3293, and shows improvement from the initial model (0.5696 and 0.3228 respectively).

The low accuracy can be partially explained by the feature importance generated from the final GBDT model (Figure 5). The chart shows that although geographic coordinate and air pressure contribute the most for the model, other features lack distinctive correlations with the predicted labels. Therefore, existing features in the dataset limits the maximum prediction accuracy that can solely be achieved by optimizing the prediction models.

B. DNN

Our implementation of the Deep Neural Network is based on Tensorflow [8]. The model features consecutive dense and dropout layers. The exact number of dense layers and neurons within each hidden layer are optimized using Keras Tuner.

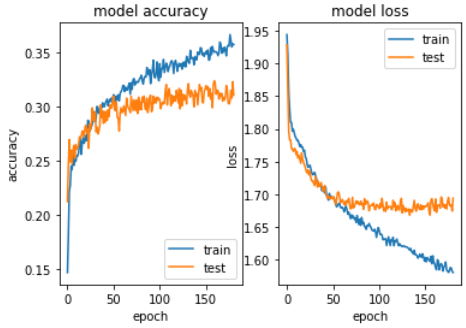

Fig. 6. DNN training process

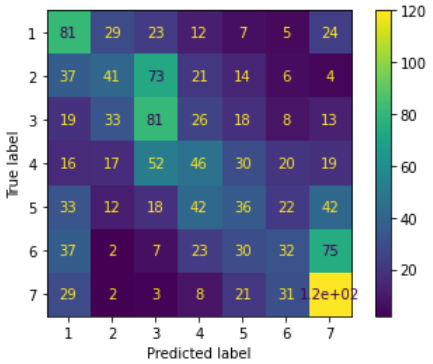

Fig. 7. DNN confusion matrix

We limit the max number of dense layers to five and use ReLU (Rectified Linear Units) as activation function for each dense layer except the output. The dropout layers are added after each dense layer with a dropout rate of 0.2 for optimal regularization on the DNN model. The default loss function categorical_crossentropy is used here considering the multi-class identity of fire class prediction. The improved stochastic gradient descent method, Adam is utilized as a training optimizer for the model to maximize model performance.

The training process of the final model is shown in Figure 8. The training process stops at around 180 epochs when validation loss stops improving. The trained model shows a final training loss of 1.5804 with 35.76% accuracy. On test data, the prediction observers a loss of 1.6749 with 31.71% accuracy. The confusion matrix on the test data is shown in Figure 7.

Notice that, from the confusion matrix, prediction precisions are higher on extreme cases (Fire Class 1 and 7) than others. We think the reason is that extreme fire weather such as lack of rain and high temperature tends to increase the probability of wildfire occurrence. And thus makes extreme fire class cases relatively easy to be distinguished from other fire classes, resulting in higher prediction precision.

C. Other models

Figure 8 shows other ML methods we attempted on wildfire prediction. The reasons for low accuracy results might include highly randomized wildfires occurrences. Researches have shown that human behaviors are the major factor for wildfires [9]. Without valid predictions on human behaviors, it is difficult to only rely on climate data. Another reason might be some climate features themselves are not well correlated to wildfire occurrences.

Model	Accuracy	CV Accuracy
Gradient Boosting Trees	31.90	32.73
Random Forest	31.52	31.70
Ada Boost	28.71	29.08
Decision Tree	28.19	26.93
Naive Bayes	26.86	26.65
Logistic Regression	25.86	25.31
KNN	21.14	20.84
Linear SVC	20.57	18.13
Stochastic Gradient Decent	15.19	14.75

Fig. 8. Results from other models

VI. Conclusion

In this project, we implemented multiple machine learning methods to predict wildfire occurrence and perimeter based on climate data. The models we attempted to apply include Gradient Boosting Decision Tree, Deep Neural Networks, and other conventional machine learning methods. Although fine tuning of hyperparameters was also conducted, we observed limited accuracy among all models on the current dataset. GBDT and DNN achieve the highest prediction accuracy at around 32%. The reasons behind prediction inaccuracy might include unpredictable human-caused wildfires, low correlations between climate data and wildfire, and highly biased fire severity data. For the future development of wildfire prediction, we think it is crucial to include human factors, such as regional population and industrial activities to establish a more robust model. Also, more reliable geographical features including elevation, slope, and soil type may also help improve the performance of prediction models.

CONTRIBUTIONS

- Yujian Xiong: Data cleaning, extracting features, implement models, making presentation, writing report
- Jie Wu: Parameter tuning, result analyze, implement models, making presentation, writing report
- Zizhan Chen: Algorithm introduction, implement models, making presentation, writing report

REFERENCES

[1] Vega-Garcia, C., Lee, B., Woodard, P., & Titus, S. (1996). Applying neural network technology to 61 human-caused wildfire occurrence prediction. AI Applications, 9-18.

[2] Alonso-Betanzos, A., Fontenla-Romero, O., Guijarro-Berdinas, B., Hernandez-Pereira, E., Paz Andrade, M. I., Jimenez, E., . . . Carballas, T. (2003). An intelligent system for forest re risk prediction and fire fighting management in Galicia. Expert Systems with Applications, 25 (4), 545-554. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0957417403000952

[3] P. Jain, et al., (Mar, 2020) "A review of machine learning applications in wildfire science and management." Retrieved from https://www.researchgate.net/publication/339641999_A_review_of_machine_learning_applications_in_wildfire_science_and_management/citation/download

[4] Stojanova, D., Kobler, A., Ogrinc, P., Zenko, B., & Dzeroski, S. (Mar, 2012). Estimating the risk of fire outbreaks in the natural environment. Data Mining and Knowledge Discovery, 24 (2), 411-442. Retrieved from http://link.springer.com/10.1007/s10618-011-0213-2

[5] Dutta, R., Das, A., & Aryal, J. (Feb, 2016). Big data integration shows Australian bush-re frequency is increasing significantly. Royal Society Open Science, 3 (2), 150241. Retrieved from https://royalsocietypublishing.org/doi/10.1098/rsos.150241

[6] De Angelis, A., Ricotta, C., Conedera, M., & Pezzatti, G. B. (Feb, 2015). Modelling the Meteorological Forest Fire Niche in Heterogeneous Pyrologic Conditions. PLOS ONE, 10 (2), e0116875. Retrieved from https://dx.plos.org/10.1371/journal.pone.0116875

[7] Pedregosa et al. (2011). Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830.

[8] Martín Abadi, et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org

[9] Grala, K, et al. (Apr, 2014) "Impact of human factors on wildfire occurrence in Mississippi, United States." Retrieved from https://www.researchgate.net/publication/316908932_Impact_of_human_factors_on_wildfire_occurrence_in_Mississippi_United_States

[10] Stathakis, & D. (2009). How many hidden layers and nodes?. International Journal of Remote Sensing, 30(8), 2133-2147.

[11] Friedman, J. H. . (2001). Greedy function approximation: a gradient boosting machine. Annals of Statistics, 29(5), 1189-1232.

[12] US wildfire dataset
https://www.kaggle.com/rtatman/188-million-us-wildfires

[13] Greenhouse gas emission dataset
https://www.kaggle.com/unitednations/international-greenhouse-gas-emissions

[14] Precipitation dataset from NOAA Physical Science Laboratory
https://www.esrl.noaa.gov/psd/data/gridded/data.cmap.html

[15] Temperature dataset from NOAA Physical Science Laboratory
https://psl.noaa.gov/repository/a/psdgrids

[16] Various ERA5 monthly averaged climate data from ECMWF
https://www.kaggle.com/unitednations/international-greenhouse-gas-emissions

[17] Official glossary of fire class category
https://www.nwcg.gov/term/glossary/size-class-of-fire

Replies to Critical Review

Critiques by group 22.

1. The greenhouse gas correlation map is a bit confusing to me. For example, CO2 has a 0.52 correlation with wildfire size. What does a 0.52 correlation mean?

Response: This is the correlation between annual gas emissions of a certain kind of gas (such as CO2) and annual total burned down area (which is calculated by summing up all fire case's fire size).

2. The explanation for "Why ML" seems to mainly be that "whether conditions are featuralizable and human activities are related to whether conditions". This explains why you can use ML, but not why you should use machine learning.

    Recommendation: It may help to emphasise more on the difficulty and unpredictable nature of wildfires and then explain why ML is better than other solutions (are there any other types of approaches to predicting wildfires?). You could say that the problem of predicting wildfires ultimately boils down to a complicated pattern recognition problem and that machine learning is exceptional for complicated pattern recognition. Another solution to the predicting wildfires/weather problem is physics simulations, but maybe those are expensive and implementing ML techniques may be less expensive, quicker and just as accurate?

Response: Good suggestion! Actually we think this part is just recommended but not compelling, it might be boring if every group spends a lot of time on why ML is better than others in their topic. However, it is a good suggestion to compare ML predicting and non-ML predicting, as we discussed in group 89's question 5.

3. Why did you choose these models? Were there any other models you considered trying? What hyperparameters did you have the tuner select? Was it just the layer and node number? If so, how did you go about selecting other hyperparameters?

Response: See "Experiment and Results" part in our report.

Critiques by group 44.

1. For the dataset part, group 75 selects 14 weather features and 20k random samples. Here I think the number of features seem not sufficient. It might be the limitation of dataset that causes the low accuracy in the later training. If we can have time-series weather features (e.g. different temperatures within 30 days before the fire) and combine them into temporal features, it might perform better during the training.

Response: That is a quite insightful suggestion! Although some of our data does not have continuous records of weather conditions, adding time-series features and some model structures like RNN or LSTM are likely to provide more reliable features for the model.

2. At first, it is mentioned there're 2 labels of the dataset: fire class and fire size. It seems training was only performed on classifying the fire class. Since you don't have a very good result on fire class classification due to human factor or dataset limitation, it might be also helpful to do regression on fire size given a certain class fire. (might be a interesting topic as well)

Response: It turns out that fire size is also largely biased and our initial attempt on regression did not return better results than fire class prediction. This is exactly the inspiration for us to divide fire cases into different class, and do classification instead of regression.

3. For some of the complex models, there might be overfitting problem according to the training/validation loss plot. Since you only have 14 features, a boosting tree with too many leafs is very easy to overfit on your dataset.

Response: We limited the number of leaves by using optimized min_samples_split, min_samples_leaf parameter in GBDT. To completely eliminate overfitting in decision tree or similar models is not very realistic.

Critiques by group 89:

1. Features in the dataset should be explained in further detail. What is the data exactly? Is it mean monthly data, mean of data over a certain period around the fire, or something else?

Response: The climate data (from precipitation to vegetation) are all monthly-averaged data, as we mentioned in our presentation.

2. Based on the details of features, you can modify the features like instead of taking absolute values, can you use something like the difference between successive values of features around the time of the fire?

Response: That's a very good point! Actually we have though and even made some plots on analyze based on relative increase/decrease instead of absolute value, but the difference between them and our current method is very small.

3. Your dataset is highly unbalanced. You showed that the distribution of fires in different classes is highly uneven. Based on my understanding, you should have a balanced dataset to train your model.

Response: As we mentioned in our presentation, we randomly sampled from the whole fire dataset, but it is not totally random, we kept the same value of sample number of each fire class. If randomly selected our samples, most fire cases will be of very small fire size, which is not good for model training.

4. You did not mention the accuracy of the models found in the literature review. So, it is difficult to say how well your models performed.

Response: We found that the prediction accuracy in literature varies significantly (43%-97%) in considering different features and models they applied.

5. Also, apart from ML, are there any other standard methods used for this purpose? If so, how well do they perform?

Response: Conventional fire predictions have been mostly based on fire indexes such as Canadian Fire Weather Index (FWI) system, which uses output from weather prediction models. The errors in weather prediction can accumulate in the fire predictions and the FWI seems to lack the ability of self-correction.