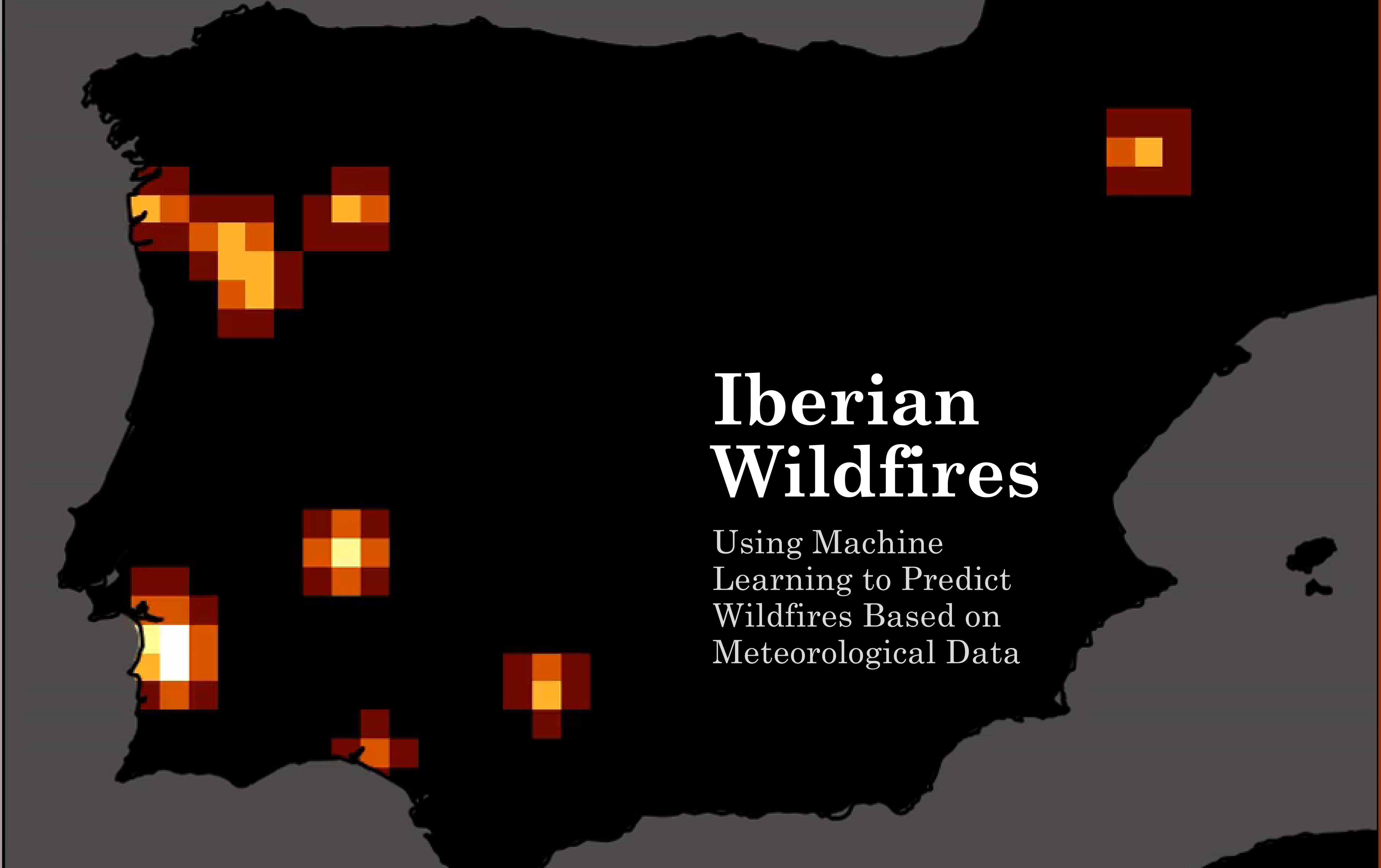


Iberian Wildfires

Using Machine Learning to Predict Wildfires Based on Meteorological Data

Motivation and Goal

‘Climate change is playing an increasing role in determining wildfire regimes alongside human activity (medium confidence), with future climate variability expected to enhance the risk and severity of wildfires in many biomes such as tropical rainforests (high confidence)’ – IPCC, 2019 [1]

Main Goal:

Predict wildfires across the Iberian Peninsula from meteorological data.

Data:

Fire data (NASA VIIRS) [3]
https://firms.modaps.eosdis.nasa.gov/

Weather data (ERA5 reanalysis) [2]
https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels

FORBES > BUSINESS

BREAKING

New York Goes Martian: Wildfire Smoke Engulfs City In Eerie Orange Haze

Ty Roush Forbes Staff
I cover breaking news.

Follow

Jun 7, 2023, 03:21pm EDT

Updated Jun 7, 2023, 04:12pm EDT

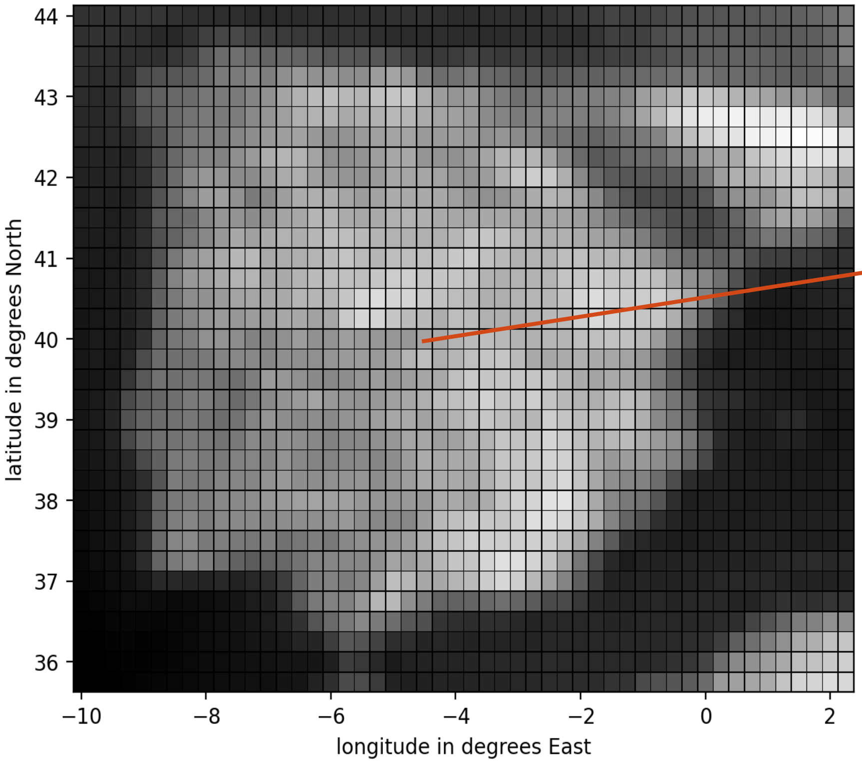
TOPLINE The New York City skyline was swept into an orange haze Wednesday, as air quality in the city—which canceled all outdoor activities for local schools—fell to “unhealthy” levels caused by smoke from hundreds of Canadian wildfires.

Smoky haze covers New York City on Wednesday. COPYRIGHT 2023 THE ASSOCIATED PRESS. ALL RIGHTS RESERVED

[4]

Data Structure

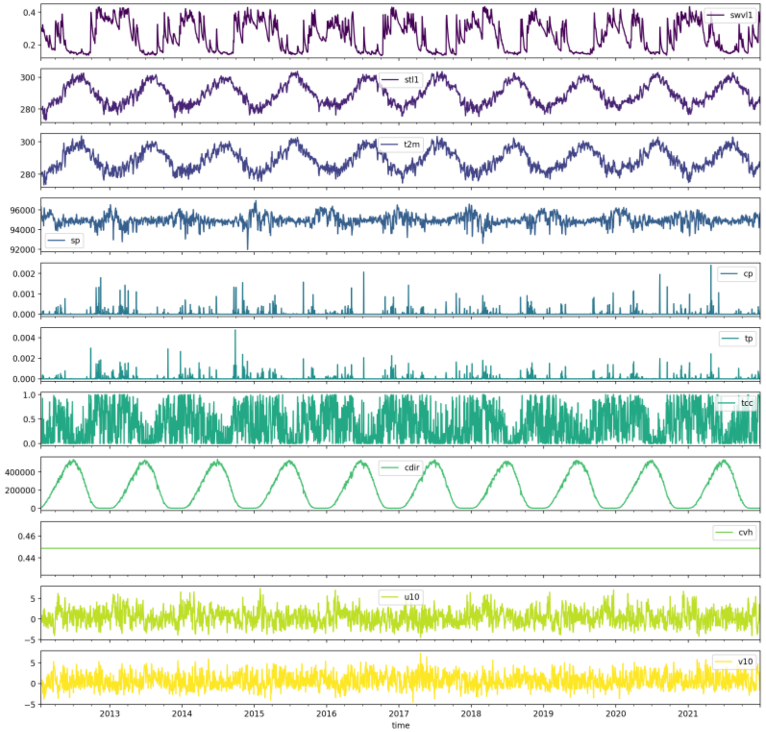
0.25° x 0.25° boxes over Iberia



33 x 55 = 1815 boxes

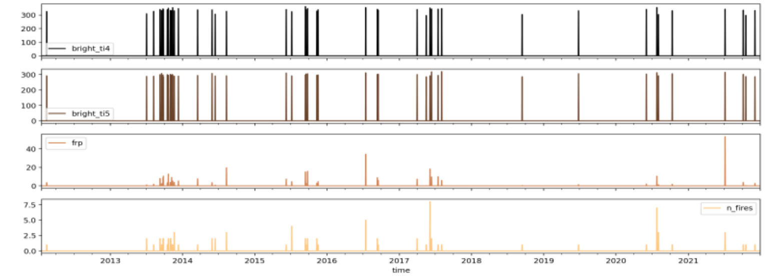
Timeseries with daily averages from 2012-01-20 to 2021-12-31 → 3643 days

Meteorological data: ERA5 reanalysis



11 variables

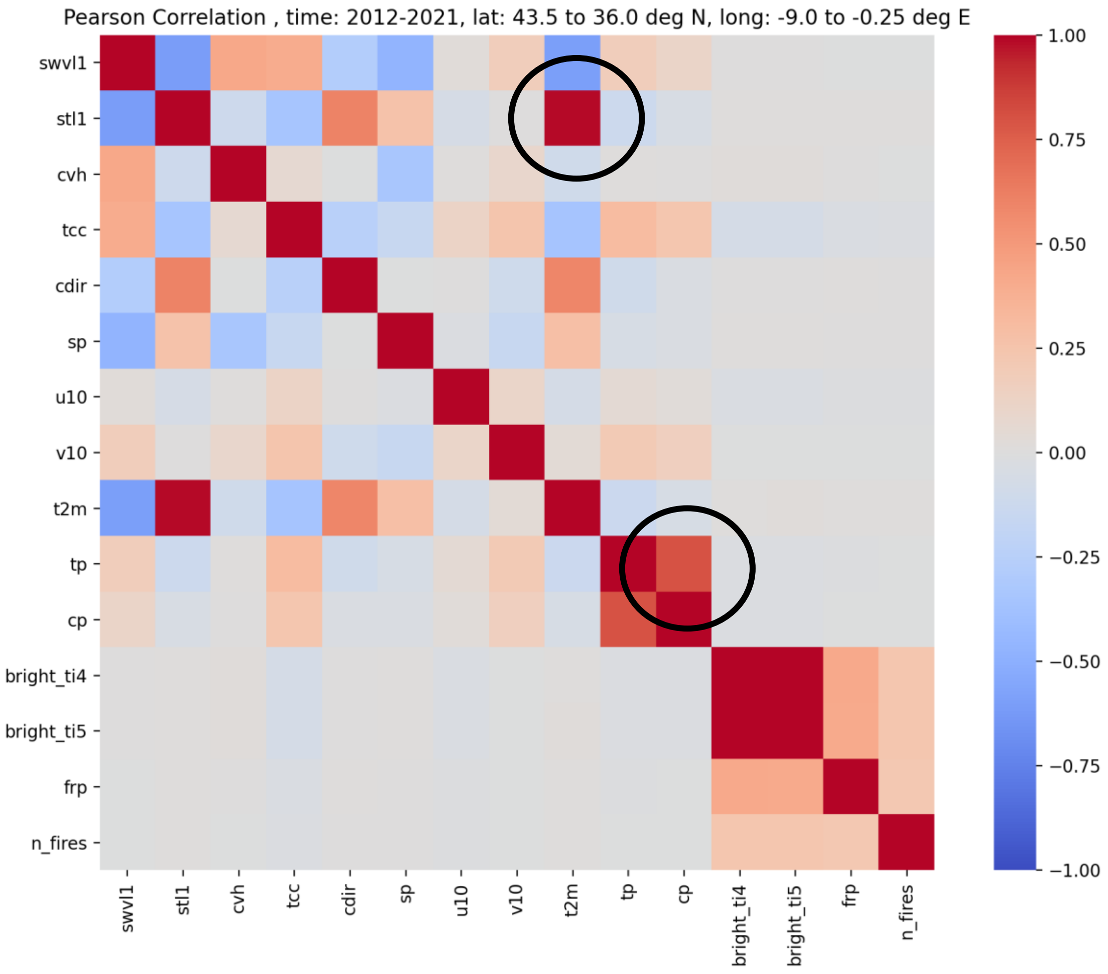
Fire data: VIIRS Nasa



4 variables

Data Correlations

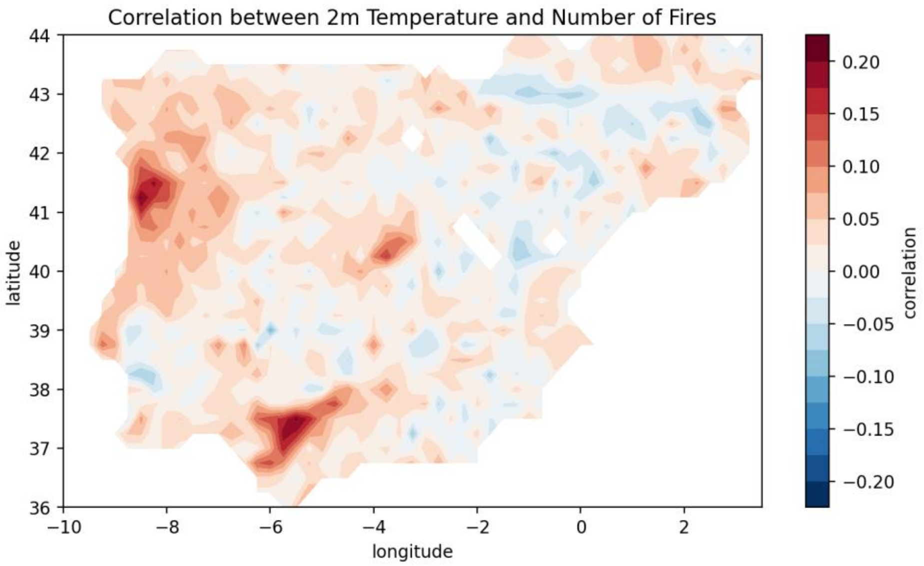
Correlations over the whole area and time



- High correlation (>0.8) between the temperature variables (t2m and stl1) and the precipitation variables (cp and tp)

- Low correlation between fire data and meteorological data for the whole area

  … but there is hope when looking at correlations in single boxes:

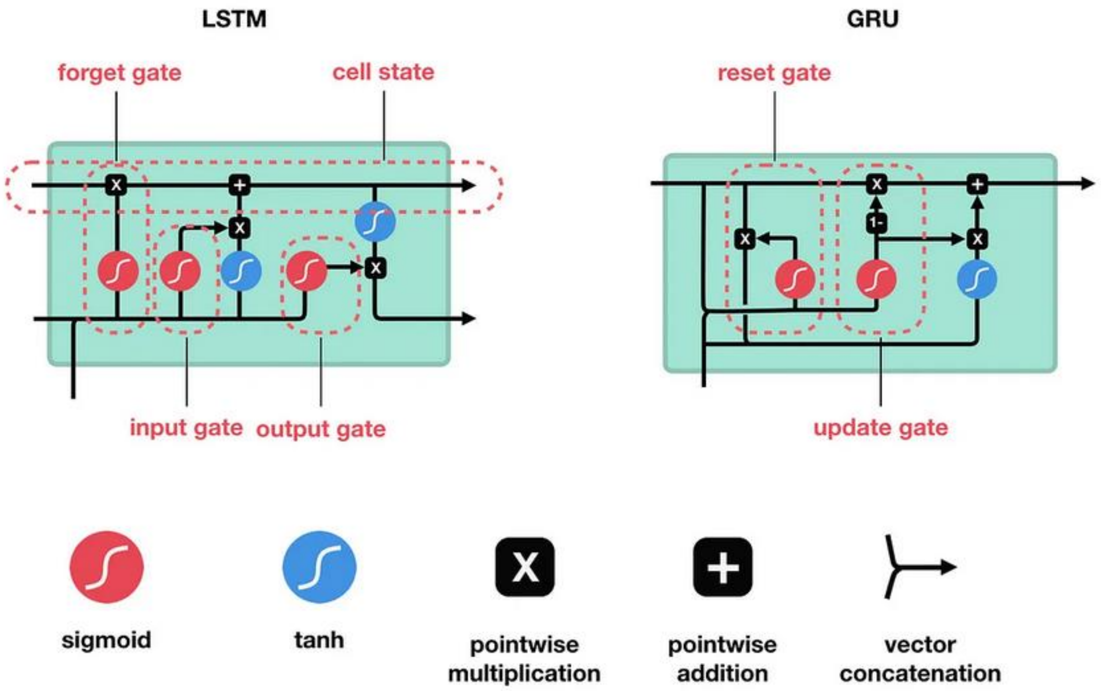


Models
RNN with LSTM and GRU

Recurrent Neural Networks (RNNs) are very useful for time series, as their cells both take the prediction of the previous time step and the input of the current time step into account to generate the output [5].

They have a short-term memory problem, 'forgetting' the first time steps due to vanishing gradients.

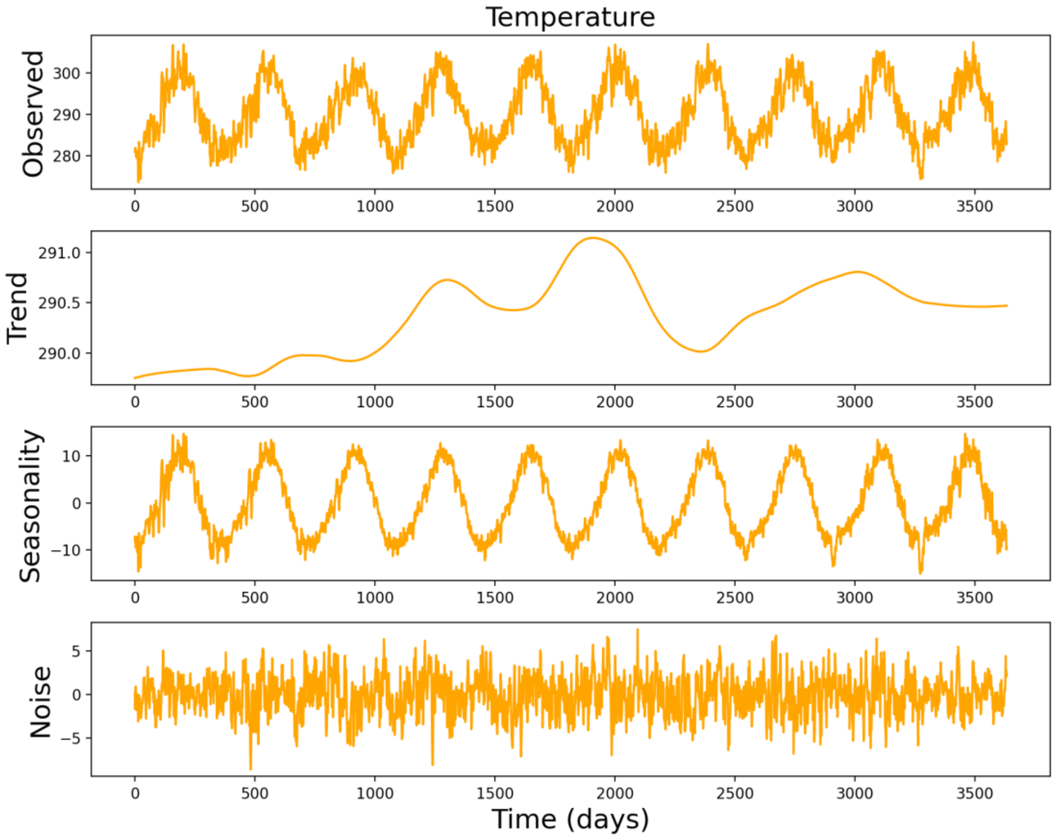
LSTM and GRU cells are two approaches to solve this problem by adding (long-term) memory cells.



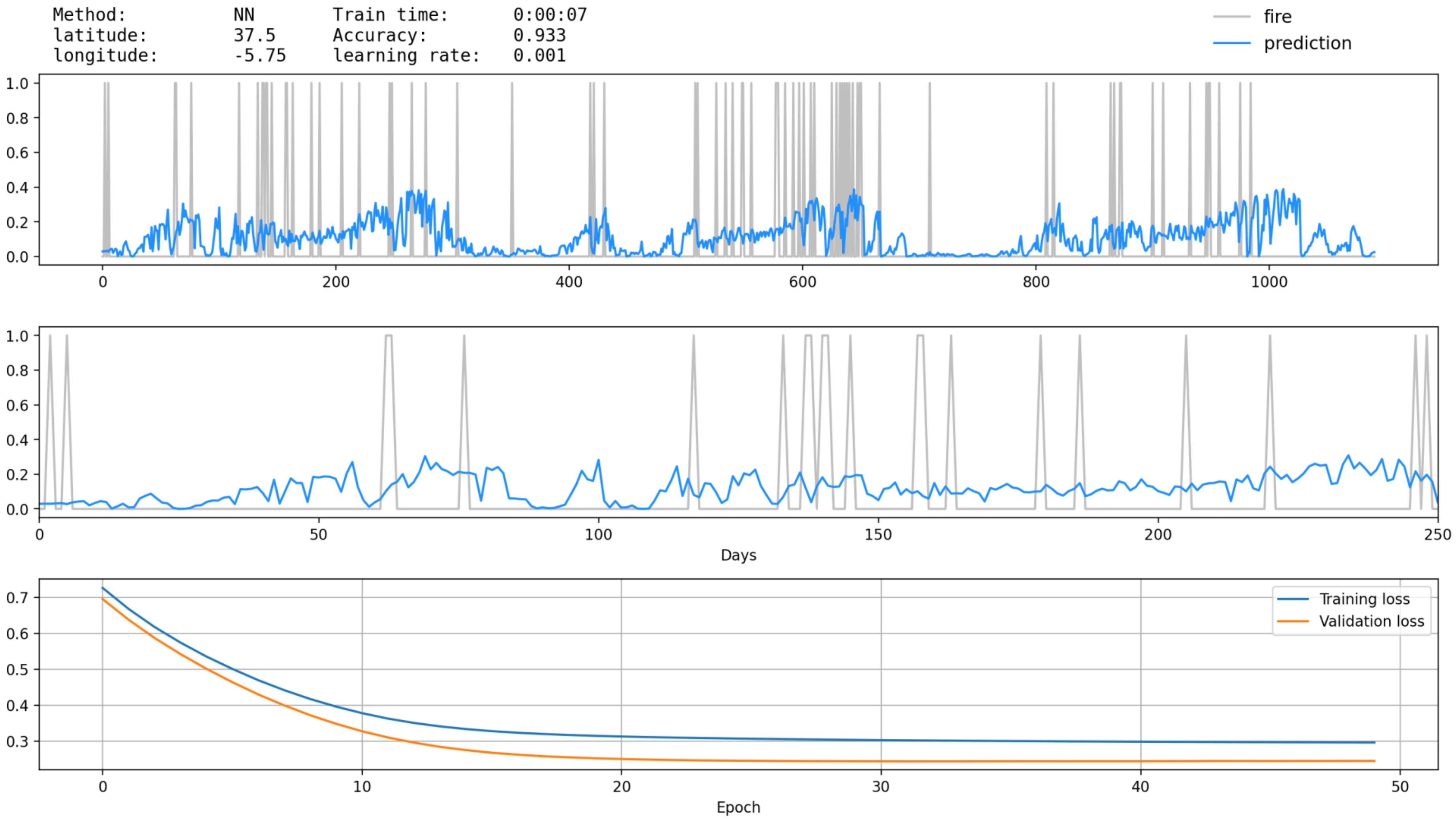
[6]

Trend and Seasonality

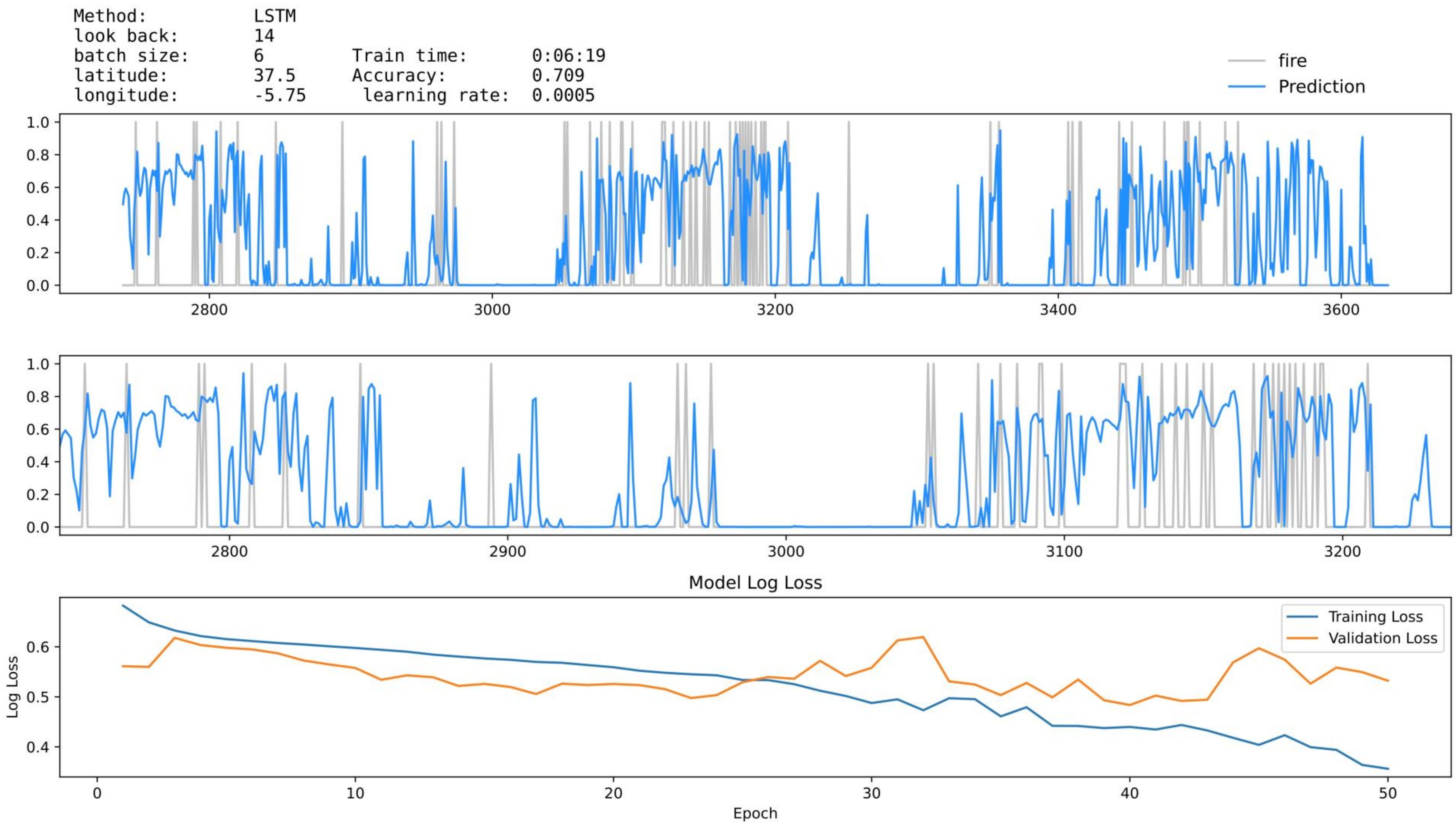
- Most of our features and the target are s easonal, with a period of approximately a year.
- Although LSTM and GRU have long memory, they may not have it long enough to learn the seasonality in the data.
- Therefore, it is sometimes better to extract noise and seasonality from the data, train the model with the trend, and add again the seasonality to the predictions.



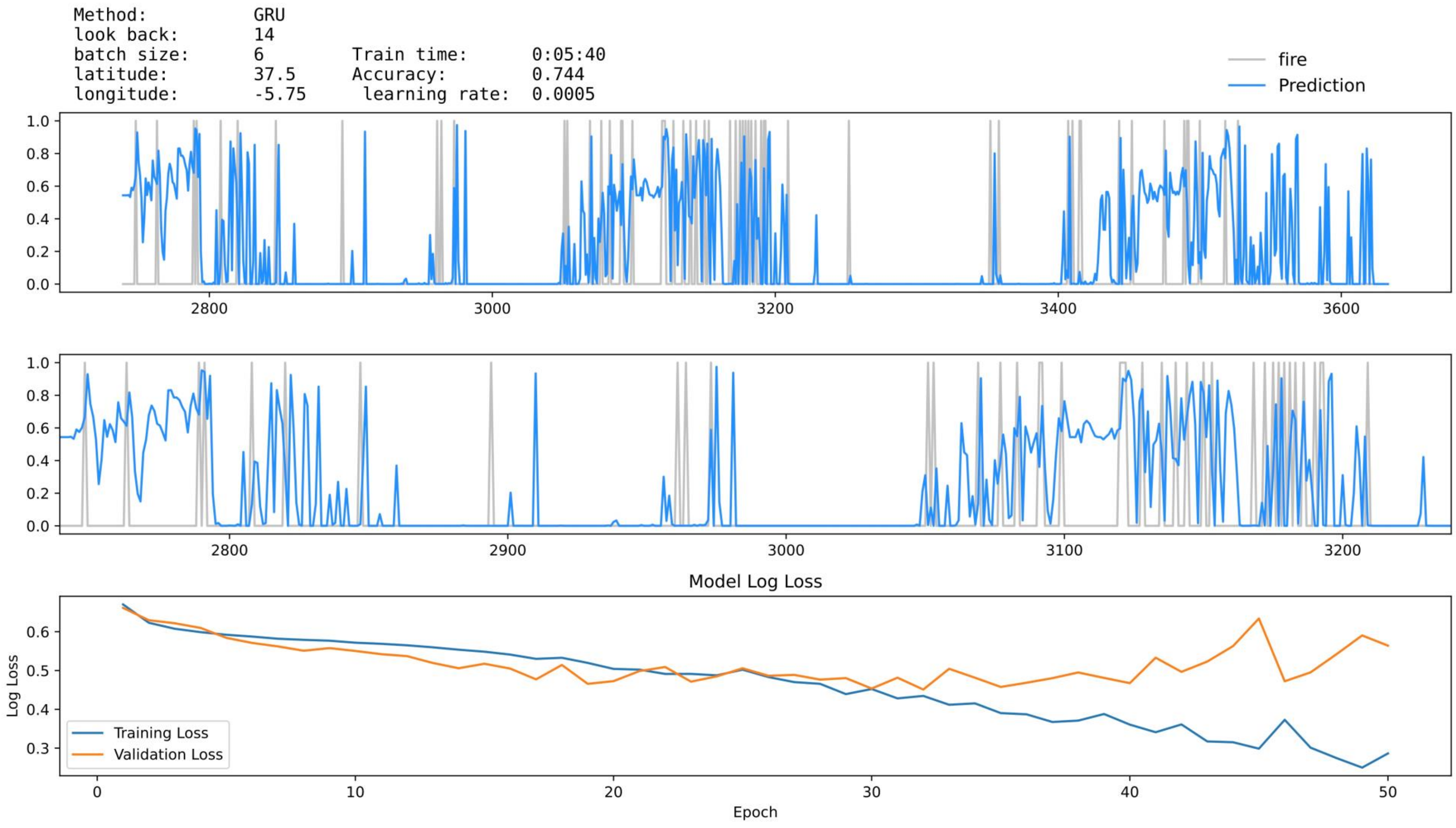
Neural Network



LSTM

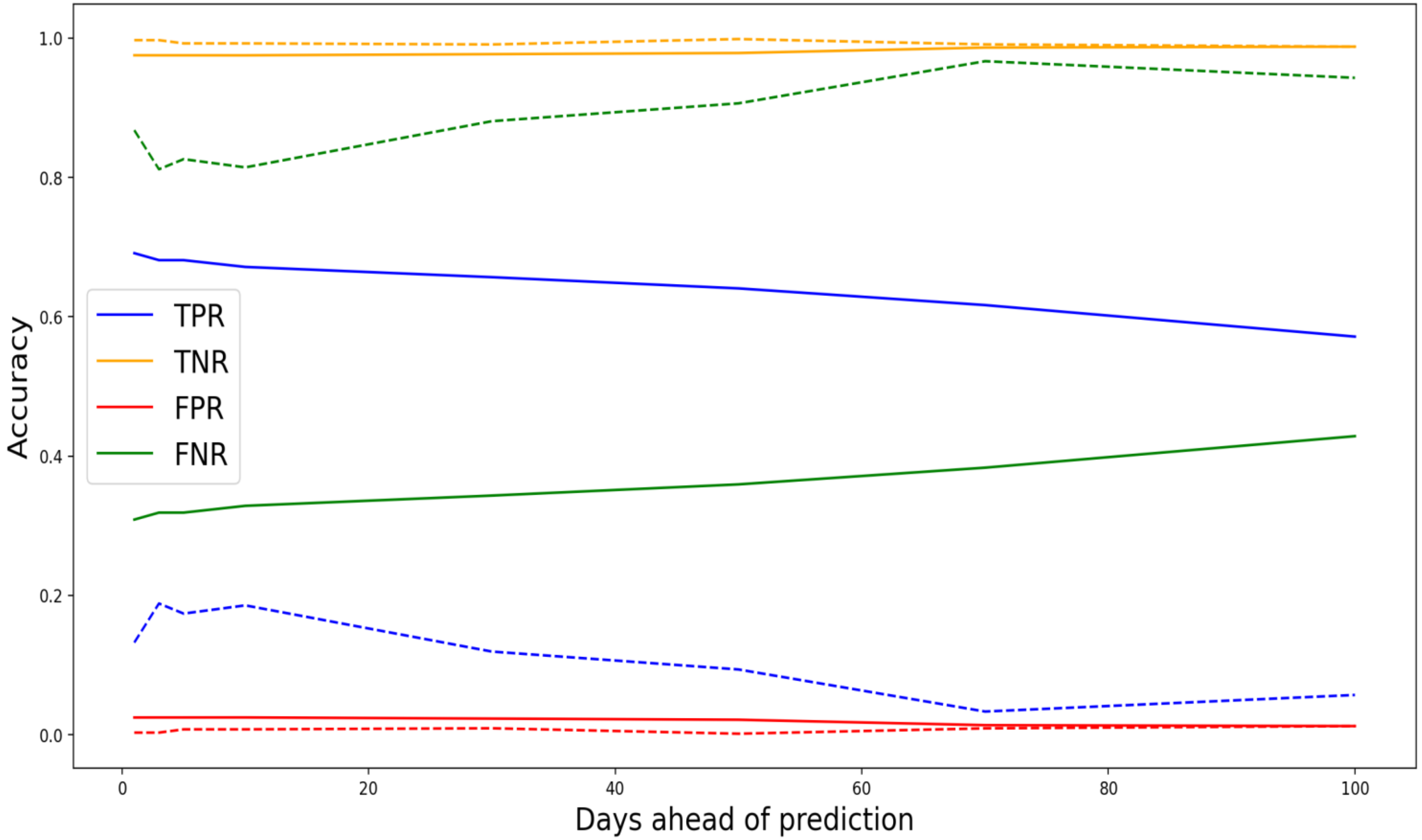

Method: LSTM
look back: 14
batch size: 6
latitude: 37.5
longitude: -5.75
Train time: 0:06:19
Accuracy: 0.709
learning rate: 0.0005
fire
Prediction
Model Log Loss
Training Loss
Validation Loss
Log Loss
Epoch

GRU



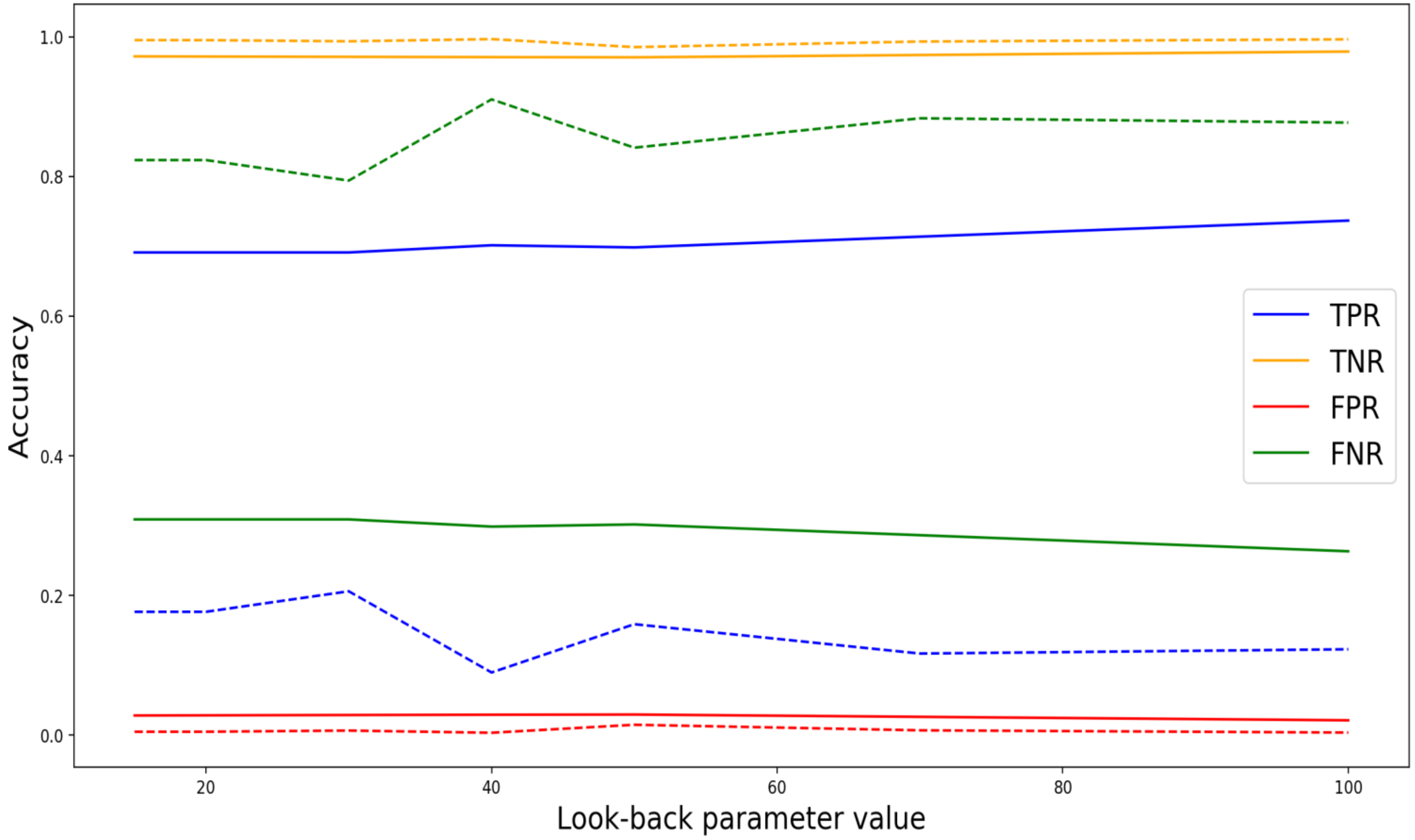
Method: GRU
look back: 14
batch size: 6
latitude: 37.5
longitude: -5.75
Train time: 0:05:40
Accuracy: 0.744
learning rate: 0.0005
fire
Prediction
Model Log Loss
Log Loss
Epoch
Training Loss
Validation Loss

Seasonality correction



- Extracting the seasonality and training the model with just the trend (Thick lines) shows in a significantly improved results compared to training the model without extracting seasonality (dashed lines)
- Results of the model trained with just the trend are better no matter how many days ahead we want to predict.

The look-back parameter



- The more days we look back to make the prediction, the more beneficial it is to train the model with just the trend

- In fact, the best choice to train the model is to choose 100 days of look-back and extract the seasonality. (We could not add more days of look-back because of computation time)

Conclusions

‘What did we learn?‘

- RNN‘s might be a powerful tool for predicting timeseries when the hyperparameter are optimised in the right way
- We assume that we would have needed more fires in order to make good predictions
- Organisation of the models and well structured code is key

‘What else would have been cool to investigate?’

- Using information of neighboring cells (e.g. whether there’s a fire or not and/or taking the wind direction into account) to improve the fire prediction
- Trying an algorithm that combines LSTM and some spatial correlation weights, such as a GNN. We tried ConvLSTM that combines CNN and LSTM but it did not fit our purpose.
- It would be very interesting to know what is the optimal combination of look-back parameter and how many days ahead to train the model for.

Bibliography

[1] Jia et al., 2019: Land–climate interactions. In: Climate Change and Land: an IPCC special report on climate change, desertification, land degradation, sustainable land management, food security, and greenhouse gas fluxes in terrestrial ecosystems

Meteorological Data

[2] Hersbach, H., Bell, B., Berrisford, P., Biavati, G., Horányi, A., Muñoz Sabater, J., Nicolas, J., Peubey, C., Radu, R., Rozum, I., Schepers, D., Simmons, A., Soci, C., Dee, D., Thépaut, J-N. (2023): ERA5 hourly data on single levels from 1940 to present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS), DOI: 10.24381/cds.adbb2d47 (Accessed on 26-05-2023)

Fire Data

[3] https://firms.modaps.eosdis.nasa.gov/ (Accessed 12-06-2023)

[4] https://www.forbes.com/sites/tylerroush/2023/06/07/images-of-new-york-city-engulfed-by-canadian-wildfire-smoke/?sh=ec9841036412 (Accessed 12-06-2023)

[5] Géron, Aurélien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems*. Second edition. Beijing [China] ; Sebastopol, CA: O'Reilly Media, Inc, ISBN 978-1-4920-3264-9, 2019.

[6] https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21 (Accessed 13-06-2023)

Statement

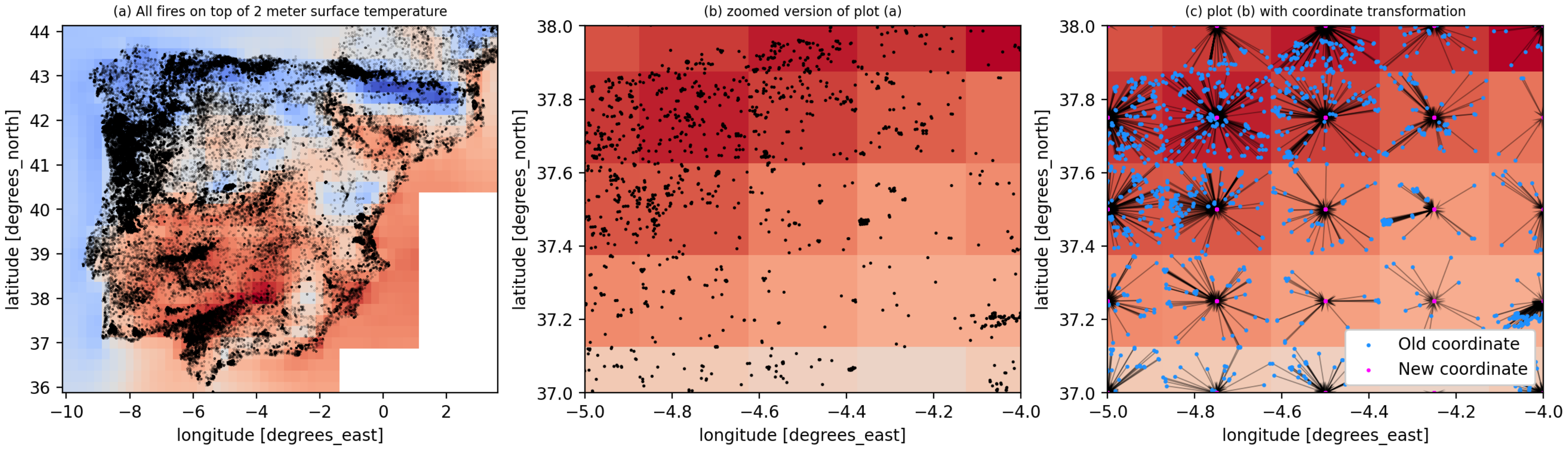
All participants contributed evenly

Appendix

The following slides describe some of our decision making regarding variable choice, data preproccessing and model choice in more detail.

Details – Fire Data Preprocessing

The fire data contains lat, lon coordinates for each fire event with an accuracy of ±375 meters. This is a lot smaller than our ERA5 datagrid. To collect the two data sets in a common format, we have done the following: coordinate transform fire events to the center of the ERA5 grid cell to which they belong. Then we save new variables to the ERA5 dataset, the most important of which is the number of fires in each grid cell (for each day). From this we later make a new binary variable stating for a given day if a given cell had at least one fire. Plotted below (left) are all fires in the 10-year period on top of a random day's surface temperature. The middle plot is the same but zoomed in, such that you can see the individual ERA5 grid cells, and the right plot shows the coordinate transformation for each fire event (blue) to their corresponding ERA5 coordinate (magenta). Code: aggregate_data.py



Details – Fire Data Variables

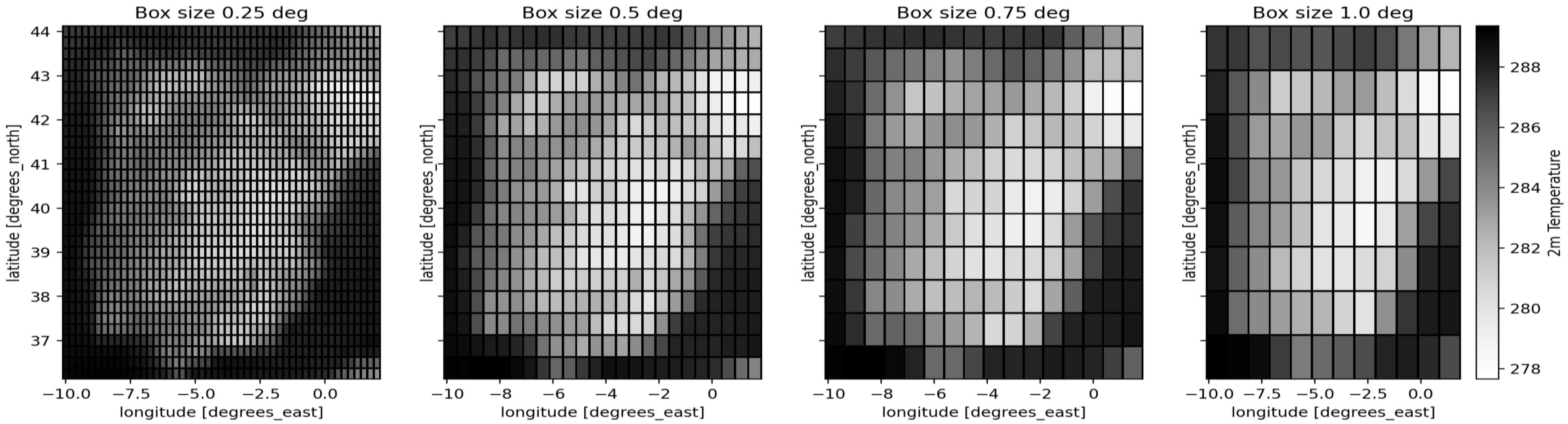
Variables of the Fire Dataset	Description
bright_ti4	• I-4 Channel brightness temperature of the fire pixel [K]
bright_ti5	• I-5 Channel brightness temperature of the fire pixel [K]
frp	• pixel-integrated fire radiative power [MW]
n_fires	• Number of fires in each grid cell of the era5 data

We considered to use these variables of the fire dataset for the models. All variables are averaged daily and spatially for the era5 gridcells.
In the end we decided to make classification models and ended up just using the number of fires as described on the previous slide.

Details – Met Data Preprocessing

We downloaded hourly data as netCDF files from [2]. We made daily averages and combined datasets, since we had to download several datasets for different variables because otherwise they became too big to download. We spatially averaged the data for different box sizes but decided to use the smallest box size. In a future project one could think about using bigger box sizes, since there would be more fires per box to train the model. On the other hand one might loose important details of the data when averaging spatially over bigger areas.

Code: met_data_preprocessing.py



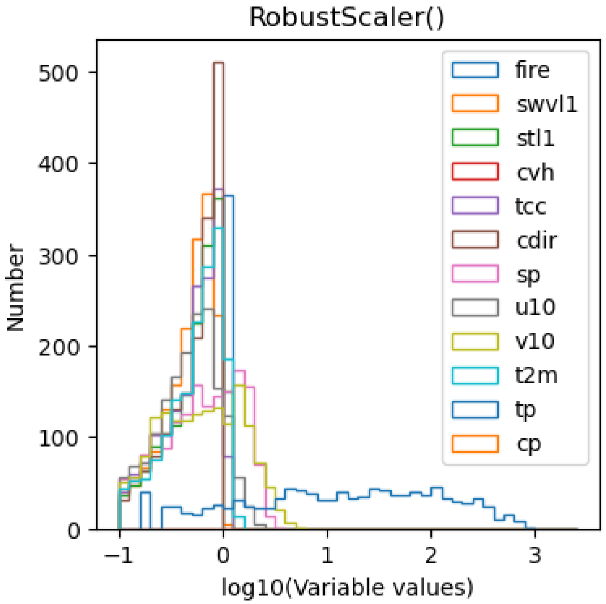
Details – Meteorological Variables

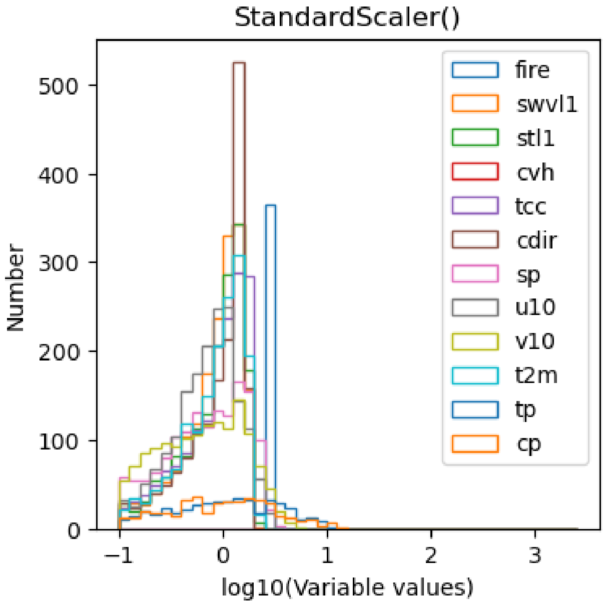
We decided to download the following variables of the meteorological dataset for the fire predictions.

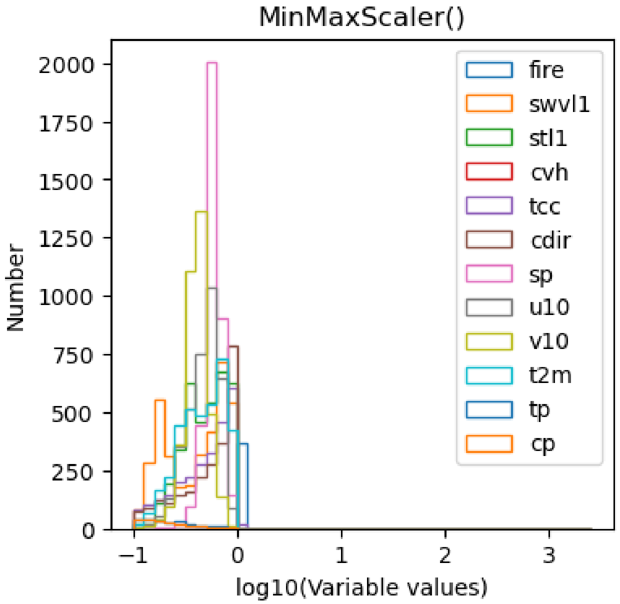
Variables that affect wildfires	Data we used [2]
Temperature	• 2m Temperature [K]
Solar radiation	• Direct Solar Radiation [J/m^2] • Total Cloud Cover [%]
Wind speed and direction	• U Component of the Wind [m/s] • V Component of the Wind [m/s]
Atmospheric stability	• Convective Precipitation [m] • Surface Pressure [Pa]
Precipitation	• Total Precipitation [m]
Fuel availability	• High Vegetation Cover [%]

Details – Data Preprocessing

Several Scaler from sklearn were tested on the data in order to generate similar data distribution in the same ranges.







The MinMaxScaler() was used for the RNN GRU Classification algorithm.

Details – Feature Importance

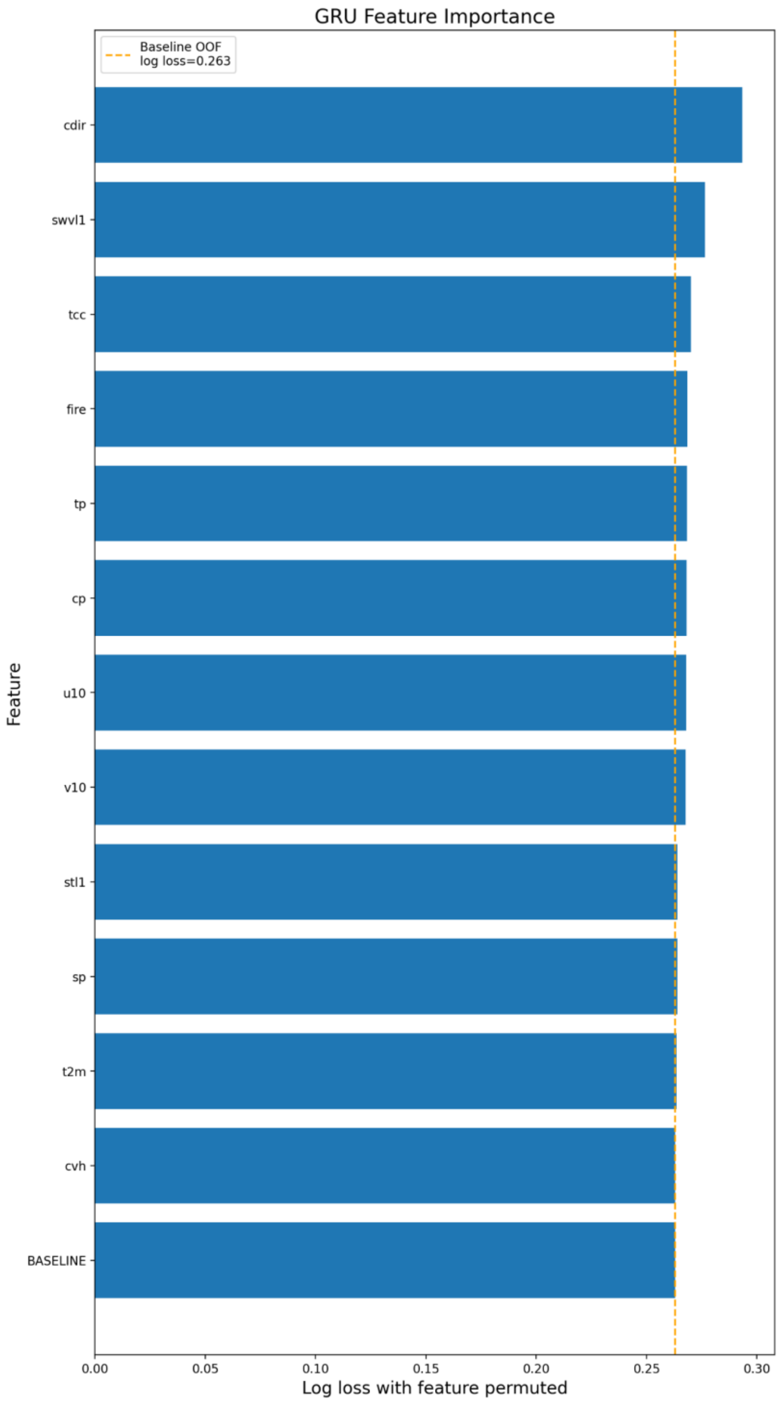
We took a look at the Feature Importances for the RNN Classification model using GRU.
The values of single features were randomnly permuted and the logloss calculated.

Results:

- The log loss did not change a lot (max change of ~ 0.02)
- The variables that increase the loss the most when randomnly permutted are:
    - cdir (Clear-sky direct solar radiation at surface)
    - swvl1 (Volumetric soil water)
    - tcc (total cloud cover)
- The temperature (t2m) and the high vegetation cover (cvh) didn't seem to be too important for this classification model

In the end we used all meteorological variables for the models.

Code: RNN_GRU_and_Feature_Importance.py



Details – Hyperparameter Optimization

- HP optimization on the NN, LSTM, and GRU did not yield significant improvements.

Details – Training on more than one box

- Training on many cells (tried up to 64) did not make any significant improvement to the NN model.
- We find enough time to try this with the RNN (LSTM, GRU) because it was more complicated to code, as the time dependency of the RNN prevents us from simply concatenating data from different cells.