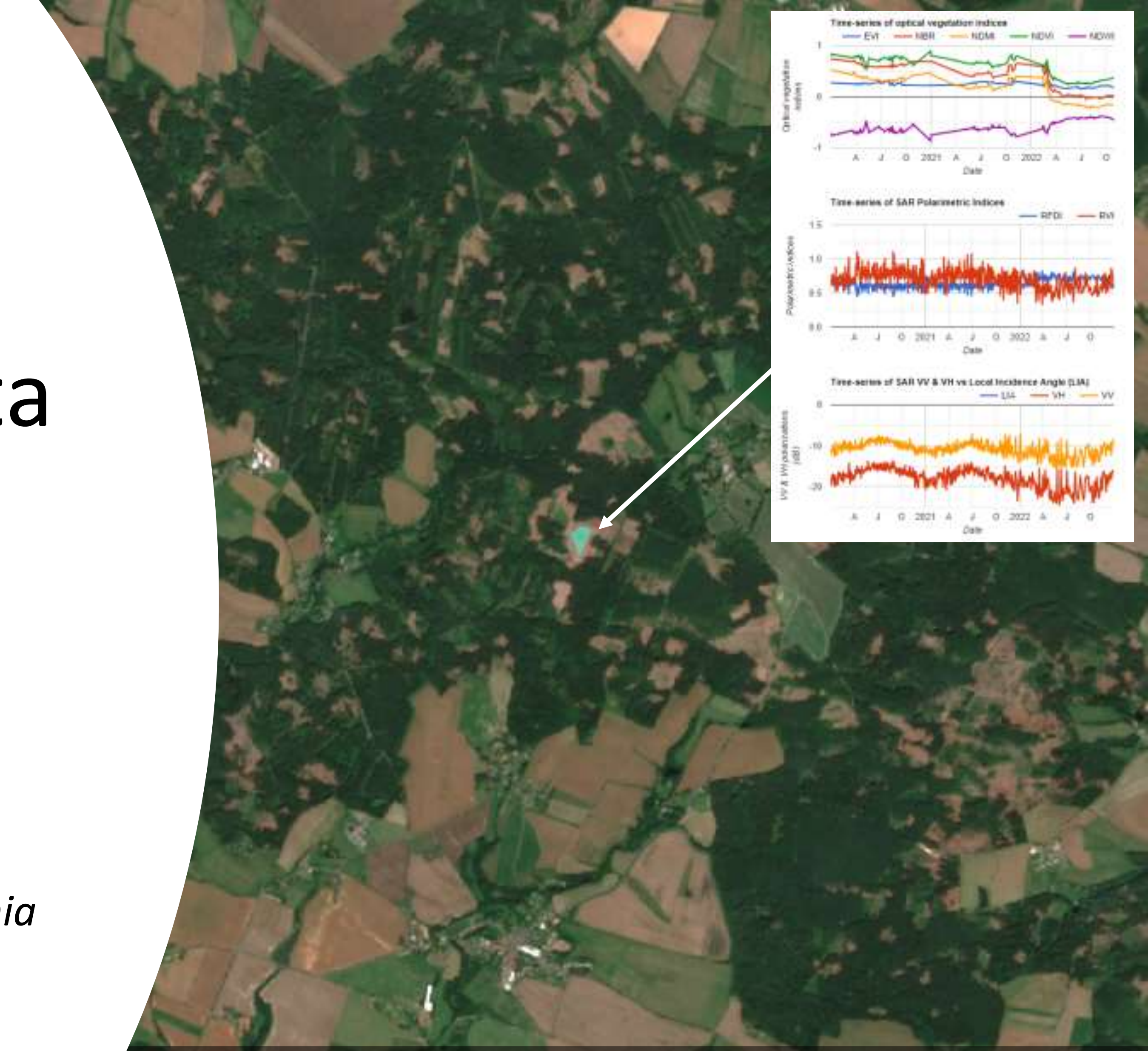
Applications of SAR data (time series) for forest monitoring

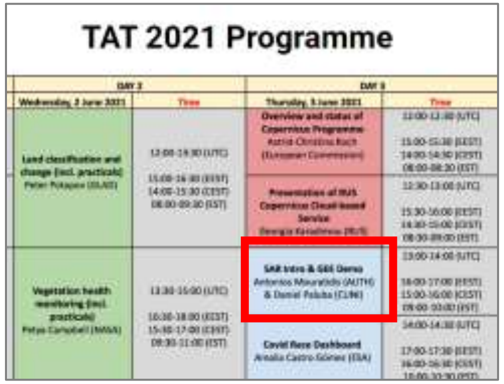


Daniel Paluba
PhD Candidate at *Charles University, Prague, Czechia*

CHARLES UNIVERSITY
Faculty of science

My way to TAT

- TAT 2017, Hungary – not accepted
- TAT 2019, Serbia – accepted as participant
- TAT 2021, Greece, Online – participant & I had a short demo on terrain effects on SAR data and its correction in GEE
- TAT 2022, Prague – organizing committee & teacher assistant
- TAT 2023, Prague & Brno – speaker of a session & organizing committee



Outline of this lecture

Theoretical part I.

- General applications on forest monitoring using SAR (focused on C-band)
- Current research efforts by EO4Lanscape group on using SAR and machine learning in forest monitoring

Practical part I.

- Optical and SAR time series in GEE
- Showcase of the "*SAR & Optical Time Series Explorer*" GEE app

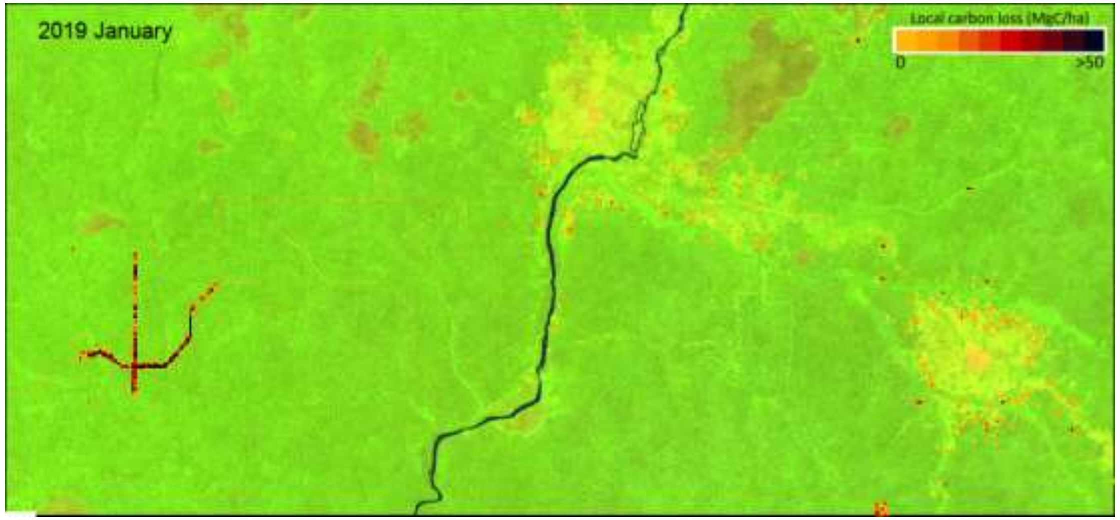
Theoretical part II.

- Estimation of optical vegetation indices using SAR data using machine learning
  - With a focus on data pre-processing and Automatic Machine Learning (AutoML)

Practical part II.

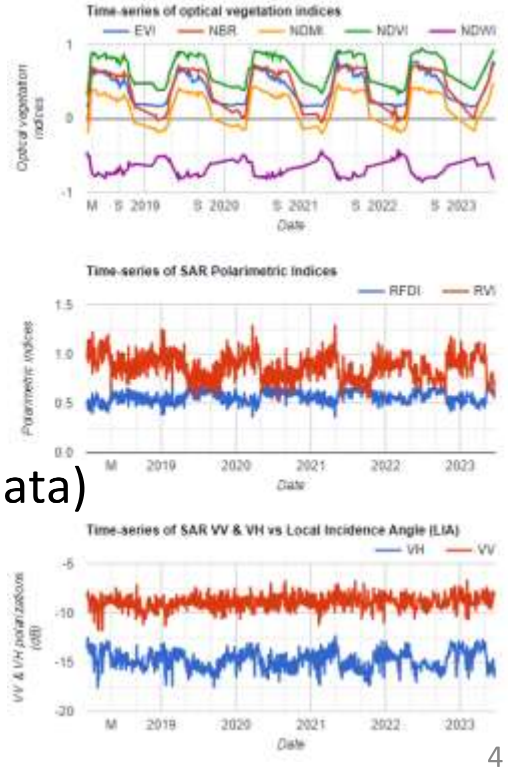
- Classical ML vs AutoML demonstration in Google Colab using Python

SAR applications on forests (focused on C-band)



Local carbon loss in the Central African Republic. Source: Reiche et al. 2021. URL.

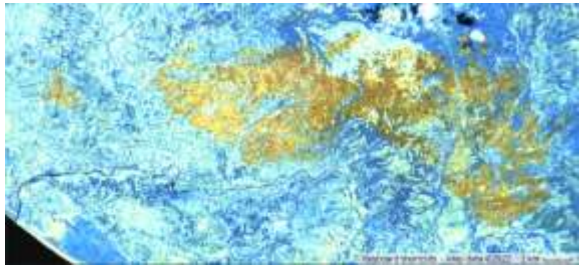
- Forest Change Detection and near-real time monitoring
  - Deforestation and Forest Degradation Monitoring (Reiche et al. 2021)
  - Forest change drivers monitoring (Slagter et al. 2023, Welsink et al. 2023)
  - Forest fires (e.g. Ban et al. 2020, Lasaponara et al. 2019)
  - Forest recovery monitoring after fires (De Petris et al. 2022)
  - Windthrows (e.g. Rüetschi et al. 2019)
  - Drought monitoring (Schellenberg et al. 2023, Kaiser et al. 2022)
- Forest seasonality and phenology monitoring (Soudani et al. 2021, Frison et al. 2018)
- Vegetation and forest type classification (Lechner et al. 2022, Dostálová et al. 2021)
- Forest Aboveground Biomass Estimation (Li et al. 2020 – combination with Landsat 8 data)
- Forest Height Mapping (Ge et al. 2022, Kumar et al. 2019)
- Fusion with other data, typically with other SAR bands or with optical data to improve forest monitoring … and much more ☺

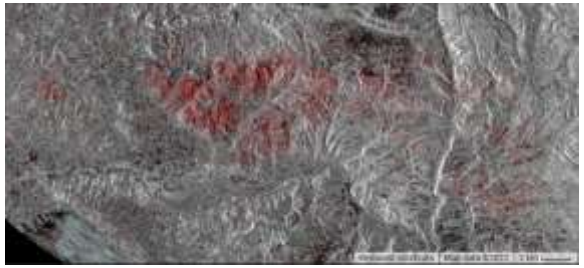


Current research efforts by EO4Landscape: enhancing forest monitoring through SAR data analysis and machine learning

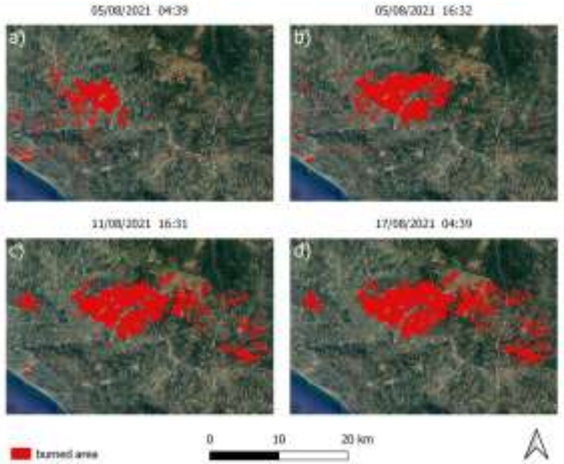
Wildfire monitoring using Sentinel-1 and unsupervised learning in GEE

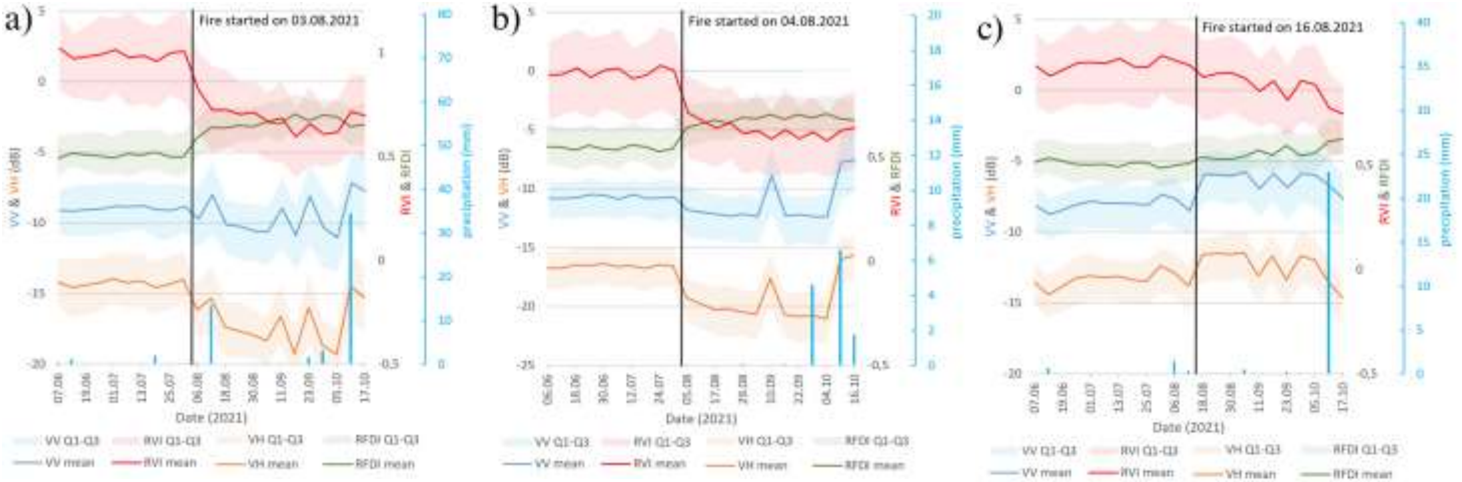
- Unsupervised k-means clustering with 2 clusters
- Reference data: Difference Normalized Burn Ratio Index (dNBR) from Sentinel-2
  F1-score, Overall Accuracy - OA, Omission Error (OE), Commission Error (CE)
- Various speckle filters and post-processing filters were tested
- Accuracies: F1 0.77-0.92, 85-95% OA, OE 8-28%, CE 5-17%





Spread of the fire in Olympia area





SAR time series for a) Evia, b) Olympia and c) Athens areas

Input features:

$$Kmap = \left(\frac{log\text{-}ratio}{\sigma}\right)$$

$$\sigma(i,j) = \sqrt{\frac{\sum_{t=1}^{N}(\beta^t(i,j)-\bar{\beta}(i,j))^2}{N-1}}$$

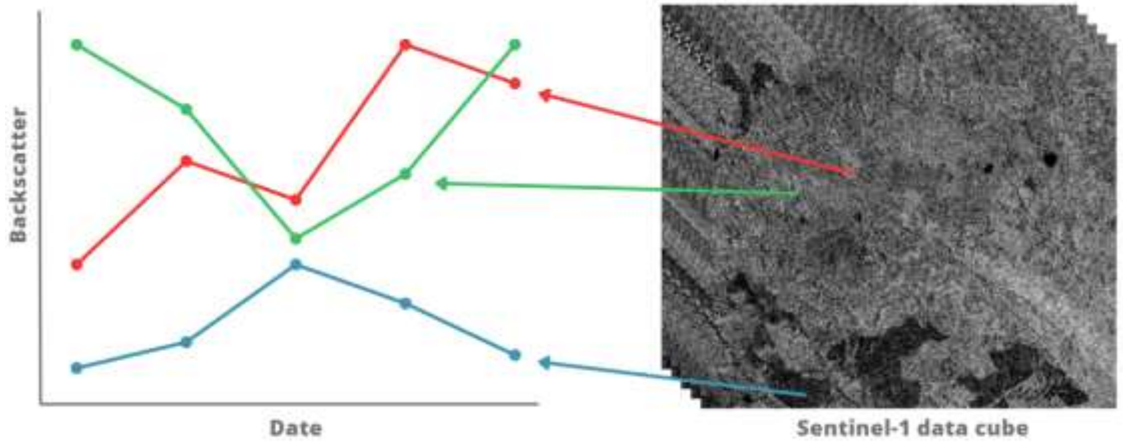
$$log\text{-}ratio_{pg} = 10log10\left(\frac{\delta^0_{post_{pg}}}{\delta^0_{pre_{pg}}}\right)$$

$$RVI = 4VH\left(\frac{4VH}{VV+VH}\right)$$

$$\Delta RVI = RVIpost - RVIpre$$

$$RFDI = \left(\frac{HH-HV}{HH+HV}\right) \sim \left(\frac{VV-VH}{VV+VH}\right)$$

Differentiation of land cover types using backscatter coefficient time series for Sentinel-1 time series clustering



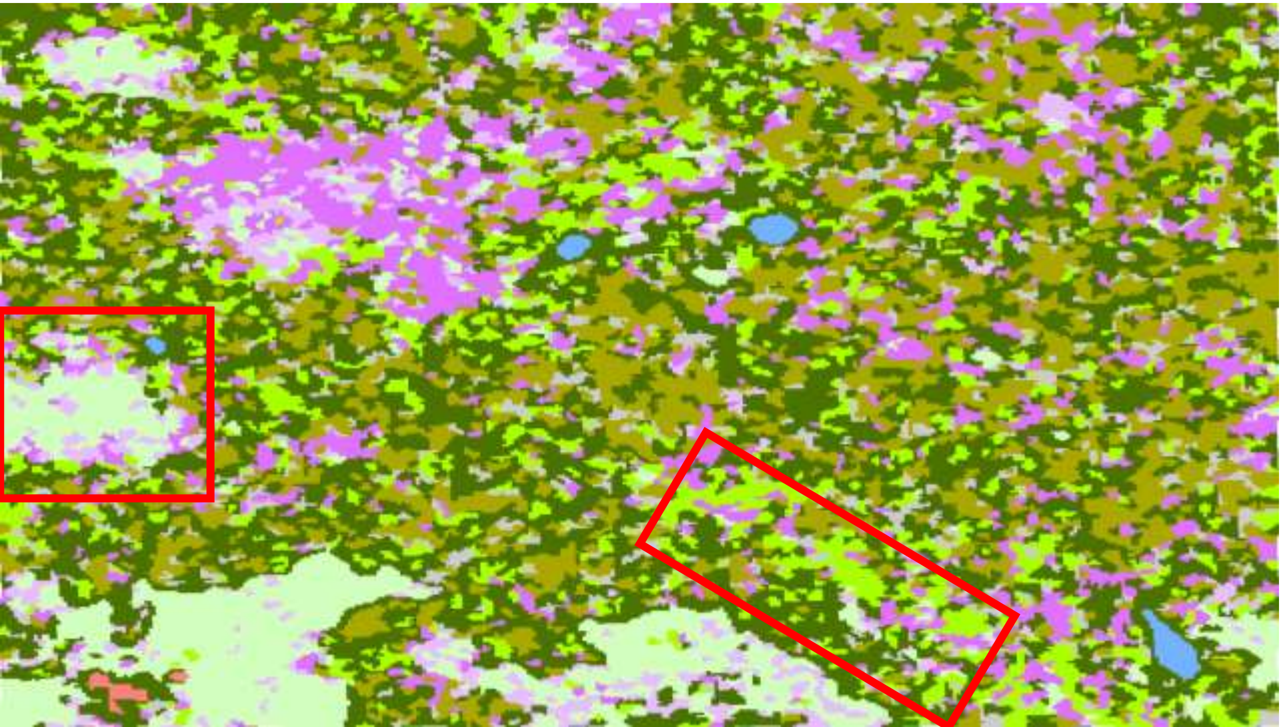
Why Time Series Clustering?

- Unsupervised machine learning does not require labeled data
  - Often unavailable or not in desired resolution / details
- Analyzing temporal patterns of data clustering based on similarities and differences in time series
- Accounting for seasonal and interannual variations and dynamics monitoring trends
- Can identify complex temporal patterns and changes in time use in change or anomaly detection
- Data pre-processing in GEE *sktime* time series ML library was tested

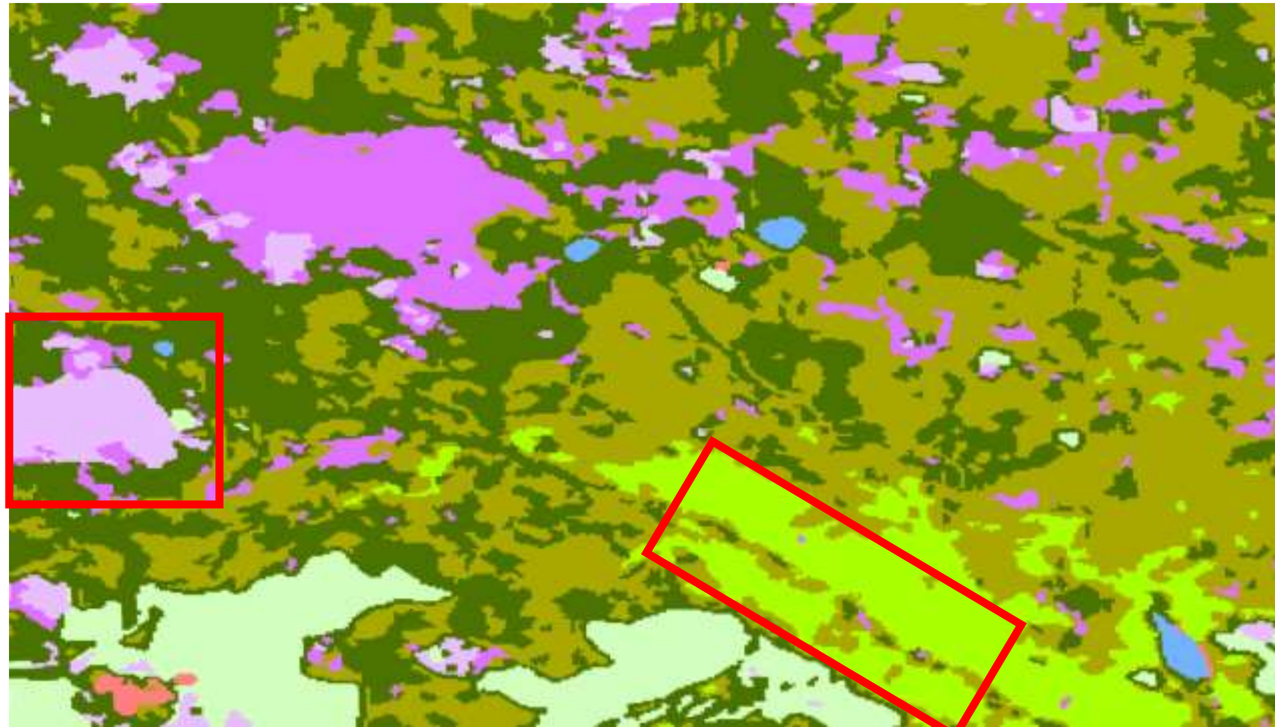
Mono-temporal vs time series clustering

- Problematic: grassland vs shrublands; deciduous vs coniferous vs young forest
- Salt-and-pepper noise left in mono-temporal clustering
- Good differentiation only in built-up and water classes

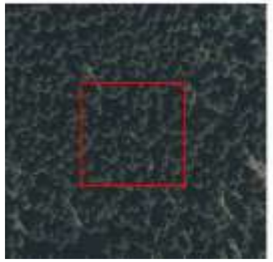
mono-temporal (3. august 2021)



time series clustering



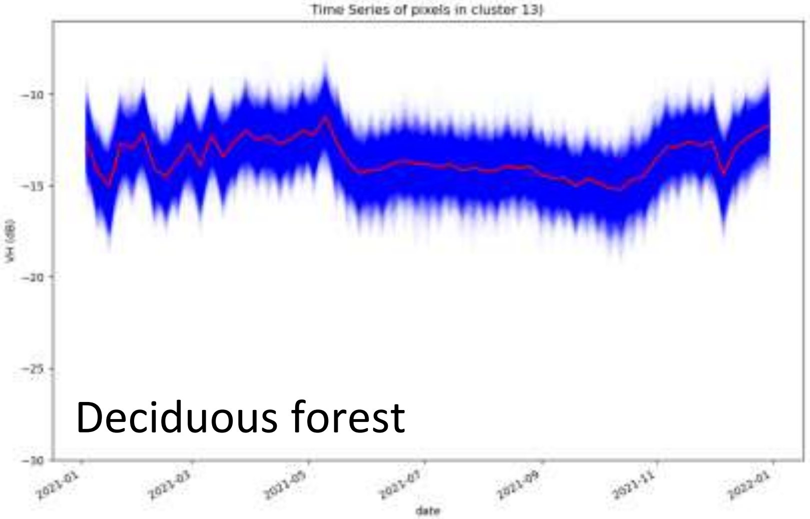
Sentinel-1 time series of clusters

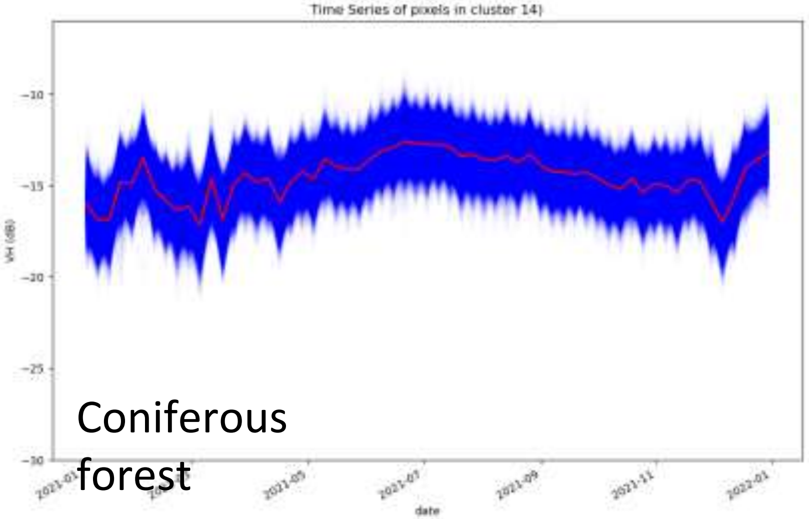

Coniferous f.

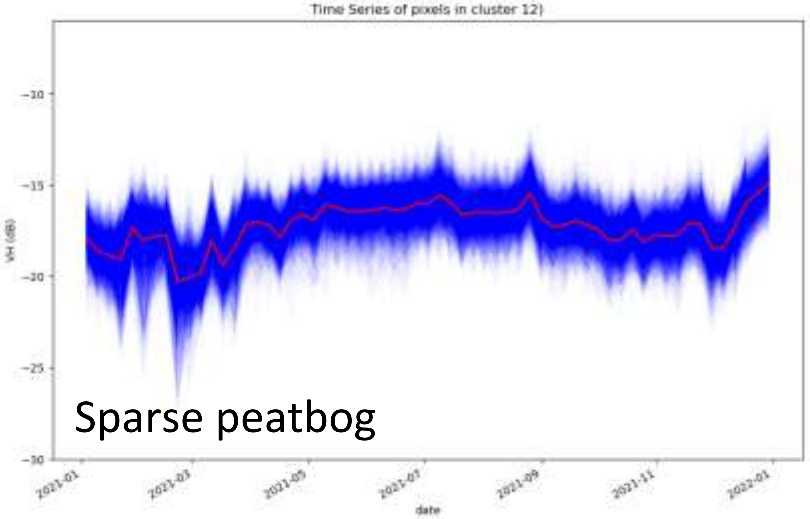

managed f.

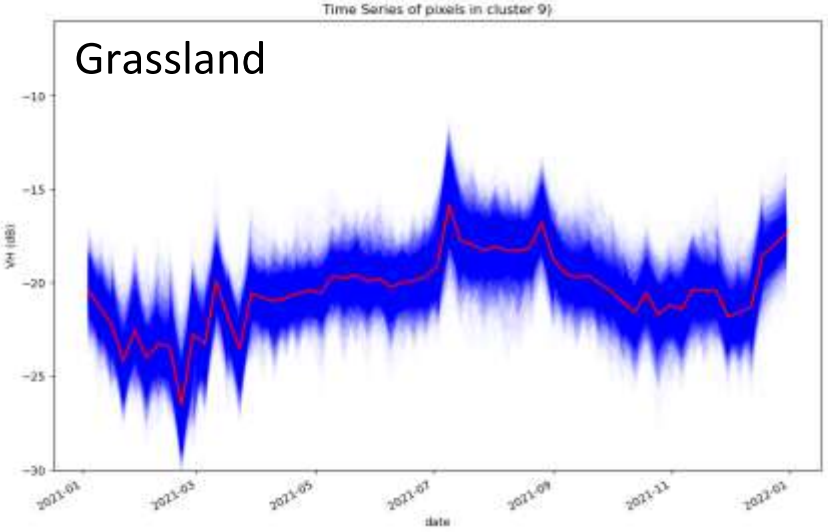

dense p.

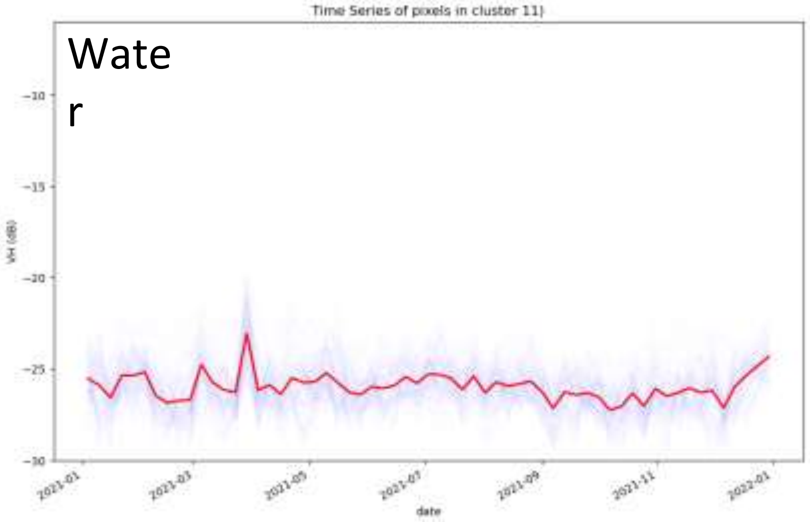

spase p.

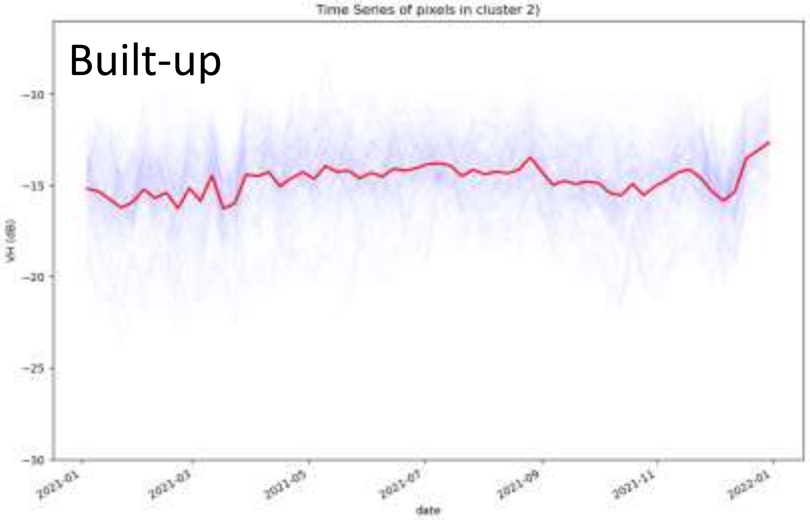




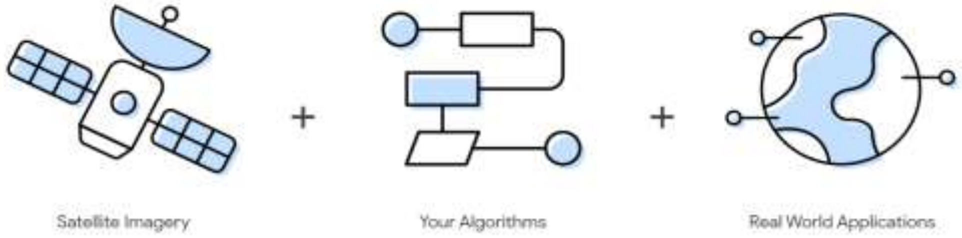






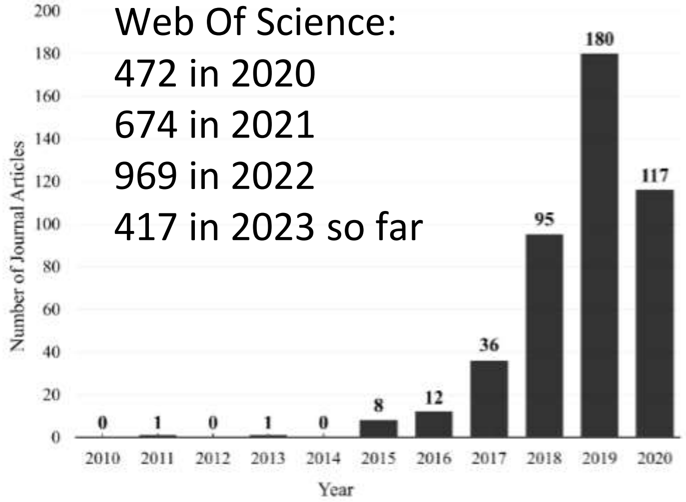


Why Google Earth Engine ?



Pros

- Huge data catalog available on cloud + processing power → no need to download data or software
  - You just need a web browser and internet connection
- Simple but powerful API (Python or JavaScript)
- You can use your own data or algorithms or create apps
- Great user community



Use of GEE in impact factor journals.
Source: Amani et. Al 2020 (May 2020)

Cons

- Knowledge of JavaScript or Python needed
- Not extendable with third-party libraries – use only what is available in GEE or create your our own
- Not open-source
- Free only for non-commercial use → we are optimistic that it will remain free in the future ☺

Practical part I.
GEE time series

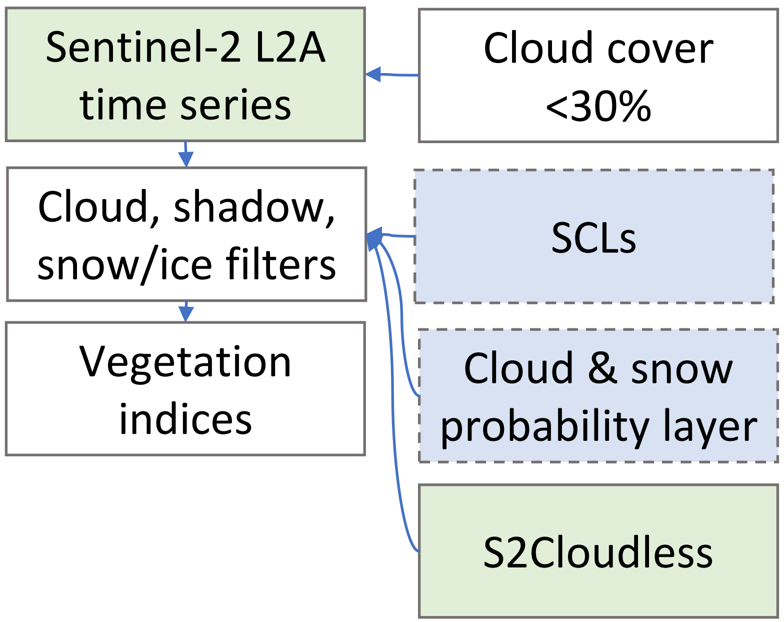
Google Earth Engine time series

QUIZ code:
https://code.earthengine.google.com/d800ad27238518cf8763eedda9a64f3f?noload=true

Solution code:
https://code.earthengine.google.com/868f4448c6d5d9913ea16e7887997d30

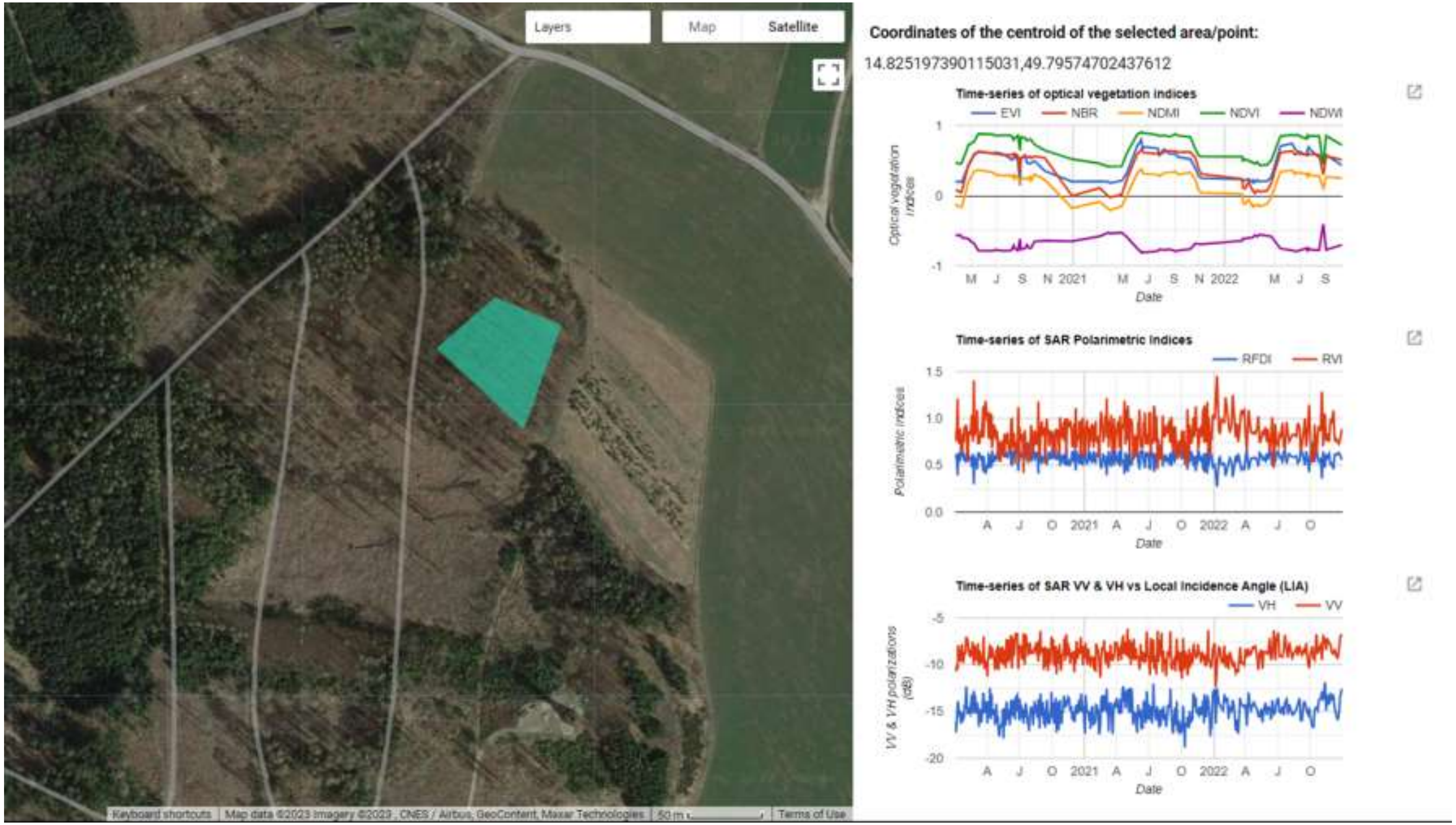
Available also from GitHub:
https://github.com/palubad/TAT2023

Sentinel-2 cloud/shadow/snow mask



SAR & Optical Time Series Explorer

https://danielp.users.earthengine.app/view/saropticaltsexplorer



Estimating forest health: Can we get standard optical vegetation indices from SAR data using AutoML?

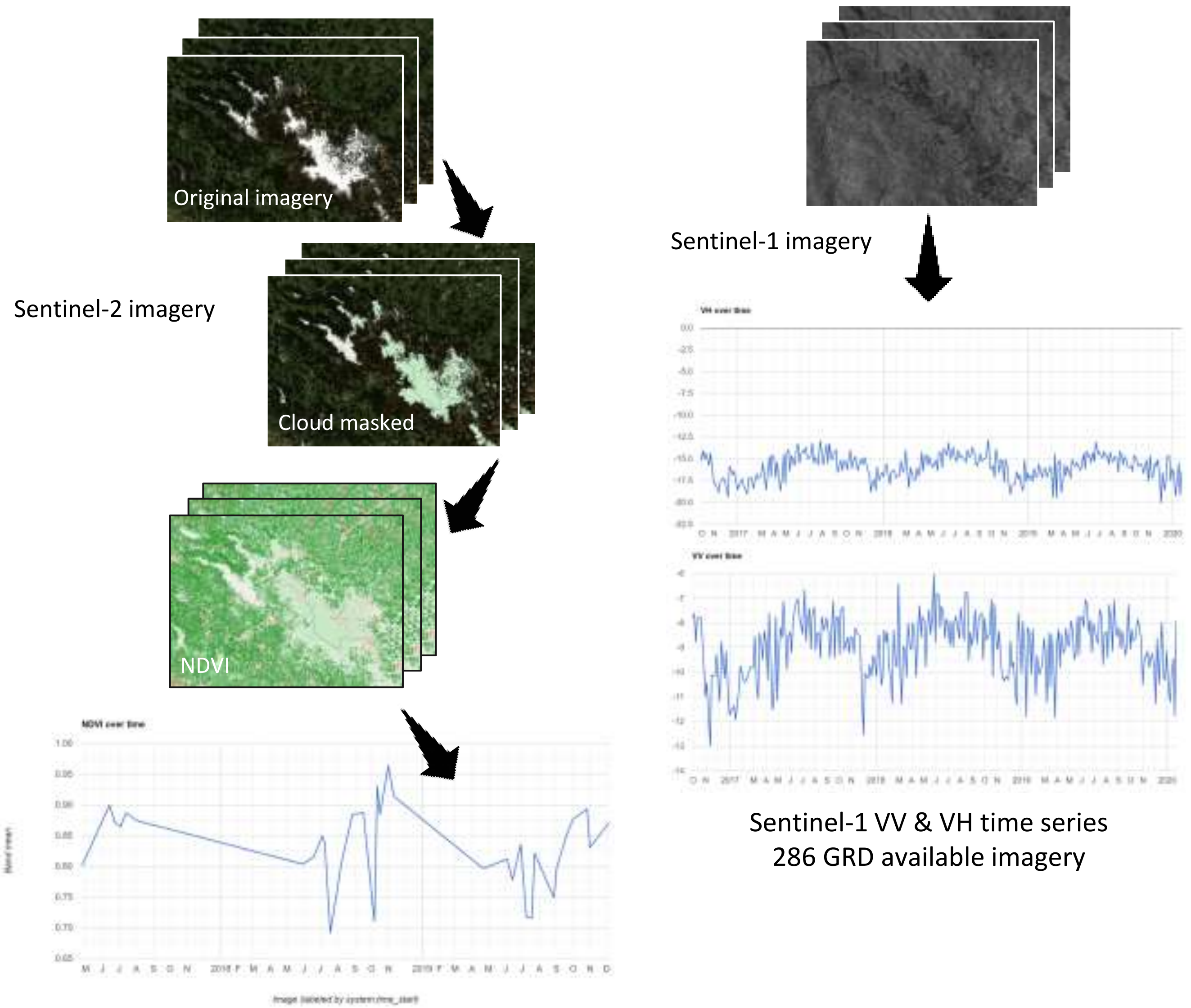
Φ-Lab: Daniel Paluba, Bertrand Le Saux,
ESRIN: Francesco Sarti (EOP-SD) & other ESRIN collaborators
Charles University: EO4Landscape research team, Přemysl Štych

CHARLES UNIVERSITY
Faculty of science

Motivation

Time series of an old, healthy forest in Czechia (24.04.2017 – 19.01.2020)

- **Forest health indicators come from optical satellite imagery:** Techniques to monitor vegetation with optical imagery are well-founded and often used by decision-makers in forestry, agriculture etc.
- **Problem**: Clouds – not only in tropics □ Central European forests are located mostly in mountainous areas
- **Workaround solution**: Conversely SAR imagery offers insightful and complete time-series!



Sentinel-2 NDVI time series (174 / 25 cloud free)

Objectives

1) **Go from SAR to optical**
   - Using ML can estimate standard vegetation indicators (inc. optical vegetation indices) from SAR C-band (Sentinel-1) data for various forest types

2) **Support** the current **Essential Climate Variables** (ECV): Apply the developed methodology in generating ECVs in a **higher temporal** and **better spatial resolution**

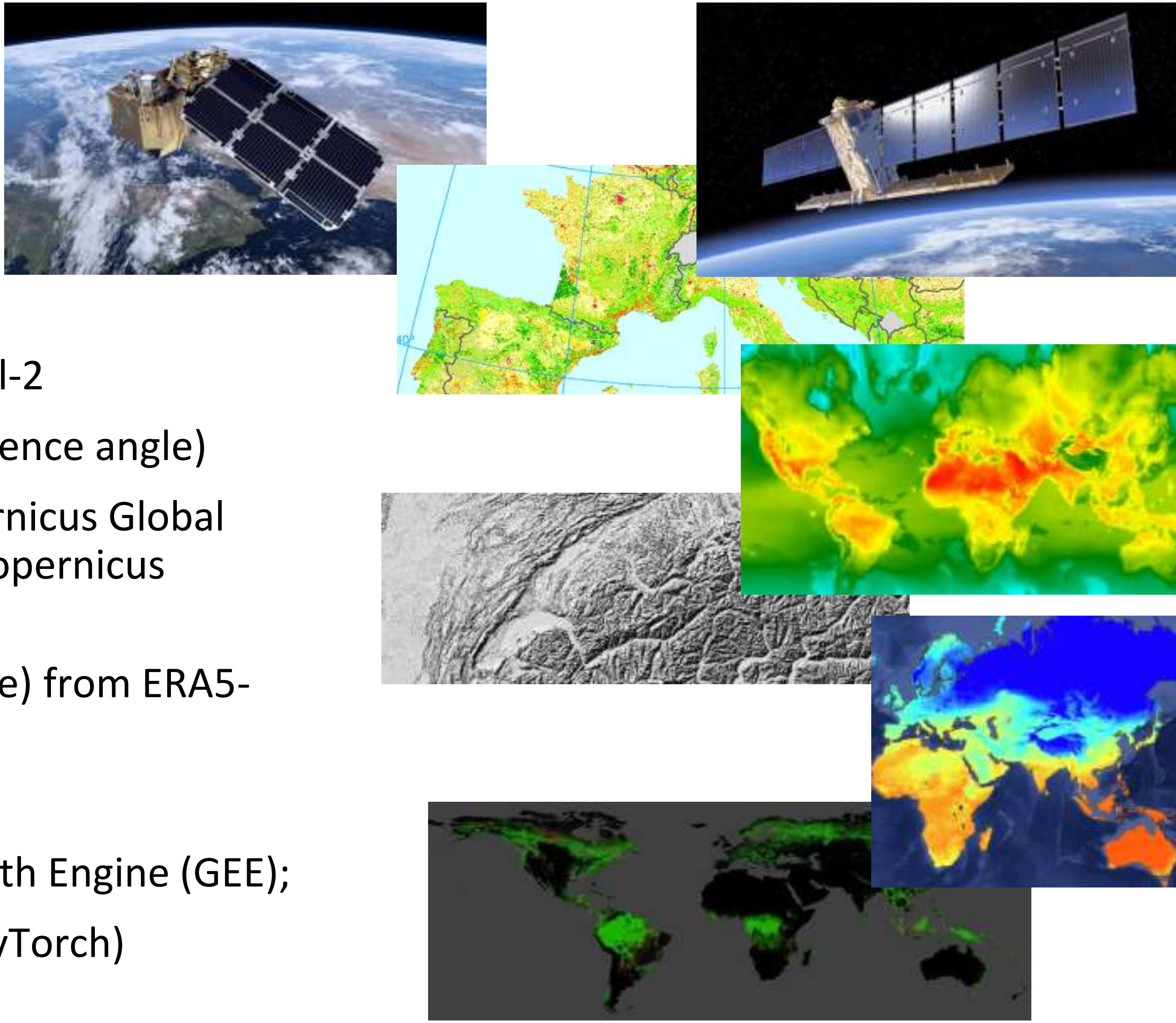
3) **Create materials for future ESA EO Education & Training**

4) (*longer-term*) **Investigate the approach for other SAR bands and polarizations** (X-, L- and P-band (airborne so far) & quad-pol.) and prepare future missions

Data

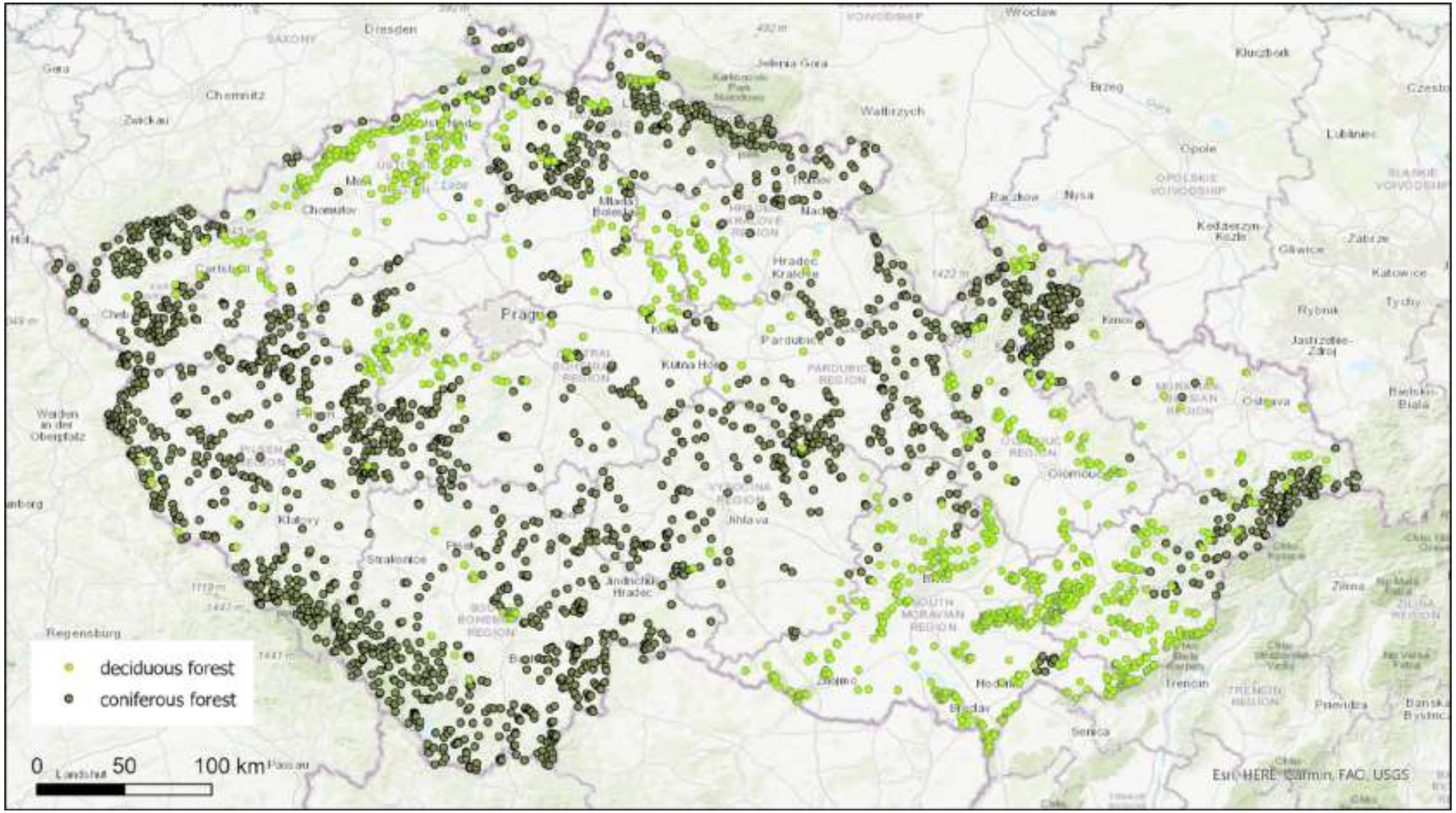
- Main focus on open access data:
- C-band SAR Sentinel-1, multispectral Sentinel-2
- Copernicus DEM (elevation, slope, local incidence angle)
- Land cover datasets: ESA World Cover, Copernicus Global Land Cover, Hansen Global Forest Change, Copernicus CORINE Land Cover
- Weather datasets (precipitation, temperature) from ERA5-Land

- Time series data preprocessing in Google Earth Engine (GEE);
- **AutoML approach** (Auto-sklearn and Auto-PyTorch)



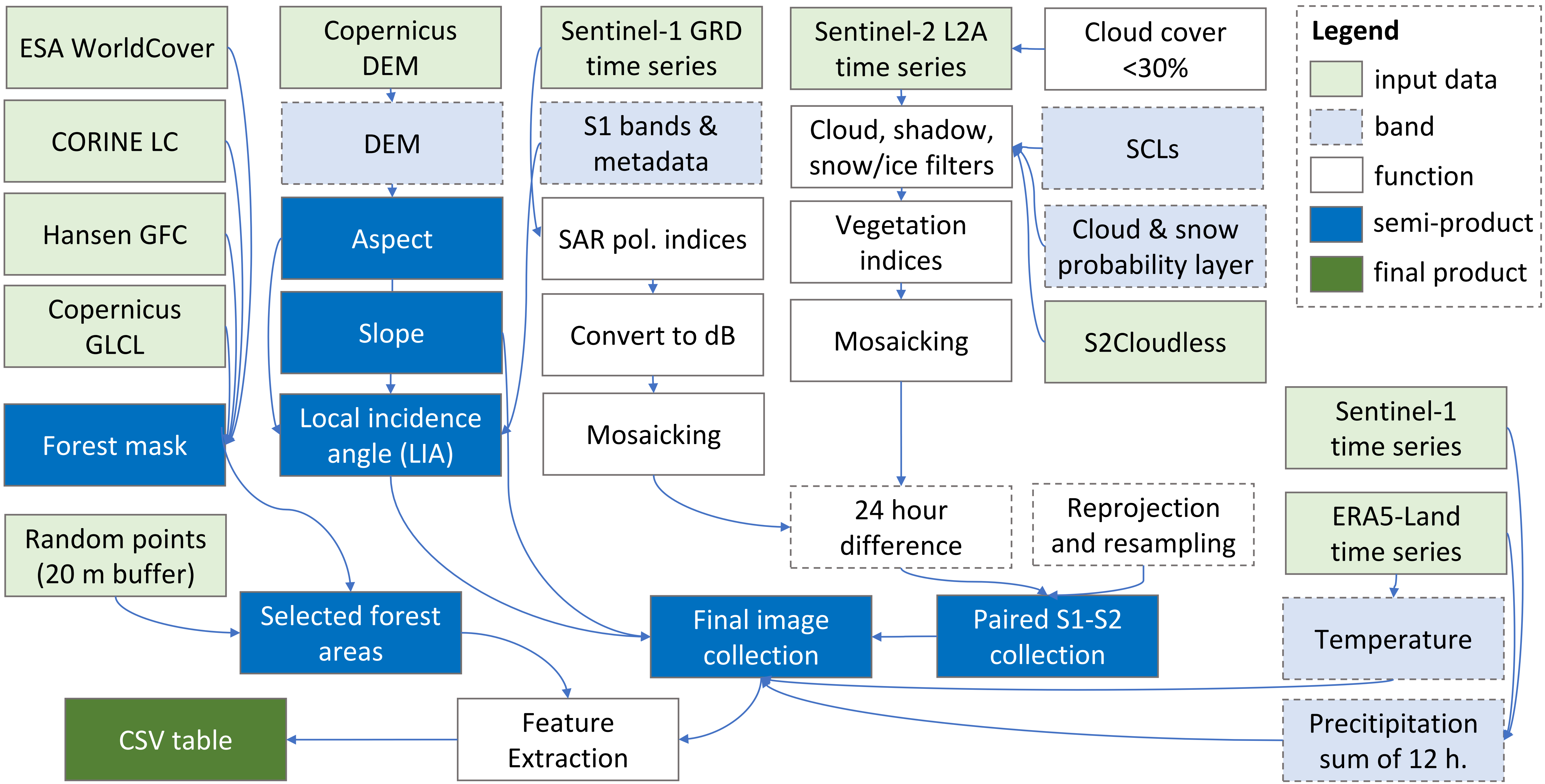
Study areas

- 2100 coniferous forests
- 1300 deciduous forests
- 1200 forest loss areas (2018-2021)



- Intersection of 4 land cover / forest databases in Czechia ⍰ generation of random points
  - Hansen Global Forest Change → 2000 base layer (forest cover >50%), forest loss (2000-2021) was masked out
  - Copernicus Global Land Cover Layer 2019 - **deciduous vs coniferous**,
  - CORINE Land Cover 2018 - **deciduous vs coniferous**,
  - ESA WorldCover 2021

Data preprocessing and preparation in GEE



Feature selection

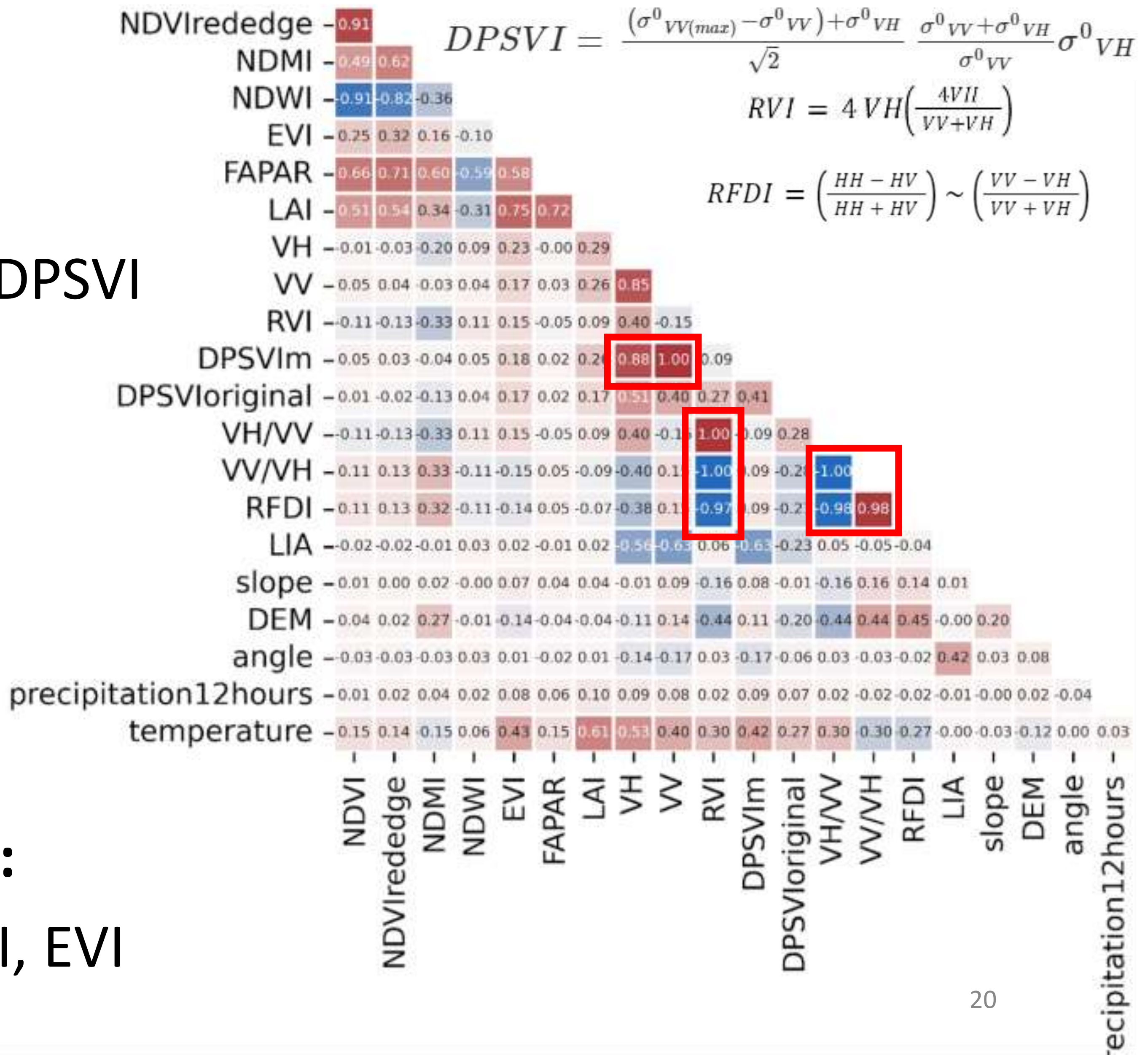
Input features:

- VV, VH
- ~~VH/VV, VV/VH, RFDI, DPSVIm,~~ RVI, DPSVI
- LIA, slope, DEM
- Precitipitation 12 h., temperature
- Day of the year (sin&cos),
- X,Y coordinates of the centroid
- Forest type

Estimated optical vegetation indices:

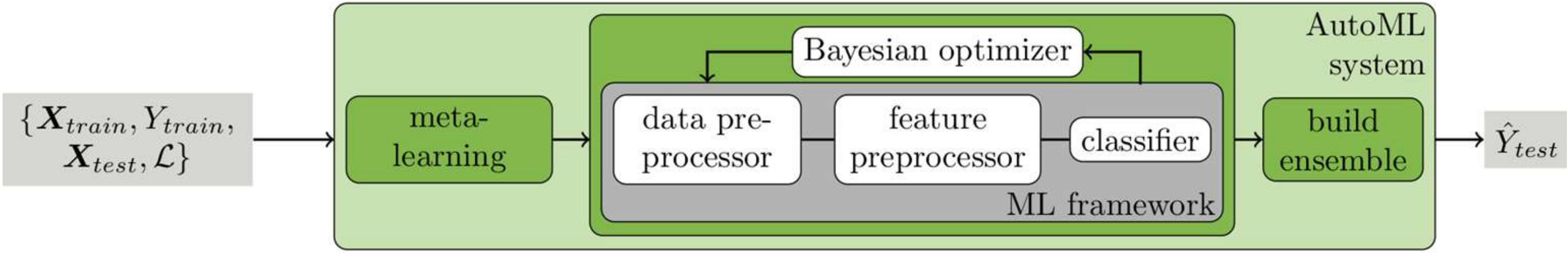
- LAI, FAPAR , NDVI, NDVIrededge, NDMI, EVI

Pairwise correlation between features



What is AutoML? Auto-sklearn as an example

- Out-of-the-box supervised machine learning – lower the barrier to use ML
- **Algorithm selection and hyperparameter tuning** trough **Bayesian optimisation**
  - 15 classification and 11 regression algorithms, 14 feature preprocessing algorithms (scikit-learn)
- **Builds an ensemble** with the best solution
- Uses meta-learning to identify similar datasets and use knowledge gathered in the past
  - Initialize the hyperparameter optimization algorithm configurations which worked well on previously seen datasets.
- **Other AutoMLs, e.g. Auto-PyTorch** □ Neural Architecture Search (NAS) - Fully automated deep learning (AutoDL)



Regression models:
- adaboost
- ard_regression
- decision_tree
- liblinear_svr
- libsvm_svr
- Mlp

In this work - training:
- 30% for testing
- 10-fold cross-validation
- Loss function: MAE

Why AutoML? Benefits, limitations, use cases

Benefits:

- Saves your time and increases your productivity - automates repetitive and time-consuming tasks, efficiently explores the algorithm and hyperparameter space.
- Improves the performance – uses ensembles to combine the best models.
- Makes ML accessible – user-friendly tools, enables non-experts to develop competitive models.

Limitations:

- May not capture domain-specific knowledge as effectively as manual approaches.
- Limited transparency and interpretability in model selection and hyperparameter optimization.
- Lacking customization - limited support for highly specialized or novel techniques.
- Most of the AutoML libraries prepared for tabular data, but the are some for image data, e.g. Auto-Keras, Ludwig.

Use Cases and Considerations:

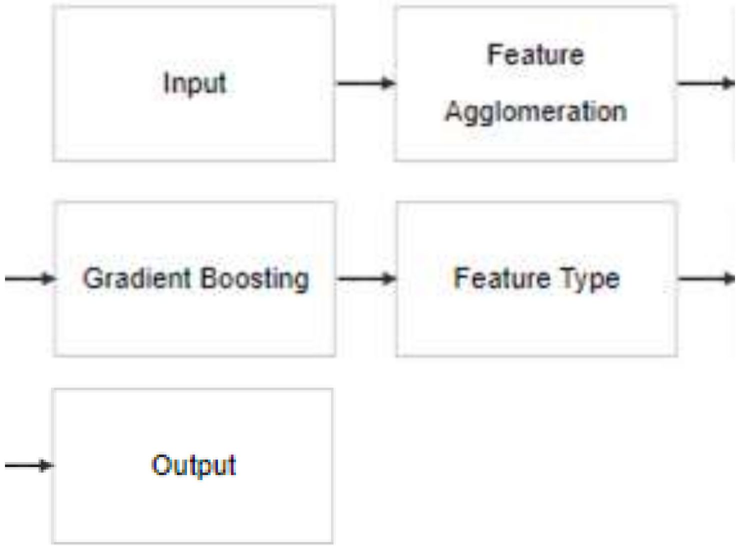
- Ideal when time and computational resources are limited.
- Well-suited for large-scale datasets where manual optimization becomes infeasible.
- Great also for exploratory analysis and prototyping – providing insights and direction for further analysis.
- Useful for researchers who are less experienced in machine learning or lack domain expertise.

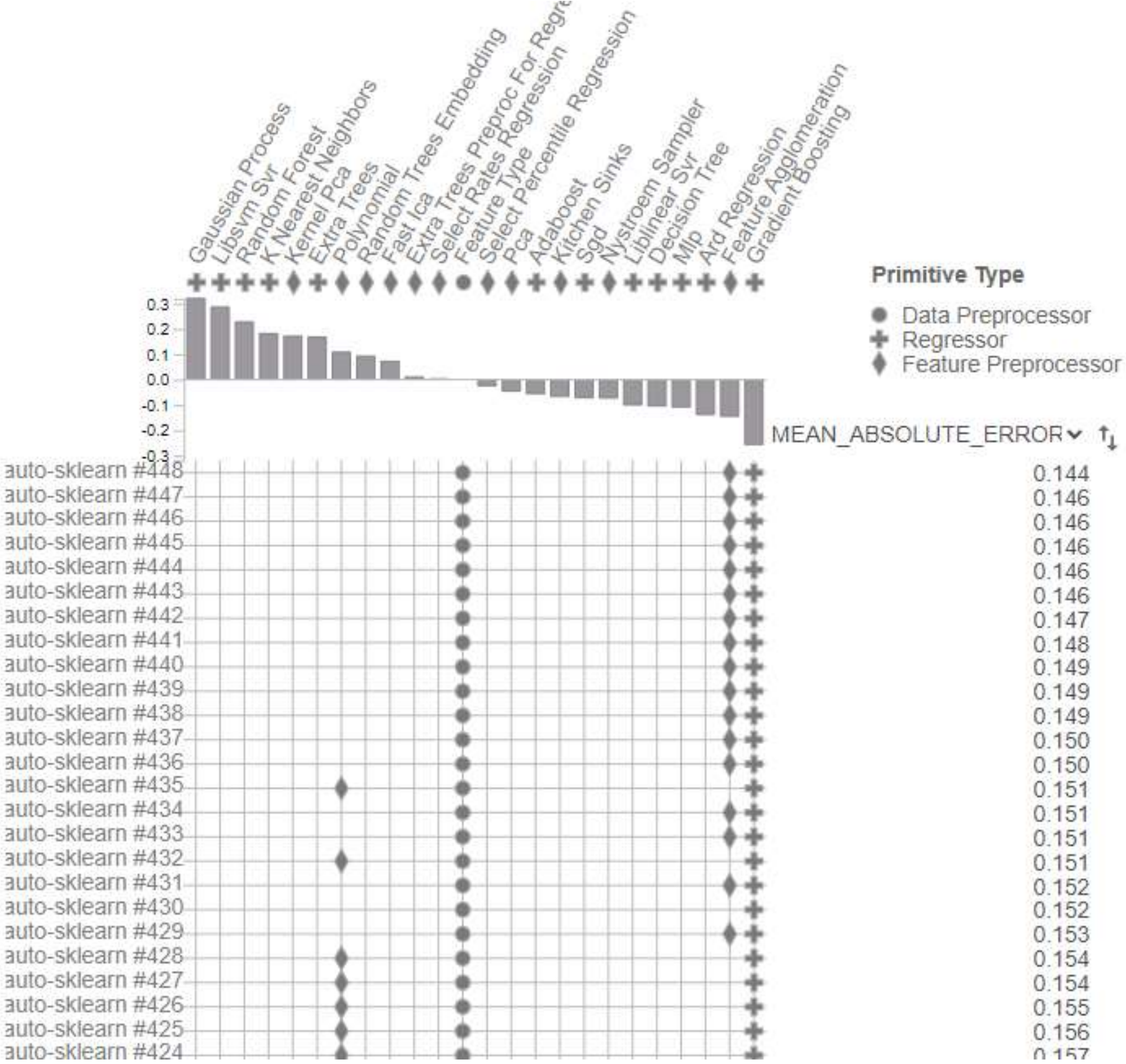
Auto-sklearn ouputs

Check out the final ensemble and their weights:

model_id	rank	ensemble_weight	type	cost	duration
346	1	0.18	gradient_boosting	0.144000	815.199952
404	2	0.16	gradient_boosting	0.145509	664.683397
431	3	0.10	gradient_boosting	0.145904	575.139516
389	4	0.04	gradient_boosting	0.146090	623.569639
343	5	0.18	gradient_boosting	0.146346	603.111197
398	6	0.04	gradient_boosting	0.147216	1229.422719
428	7	0.10	gradient_boosting	0.148389	1135.259109
205	8	0.06	gradient_boosting	0.148677	535.995555
180	9	0.04	gradient_boosting	0.150514	4599.399399
261	10	0.04	gradient_boosting	0.151253	4029.990861
392	11	0.06	gradient_boosting	0.156059	4081.637630

Explore the pipeline of each model tested model:





Explore each pipeline using the PipelineProfiler library

Auto-sklearn results

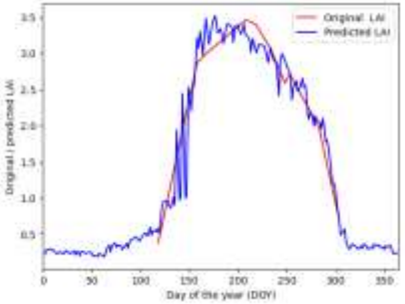
Statistical results (for LAI estimation):

- Deciduous forests
  - MAE = 0.296
  - RMSE = 0.163
- Coniferous forests
  - MAE = 0.256
  - RMSE = 0.113
- **Mixed model** - LAI
  - MAE = 0.268
  - RMSE = 0.137
- FAPAR
  - MAE = 0.054
  - RMSE = 0.005
- NDVIred-edge
  - MAE = 0.037
  - RMSE = 0.003

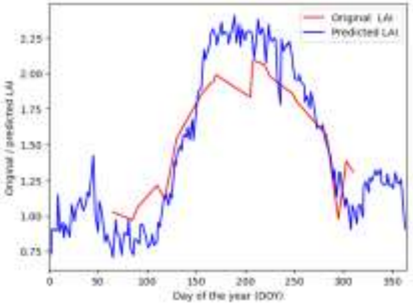
Optical: 9-25 in 2021
SAR: 70-170 in 2021

Separate models (healthy forest)

Deciduous forests

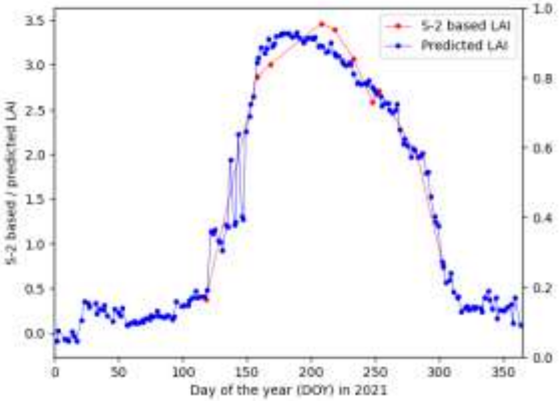


Coniferous forests

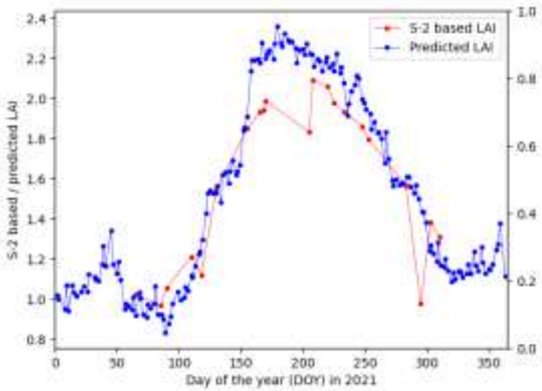


Mixed model (healthy forest)

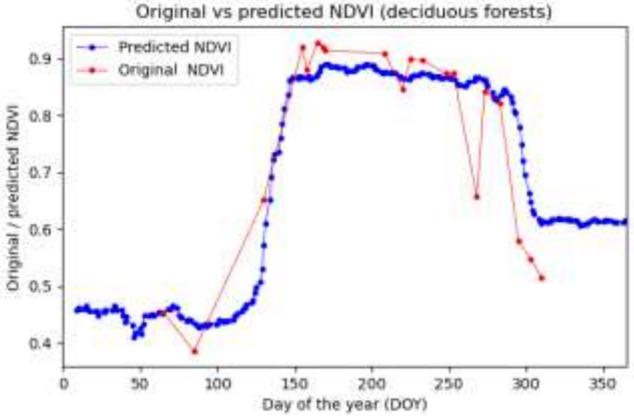
Deciduous forests

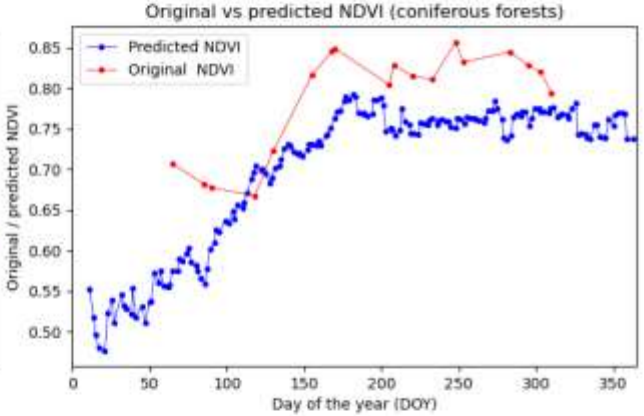


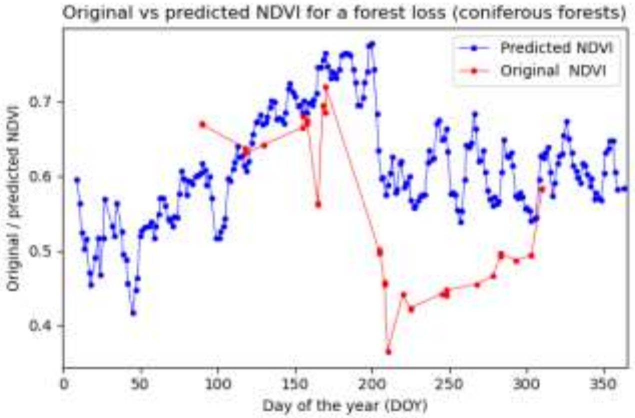
Coniferous forests

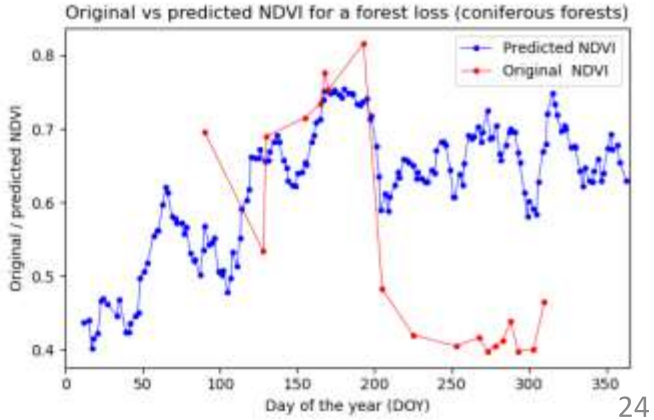


Added forest loss training (NDVI)









Conclusion / take away messages

- Best Auto-sklearn results using **Gradient Boosting** algorithm
- Slightly better results with Auto-PyTorch
  - Longer training (few hours) on a powerful computer is required
- Auto-Sklearn can be run on an “normal” computer with 4-8 cores (or even on Google Colab’s 2 cores) while a shorter training time is enough to found a sufficient number of successful runs **useful for training and educational purposes**
- Better temporal resolution was achieved compared to S-2 and other products (e.g. Copernicus GLMS)
  - Consistent vegetation index time series – up to 170 measurements/year
- Better spatial resolution (20 m) compared to e.g. Copernicus GLMS (300 m)

References

Ban, Yifang, Puzhao Zhang, Andrea Nascetti, Alexandre R. Bevington, and Michael A. Wulder. 2020. “Near Real-Time Wildfire Progression Monitoring with Sentinel-1 SAR Time Series and Deep Learning.” Scientific Reports 10 (1): 1322. doi:10.1038/s41598-019-56967-x.

De Petris, Samuele, Evelyn Joan Momo, Filippo Sarvia, and Enrico Borgogno-Mondino. 2022. “Multitemporal Dual-Pol Sentinel-1 Data to Support Monitoring of Forest Post-Fire Dynamics.” Geocarto International 37 (27): 15463–15484. doi:10.1080/10106049.2022.2098388.

Dostálová, Alena, Mait Lang, Janis Ivanovs, Lars T. Waser, and Wolfgang Wagner. 2021. “European Wide Forest Classification Based on Sentinel-1 Data.” Remote Sensing 13 (3): 337. doi:10.3390/rs13030337.

Ge, Shaojia, Weimin Su, Hong Gu, Yrjö Rauste, Jaan Praks, and Oleg Antropov. 2022. “Improved LSTM Model for Boreal Forest Height Mapping Using Sentinel-1 Time Series.” Remote Sensing 14 (21): 5560. doi:10.3390/rs14215560.

Kaiser, Philipp, Henning Buddenbaum, Sascha Nink, and Joachim Hill. 2022. “Potential of Sentinel-1 Data for Spatially and Temporally High-Resolution Detection of Drought Affected Forest Stands.” Forests 13 (12): 2148. doi:10.3390/f13122148.

Kumar, Praveen, and Akhouri Pramod Krishna. 2019. “InSAR-Based Tree Height Estimation of Hilly Forest Using Multitemporal Radarsat-1 and Sentinel-1 SAR Data.” IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 12 (12): 5147–5152. doi:10.1109/JSTARS.2019.2963443.

Lasaponara, Rosa, and Biagio Tucci. 2019. “Identification of Burned Areas and Severity Using SAR Sentinel-1.” IEEE Geoscience and Remote Sensing Letters 16 (6): 917–921. doi:10.1109/LGRS.2018.2888641.

Li, Yingchang, Mingyang Li, Chao Li, and Zhenzhen Liu. 2020. “Forest Aboveground Biomass Estimation Using Landsat 8 and Sentinel-1A Data with Machine Learning Algorithms.” Scientific Reports 10 (1): 9952. doi:10.1038/s41598-020-67024-3.

Reiche, Johannes, Adugna Mullissa, Bart Slagter, Yaqing Gou, Nandin-Erdene Tsendbazar, Christelle Odongo-Braun, Andreas Vollrath, et al. 2021. “Forest Disturbance Alerts for the Congo Basin Using Sentinel-1.” Environmental Research Letters 16 (2): 024005. doi:10.1088/1748-9326/abd0a8.

Rüetschi, Marius, David Small, and Lars Waser. 2019. “Rapid Detection of Windthrows Using Sentinel-1 C-Band SAR Data.” Remote Sensing 11 (2): 115. doi:10.3390/rs11020115.

Schellenberg, Konstantin, Thomas Jagdhuber, Markus Zehner, Sören Hese, Marcel Urban, Mikhail Urbazaev, Henrik Hartmann, Christiane Schmullius, and Clémence Dubois. 2023. “Potential of Sentinel-1 SAR to Assess Damage in Drought-Affected Temperate Deciduous Broadleaf Forests.” Remote Sensing 15 (4): 1004. doi:10.3390/rs15041004.

Slagter, Bart, Johannes Reiche, Diego Marcos, Adugna Mullissa, Etse Lossou, Marielos Peña-Claros, and Martin Herold. 2023. “Monitoring Direct Drivers of Small-Scale Tropical Forest Disturbance in near Real-Time with Sentinel-1 and -2 Data.” Remote Sensing of Environment 295 (September): 113655. doi:10.1016/j.rse.2023.113655.

Soudani, Kamel, Nicolas Delpierre, Daniel Berveiller, Gabriel Hmimina, Gaëlle Vincent, Alexandre Morfin, and Éric Dufrêne. 2021. “Potential of C-Band Synthetic Aperture Radar Sentinel-1 Time-Series for the Monitoring of Phenological Cycles in a Deciduous Forest.” International Journal of Applied Earth Observation and Geoinformation 104 (December): 102505. doi:10.1016/j.jag.2021.102505.

Welsink, Anne-Juul, Johannes Reiche, Veronique de Sy, Sarah Carter, Bart Slagter, Daniela Requena Suarez, Ben Batros, Marielos Peña-Claros, and Martin Herold. 2023. “Towards the Use of Satellite-Based Tropical Forest Disturbance Alerts to Assess Selective Logging Intensities.” Environmental Research Letters 18 (5): 054023. doi:10.1088/1748-9326/acd018.

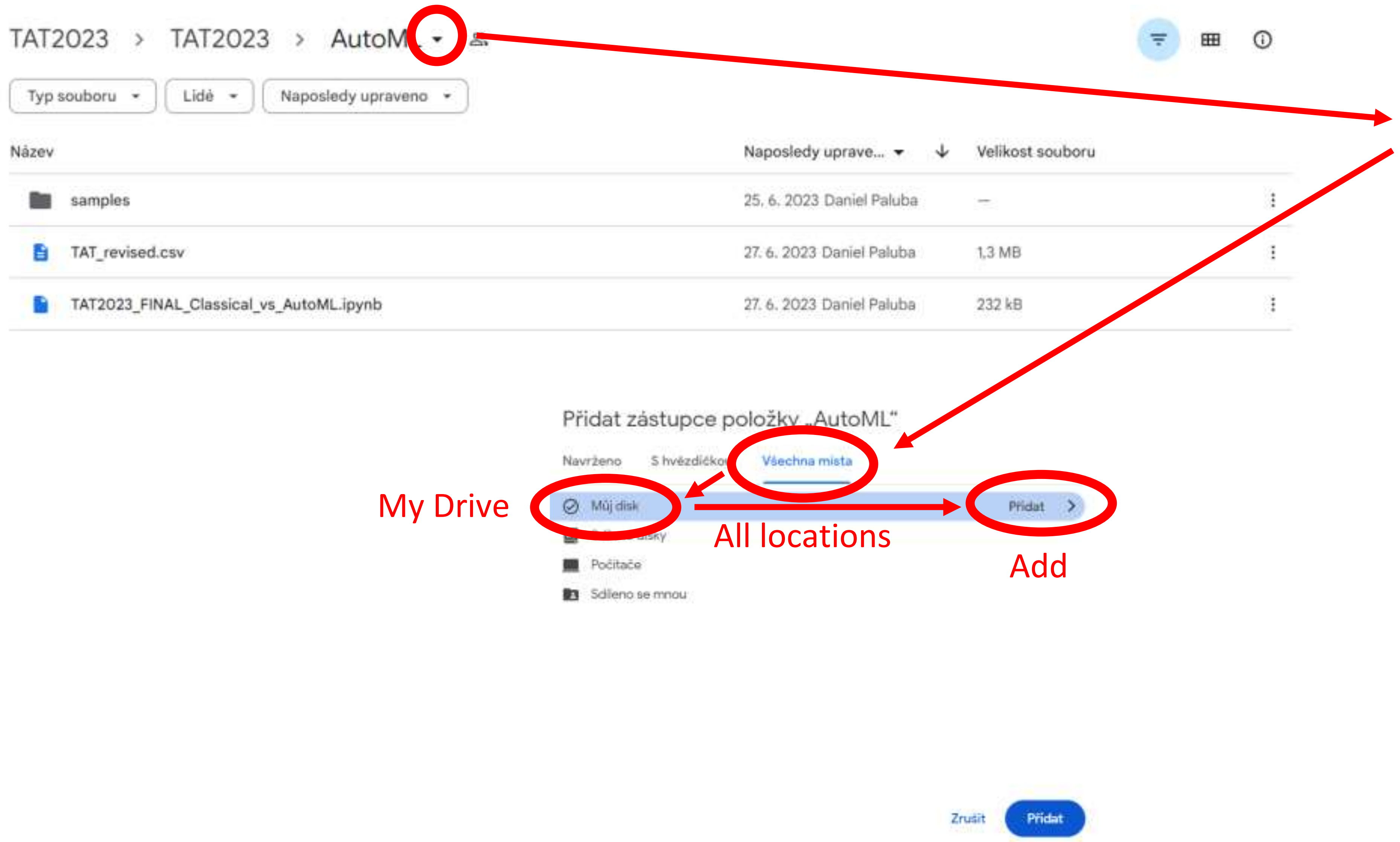
Thank you for your attention!

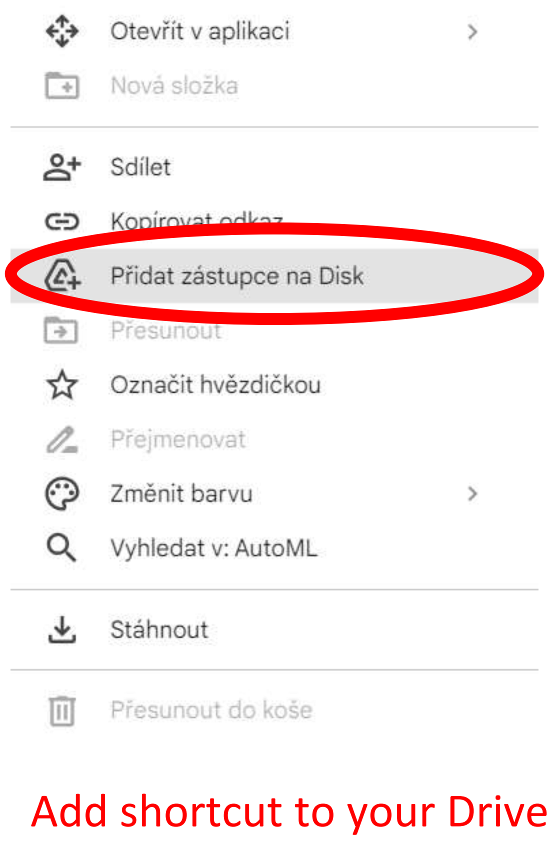
Daniel Paluba

palubad@natur.cuni.cz

Practical part II.
GEE time series & classical ML vs AutoML

Add folder to your Drive



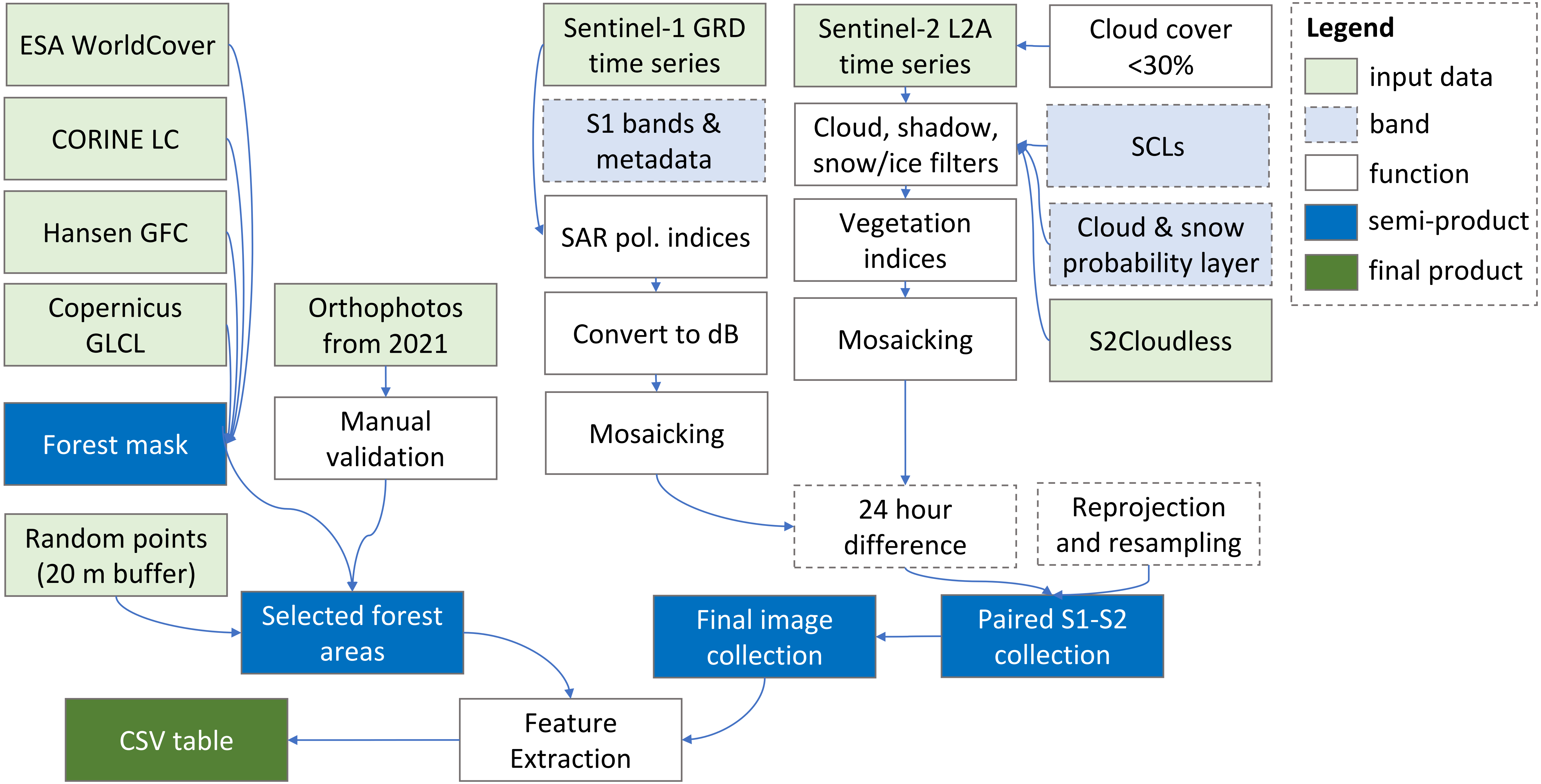


Classical ML vs AutoML

- Data: https://drive.google.com/drive/folders/1Vrtd2XAE5S7bZW7aSitMVwZCcQkLV8Mk?usp=drive_link

- Code: https://drive.google.com/file/d/1qcji9NfuKai_2TtoIq-vUVtaJk4ncpzx/view?usp=drive_link

Classical ML vs AutoML: A simplified data preprocessing and preparation in GEE



Google Colab: pros & cons

Pros:

- Cloud-based: no need to install any software & no worries about system requirements
  - Ideal for AutoML (Auto-sklearn) demonstration, while it needs a Linux OS or a Linux subsystem installed on your Windows
- Available from any device with an internet connection
- Pre-installed libraries and dependencies
- Collaborative features: collaborative editing, enabling multiple users to work together on the same notebook simultaneously, easily shareable
- Integration with Google Drive and GitHub

Cons:

- Limited computational resources for free: 2 cores with 12 GB of RAM.
- Dependency on internet connectivity: As a cloud-based service, Colab requires a stable internet connection. If you have limited or unreliable internet access, it can impact your ability to work on your projects. Disconnected after 90 minutes of inactivity.
- **Limited session duration and idle timeout**: Colab sessions have a maximum duration of around 12 hours, after which the session may be terminated.
- Harder to customize and control, e.g. library versions, etc.