1 Requirement

Forests play the main role in maintaining the ecological balance of Earth. Unfortunately, such balance has been disrupted by the frequent forest fire occurring everywhere (Chowdary et al., 2018). It is urgent to detect and monitor the forest fire. By understanding the potential causes and studying the forest's response and expansion patterns, we can not only prevent future disasters but also assist in promoting forest regrowth.

To detect the fire, first, the project needs information about the temporal changes of the forest. Then, an efficient and effective way of detecting fire is essential, with official true fire boundaries for validation. After that, the easily understandable visualization of fire detection should be presented. All steps must be deployed in a consistent efficient and independently executed code structure, with the ability to handle different areas, times, and regions easily and swiftly.

2 Design

2.1 Data

Table 1: Overview of Data

Type	**Description**	**Spatial resolution**	**Temporal resolution**	**Source**
National Land Cover Database (NLCD)	U.S. land cover with a 16-class legend based on a modified Anderson Level II classification system	30m within US	Synthesis of 2020	U.S. Geological Survey (USGS)
MOD09AQ1 Remote Sensing Product	MODIS Terra satellite product capturing surface reflectance in seven spectral bands. Used for vegetation and land monitoring.	250m with global coverage	8-day	NASA
Official Fire Boundary	Shapefile of study area	-	-	National Interagency Fire Center

It is possible to access a comprehensive set of open data including remote sensing data, land cover data, and true fire boundaries provided by institutions from the United States. Therefore, at this moment this project supports the study within the region of the United States.

As Table 1 shows, the NLCD data was used to locate the lands with forests. Remote sensing time series data handle a sequence of images collected over an interval time, decipher their inner pattern, and forecast the future situation. The focus of their temporal changes aligns with the analysis of forest fire detection and monitoring (Shumway and Stoffer, 2017). With change point detection algorithms and sufficient support of remote sensing images, it is efficient and economical to detect whether the fire is happening as well as its expansion. Hence, MODIS MOD09Q1 data were used for the detection with a moderate 250m resolution (in theory, any MODIS data is compatible with the project). The Near-infrared and red bands were used to calculate the Normalized Difference Vegetation Index (NDVI). The high correlation between NDVI and the leaf areas of the plants could effectively reflect the patterns of plants' changes (Huang et al., 2014). Therefore, the NDVI was used to represent the forest in the study. Official fire boundaries were used for the validation of detection results.

2.2 Method

Many technologies have been applied to forest fire detection. Popular methods include sensor-based techniques, camera-based techniques, air-borne techniques, satellite-based techniques, etc. Some of them have reached a high degree of accuracy. However, the cost is relatively high and it is expensive to perform the monitoring over a given period. BEAST (Bayesian Estimator of Abrupt change, Seasonality, and Trend) is a fast, generic averaging algorithm based on the Bayesian model. BEAST is implemented in Python and the package is provided. It is the most suitable choice to detect fire for the project, which will focus on detecting both whether and when the fire happened.

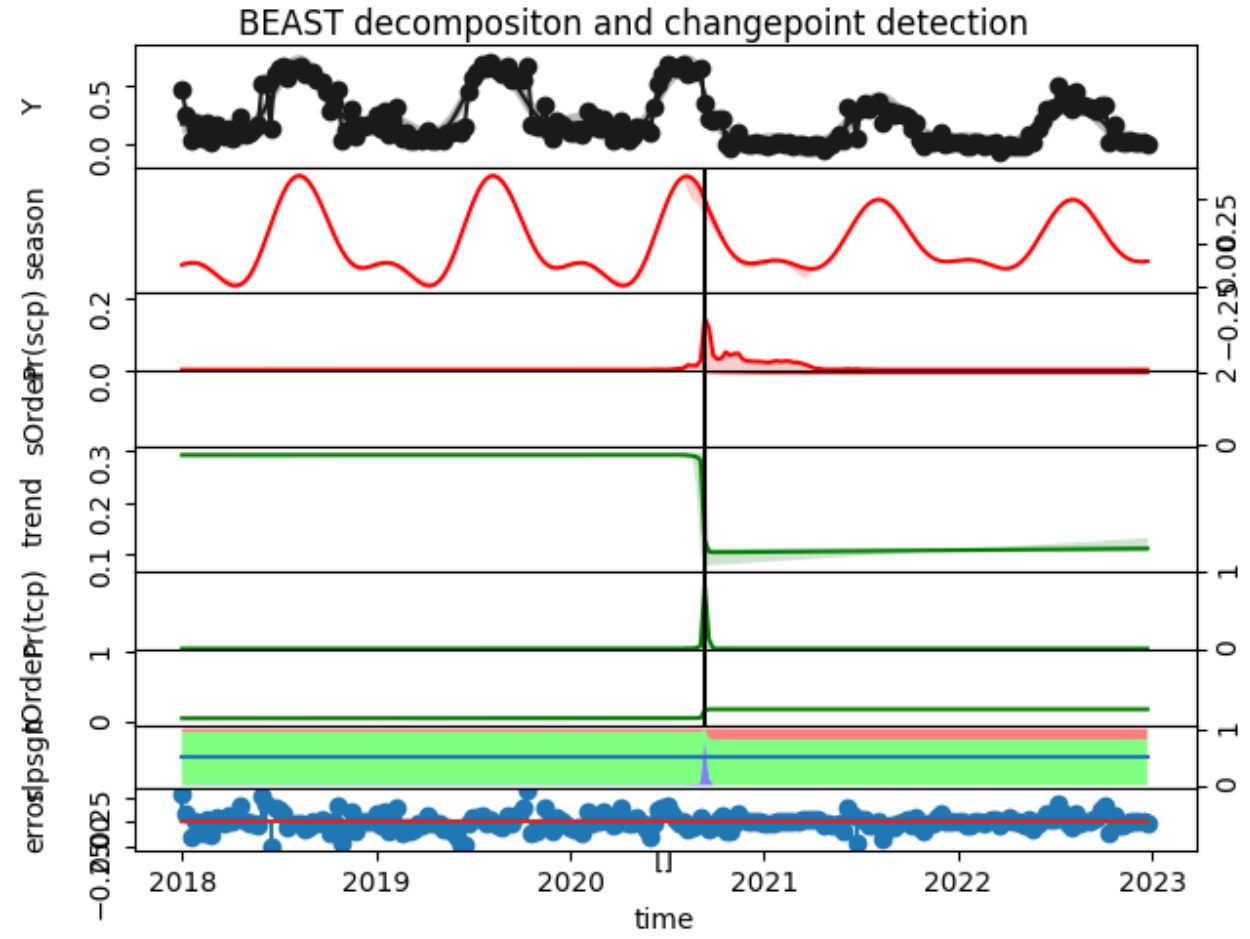

Figure 1: Overview of Method

As figure 1 shows, BEAST breaks down time series into trend, seasonality, and noise first, and then detects the change point based on the trend (Zhao et al., 2019). Afterward, a probability curve for the change point will be generated, and the point with a value over the designated threshold will be regarded as the change point.

2.3 Project Structure

The project will consist of three parts:

- Data download and pre-process part: The project aims to connect to Google Earth Engine (GEE). It is a cloud-based platform designed to support the storage, processing, and analysis of large-scale geospatial data. This part can fetch the data, clip and mask the MODIS data in Google Earth Engine, and download it to the local files.
- Fire detection part: The project first re-structures the standard MODIS data into a time-series list, and then detects the pixel of fire based on the generated time-series list with BEAST algorithm and finally visualizes the results. The detection result will be a list of pixels with each one's time series and a mark of fire happening or not (if happening, also recording the date). The result will be passed to the validation part.

- Validation part: The project validates the detection result with the true fire boundaries and shows the users the related statistics (confusion matrix, accuracy, precision, recall, F1 score, and Kappa index).

Besides, the project needs to consider an independent file to set all parameters ahead flexibly to avoid any hard-coded vars. Version control and a logbook should also be arranged for project history records and misbehavior prevention.

2.4 Test Area and Period

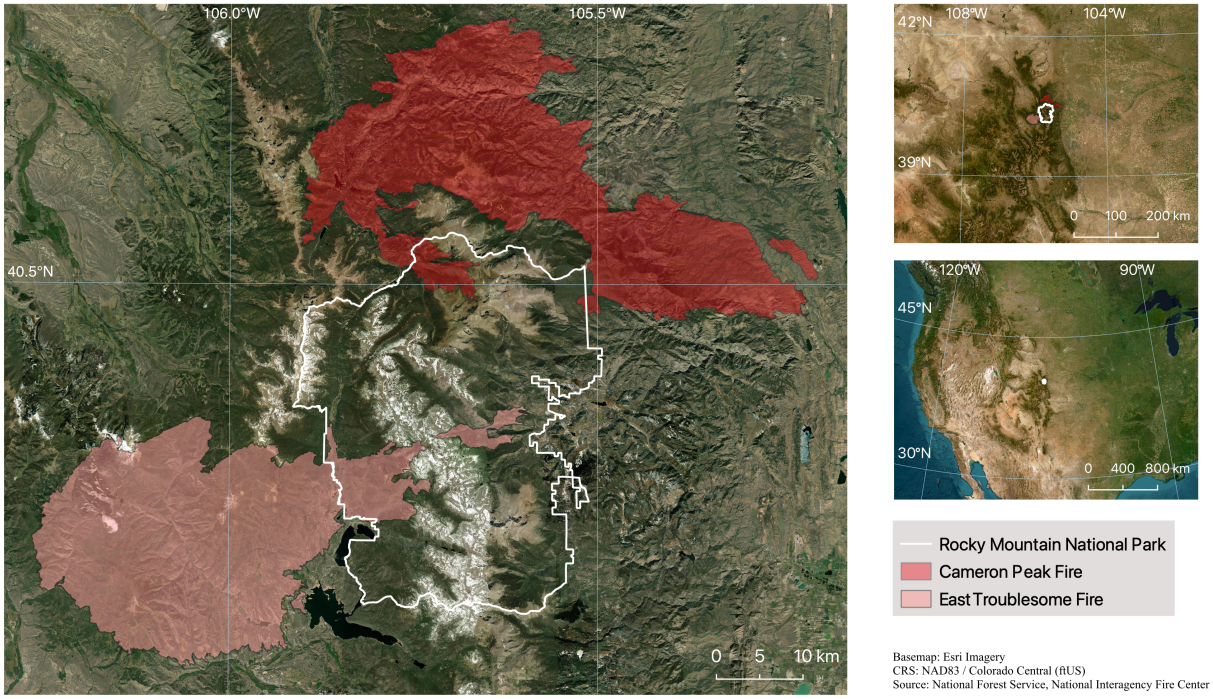

Figure 2: Overview of Test Area

As figure 2 shows, the Rocky Mountain National Park (RMNP) region was selected as the study area for testing on the project. With an area of 1074 square kilometers, RMNP is located in the middle of Colorado, United States. RMNP is blessed with abundant forest resources, with a diverse range of tree species, which offer a proper study area. RMNP has long been under threat of forest fire. The Cameron Peak Fire happened from August 13 to December 2, 2020, and burned more than 208 acres of forest. The East Troublesome Fire started on October 15, 2020, and ended on November 30, 2020, torching 193 acres of land. Detecting and monitoring the forest fire of RMNP is vital for its regrowth planning and future fire tackling. The goal is to detect the above Cameron Peak Fire and East Troublesome Fire happening in RNMP. A study period of 2018-2022 was selected to catch the temporal changes caused by fire.

3 Construction

3.1 Project structure implementation

Please note that the full description of the construction has been explained in the manual provided separately. In the report here I take the key point of it.

The project is fully written in Python language. It contains two independent custom Python modules: `GEE_Processor.py` (module for class and functions to pre-process and download data from GEE) and `Fire_Detector.py` (module for class and functions to detect the fire and validate the result). An execution file is provided to make use of those modules to get the result (for users familiar with Python script, the execution file is `main.py`. For users expecting to have an interactive experience in the Jupiter Notebook, it is `main.ipynb`).

For the data I/O (input/output), a .json file (`input_parameter.json`) is constructed to set all parameters and paths ahead. The users just need to provide the shapefile for the study area and true fire boundary (for

validation), declare their paths as well as the result directory in the .json file, and set the study period and other parameters in the .json file. After the execution, the project will pre-process and download the MODIS file of the given area and time, detect the forest fire by detecting the change point of the time series, and validate the detection results by giving statistics and visualizations. Each step will output the intermediate results into the local files as the input for the next step, which could guarantee the project be run from halfway to save time and effort.

3.2 Progress Schedule

The `GEE_Processor.py` was first written and tuned to make sure that needed data can be obtained from GEE. Then fire detection part in `Fire_Detector.py` was constructed to test fire detection methods on the downloaded data, followed by the visualisation. Most of the time this phase was used to find the best hyper-parameters. The validation part was first written in an individual module but finally integrated into `Fire_Detector.py` to make the workflow smoother. After that, I tested, polished, and packed all the modules and files together for delivery.

4 Test

4.1 Unit test on functions

```python
import unittest
import pickle
import json
import ee
import pandas as pd
import numpy as np
from pandas.testing import assert_frame_equal
from GEE_Processor import gee_processor
from Fire_Detector import fire_detector

class TestMain(unittest.TestCase):
    def setUp(self):
        # Load the test variables from a file
        with open('Test/test_variables.pkl', 'rb') as f:
            self.test_variables = pickle.load(f)

    def test_file_load(self):
        expected_result = self.test_variables['config']
        with open('input.json', 'r') as f:
            actual_result = json.load(f)
        self.assertEqual(actual_result, expected_result)

```

Figure 3: Extract from Unittest class

Unittest enables developers to verify the functionality of individual components within the project and identify and fix errors early in the project development. I performed the unit test on the functions needed for the execution by constructing a class with functions to test correspondent functions (figure 3 shows part of it, and `def test_file_load` in the test class tests `def file_load` (the function to read the .json file))

```
Ran 5 tests in 5.299s

OK
```

Figure 4: Extract from Unittest result

After the running, if the prediction result matches up with the actual result, the test is passed (as figure 4 shows). The functions needed for the execution all passed the test.

4.2 Running on test area and period

The Rocky Mountain National Park (RMNP) region and study period of 2018-2022 were used to test the project. I aimed to detect the Cameron Peak Fire and East Troublesome Fire happening in RNMP in 2020. The BEAST algorithms handled the time series of 230 time stamps (8-day composite) on around 10,000 pixels (spatial resolution of 250m). The process ran for around 1 hour. Each pixel with fire happening as well as its time of fire was recorded.

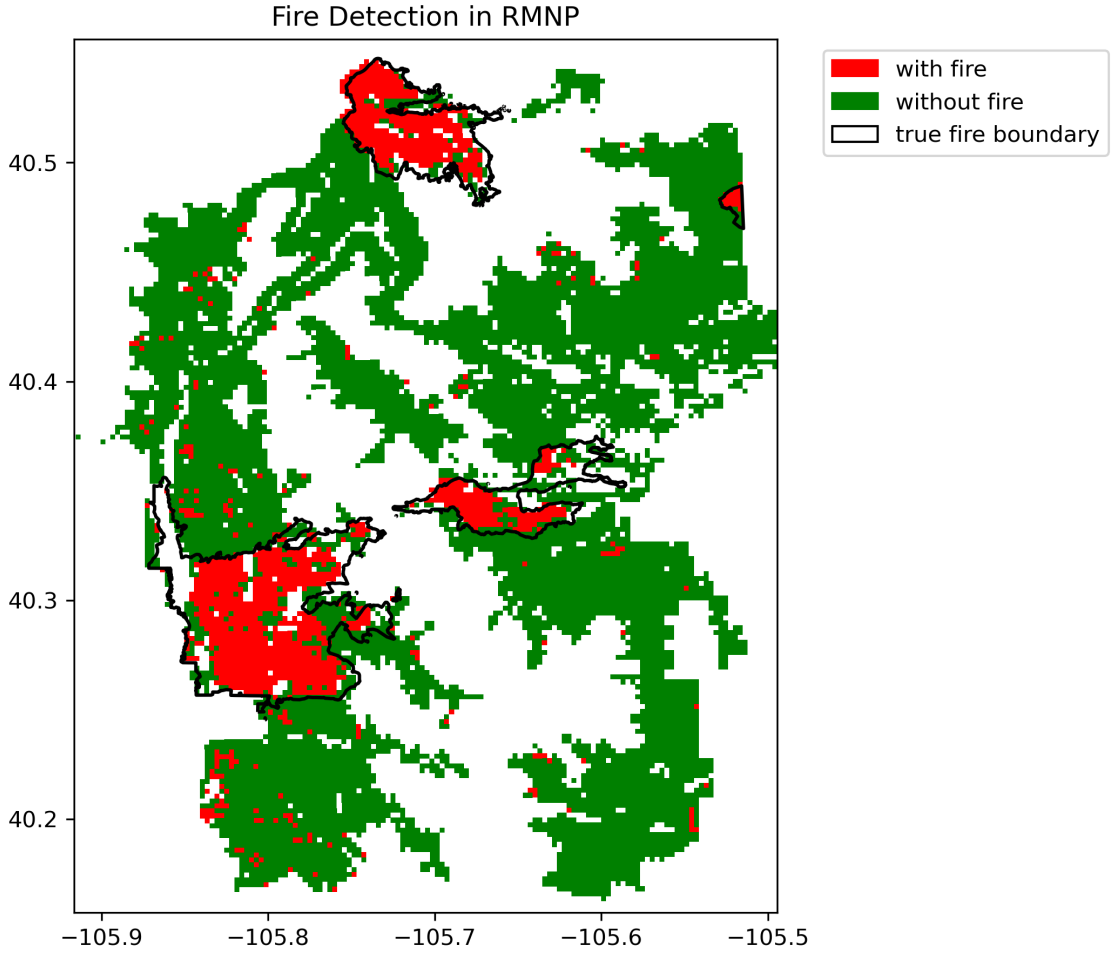

Figure 5: Detection Result of Test Area and Period

Figure 5 exhibits the result for the fire detection. Pixels with colors are forests in RMNP. Red indicates there was forest fire on the pixel and green means no fire. The black fringe on the map is the true fire boundary. It could be seen that the pixels with fire match up well with the true fire boundary, which proves the validity of the fire detection initially.

Table 2: Confusion Matrix of Detection Result

	Prediction Fire	**Prediction Non_Fire**
True Fire	1208	344
True Non_Fire	269	8299

Table 3: Evaluation Factor of Detection Result

Index	Value
Accuracy	0.94
Precision	0.82
Recall	0.78
F1 Score	0.80
Kappa Coefficient	0,76

Table 2 and 3 quantified the detection result. Most of the indices have a value around 0.8, which showed the detection result is satisfying.

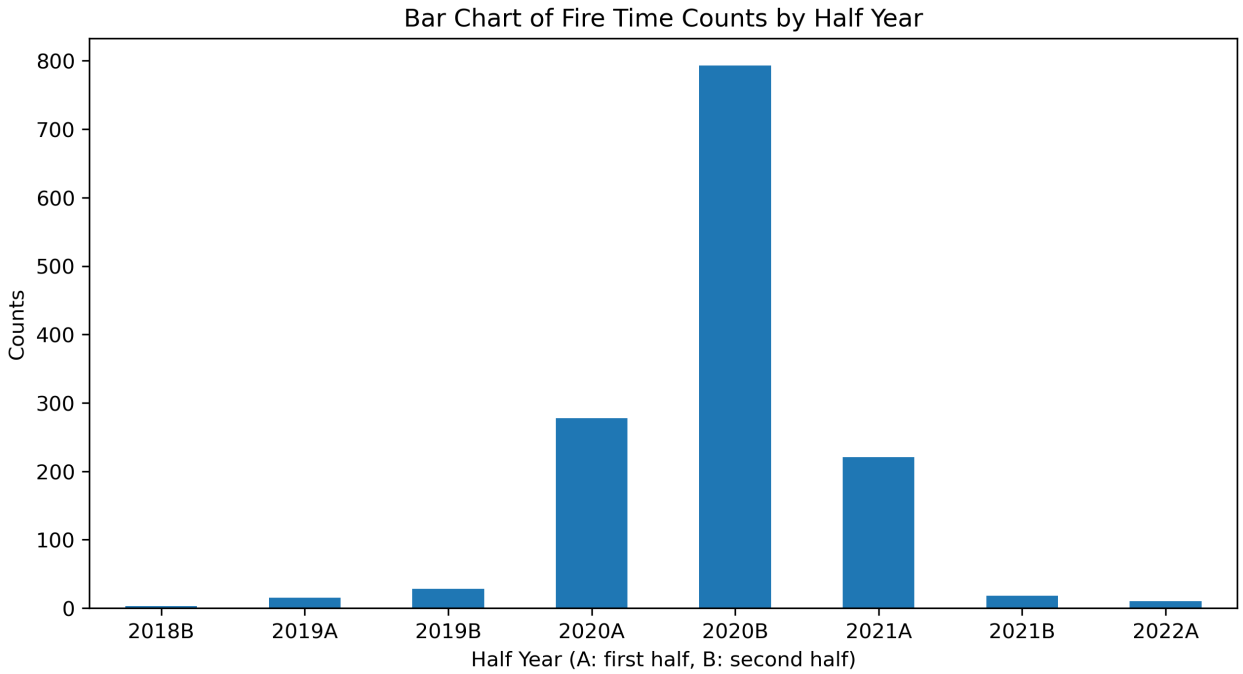

Figure 6: Detection Fire Time Counted by Half Years

Figure 6 visualizes the distribution of fire time detected. Most of the fire happened in the second half of 2020, which aligns with the official record fire time for the Cameron Peak Fire and the East Troublesome Fire. This graph proves the detection result is valid in terms of fire time.

5 Delivery

The project was finally packed in the directory containing the following files:

- `GEE_Processor.py`: data download and pre-process module.
- `Fire_Detector.py`: fire detection module.
- `main.py`: execution file of the project to refer to different modules to conduct the workflow.
- `main.ipynb`: execution file in the form of Jupyter Notebook to provide interactive working experiences (same functionality as `main.py`)
- `input_parameter.json`: set all parameters ahead flexibly to avoid any hard-coded vars.
- `Boundary_File`: sub-directory providing boundary files for research area and true fire for validation.
- `MODIS_File`: sub-directory to save the MODIS images and their band names downloaded from `GEE_Processor.py` and needed for `Fire_Detector.py`.
- `Fire_Detections_Result`: sub-directory to save the images and raster of the fire detection results and validation results as well.

- `README.md`: the manual that helps the user to understand and run the project.

In summary, this project can detect forest fire from a given study area and time by detecting the change point of the MODIS time series data within the United States. The execution of the project has been successfully tested on both Windows 10 and macOS Ventura platforms. To deliver the project to the users, a .zip file of the directory is provided and uploaded to the Canvas system. The user only needs to change the parameter in the `input_parameter.json` and arrange the boundary files. The project has also been uploaded into the GitHub platform (check the link in Appendix). Users can easily download or copy the repository, read the manual, and run the project.

6 Conclusion

The Python project requires detecting the forest fire swiftly and conveniently within the given area and time. For project design, the project applied the popular MODIS time series data, based on which the recognized BEAST algorithm was used for detection. A structure including two main functional modules ((1)data pre-processing and downloading from GEE;(2) fire detection and result validation), input data declaration file and execution file has been designed and constructed. The user can detect the fire within the area and period indicated in the declaration file by running the execution file easily. The project has been through the unit test and also detected the Cameron Peak Fire and the East Troublesome Fire happening in 2022 in the test area (the Rocky Mountain National Park) in one hour. Finally, the project has been packed as a .zip file and uploaded to the GitHub platform for delivery.

The project manages to detect fire with good validation. It is possible for users to take the project and apply different study areas and periods. However, due to the time limit, the project can be improved in some aspects, including (1) The project currently only handles MODIS data, and all of the pre-processing steps were tailored for that. The future version could introduce different data sources like Landsat or Sentinel. (2) As mentioned above, the project now applies to study areas within the United States, where the complete datasets can be collected easily. The project should also expand its application to different areas with the support of possible fire databases. (3) The data pre-processing and pre-processing were achieved in GEE and fire detection was performed with the local computer. It is possible to make use of GEE to handle the detection to save the local computation cost. (4) It took one hour to handle 230 time stamps and around 10,000 pixels. It can be accelerated by using multiple CPUs to conduct parallel computing.

7 Appendix

The link to the GitHub repository: Fire_Detection_BEAST

Overleaf was used to edit and format the document. GitHub Copilot was used to provide references for Python coding. ChatGPT was used to assist LaTEX coding and improve writing.

References

Chowdary, V., Gupta, M. K., & Singh, R. (2018). A Review on forest fire detection techniques: A decadal perspective. *Networks*, *4*, 12. Retrieved February 6, 2024, from https://www.researchgate.net/profile/Vinay-Chowdary-3/publication/332682989_A_Review_on_Forest_Fire_Detection_Techniques_A_Decadal_Perspective/links/5cc6ac0d299bf1209787544e/A-Review-on-Forest-Fire-Detection-Techniques-A-Decadal-Perspective.pdf

Huang, J., Wang, H., Dai, Q., & Han, D. (2014). Analysis of NDVI Data for Crop Identification and Yield Estimation [Conference Name: IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(11), 4374–4384. https://doi.org/10.1109/JSTARS.2014.2334332

Shumway, R. H., & Stoffer, D. S. (2017). Characteristics of Time Series. In R. H. Shumway & D. S. Stoffer (Eds.), *Time Series Analysis and Its Applications: With R Examples* (pp. 1–44). Springer International Publishing. https://doi.org/10.1007/978-3-319-52452-8_1

Zhao, K., Wulder, M. A., Hu, T., Bright, R., Wu, Q., Qin, H., Li, Y., Toman, E., Mallick, B., Zhang, X., & Brown, M. (2019). Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm. *Remote Sensing of Environment*, *232*, 111181. https://doi.org/10.1016/j.rse.2019.04.034