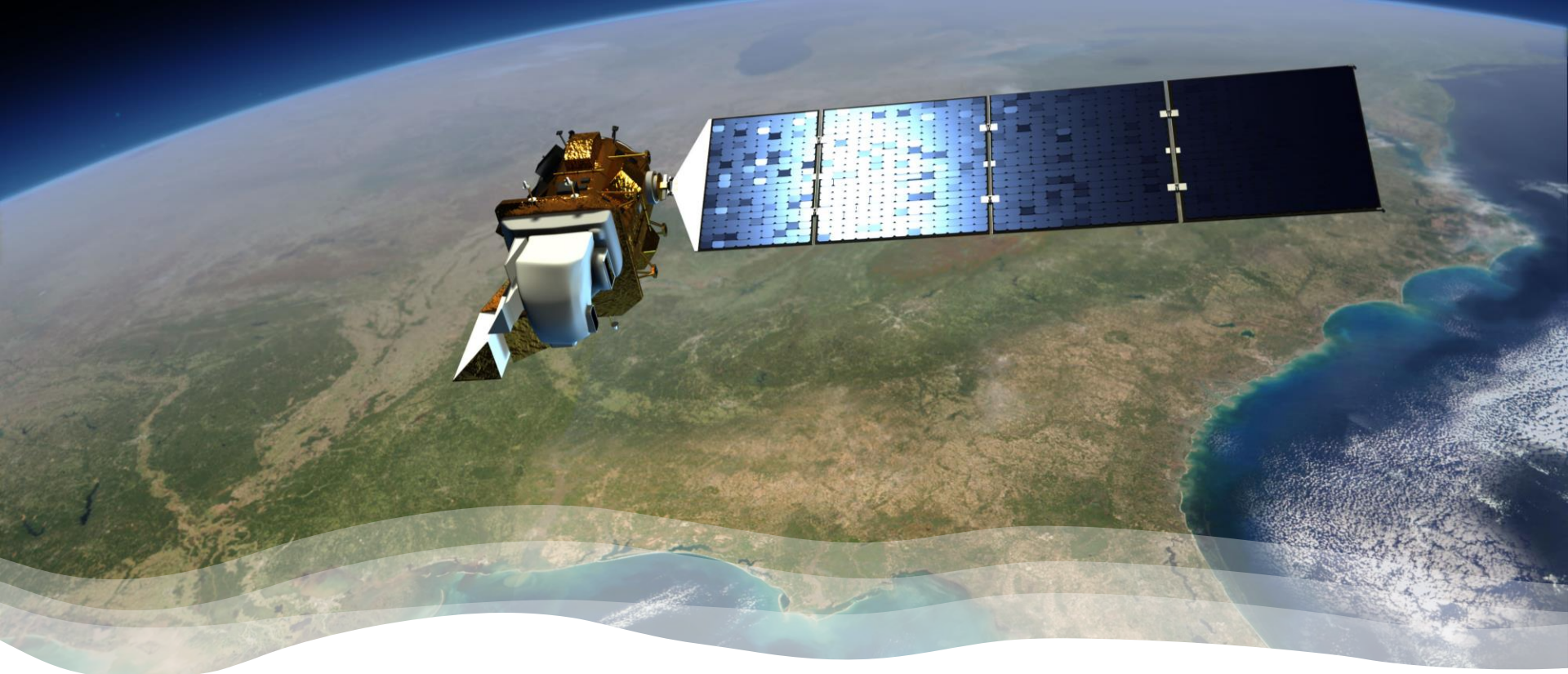


An AI-Supported Wildfire Early Warning System for the Region of Italy Using Open Data

Arbel, Jessica, Opoku, Dong

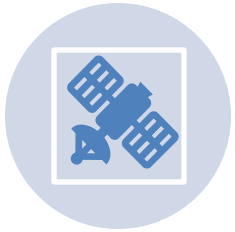
INTRODUCTION

- Build an effective and publicly accessible wildfire prediction system for the region of Italy.
- Open source data and tools

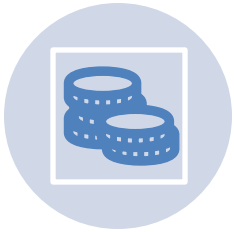
Content

- Limitations of open source data.
- Data cleaning
- Model architecture
- User Dashboard
- Conclusion.

Why Open Source Data?



Access to global satellite data.



Cost effectiveness.

Innovation and Collaboration.

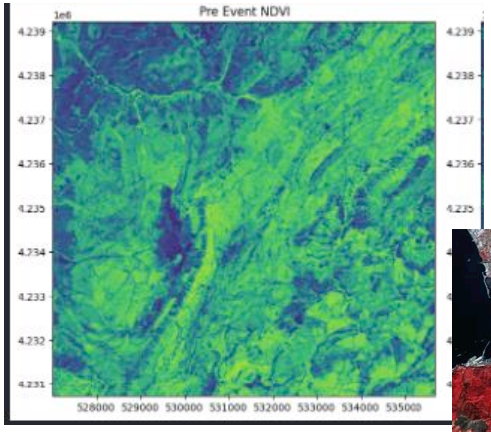
Ethical observation.

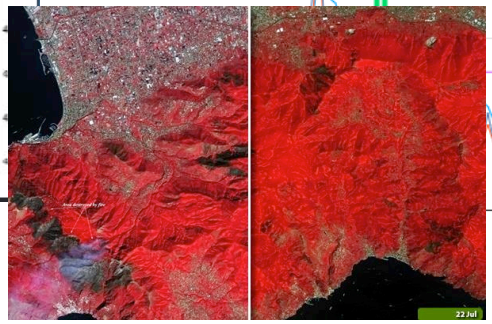
Limitation on Open source data

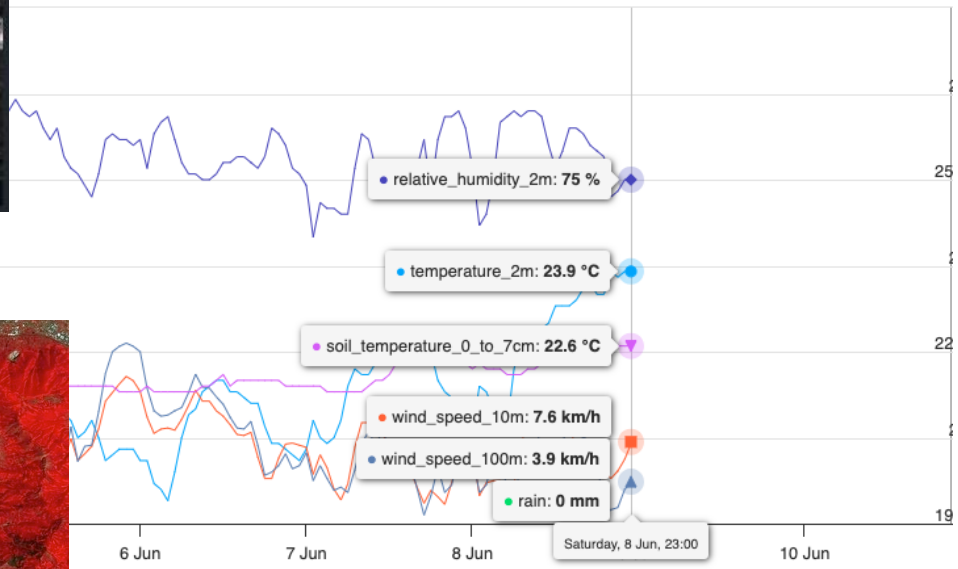
- Delays in data availability affect the relevance of open source data for time-sensitive applications. Eg. Non-geostationary satellites
- Access to certain datasets restricted due to data sovereignty issues, or proprietary rights.
- Optical satellites like Sentinel-2 are affected by cloud cover.

Satellite and weather data

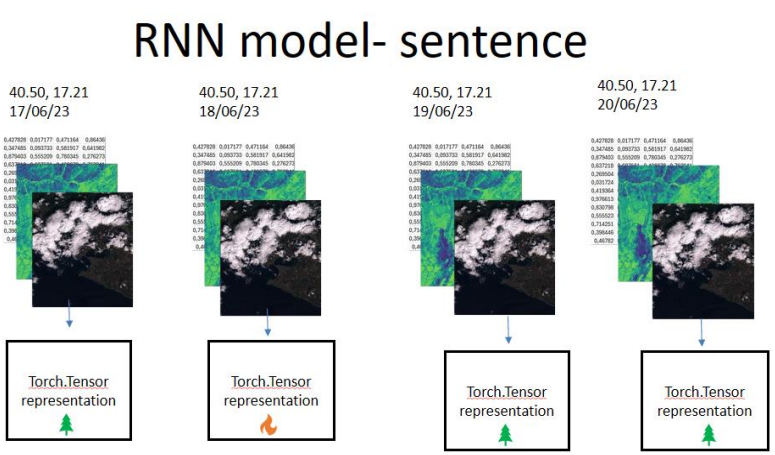
- Sentinel Hub
- Sentinel 2: higher frequency, infrared bands
- Landsat 8: thermal bands, lower resolution and frequency

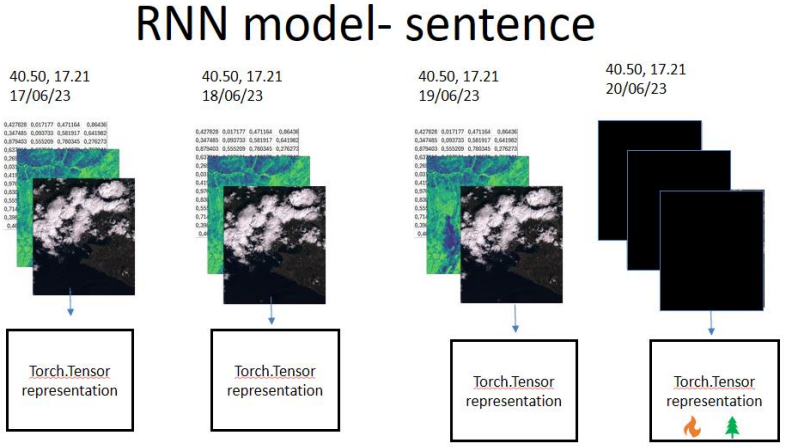






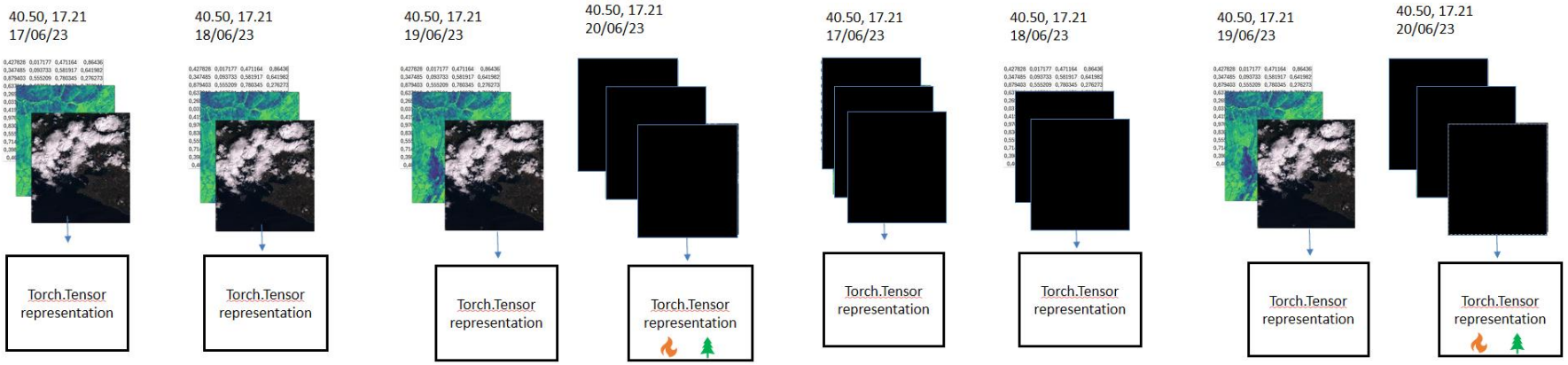
Planning \ Execution

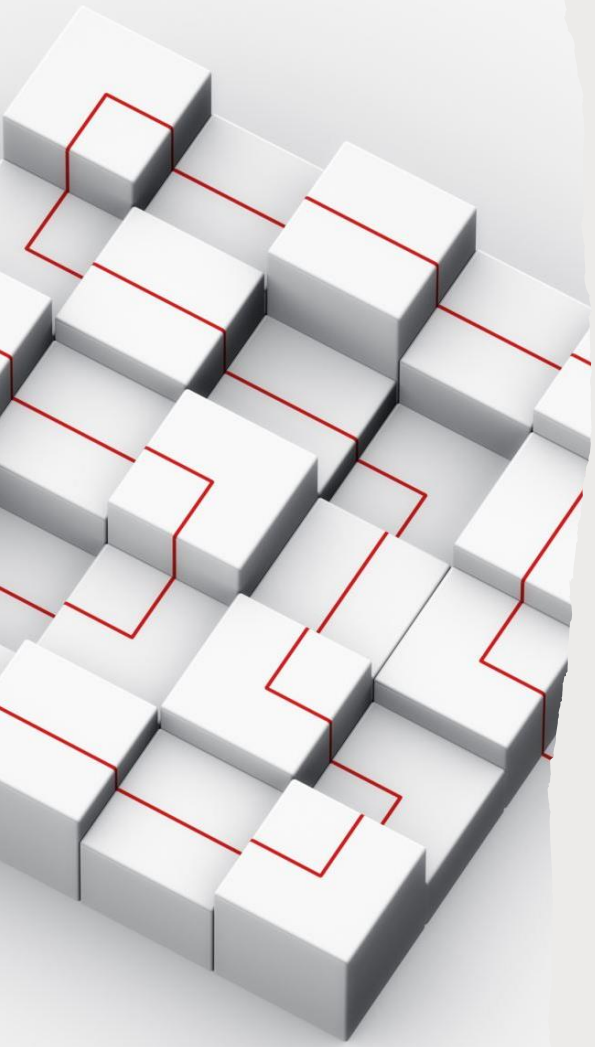




Planning vs harsh realities

- **Frequency of Images**
- **Cloud Cover**
- **Resolution**
- **Size limit**





Model Architecture – RNN

- The model is based on a Recurrent Neural Network (RNN) architecture, which is well-suited for sequential data and time-series analysis. The specific architecture used is Long Short-Term Memory (LSTM), which can capture long-term dependencies and patterns in the data.
- Layers:
  - ➢ Input Layer: Processes satellite imagery features.
  - ➢ LSTM Layers: Capture temporal dependencies in the data.
  - ➢ Dense Layer: Produces the final prediction (fire or no fire).
  - ➢ Attention Layer: Improves model focus by assigning weights to inputs, enhancing its ability to prioritize relevant information dynamically.

Inputs:

Inputs(Images)
9 days *(190*190)*13 layers

Inputs(weather)
9 days *(24*9)*15 layers

A single-layer RNN doesn't work well on complex multi-inputs

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 9, 190, 190, 13)]	0	[]
input_2 (InputLayer)	[(None, 9, 24, 14)]	0	[]
time_distributed (TimeDistributed)	(None, 9, 469300)	0	['input_1[0][0]']
reshape (Reshape)	(None, 9, 336)	0	['input_2[0][0]']
concatenate (Concatenate)	(None, 9, 469636)	0	['time_distributed[0][0]', 'reshape[0][0]']
masking (Masking)	(None, 9, 469636)	0	['concatenate[0][0]']
lstm (LSTM)	(None, 9, 50)	93937400	['masking[0][0]']
time_distributed_1 (TimeDistributed)	(None, 9, 1)	51	['lstm[0][0]']

Total params: 93,937,451
Trainable params: 93,937,451
Non-trainable params: 0

Single-layer RNNs typically only process data step by step with a time sequence, but not flexibly skip from different areas and capture long-range dependencies.

Thus limits its ability to merge two different types of data and make judgments

Why do we need multilayers?

· Enhancing the modeling ability of long-term dependencies.

· Increasing the expressive power and flexibility of the network

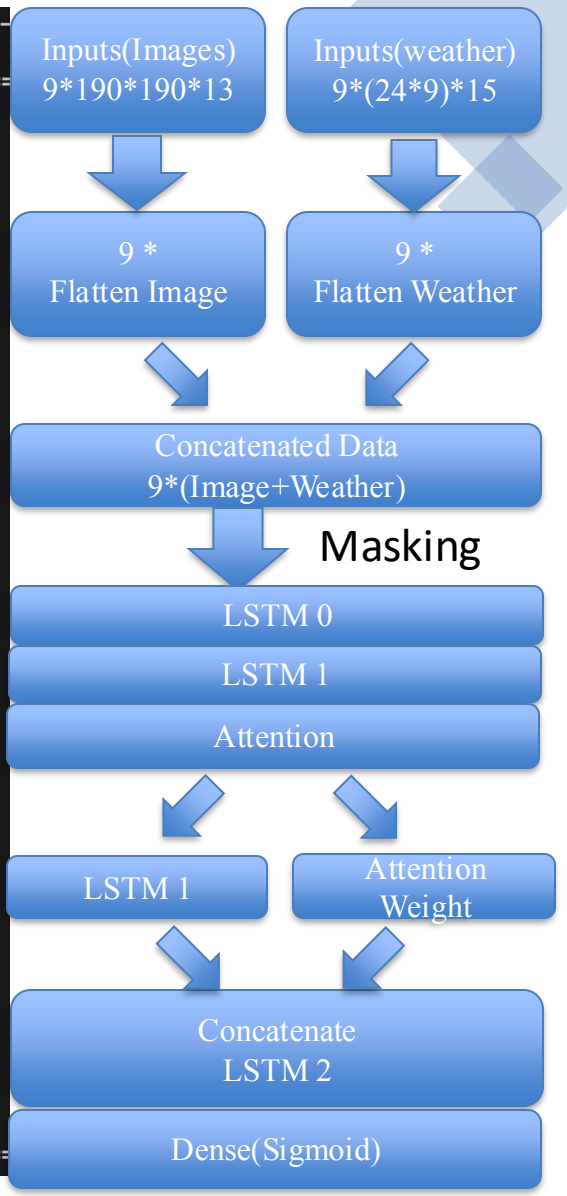
· Reducing gradient vanishing

· Attention layer can caculate the attention weight the model should pay to the current layer of output

0.47481677 0.47481677 0.9898213 0.47481677 0.47481677 0.9899491
0.9848173 0.47481677 0.9898213 0.47481677 0.47481677 0.47481677
0.47481677 0.47481677 0.47481677 0.98275995 0.9899491 0.47481677
0.47481677 0.47481677 0.47481677 0.47481677 0.47481677 0.47481677
0.47481677 0.9899491 0.47481677 0.47481677 0.47481677 0.47481677
0.9899491 0.47481677 0.47481677 0.47481677 0.47481677 0.47481677
0.47481677 0.47481677 0.9899491 0.98419976 0.98594517 0.47481677
0.47481677 0.47481677 0.47481677 0.47481677 0.9899491 0.47481677
0.47481677 0.47481677 0.47481677 0.47481677 0.47481677 0.98900247
0.9898213 0.47481677 0.98202735 0.47481677 0.47481677 0.47481677
0.47481677 0.9899491 0.47481677 0.47481677 0.47481677 0.47481677
0.47481677 0.47481677 0.47481677 0.9898213 0.47481677 0.9898044

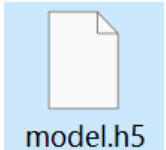
Multi layer attention RNN model

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 9, 190, 190, 13)]	0	[]
input_2 (InputLayer)	[(None, 9, 216, 15)]	0	[]
time_distributed (TimeDistributed)	(None, 9, 469300)	0	['input_1[0][0]']
reshape (Reshape)	(None, 9, 3240)	0	['input_2[0][0]']
concatenate (Concatenate)	(None, 9, 472540)	0	['time_distributed[0][0]', 'reshape[0][0]']
masking (Masking)	(None, 9, 472540)	0	['concatenate[0][0]']
lstm (LSTM)	(None, 9, 64)	120986880	['masking[0][0]']
lstm_1 (LSTM)	(None, 9, 64)	33024	['lstm[0][0]']
attention (Attention)	(None, 9, 64)	0	['lstm_1[0][0]', 'lstm_1[0][0]']
concatenate_1 (Concatenate)	(None, 9, 128)	0	['lstm_1[0][0]', 'attention[0][0]']
lstm_2 (LSTM)	(None, 64)	49408	['concatenate_1[0][0]']
dense (Dense)	(None, 1)	65	['lstm_2[0][0]']



Loss: Binary Crossentropy
Optimizer: Adam

```
1.0 VS 0 - Confidence: 0.406412 - Incorrect
0.0 VS 0 - Confidence: 0.210905 - Correct
0.0 VS 0 - Confidence: 0.206028 - Correct
0.0 VS 0 - Confidence: 0.209675 - Correct
1.0 VS 0 - Confidence: 0.423029 - Incorrect
0.0 VS 0 - Confidence: 0.205348 - Correct
1.0 VS 1 - Confidence: 0.647565 - Correct
0.0 VS 0 - Confidence: 0.229892 - Correct
0.0 VS 0 - Confidence: 0.307512 - Correct
0.0 VS 0 - Confidence: 0.224251 - Correct
1.0 VS 1 - Confidence: 0.613681 - Correct
0.0 VS 0 - Confidence: 0.332739 - Correct
0.0 VS 0 - Confidence: 0.356789 - Correct
0.0 VS 0 - Confidence: 0.358995 - Correct
0.0 VS 0 - Confidence: 0.312713 - Correct
1.0 VS 1 - Confidence: 0.632123 - Correct
1.0 VS 1 - Confidence: 0.612689 - Correct
0.0 VS 1 - Confidence: 0.582100 - Incorrect
1.0 VS 1 - Confidence: 0.505562 - Correct
0.0 VS 0 - Confidence: 0.394192 - Correct
0.0 VS 0 - Confidence: 0.331007 - Correct
0.0 VS 0 - Confidence: 0.339272 - Correct
0.0 VS 0 - Confidence: 0.205306 - Correct
0.0 VS 0 - Confidence: 0.270130 - Correct
1.0 VS 1 - Confidence: 0.644446 - Correct
1.0 VS 0 - Confidence: 0.421683 - Incorrect
0.0 VS 0 - Confidence: 0.489672 - Correct
0.0 VS 0 - Confidence: 0.322816 - Correct
1.0 VS 0 - Confidence: 0.428817 - Incorrect
1.0 VS 0 - Confidence: 0.404810 - Incorrect
0.0 VS 1 - Confidence: 0.509554 - Incorrect
0.0 VS 0 - Confidence: 0.216643 - Correct
0.0 VS 0 - Confidence: 0.264156 - Correct
0.0 VS 0 - Confidence: 0.399758 - Correct
0.0 VS 0 - Confidence: 0.491737 - Correct
accuracy_score:
0.7575757575757576
recall_score:
0.5909090909090909
precision_score:
0.65
f1_score:
0.6190476190476191
confusion_matrix:
[[37  7]
 [ 9 13]]
```



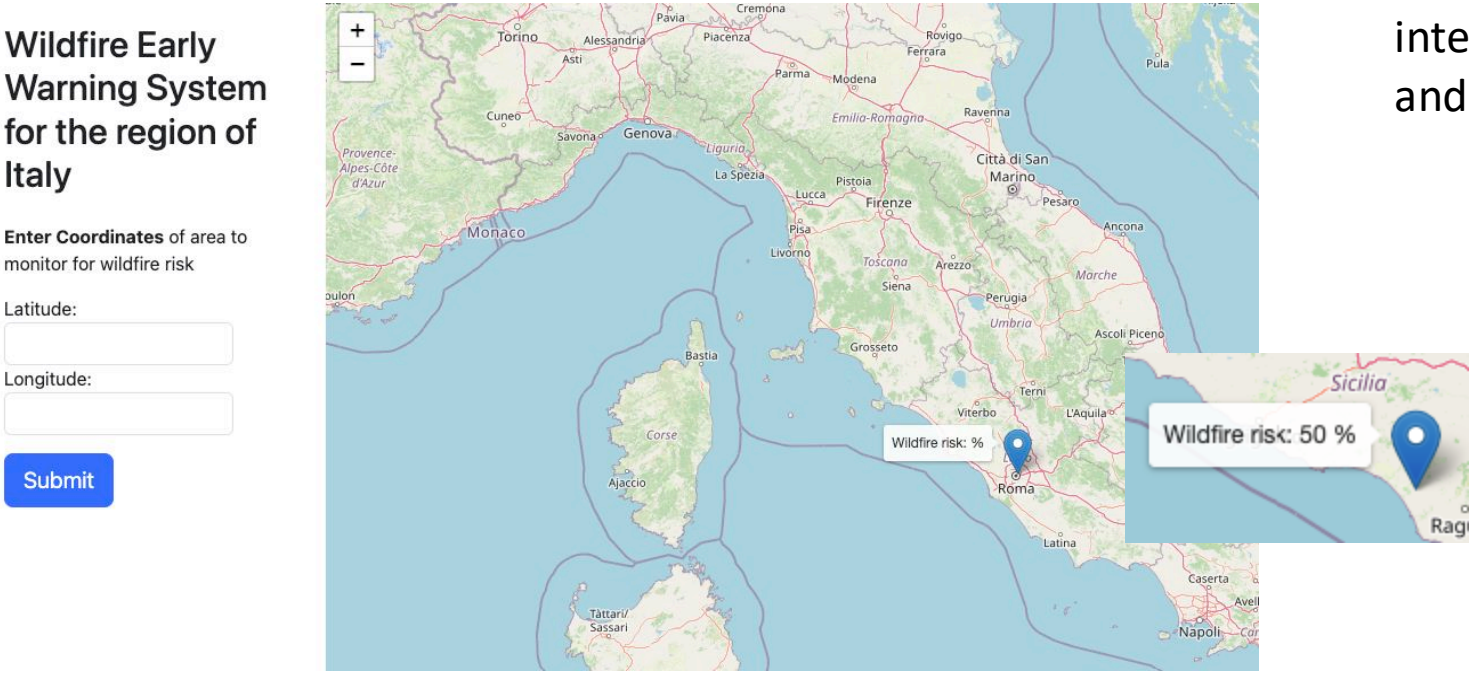
(1.35GB)

Model Performance Metrics:

- **Accuracy Score**: 0.76
  - The model correctly predicts 76% of the total instances.
- **Recall Score**: 0.59
  - This indicates that the model correctly identifies 59% of the actual positive cases.
- **Precision Score**: 0.65
  - This shows that 65% of the cases the model predicted as positive are actually positive.
- **F1 Score**: 0.62
  - The F1 score is the harmonic mean of precision and recall, reflecting a balance between the two metrics.

Dashboard

- User friendly interface and feedback



ReactJS + Leaflet + OSM
Flask API

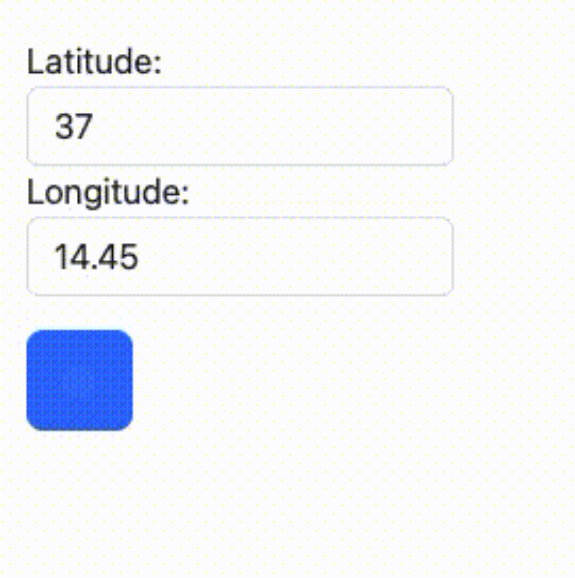
OpenStreetMap

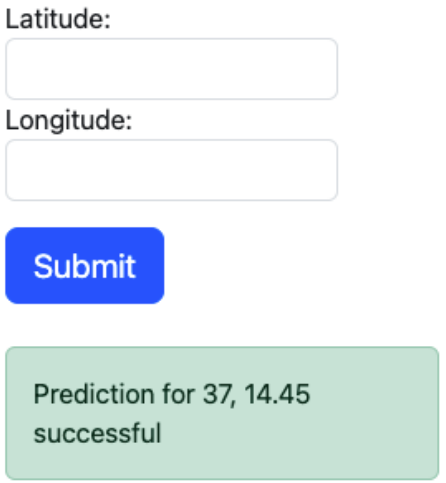
Flask

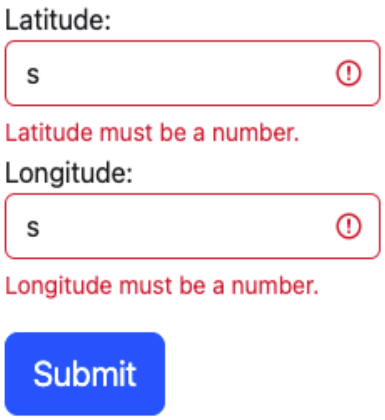
Leaflet

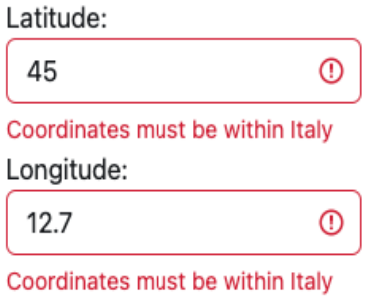
User Feedback

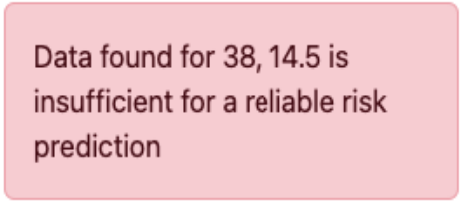
- Loading spinner
- Error and Success messages
- Input validation



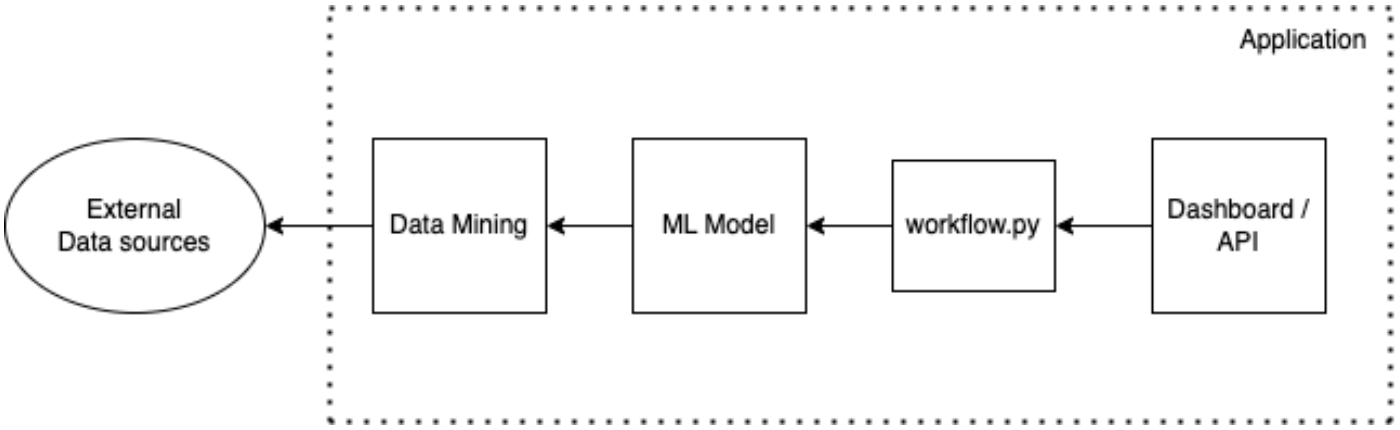








Pipeline



Latitude:

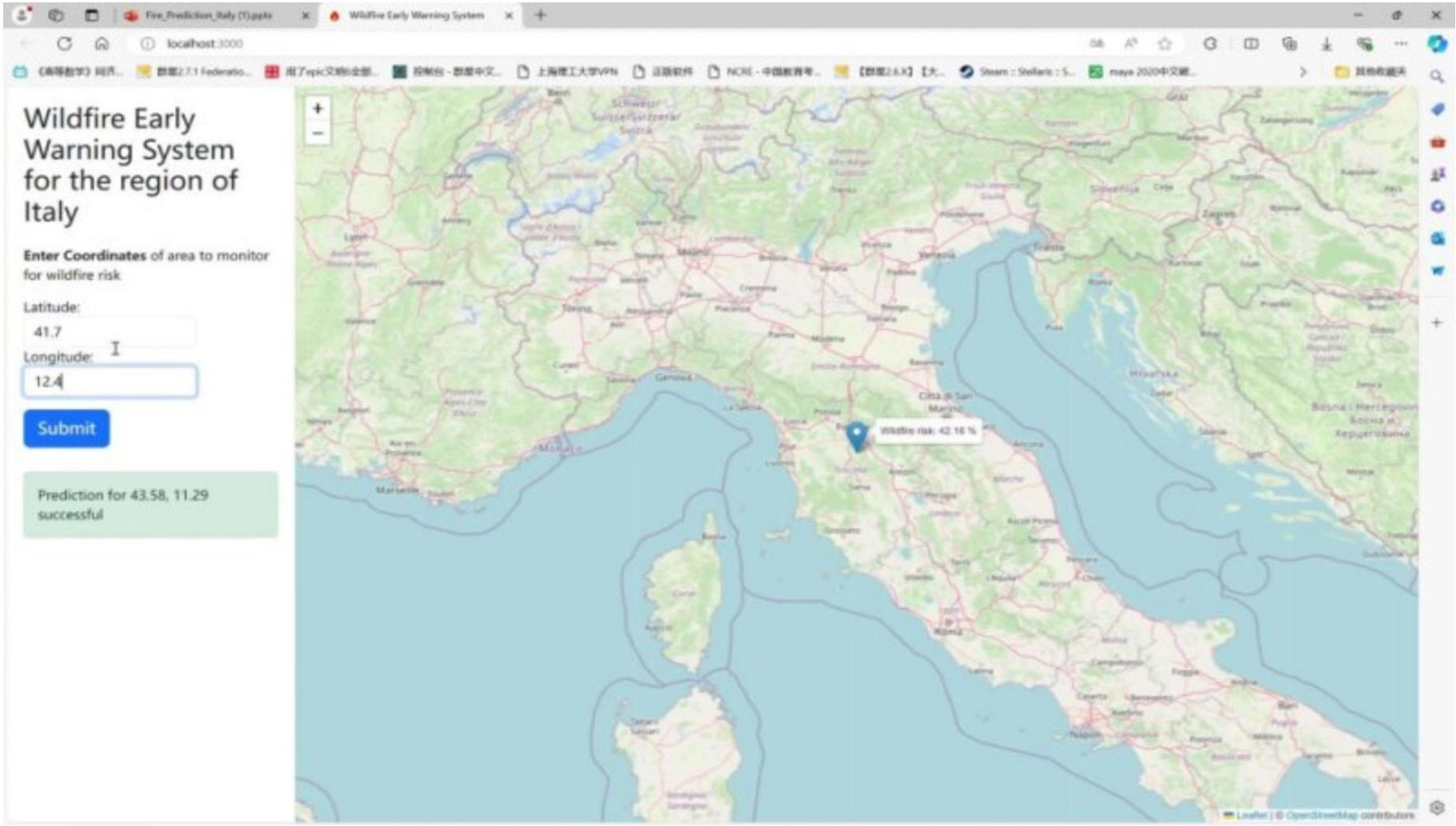
Longitude:

Submit

POST

```
response = {
    'latitude': lat,
    'longitude': lon,
    'confidence': round(prediction,2)
}
return response
```

Demo



https://github.com/R-bellH/ki_Geo_project

Conclusion

- Unique challenges of working with open source data and tools
- Availability of free and open resources / community support
- Solving real world interdisciplinary problems - creativity and experimentation
- Focus on a **prototype/proof of concept** which can be further developed and eventually used

In recent years, the number of forest fires has increased worldwide due to various factors. Experts warn that the number of forest fires will continue to increase in the coming years, mainly due to climate change. Early warning systems that use up-to-date satellite images to detect forest fires before they can spread further would be very helpful. Get an overview of the technical possibilities and the remote sensing data available for this purpose. Implement a prototype.