Wildfire Prediction Through Live Fuel Moisture Content Maps

Abstract

As a result of climate change, wildfires are becoming more common. In recent years, higher resolution remote sensing data is becoming available over longer periods of time. One such example is the live moisture content (LFMC), which are maps that show the mass of vegetation water per unit dry biomass, a key factor determinant of wildfire risk. Using this dataset in combination with ground truth wildfire data from USGS, I predict whether there will be a fire in the near future within a 8km grid using machine learning models such as SVM, random forest, and neural networks. I conduct two main experiments, using a single image within 15 days of a fire, and a concatenation of three images within 3 months of a fire. My results show that the random forest performs best with an accuracy of 71.95.% for a single image, and 73.81% for multiple images.

1. Motivation

As a result of climate change, the occurrence of wildfires worldwide is increasing. In 2019, California recorded its worst wildfire season in recorded history. A year later, the record was smashed yet again, with 2020 beating the previous all-time record. In fact, seven of the 10 most destructive fires in the state's history have burned in the last five years [1].

In general, wildfire risk depends on three quantities, which are the availability of ignition sources, the propensity of vegetation and litter fuel to ignite, and the ease of fire spread once the fuel has ignited [3]. With the increasingly higher resolution remote sensing available over longer periods of time, the use of remote sensing for wildfire prediction is increasing. Past research on wildfires have used machine learning to address questions such as fire detection and mapping, fire weather and climate change, fire behavior prediction, and fire occurrence, susceptibility and risk [2]. Though studies on fire susceptibility has been going on for almost a hundred years, only several recent studies have leveraged the availability of past wildfire data to predict the potential fire risks in the future. For example, Zhang et al. [6] used a CNN model to obtain a spatial prediction of fire susceptibility in Yunnan Provice, China while Tonini et al. [4] used random forest to obtain the susceptibility maps in Liguria, Italy.

Rao et al.[3] developed live fuel moisture content (LFMC) maps, which are maps that show the mass of vegetation water per unit dry biomass, a key determinant of all three components of wildfire risks. Using a combination of optical remote sensing, physics-assisted features, and historical field data from the National Fuel Moisture database, Rao et al. trained a deep learning model to generate LFMC maps every 15 days at 250m resolution over western US. Though the maps are useful for wildfire risk characterization, LFMC maps have not been used to predict the potential fire risks of the future.

Given this motivation, I applied machine learning models to predict potential fire risks based on the LFCM maps and historical wildfire data. In addition, I explored experimental questions such as how time windows used for training the model can influence the accuracy of the prediction. As a first step, I focused on answering the question: *Given the current moisture levels, will there be a fire within an 8km grid in the next 15 days?*

Given that the datasets have different time and geographical spans, for this project, I limited the scope of the project to the California region in a time span between January 2016- July 2020. The input to the algorithm is a single-channel image that has a coverage of 8kmx8km within 32x32 pixels. I then used a variety of different algorithms including SVM, random forest, and a neural network to output a binary classification prediction of whether there is going to be a fire in the next 15 days.

I also expanded the experiment by experimenting with concatenated images that span 3 months. That is, instead of a single-channel image, I concatenated 3 images from three different dates a month apart into a single three-channel image. Therefore the question in this case is: *Given the moisture levels of the past 3 months, will there be a fire within an 8km grid in the next 15 days?*

The rest of the paper is organized as follows. In Section 2, I describe the dataset I used, including pre-processing methods. In Section 3, I describe the methods used to approach the problem and the machine learning algorithms used. In Section 4, I discuss the results before ending the paper with the conclusions.

2. Dataset and Features

I used live fuel moisture content (LFMC) maps, which were generated by Rao et al. [3]. These maps are available at `https://github.com/kkraoj/lfmc_from_sar` and are available every 15 days at 250m resolution over western US. These maps are available from January 2016 until July 2020.

For ground truth wildfire data, I leveraged a publicly available dataset from USGS [5]. This dataset is a polygon geospatial dataset ranging from 1878-2019 that contains all the wildfires that have occurred in the US.

Because the publicly available dataset from USGS is available in the format of a ESRI polygon file, I first had to convert it to a format that can be used by machine learning models. To simplify the problem, I first filtered for fires that occurred within the span of January 2016-July 2020 in California.

Then, I created a 8km x 8km grid of points (plus some random noise) in the region of interest. The points can be seen in Figure 1. I extracted points that were within a historical fire and labeled that as "fire", while points not within a historical fire polygon was labeled as "non-fire". For the "fire" points, I also included the date of fire.

Once the points were obtained, I extracted 8km x 8km (32x32 pixels) images from the LFMC maps to create our "fire" and "no fire" training samples with the point as the center of the image. For the "fire" samples, I made sure to extract the image from the nearest date to the fire. For example, if the fire occurred in January 6th 2018, then I extracted the LFMC map of January 1st 2018 (because this is the closest pre-image to the fire).

For the non-fire samples, I employed two different sampling techniques. For points that had no fire historically, I randomly sampled a date to extract the image from. I also sampled from points that had a fire historically. However, I made sure to put a buffer between the fire date that occurred. For example, if the fire occurred in January 6th 2018, then I made sure to sample a random date that is not within one year of the fire (both before and after).

For my second main experiment, I concatenated images that spans 3 months. For example, if the fire occured in January 6th 2018, I extracted the LFMC map of the closest image (January 1st 2018), and took two other images two months prior. In this case, December 1st 2018 and November 1st 2018. Therefore, the input size of the image for this experiment is 3x32x32 instead of just 1x32x32 as before. By doing this, I wanted to see if more moisture information prior to the the fire will enable us to have more accurate predictions.

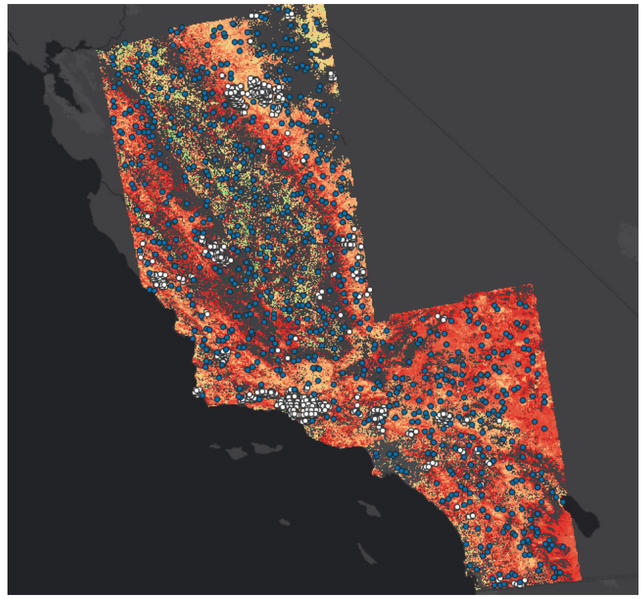

Figure 1. Sampled data points overlayed over a sample LFMC map. White points are classified as fire, because they are within a polygon where there was historically a fire, while blue points are classified as non-fire.

The final dataset contains 1928 fire samples and 1600 non-fire samples. Figure 2 shows several sample data. I split the dataset into 80% training set and 20% validation set.

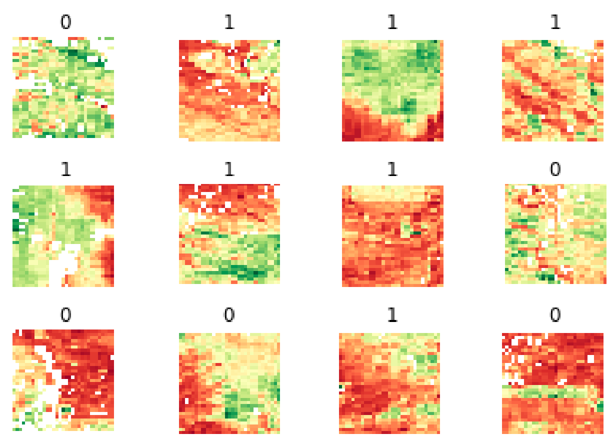

Figure 2. Sample data with 1 being there is a fire in the next 15 days.

3. Methods

In this section, I describe the machine learning models I used and the error metrics that I used to quantify how well my models were doing.

3.1. Machine Learning Models

I experimented with several machine learning algorithms to perform our binary classification (fire or no fire), such as SVM, Random Forest, and CNN.

SVM: The SVM is a supervised classification algorithm that uses a subset of training points in the decision function. The advantage of SVM is that it is effective even in high dimensional cases, and more robust to outliers. In this study, I used the radial basis function (RBF) kernel to produce more non-linear boundaries. I performed cross validation to obtain the optimal kernel coefficient γ and the regularization parameter.

Random Forest: Random Forest is an ensemble learning method that works for many different types of machine learning application, including classification. Random Forests has many advantages as it is a simple algorithm that works well and can handle thousands of input variables. There are many different hyperparameters that can be tuned in a Random Forest. For this project, I tuned the number of estimators, the maximum depth, the minimum samples before the tree needs to be split, and the minimum samples for each leaf. To obtain the optimal hyperparameters, I performed a gridsearch CV.

CNN: In this paper, I experimented by using a pretrained ResNet-18 as the architecture. Given that ResNet is trained for images with 3 input channels, and my input data is only 1 channel for the first experiment, I duplicated the channel 3 times to feed into the network. I trained for 25 epochs.

3.2. Evaluation Metrics

The evaluation metrics used to reflect the overall performance of the binary classification model is accuracy, which is the ratio of number of correct predictions to the total number of input samples. It can also be written in the following equation:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

In addition, I also look at the precision and recall values which are defined as the following:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4. Results and Discussion

In this section, I discuss my results for the two main experiments I conducted, using single image and multiple image.

4.1. Single Image

In this experiment, I used only a single one-channel LFMC image as the input to the model. The image corresponds to the closest date the image was captured to the actual date of the fire. All the experiments were implemented in Python using Sklearn and Pytorch. First, I experimented by using SVM as the classifier. I performed cross validation to obtain the optimal hyperparameter on 3 folds with an RBF kernel. Using cross validation, the optimal SVM parameters is $\gamma = 0.001$ and the regularization $C = 1$. The optimal validation set accuracy is 65.86% while the training set accuracy was 100%. I thought that this was an interesting result because this means that the model overfit to the training data, but still performed the best on the validation set. The confusion matrix can be seen in Figure 3. Here, the SVM model actually never made the error of predicting "fire" when there was actually no fire. However, in reality, it may have been better if our model would predict more false positives versus false negatives. That is, it is better if the model predicts that there is fire, while there is none, instead of the other way around.

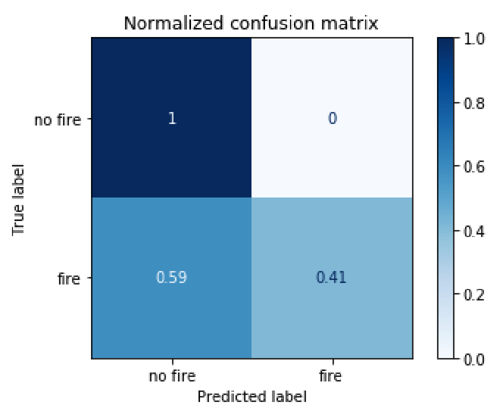

Figure 3. Confusion Matrix of SVM results

Next, I experimented with Random Forest as it is known to be a very simple algorithm but can handle large sizes of input variables. Similar to the SVM, I performed cross-validation with 3 folds to obtain the optimal hyperparameters of the the Random Forest model. The optimal parameters for the random forest are: Maximum depth = 25, Minimum Samples per Leaf = 2, Minimum samples before a split = 5, and number of estimators = 500. Using these optimal hyperparameters, the model achieved a validation score of 71.95%, slightly higher than the SVM model. The model had a precision of 73% and recall of 82% for the fire (positive) label. The confusion matrix can be seen in Figure 4.

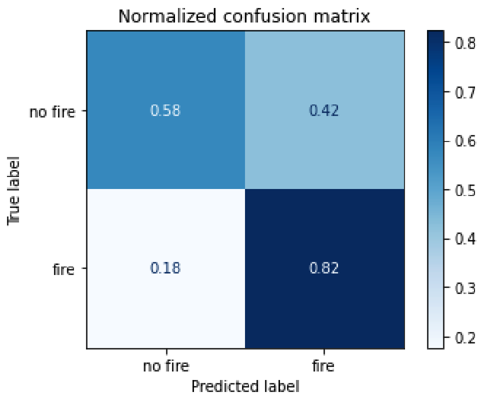

Figure 4. Confusion Matrix for Random Forest Results using 1 image

Finally, I experimented with a ResNet-18 CNN model to predict whether or not there is going to be a fire in the next 15 days. As mentioned previously, I duplicated the input data, which is only 1 channel, three times in order to feed it into the network. Given the limited training data, I performed transfer learning for the ResNet-18, pre-trained of ImageNet. I only retrained the last fully connected layer and changed the output to be a binary classification. The ResNet achieved an accuracy 65.43%.

A summary of the best performing models I experimented with can be seen in Table 1.

Method	Validation Acc.
SVM	65.86%
Random Forest	71.95%
ResNet-18	62.46%

Table 1. Preliminary Results

From the accuracy results, Random Forest actually performs best out of the three models. SVM did not perform as well because perhaps I was using the wrong kernel, which might not be representing the data well. On the other hand, I also think that ResNet did not perform well because the model was pretrained on ImageNet, which is very different images to that of the LFMC maps. Another reason why the models might not perform well is because the maps have too much noise, and might not have that much information that can actually predict the fires. Furthermore, there are tiles in which there are missing pixels, which might cause the accuracy to go down. Examples of such samples can be seen in Figure 5

4.2. Multiple Images

To see if adding additional information would help improve my results, I tried experimenting with input data consisting of multiple images concatenated together. The aim

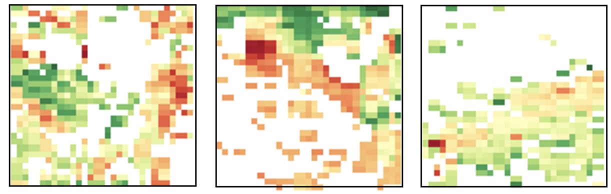

Figure 5. Sample of missing pixels

of this is to see whether or not adding information 3 months before the fire can help us predict better. Here, I only focused on using Random Forest and the CNN.

For the Random Forest, I flattened each datapoint into a single array and fed that into the model. Therefore, the input size of each sample is $3 \times 32 \times 32 = 3072$. Using this model, the accuracy of the validation set is 73.80%, slightly higher than just using a single image, which suggests that adding additional information over time is slightly helpful when trying to predict a future fire. The confusion matrix can be seen in Figure 6. For the positive class, the precision is 76% and the recall is 80%. In this case, the recall is better than the precision, which is something we would want to prioritize in a fire predicting model. Only a very few percentage of the images were classified as no fire, when there was actually a fire. However, given that I only flattened the image into a single array, there may be too many features for the model to learn. Perhaps pruning the features might increase the accuracy even more.

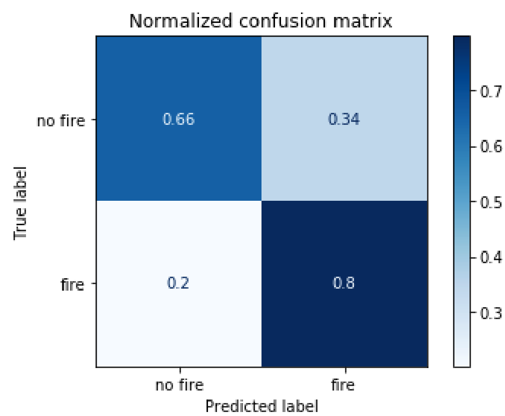

Figure 6. Confusion Matrix for Random Forest Results using 3 images

Using similar methods as the previous subsection, I also trained a ResNet-18 model for 25 epochs using this 3-channel concatenated image. Unfortunately, the model was unable to train well, and only achieved a validation accuracy of 54.97% which is only slightly better than random guess. I think that the reason why the model performed poorly is because the model was unable to learn any distinguishing features. Perhaps for the CNN model, concatenating the images directly was not the best method, but instead, other feature construction should be performed such as the difference between the images.

5. Conclusion and Future Work

In this project, I aimed to answer the question: *Given current moisture levels, can we predict a fire occurring in the near future?* I used publicly available live fuel moisture content (LFMC) maps and ground truth wildfire data from USGS as my input data. The output is a binary classification prediction of whether or not there will be a fire in the near future. I used several different algorithms including SVM, random forest and ResNet-18 for predicting. In this project, I also experimented by concatenating the LFMC maps to include more information over time.

For the first experiment using only images within 15 days of a fire, the Random Forest algorithm performed best, acheiving 70.40% accuracy. On the other hand, using a combination of three images within 3 months of the fire, the Random Forest algorithm was able to perform slightly better with an accuracy of 73.81%. Very few images were predicted to be no fire, when there was actually going to be a fire within 3 months, suggesting that having additional information over time is useful in predicting fire or no fire. SVM and the ResNet-18 didn't perform as well as the Random Forest.

Given the poor results by the CNN, future research can focus on experimenting with a simpler model that would be able to capture non-linearities, but not as complex as photo images would be. Furthermore, maybe using different features would work better as input data, instead of concatenating. Finally, adding more information that would help indicate whether there will be a fire might be helpful. This means including other types of data, such as climate data, or DEM maps.

6. Data and Code Availability

Data and code used to conduct this analysis is publicly available at: `https://www.github.com/NemaOO7/ForestFirePred`

References

[1] 2020 california fires are the worst ever. again.

[2] P. Jain, S. C. Coogan, S. G. Subramanian, M. Crowley, S. Taylor, and M. D. Flannigan. A review of machine learning applications in wildfire science and management. *Environmental Reviews*, 28(4):478–505, 2020.

[3] K. Rao, A. P. Williams, J. F. Flefil, and A. G. Konings. Sar-enhanced mapping of live fuel moisture content. *Remote Sensing of Environment*, 245:111797, 2020.

[4] M. Tonini, M. D'Andrea, G. Biondi, S. Degli Esposti, A. Trucchia, and P. Fiorucci. A machine learning-based approach for wildfire susceptibility mapping. the case study of the liguria region in italy. *Geosciences*, 10(3):105, 2020.

[5] J. L. Welty and M. I. Jeffries. Combined wildfire datasets for the united states and certain territories, 1878-2019, 2020.

[6] G. Zhang, M. Wang, and K. Liu. Forest fire susceptibility modeling using a convolutional neural network for yunnan province of china. *International Journal of Disaster Risk Science*, 10(3):386–403, 2019.