USING DATA ANALYTICS AND MACHINE LEARNING IN SUSTAINABLE FOREST MANAGEMENT FROM REMOTE SENSING DATA

POLINA SYSOEVA

A THESIS SUBMITTED TO

THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF ARTS

GRADUATE PROGRAM IN INFORMATION SYSTEMS AND TECHNOLOGY

YORK UNIVERSITY

TORONTO, ONTARIO

MAY 2023

Abstract

Nowadays, remote sensing has become a widely used technique to acquire data for ecosystem service assessment (ESA) and other sustainable management practices. Remotely Sensed Data (RSD) is particularly crucial in locations where *in situ* observations are either limited or completely impossible due to their inaccessibility, such as mountainous areas. However, due to the unique features of the RSD, obtaining substantial insights requires specific preprocessing steps and strong computational algorithms, such as machine learning (ML). In the research, we present a methodology integrating RSD with data analytic and machine learning techniques for the needs of ESA. A pipeline for preprocessing EOS data, transforming into features, and experimenting with tuning of the ML algorithms is developed. A practical application of the proposed approach is demonstrated through assessing the impact of extreme weather events on forest ecosystems and their carbon sequestration abilities in two areas of the Kashmir Valley, Jammu & Kashmir, India.

Acknowledgements

Firstly, I would like to thank Prof. Peter Khaiter for his supervision throughout my master's studies and especially during the completion of this thesis. His expertise and guidance in using information technology, data science, and machine learning in the domain of sustainable environmental management was crucial to my exploration and understanding of this field of science, which was novel to me. An advice of the Data Services Librarian at York University, Rosa Orlandini, on the sources of remote sensing data was very helpful. I would also like to express my gratitude towards the members of my examination committee, Prof. Ling Jiang and Prof. Hyunwoo Lim, for their time, expertise and feedback. Lastly, I would like to thank my family and friends for their continuous support that allowed me to complete this milestone successfully.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

According to the UN Food and Agriculture Administration (UN FAO, 2023), sustainable forest management is "the process of planning and implementing practices for the stewardship and use of forests to meet specific environmental, economic, social and cultural objectives. It deals with the administrative, economic, legal, social, technical and scientific aspects of managing natural and planted forests". Khaiter & Erechtchoukova (2022) indicate that sustainable forest management is crucial, as the forest is an ecological-economic-social system (EESS) and the fulfillment of the UN Sustainable Development Goals (SDG) relies upon its preservation. Being an EESS, the forest provides ecosystem services that can be beneficial to all three pillars of SDG: economy, society and environment. One such service is carbon sequestration, the process of capturing and storing carbon in carbon pools, such as the leaves, roots and soil of a forest or a body of water. An accurate assessment of carbon sequestration is a crucial task for sustainable forest management, as the amount of carbon a forest can sequestrate is an indicator of the health of an individual ecosystem and a contribution to maintaining global environmental sustainability (Khaiter & Erechtchoukova, 2007).

Previously, scientists had to rely on gathering data on site, which meant it was small in quantity and became obsolete quite fast. Nowadays, remote sensing has become a widely used technique to acquire data for ecosystem service assessment (ESA). However, due to specific features of the data, researchers cannot get insights from it without using powerful computational algorithms.

This research aims to explore how methods of Data Science and Machine Learning can be used with Remote Sensing Data (RSD). The objective of the research is to derive a methodology for working with RSD and use it for sustainable forest management.

This chapter will provide an introduction to the study by first discussing the background and context, followed by the research problem, the research aims, objectives and questions, the significance and finally, some limitations.

According to NASA (2022), the Earth Observing System (EOS) is a system of satellites that orbit the Earth and record data about agricultural and water resources, the atmosphere, biodiversity, and wildfires. Researchers derive information from RSD by various methods, such as Machine Learning (e.g., Kankare et al, 2013), simulation modelling (e.g., He, 2008), and optimization (e.g., Pascual et al., 2016).

It is important to note that Remote Sensing Data cannot be used directly as recorded and require multi-stage transformations, which can be done with Geographic Information Systems (GIS) that allows its user to manipulate, analyze and visualize RSD. Modern GIS can solve various scientific problems by using 3- or even 4- (including time) dimension presentations and come in various forms.

Diverse computational techniques derive knowledge or predictions from remotely-sensed data. One particular technique deployed in research is Machine Learning which can predict events, such as disturbances caused to the environmental objects or ecosystem services they produce. Another common technique is optimization, particularly heuristic and metaheuristic methods, such as Genetic Algorithms or Simulated Annealing. As to the choice of a particular Machine Learning algorithm, the ensemble learning methods, such as Random Forest (RF), XGBoost or K-Nearest

Neighbors have been applied in several studies (e.g., Freeman et al., 2016; Sandino et al., 2018; McRoberts et al., 2011).

Khaiter & Erechtchoukova (2022) suggest that a methodology for working with remotely sensed data should include data acquisition, data preprocessing, data analysis and, depending on the goal of the project, modeling or optimization, and decision-making.

Khaiter & Erechtchoukova (2022) describe a decision-making system based on data and Machine Learning or optimization. Stakeholders may require to assess ecosystem services to estimate their monetary value but, in so doing, they encounter several challenges, such as lack of direct access to the area of interest due to its remote location and instrumentation mismatch, or lack the expertise to process remotely sensed data and interpret the results. The solution to these issues is to introduce the EOS into the assessment, as argued by Ramirez-Reyes et al. (2019).

Khaiter & Erechtchukova (2020) state that the quantification of ecosystem services must form the basis of sustainability assessment. For this purpose, they propose a theoretical framework and conceptualize an information system to assist policy- and decision-makers in their work. The underlying software architecture includes data source and data-warehouse layers which shows the necessity of integrating big data into the problem domain.

While the publications outline the necessity of incorporating data analytic approaches for working with RSD, they do not consider certain aspects of data availability, such as its sparseness due to the missions discontinued or only recently launched or its low quality because of the technical features of the sensors. Despite the satellite data becoming more available for research purposes, prior research (e.g., Zhan et al., 2022) has derived its data from the *in-situ* observations, for instance by using FLUXNET sensors. FLUXNET towers measure atmospheric parameters through eddy covariance methods (Montgomery, 1948). As mentioned above, accessibility issues

may prevent the researchers from conducting *in situ* observations, so we are proposing an approach that uses exclusively RSD to make predictions about carbon storage capacity of forest ecosystems and their health.

Thus, the current gap in knowledge gives a potential of using the combined methods of remote sensing data fusion, Data Science and Machine Learning to create a new instrument fit for policy- and decision-making.

This research aims to explore how methods of Data Science and Machine Learning can be used in combination with RSD. The objective of the research is to develop a methodology for working with RSD and to use it for forest ecosystem service assessment in remote and hardly accessible (e.g., mountainous) locations.

The research aims to answer the following questions:

- [RQ1] What are the steps necessary to pre-process RSD for using it in combination with Data Science and Machine Learning in the problem domain?
- [RQ2] What are the methods of transforming RSD into features appropriate for Data Science and using it in Data Science and Machine Learning in the field of study?
- [RQ3] What predictive modeling techniques are suitable for working with RSD time series characterizing the state of forest ecosystems?
- [RQ4] What are the specific techniques that are suitable for forest ecosystem service assessment based on predictions obtained from RSD?

The study contributes to the current research by providing a methodology that defines a six-phase roadmap: Feature Extraction, Data Preprocessing, Data Exploration, Time Series Analysis, ML Modeling and Ecosystem Service Assessment. Also, we plan to enrich the discourse by recommending practices of working with RSD in Data Science, such as conducting descriptive

statistical tests (the Five-Number Summary) and time series analysis tests (Stationarity tests, Lag Plot tests, Granger Causality test, Correlation tests and the Variance Inflation Factor test). Concerning the body of knowledge on using Machine Learning with RSD, we plan to test the performance of various ML algorithms, such as Random Forest Regressor, XGBoost Regressor and LSTM and prove their efficiency in this context. The idea of sustainable environmental management is approached through forest ecosystem service assessment and the corresponding monetary estimate through ETS pricing. As a practical implication of this study, a recommendation was made to introduce an Indian National ETS. It is anticipated that the decision-makers will be able to conduct a more accurate assessment of ecosystem services, especially crucial in light of ongoing climate change.

The research concerns selected areas of the Kashmir Valley in Jammu & Kashmir, India and accounts for the conditions of the region. At the same time, there are good reasons to believe that the solutions worked out in this research are universally applicable in similar regions, though the results may differ for areas with different climactic characteristics or types of ecosystems.

The remainder of the thesis has the following structure. Chapter 2 presents a literature review and bibliometric analysis of the current state of research and definition of the concepts crucial for the project. Chapter 3 outlines the region of study and research methodology including the sources and scope of the data collected, the transformation applied to the data and the methods of Data Analysis used to study the properties of the data and presents the research design. Chapter 4 states the results of the experiments and discusses their implications as well as potential applications for Ecosystem Service Assessment. Chapter 5 concludes the thesis by summarizing the work and discussing its limitations as well as future research directions.

Chapter 2: Literature Review and Bibliometric Analysis

The purpose of this chapter is to outline the current state of research in a form of literature review as well as to provide definition of the concepts crucial for the study. The review covers the studies carried out in the domain of assessing various ecosystem services produced by forests and outlines techniques of using RSD and machine learning. The goal of the review is to determine gaps in the knowledge within the domain.

The structure of this chapter is as follows. Firstly, we provide the definitions of the key terms pertaining to the domain. Then, we conduct a bibliometric analysis of the publications on the topic to assess the state of knowledge quantitatively. Finally, we summarize the research gaps found during the literature review and bibliometric analysis.

2.1 Definition of Ecosystem and Forest Ecosystem

We will begin by identifying the concept of ecosystem. A natural ecosystem can be defined as an interacting and interrelated combination of living biotic component and their non-living abiotic physical and chemical habitat (= biotope) that form a unitary whole within a certain spatiotemporal scale (Khaiter & Erechtchoukova, 2007).

One distinctive type of ecosystems is the forest. A forest ecosystem describes plants and other organisms inhabiting a common environment and interacting with its physical and chemical features. The mean annual temperature and the mean annual precipitation divide forests into subcategories, such as rainforests, temperate forests, boreal forests, shrubland *etc*. Changes in mean annual temperature and precipitation also cause changes in ecosystems. For example, Tyagi & Kumar (2022) describe this process in the Himalayan forests. Human industrial and agricultural activity has significantly influenced the ecosystems as well. Anthropogenic factors connected with

these activities include increased resource consumption, waste production, pollution and a shift in chemical and biological thresholds (Pegov, 2008).

2.2 Definition of Ecosystem Services, Ecosystem Services of the Forest

According to Millennium Ecosystem Assessment (2003), ecosystem services are goods (material) and services (immaterial) that an ecosystem can provide to the human population.

Khaiter & Erechtchoukova (2010) point out that forests have three groups of ecosystem services – economic, ecological and social. This is reflected in the FEES framework proposed by Khaiter (1993a). The authors indicate that not all ES are mutually possible, since active production of one set may negatively affect the others. The same source lists wood and non-timber goods production as an economic ecosystem service; erosion regulation, climate regulation, carbon sequestration and habitat provision as ecological ecosystem services; and recreation and tourism as social ecosystem services. Ecosystem service assessment is a non-trivial task and requires building models of each ecosystem service (Khaiter & Erechtchoukova, 2010). In this thesis, we will build a model for assessing ecological services of forest, namely carbon sequestration.

2.3 Ecosystem Service Assessment

Ecosystem services can be assessed in various ways (Millennium Ecosystem Assessment, 2003), as shown in Figure 1.

It is not always possible to measure the value of an ES as a value of a conventional service or good. However, if a service yields something that can be transformed into a service or good, the utilitarian approach is used. The Total Economic Value Framework (TEV) (Costanza et al., 1997a) is an example of a utilitarian assessment. TEV disaggregates the values of an ES into use and non-use values. Use values mean the ability of the ecosystem to provide services that can be used for production or construction. Non-use values are the value of the ecosystem's services under the

conservation scenario, which is a situation when the existence of a resource is known to a society but its members abstain from its exploitation.

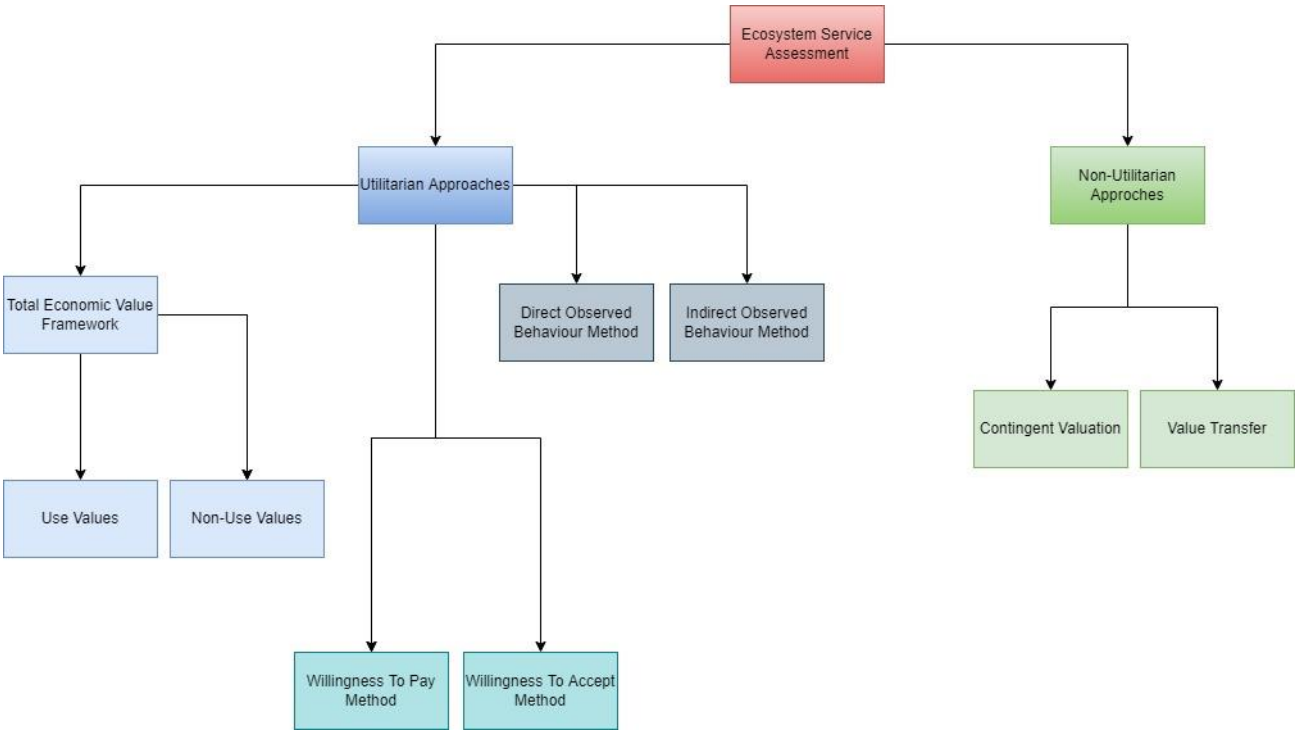


Figure 1 Methods of Ecosystem Service Assessment (adopted from: the Millennium Ecosystem Assessment, 2003).

Willingness to Pay (WTP) and Willingness to Accept (WTA) are two other approaches and indicators of society's welfare. WTP is applied when the community does not own the resource and is more often used when the volume of the ES increases. WTA, on the other hand, is used when the community controls the resource but has to accept its declining rates of production. WTP estimates are utilized more frequently because they provide a more conservative evaluation, as they are usually lower than WTA estimates.

When the researchers have a chance to observe the dynamics of ecosystem behavior, they can use two types of methods. Direct observed behavior methods measure the value of ES from the behavior of consumers by applying market prices. These methods are suitable for privately-

owned, market-traded resources. Indirect observed behavior methods apply surrogate methods to derive ES values such as hedonic pricing methods, travel cost methods or replacement costs methods.

Non-utilitarian values are applied to cultural ES. Since cultural ES are important for a community's identity and heritage, the evaluation methods often use group valuations, the result of which is a Contingent Valuation (CV) or benefit transfer. Another approach is rooted in the notion of intrinsic value in a sense that the nature was creates by a supreme being, such as God in Abrahamic religions or Forces of Nature in North American Indigenous tradition. The non-utilitarian approach also involves assessing social, economic, and ecological consequences of ES usage to a community.

In this study, we apply a utilitarian approach whereby the value of Carbon Sequestration ES is expressed in the corresponding market price and a detailed consideration of the method is given further in Chapter 3.

2.4 Sustainable Management and its Application to Forest Ecosystems

The UN Food and Agriculture Administration (UN FAO, 2023) describes sustainable forest management as "the process of planning and implementing practices for the stewardship and use of forests to meet specific environmental, economic, social, and cultural objectives. It deals with the administrative, economic, legal, social, technical, and scientific aspects of managing natural and planted forests". Although the notion of sustainability has been assigned different meanings (Kumar et al., 2019), the bottom line is that sustainability is the preservation of the ecosystems in their optimum state, so that they are able to withstand stresses. Sustainable management takes into account possible consequences of the decisions being made to the three pillars of SDG: the economy, the society and the environment (UNGA, 2005). A methodology linking together the

three pillars of sustainability is rooted in the concept of ecological-economic-social systems (EESS) and ecosystem Services they produce (Khaiter 1986, 1993a, 1996, 2005; Gorstko and Khaiter 1991). An idea of managing the economic and societal development of forested territories with a view of environmental constraints had been suggested and developed since the late 1980s (Khaiter 1989, 1990, 1991, 1993b, 2005; Khaiter and Erechtchoukova 2012, 2013, 2017, 2020).

According to Khaiter & Erechtchoukova (2022), sustainable forest management is crucial as the forest is an ecological-economic-social system (EESS) and the fulfillment of the UN Sustainable Development Goals (SDGs) relies upon its preservation. Being an EESS, the forest provides ecosystem services that can be beneficial to all three pillars of SDG. The authors also indicate an important role of Information Technology including Geographic Information Systems and Data Science in practical Sustainable Management.

2.5 NASA's Earth Observing System

Earth Observation System (EOS) is a system of satellites that orbit the Earth and record data about agricultural and water resources, the atmosphere, biodiversity, and wildfires (NASA's Earth Observation System, 2022). The System has nearly 30 active missions with different types of satellites carrying various types of sensors. For example, the Aqua gathers information about the planet's water resources, the Aura observes atmosphere's chemistry and dynamics, the CALIPSO collects data on cloud and aerosol interactions, LANDSAT 7, LANDSAT 8, and LANDSAT 9 provide images of Earth surface, including thermo-infrared imagery, the OCO-3 gathers data on atmospheric carbon dioxide, and SMAP measures the soil moisture and freeze-thaw conditions. EOS plays an important role in observing the state and dynamics of ecosystems as well as assessing their services. EOS can effectively solve the problem of access into remote areas and data provisioning (Ramirez-Reyes et al., 2019).

Satellites carry several types of sensors, the RADAR (Radio Detection and Ranging) and LIDAR (Light Imagery Detection and Ranging) being the most popular ones. In these sensors, the pulse sent from the spacecraft is reflected from the surface while the sensor picks it up. The data is recorded in a form of bands which capture various parts of the light spectrum. According to ESRI (2023), a band is "a wavelength range in the spectrum of reflected or radiated electromagnetic (EM) energy to which a remote sensor is sensitive. The file or portion of the file devoted to that range is also commonly referred to as a band." The imagery produced by the sensors can be high-resolution or low-resolution. In Remote Sensing, resolution means "the dimensions represented by each cell or pixel in a raster" (ESRI, 2023). High resolution data means that one pixel of the image covers a small area on the ground, and the larger that area is the lower the image resolution is.

Indexes are specific indicators that are calculated from bands. The most widely used indices include the Normalized Differentiated Vegetation Index (NDVI) calculated from a combination of the near-infrared and infrared bands, the Normalized Difference Moisture Index (NDMI) and Normalized Burned Ratio Index (NBRI) calculated from shortwave infrared and near-infrared light in different combinations, and some other indices.

While indices and imagery are an appealing material to work with, there are specific issues like with any other types of data. For example, they contain noise from reflectance and interference from clouds. Alvarez et al. (2017) investigated algorithms for LANDSAT images that eliminate the influence of clouds, specifically the Automatic Cloud Removal Method (ACRM) for the regions with high elevation, prone to an increased number of clouds. It is important to note that areas with extensive cloud coverage often also have dense forest presence. In order to observe the forest ecosystems in such locations, it is vital to eliminate the impact of clouds to increase the

accuracy of Normalized Difference Vegetation Index (NDVI) or other indices. In their study, Alvarez et al. (2017) used images acquired by the two types of LANDSAT sensors - Land Imager (OLI) and Thermal Infrared Sensor (TIR).

There is an important issue of fitness for use in working remotely-sensed data. According to ESRI GIS Dictionary (2023), fitness for use is "the degree to which a dataset is suitable for a particular application or purpose, encompassing factors such as data quality, scale, interoperability, cost, data format, and so on." The researchers must take all these factors into consideration before initiating projects. Additionally, for policy- and decision-makers, the cost of data will be an significant factor in planning the projects on Ecosystem Service Assessment (ESA).

2.6 Geographic Information Systems, their definition and applications

Data acquired from remote sensing technologies cannot be used in its initial state and requires manipulations. Traditionally, Geographic Information Systems (GIS) are used to perform them. Savvaidis & Stergioudis (2012) describe the history and applications of GIS which they determine as computer systems used to collect and analyze geospatial imagery. The authors state that the first GIS systems were created for conducting research in forest science and were desktop applications. The modern GIS applications can solve a variety of scientific problems by using 3 or 4 dimensions, including time and come in a variety of forms. The most famous and widely-used GIS software include ArcGIS, QGIS and Google Earth. ArcGIS is a software developed by ESRI, Google Earth is a product of Google and QGIS is an open-source, volunteer-maintained project.

Grigolato et al. (2017) further explore the subject of GIS applications, specifically on forest operations. They conducted a literature review which showed that GIS has been used for road engineering and management, harvesting, supply chain management, *etc*. GIS operate on Digital Elevation cartographic, and spatiotemporal models. The types of image data include raster and

pixel. In the early period of development, GIS were used to compute the distances of the forest road network or slope based on raster or vector analysis. Later, GIS were applied for decision support based on mathematical models. Nowadays, more research conducted with GIS involves 3D modeling for forest engineering operations.

One of the applications of GIS is in support of the assessment of ecosystems services. Baniya et al. (2019) demonstrate the role of NDVI in assessing ecosystem services in Nepal. Medium Resolution Imaging Spectroradiometer (MODIS) data for over seven-year period was used in the study along with several NDVI products. GIS was applied to preprocess imagery and calculate the index. After respective calculations, Ecosystem Service Value (ESV) was computed using the value transfer approach. The total economic value of ES for forests amounted to several billion dollars and showed a significant increase over the observed period. The authors observe that the increase in economic output may be explained by conservation programs. Moreover, the increasing amount of atmospheric CO2 also causes several forest ESVs to grow, particularly, the volume of carbon sequestration. Said trend calls for an instrument or methodology to accurately assess this forest ecosystem service.

Remote Sensing and GIS were also used for assessing the impact of anthropogenic factors on the underlying ecosystems. Mishra & Mainali (2017) conducted a time-series analysis to identify trends in carbon sequestration productivity in the Himalayas. The researchers note the lack of such studies for mountainous areas. The data for their study was obtained from MODIS NDVI series collected over 2000-2016. Using GIS, the authors applied the Savitzky-Golay filter to isolate pixels that matched the criteria of topographic elevation and mean-NDVI. The majority of pixels demonstrated an increase in at least one of the NDVI-mean or NDVI-max parameter which, among other effects, signifies an ongoing afforestation process.

2.7 Machine Learning and its Role in ESA

While EOS can provide data for research and GIS can assist in its preprocessing, the studies mentioned above demonstrate that GIS alone is insufficient to successfully answer the questions posed in this research and that additional theoretical concepts and computational techniques are required.

According to Mitchell (1997), machine learning (ML) is a name for a group of algorithms that have the ability to improve their own performance in the course of running, which is called "learning". Machine learning has several applications, such as in Artificial Intelligence, Optimization, and Statistics. Machine Learning algorithms come in a variety of forms and can be grouped together to form Neural Networks that are able to make even better predictions.

One of the most popular ML algorithms used to monitor forest state and solve problems related to ecosystem service assessment is the Random Forest algorithm. Random Forest (RF) was described by Breiman (2001). It is a supervised machine learning algorithm, which means that it requires the data to be split into a training and testing samples. The algorithm is part of the so-called Ensemble Learning algorithms because it incorporates several Decision Trees and uses randomized node optimization and bagging. The measure of performance for this algorithm is out-of-bag error and the measure of variable importance is permutation. Using several trees together allows the algorithm to avoid overfitting and produce more accurate results, while using several bagging techniques on training allows it to derive predictions even from a limited amount of data. Random Forest uses the bootstrap sample of the training data meaning that the data is sampled at each node of the tree, and the best split is made between training and testing samples until the tree reaches its largest possible size. The final prediction is a vote or an average from all trees in the forest. (Breiman, 2001).

Another frequently used ML algorithm is Extreme Gradient Boosting (XGBoost) suggested by Chen & Guestrin (2016). XGBoost is a supervised algorithm that operates based on the Newton-Raphson search procedure with a second order Taylor approximation in its loss function. The impressive number of hyperparameters allows the algorithm to generate very accurate predictions through setting tree-growing and pruning rules. The goal of the algorithm is to turn a set of weak learners into a single strong learner, which will produce the predictions. We further discuss some of the studies that applied these algorithms.

2.7.1 Machine Learning in the Context of Remotely Sensed Data

Freeman et al. (2016) tested the sensitivity of random forests (RF) and stochastic gradient boosting (SGB) to the tuning parameters and also assessed their performance on data acquired from aerial photography. The authors emphasize the importance of tuning and specifically constructing a robust tuning dataset. SGB is a boosting ML technique proposed by Friedman (2002) that incorporates several weak-learner decision trees just like Random Forest and operates as a gradient descent. However, because of the iterative learning process which includes gradual minimization of the Mean Squared Error, it often outperforms the RF. In the course of research, the accuracy of each model was assessed through MSE, Pearson and Spearman correlation coefficients.

Data acquired from remote sensing is very noisy, which can negatively impact the accuracy of predictions. The high number of hyperparameters can drastically improve the performance of the algorithms but can also become a nuisance to select. Fortunately, various heuristic methods can assist in model tuning, and the choice of the most powerful one should become the focus of future research. It is unadvisable to use the same data for testing and validation. Data-wise, this

means that research involving ML algorithms requires sufficient enough data samples and appropriate research designs.

Sandino et al. (2018) propose a framework to assess forest health and detect patterns in its deterioration due to pathogen involvement. They selected data from unmanned aerial vehicles (UAVs) and hyperspectral image sensors. The predictor variables included the normalised difference vegetation index (NDVI), the green normalized difference vegetation index (GNDVI), the soil-adjusted vegetation index (SAVI), and the second modified adjusted vegetation index (MSAVI2). The data was split into training and testing samples in an 80/20 proportion. The algorithm used for the study was XGBoost (XGB) implemented through Python Scikit-Learn library. The hyperparameters of the model, namely estimators, learning rate and maximum depth, were selected by applying the grid search optimization technique. The model was cross-checked with the k-fold cross-validation. Overall, the success of the model was 97.35%.

This research demonstrates successful use of tuning for problems related to Remotely Sensed data. However, grid search is a heuristic procedure, and it is not always fit for finding the global optimum of the objective function. Instead, metaheuristic techniques can be tested in the future studies.

2.7.2 Machine Learning in the Context of GPP Estimation and its Economic Assessment

Sarkar et al. (2022) propose a framework for assessing GPP from RSD, meteorological and topographical data with the help of Random Forest Regression (RFR) and XGBoost Regression. Some predictor variables such as Land Surface Water Index (LSWI), Enhanced Vegetation Index (EVI), Leaf Area Index (LAI) and other parameters related to forest health were derived from MODIS sensor on board of NASA's Terra satellite. However, the target variable (i.e., GPP) was derived from the FLUXNET Eddy Covariance observations. During the preprocessing, all data

was converted to an 8-day average. RFR outperformed XGBoost with Pearson R coefficient of 0.82 as compared to 0.8.

FLUXNET are probes positioned on the ground in selected locations. As was mentioned above, the lack of accessibility in certain areas may prevent the placing of probes, thus making studies involving collection of the on-the-ground data impossible. This gap is being addressed by proposing an approach that relies exclusively on RSD.

Another obvious limitation of Sarkar et al. (2022) is the averaging of data to form one observation over an 8-day interval. This approach will not yield predictions sensitive to daily changes in the target variable and may miss the point of sudden bifurcation.

Zhu et al. (2023) investigate the prediction of Annual Gross Primary Productivity (AGPP) of forests in China. The target variable was also derived from FLUXNET. Four machine learning algorithms were compared: the BP Artificial Neural Network (BPANN), Support Vector Machine (SVM), Random Forest Regression (RFR) and the Boosted Regression Tree (RFT). The predictions were performed on four distinct sets of variables. Just like reported by Sarkar et al. (2022), RFR performed better than other candidates with a Pearson R coefficient of 0.85. The authors highlight the management implications of their research, such as assessing GPP in various regions of China and achieving carbon neutrality. The research design may have caused bias in results as the majority of FLUX towers were located only in the Eastern China. Moreover, the Annual GPP, while suitable for ESA in principle, cannot illustrate the full dynamics in its values that occur throughout the year.

Bandopadhyay et al. (2021) propose a machine learning-based approach that uses MODIS-derived GPP to make predictions. They compare the GPP of the Indian forests before and after the monsoon seasons. Three algorithms applied are: RFR and its variants - conditional inference trees

and quantile regression forest. Random Forest also performed better than its counterparts. A significant drawback of the research design is that it completely ignores the forest conditions during the monsoons. As a result, the decision-maker are unable to estimate the impact of the monsoon seasons and damage it could inflict on the forest ecosystems and related decrease of GPP.

2.8 Bibliometric Analysis

Bibliometrics is a research methodology that uses quantitative analysis and descriptive statistics to survey authorship, citation, and publication patterns in the existing body of knowledge on a particular topic.

Bibliometrics has been used in various cases to analyze publications on ecosystem service assessment. For example, Wang et al. (2009) analyzed publications in all editions of the Water Research journal between 1967 and 2008. Another water-related bibliometric work is Zare et al. (2017) where authors explored publications on integrated water assessment and modelling. Li & Zhao (2014) observed global environmental assessment research patterns. Xu et al. (2023) reviewed the development of carbon accounting systems and their role in tracking production and consumption.

2.8.1 Approach

The bibliometric study may assist in identifying trends in using data analytics to predict carbon sequestration by forest ecosystems from remote sensing data, as well as showcasing the gaps in the domain. Figure 2 outlines the overall flow of the analysis.

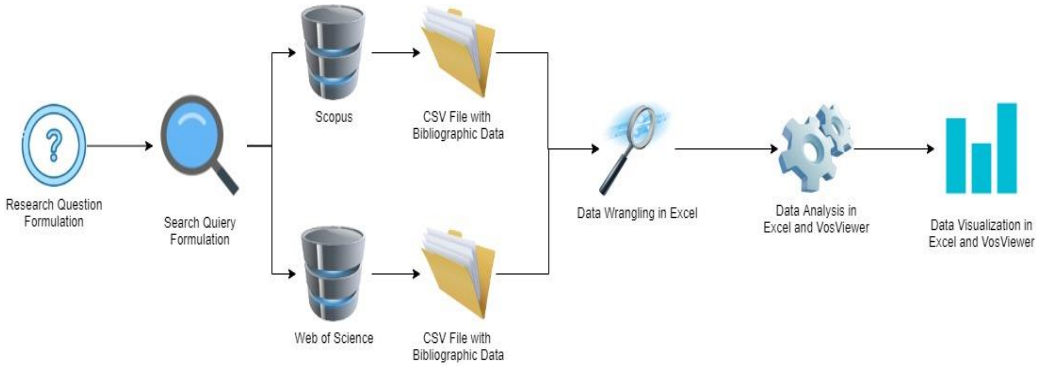


Figure 2 Workflow of Bibliometric Analysis

2.8.2 Roadmap of the Bibliometric Analysis

Defining the questions of interest is the first step in the analysis. They are presented in Table 1 along with their corresponding methods.

Table 1 Bibliometric Questions and Methods

Number	**Research Questions for the Bibliometric Analysis**	**Bibliometric Method**
BRQ1	How does the number of publications on the topic change from year to year?	Year-by-year trend of publishing
BRQ2	Which research areas pertain to the topic?	Most frequently used keywords for area of study
BRQ3	What are the most used keywords and how are they connected?	The network of keywords
BRQ4	What are the most influential publications?	The publications with most citations
BRQ5	Who are the most influential authors?	Summary of authors with most articles
BRQ6	Which journals usually publish relevant research?	Summary of most frequent journals

The second step in the analysis involves the selection of resources and formulation of an adequate search query. We used the Scopus and the Web of Science (WOS) core collections to export the complete bibliographic records as a tab-delimited text file. The search criterium created through the Advanced Search feature was:

((TS = ("Gross Primary Productivity" and "Ecosystem Service" and "remote sensing" or "Ecosystem Service Assessments")) AND DT = (Article OR Review)) AND LA=(English).

The TS field implemented an advanced search of the provided keywords in the title, the topic, the keywords and the abstract sections of each publication. The DT field specified the type of the publication and the LA field specified the language of the publication. The AND, OR logic operators provided the inclusion or exclusion of the terms. The total number of sources thus yielded was 980. The annual distribution of publications is presented in Figure 3.

The data wrangling was performed using the MS Excel and VOSViewer software (Van Eck & Waltman, 2007).

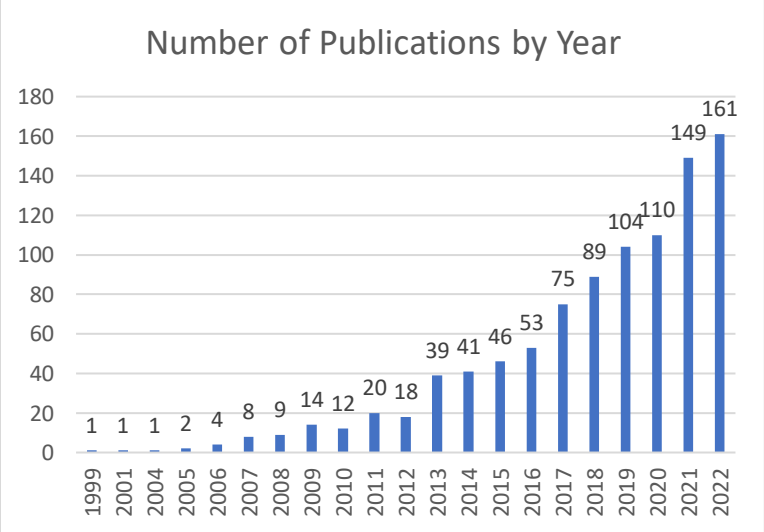


Figure 3 Annual Number of Publications in Research Domain

Now we can answer questions formulated in Table 1, starting with BRQ1. It is evident from Figure 3 that the scientific community first showed interest towards predicting GPP from satellite

data in 1999. Since then, the number of publications steadily grew, most likely due to increase in the availability of RSD and introduction of new data analytic techniques. In the previous four years (2019-2022), the number of publications in a single year ranged from 104 to 161.

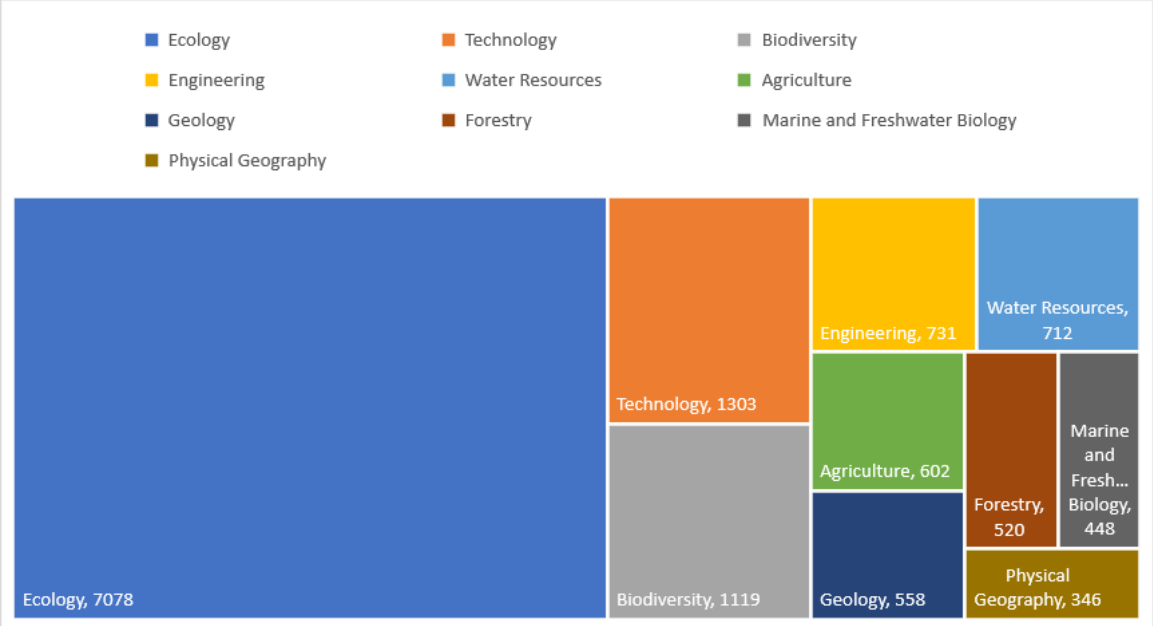


Figure 4 The Most Popular Research Areas for Research Involving RSD

In answering BRQ2, the 10 most important research areas are presented in Figure 4, where the size of a rectangle corresponds to the number of publications attributed to the study area. Among the areas, some are related to nature, ecology, agriculture, and their various applications as well as to engineering and technology. Ecology, Technology and Biodiversity seem to be the areas where data analytic methods are most frequently used for predictions.

To get an answer to BRQ3, we need to analyze Figures 5 and 6.

Figure 5 presents the network of keywords parsed from the bibliometric records exported from Scopus and WOS. The VOSViewer determined 10 distinct clusters of which five clusters seem to be of a major importance. The “Blue” cluster mostly concentrates on Gross Primary Productivity and the related parameters in accordance with the Light Use Efficiency theory (Monteith 1972).

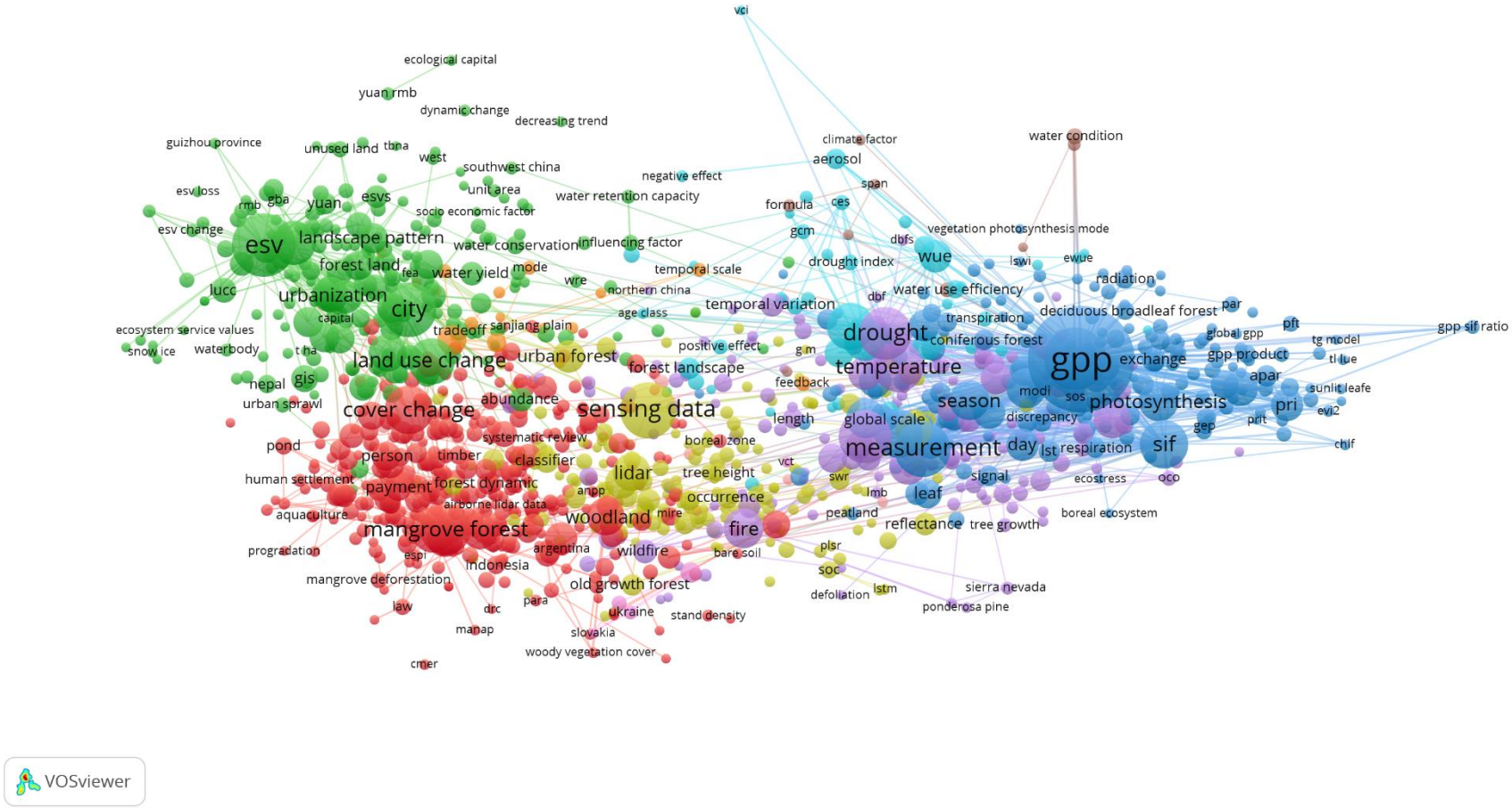


Figure 5 The Network of Publication Keywords

The “Purple” cluster represents the factors that affect GPP such as seasonality, temperature, drought, *etc*. The “Yellow” cluster mostly focuses on the Remote Sensing aspect of the studies with keywords like “LIDAR” and “reflection”, however, it also includes such keywords as “classifier” and “LSTM” (Long Short-Term Memory models). The “Red” cluster emphasizes the forest, its various types and aspects like afforestation. Finally, the “Green” cluster accentuates the Ecosystem Service Value (ESV) and its relation to different types of settings, such as urban areas, waterbodies and so on.

The presence of certain data analytic terms in the “Yellow” cluster indicates that some techniques of machine learning have already been used, however other characteristics of the clusters are to be regarded. In the network theory, the distance between individual clusters is an important indicator as the shorter the distance is, the more closely the clusters are related and the more links they have. The “Green”, “Red” and “Yellow” clusters are positioned closely to each other but quite far away from the “Blue” and the “Purple” clusters. This may indicate the lack of studies in which data analysis and RSD were jointly used for prediction of GPP and related parameters.

Figure 6 shows the same network but coloured in accordance with the time trends of the keywords. The “Blue” and “Purple” clusters have GPP and its measurements among the oldest trends, however in the latest years, the relation between GPP and SIF (Solar-Induced Chlorophyl Fluorescence), estimation of drought indices, as well as calculation of these parameters in the presence of atmosphere aerosols are gaining more attention of the researchers.

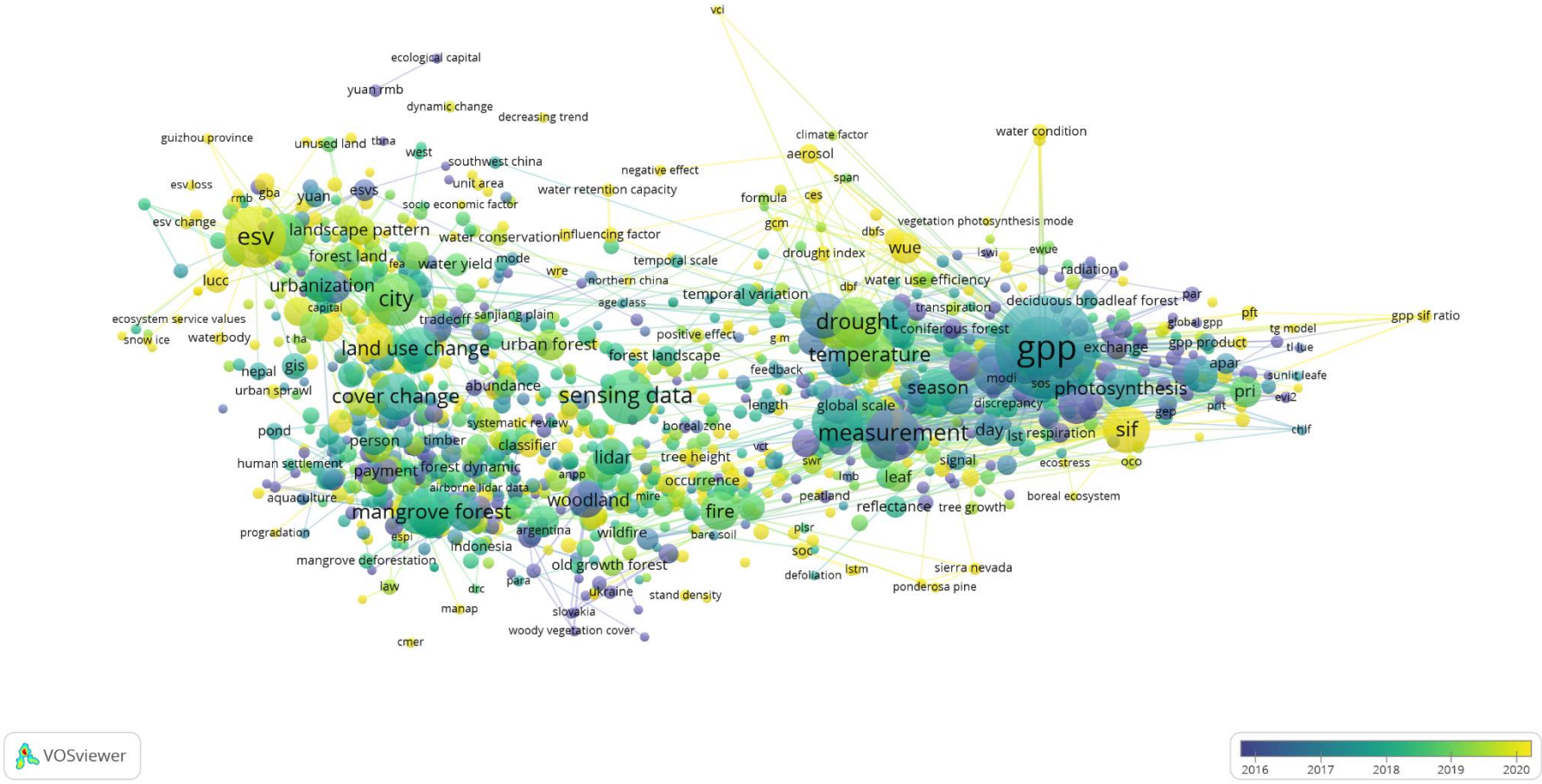


Figure 6 The Network of Publication Keywords in Overlay Projection

Table 2 Ten Most Cited Publications

Title	Authors	Year	Source Title	Citation Count
Considering landscape-level processes in ecosystem service assessments	Metzger et al.	2021	Science Of the Total Environment	796
Reimagining the potential of Earth observations for ecosystem service assessments	Ramirez-Reyes et al.	2019	Science Of the Total Environment	665
Assessing top- and subsoil organic carbon stocks of Low-Input High-Diversity systems using soil and vegetation characteristics	Ottoy et al.	2017	Science Of the Total Environment	589
Rates and drivers of mangrove deforestation in Southeast Asia, 2000-2012	Richards & Friess	2016	Proceedings Of the National Academy of Sciences of the United States of America	567
Creation of a high spatio-temporal resolution global database of continuous mangrove forest cover for the 21st century (CGMFC-21)	Hamilton & Casey	2016	Global Ecology and Biogeography	527
The structure, distribution, and biomass of the world's forests	Pan et al.	2013	Annual Review of Ecology, Evolution, and Systematics	490
Near-surface remote sensing of spatial and temporal variation in canopy phenology	Richardson et al.	2009	Ecological Applications	379
Drought impacts on ecosystem functions of the US National Forests and Grasslands: Part I evaluation of a water and carbon balance model	Sun et al.	2015	Forest Ecology and Management	353
Characterizing landscape pattern and ecosystem service value changes for urbanization impacts at an eco-regional scale	Su et al.	2012	Applied Geography	294
Urban warming reduces aboveground carbon storage	Meineke et al	2016	Proceedings Of the Royal Society B-Biological Sciences	283

In the “Yellow” cluster, the terms “classifier” and “LSTM” are the newest additions to the cluster, which can be explained by the latest technological advancements in the area. In the “Green” cluster, the keyword “ESV” has appeared relatively recently, however the latest additions are the terms that show its connection to snow, ice and efficiency of ESV per unit of area.

Table 2 presents ten most cited articles in the sample sorted by citation count (BRQ4). The oldest article dates back 2009 and the most recent was published in 2021. The topics covered by the articles range from Ecosystem Service Assessment to estimation of carbon storage and vegetation characteristics. Most of the journals focus on some aspect of Environmental Sciences.

The authors and the journals with the highest publication numbers (BRQ5 and BRQ6) are presented in Figure 7 and 8, respectively. The most prolific author has written seven articles. It is important to note that the total amount of individual authors amounted nearly 6000 people and the majority of articles have multiple authorship. The Journal of Remote Sensing has published 99 articles, which is nearly three times more than its closest contestant, the Journal of Remote Sensing of the Environment. Among other outlets, the majority of publications is dedicated to the environment rather than remote sensing.

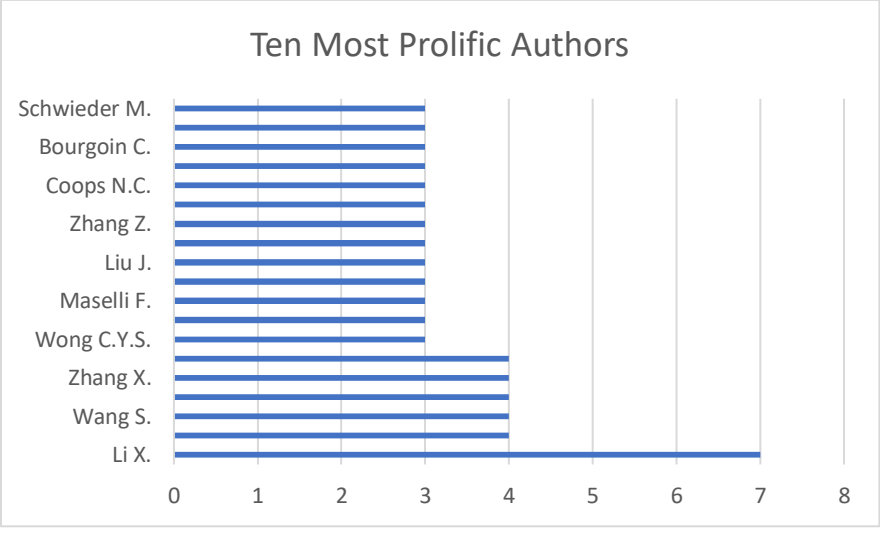


Figure 7 Ten Most Prolific Authors

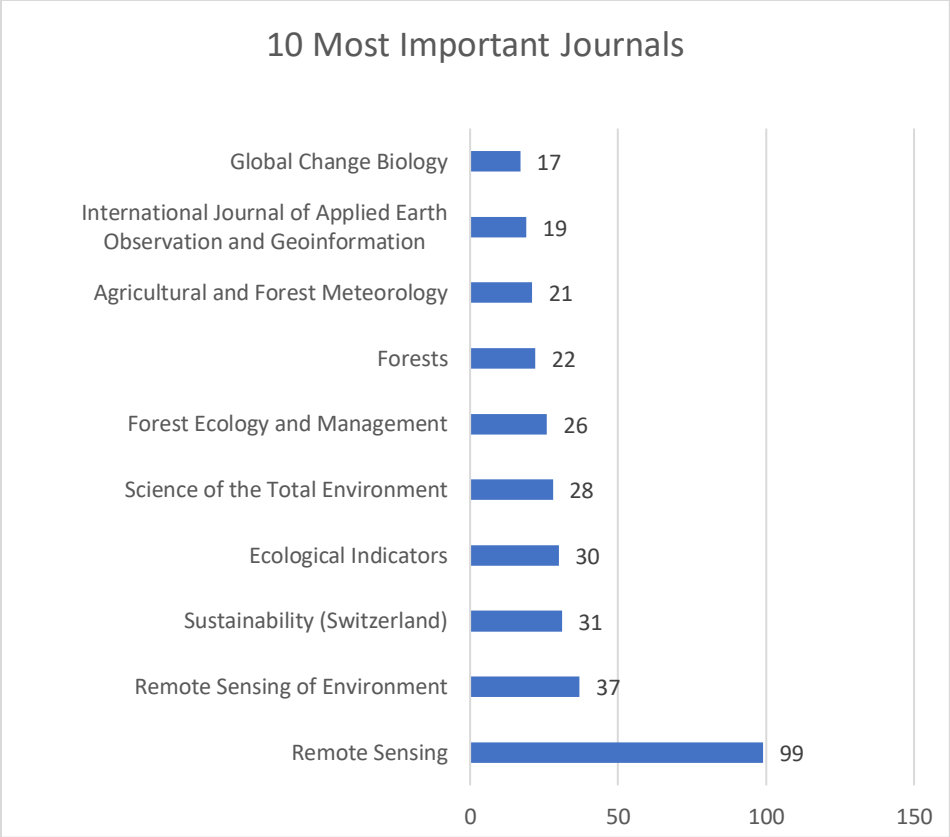


Figure 8 Ten Most Significant Journals

Conducted bibliometric analysis shows that the research field is growing rapidly. A significant number of studies is dedicated to the subject of ESA through RSD in various contexts. The network charts of the keywords (Figures 5 and 6) demonstrate the lack of studies closely linking RSD to Gross Primary Productivity through the application of Machine Learning techniques.

2.9 Research Gaps

Our literature review (subsections 2.1-2.7) has revealed several important gaps in the body of research. The summary of these gaps and the number of corresponding articles are presented in the Table 3. The results of Bibliometric Analysis, especially Cluster Analysis, confirm their relevance.

Table 3 Summary of Literature Gaps

Literature Gap	**Number of Publications Indicating the Gap**
Requires development of methodology to adequately assess ESV of carbon sequestration	1
Does not perform data preprocessing according to quality standards	1
Requires application of data imputation techniques to make up for sparce observations	2
Requires development of methodology that uses only remotely-sensed data for assessing the value of carbon sequestration	3
Uses the Random Forest algorithm, omitting XGBoost and requiring the validation or refutation of RF's superior performance	4
Calls for adoption of metaheuristic algorithms for model tuning	1

This chapter contains a review of literature and a bibliometric analysis on using RSD in the context of ESA. The insights obtained provide ample evidence of the absence of any study which would combine the use of remotely-sensed data with adequate data preprocessing practices and apply the full scope of the modern ensemble machine learning techniques. In the subsequent chapter, we propose a methodology bridging all of the aforementioned gaps within a common framework.

Chapter 3: Research Methodology

The objective of the research is to propose a methodology for working with Remote Sensing Data (RSD) and using it for forest ecosystem service assessment. However, the way of processing RSD differs from standard procedures of working with business or other types of datasets. There are certain issues stemming from the technical aspects of the sensors, the satellite orbit and speed of rotation, the image resolution and other factors that must be taken into account and incorporated into the developed framework. Furthermore, fitness for use problem needs to be considered as well.

Khaiter & Erechtchoukova (2022) indicated that a methodology for working with remotely sensed data should include data acquisition, data preprocessing, data analysis, and, depending on the goal of the project, modeling or optimization, and decision-making. In the previous chapter, we have outlined several gaps in the research literature which serve as requirements for the suggested methodology.

This chapter also presents the region of study, the motivation behind the choice of research variables and the sources of data, the processes of data collection, preprocessing and exploration as well as experimental design.

3.1 Region of Study

The region of interest for the study is the province of Jammu and Kashmir in India. Jammu and Kashmir is a former state located in the North-West of the country. It borders on the Himalayas range in the North, which is a significant factor in its climate.

The Himalayas are the Earth's highest mountain range and pose unique challenges for researchers. Opportunities of *in situ* observations remain limited due to altitudes and climactic conditions, but the state of forest ecosystems must, nevertheless, be monitored, making RSD the

only option of data collection. An obvious benefit of adopting Jammu & Kashmir region in the Himalayas as the study area is rather the universal nature of the developed methodology suitable for geographical locations with similar conditions.

Figure 9 The Map of India showing the location of Jammu and Kashmir (JK)

The main meteorological event in India is the Monsoon, a pattern of seasonal winds that bring rainclouds from the Arabian Sea or the Bay of Bengal. The Monsoon is divided into two phases – the Summer Monsoon that blows from the South-West between May and September and the Winter Monsoon blowing from the North-East from October to November (Ramage, 1971). Because the Valley in located between several branches of the Himalayan Mountains, it is much affected by the event whereby the Himalayas act as an orographic barrier to the clouds (Stull, 2017). An orographic barrier is when the high mountains cause the clouds to rise, cool down and

produce rain. Moreover, before the onset of the Monsoon, as the temperature begins to rise in March, the snow in the mountains starts to melt. The water rushes down the slopes and joins the rivers.

The Himalayas also house many forests, predominantly of the Himalayan Subtropical Pine variety. The most wide-spread species of tree in the forest is the longleaf Indian pine *(Pinus roxburghii)* shown in Figure 10. This species is mostly present in the Asian Temperate and the Asian Tropical climates and is used for its timber and resin. There are also references to this species in the folklore (USDA, Agricultural Research Service, National Plant Germplasm System, 2023).

Figure 10 Longleaf Indian Pine (*Pinus roxburghii*) on the mountain slope

3.1.1 Dachigam National Park

Dachigam National Park is located near the city of Srinagar in the Kashmir Valley. It is situated on the slopes of the Himalayas with the altitude varying from 1600 to 4200 meters above the sea level. The predominant type of land cover is a coniferous forest. The Park is an environmentally protected area open for visitation, however, no settlements are permitted on its territory. It offers recreational and cultural services to the dwellers of the near-by Srinagar that has a population of two million people as well as provides habitat for many species of animals including the endangered leopard cat and the vulnerable Black Himalayan bear. The coordinates of the rectangle encompassing the park are: 74.898, 34.151; 75.076, 34.152; 75.086, 34.063; 74.868, 34.081. The location of the park in the Valley is shown in Figure 11.

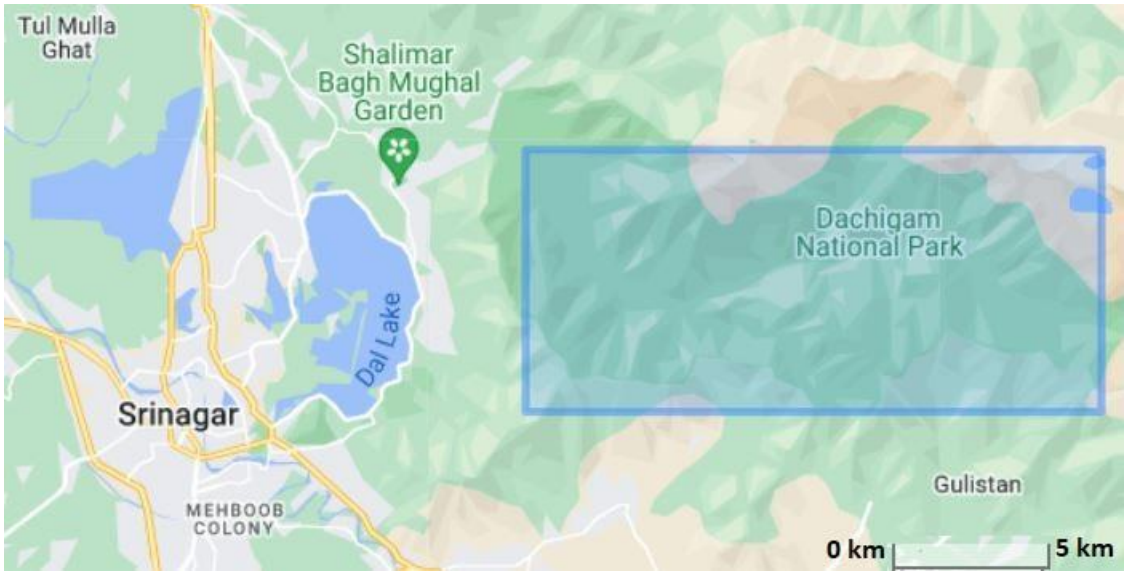


Figure 11 Location of the Dachigam National Park in the Kashmir Valley, the grey rectangle denoting the area where the data was gathered

3.1.2 Khrew, Jammu & Kashmir

Khrew is a small town in the Pulwama district of Jammu & Kashmir, located 21 km away from Srinagar and Dachigam National Park. It has an average elevation of 1600 meters above the sea level and is encompassed by two mountain spurs that are also covered by forest. The town is

famous as one of the major cement and limestone production hubs in Jammu & Kashmir. The mountainous area is rich in minerals, such as Limestone, Borax, Coal, Granite, Marble and certain gemstones (Government of Jammu and Kashmir, 2023). The coordinates of the investigated region are 74.995, 34.073; 75.059, 34.069; 75.069, 33.977; 74.986, 33.984. Figure 12 demonstrates the location of the town in the Valley.

Figure 12 Location of the Town of Khrew in the Kashmir Valley, the grey rectangle denoting the area where the data was gathered

3.2 Data Sources

3.2.1 Choice of Study Variables and their Sources – a Motivation

One of the measures of the ecosystem capacity to store carbon is the Gross Primary Productivity (GPP) (Beer et al., 2010). It is an essential component of the Light Use Efficiency Theory and its models. The concept of Light Use Efficiency (LUE) is defined as the ability of plants to acquire the light and successfully use it for photosynthesis; the LUE models estimate

GPP as the energy absorbed by plants multiplied by the actual LUE that converts energy to carbon fixed during the photosynthesis process (Monteith 1972).:

$$LUE = \frac{GPP}{PAR \times FPAR}, \qquad where \tag{3.1}$$

FPAR is Fraction of the Absorbed Photosynthetically Active Radiation; PAR is Photosynthetically Active Radiation and PAR = 0.48 x SW (Short-Wave Radiation).

Pei et al. (2022) published an excellent review on the evolution of the LUE models analyzing 23 articles from 1995 to 2019. According to the review, several models of LUE based on satellite data have been proposed estimating the amount of incoming solar radiation (e.g., total, direct or diffuse Photosynthetically Active Radiation (PAR)) and its absorption, namely the Fraction of Absorbed PAR (FPAR) by the canopy, leaves or chlorophyl. Other components of the models include vegetation characteristics, air temperature and parameters related to water environmental stresses (e.g., draughts, heavy rain or floods).

The authors dwell on the question of model uncertainties in relation to data. Firstly, they state that the accuracy of estimated LUE depends on chosen vegetation variables. While different models use different vegetation indices, we think it is reasonable to step away from a single model and collect the vegetation data in overabundance to ensure that the most accurate state of vegetation is considered.

Secondly, the review points out the issue of data quality and scale mismatch. To ensure that no missing data obscures the result, validation procedures must be adopted. Proper techniques of Remote Sensing Imagery Fusion can assist in bringing the data to a common scale.

Finally, despite the satellite data becoming more available for research purposes, some components of the LUE framework are still often derived from the ground, using GPS, FLUXNET towers or *in situ* measurements. As we have previously mentioned, accessibility issues may

prevent the researchers from GPS scanning or placing the probes, thus making RSD the only option.

Considering all these issues, we have chosen the following variables for the project as described in Table 4.

A *data product* is a piece of data available for open use that is owned and maintained by its creator. The majority of data products in this research contain observations from the Moderate Resolution Imaging Spectroradiometer (MODIS) that belongs to the group of sensors located on the NASA's Terra satellite which is a part of the NASA's Earth Observing System (EOS). In each of the products, the observations from a certain time stretch has been transformed into an average estimate. The spatial resolution ranges from 250 to 500 meters per pixel which corresponds to high resolution imagery.

One dataset in the necessary for this study has been obtained from the NASA's Prediction of Worldwide Energy Resources (POWER) project (POWER, 2023). This resource provides near-real-time climactic variables, such as solar radiation, precipitation and air temperature. The export of data is performed through an online instrument as a tab-delimited file. It is important to note a large-scale nature of the exported data with the grid size of 25 by 25 km which can be a challenge, especially when predicting GPP for smaller areas. On the other hand, daily observations of climactic variables are available.

The study is of the longitudinal nature, sampling data from January 1st, 2014 to December 31st, 2018. This timeframe has been chosen for reasons of data availability.

3.2.2 Data on Flood Events

For the purpose of Ecosystem Service Assessment, we required data on flood events in the regions of study. The Monsoon and the melting of snow often result in floods which is the most

devastating natural disaster for the Valley's economy and ecosystems. The floods cause the disruption of economic infrastructure, deaths of people and naturally detriment the forests growing on the slopes of the Valley by contributing to soil erosion. The website of the Indian Central Water Commission (CWC, 2023) lists three water level monitoring stations in the Valley (see Figure 13). One of them is located in Rammunshibagh near the city of Srinagar, close to the National Park (number 2 in Figure 13) and the other are in Safapora (number 1 in Figure 13) and in Sangam (number 3 in Figure 13). This station monitors water level in the Jhelum River which takes its origin in the Himalayas, on the South-Eastern side of the Valley and flows through it.

Over the period 2014-2018, several major flood events hit the Kashmir Valley. According to Floodlist.com, a news aggregator focused on flood events around the world, the deadliest flood of the recent times occurred on September 2nd, 2014 (Davies, 2014), resulting in the death of 200 people. Just several months after that, in March 2015, a new flood struck the region (Floodlist News Desk, 2015) bringing damage to nearly 13,000 objects of infrastructure. The disaster occurred again in July of the same year (Davies, 2015). In April 2017, the Jhelum River has overflown due to 80 mm of torrential rains and avalanches in the mountains (Davies, 2017).

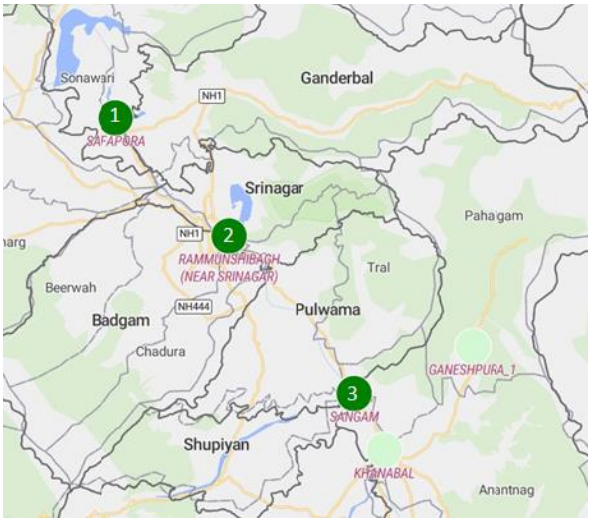


Figure 13 The Map of Water-Monitoring Stations in the Kashmir Valley (1 – Safapora station, 2 – Rammunshibagh station, 3 – Sangam station)

Table 4 Variables Chosen for the Project

Feature	**Index**	**Properties**	**Source**
Gross Primary Productivity	GPP	cumulative **8-day** composite with a **500-meter** resolution	NASA's MODIS 006 MOD17 A2H (Running et al., 2015)
Net Photosynthesis	PsnNet		NASA's MODIS 061 MCD15 A3H (Myneni et al., 2021)
Solar Radiation	FPAR (Fraction of Absorbed Photosynthetically Active Radiation)	cumulative **4-day** composite data set with **500-meter** resolution	NASA's MODIS 061 MCD15 A3H (Myneni et al., 2021)
Leaf Area Density	LAI (Leaf Area Index)		
Tree and Forest Health	NDVI (Normalized Differentiated Vegetation Index)	cumulative **16-day** composite with **250-meter** resolution	NASA's MODIS 061 MOD13 Q1 (Didan, 2021)
Canopy Thickness	EVI (Enhanced Vegetation Index)		
Plant Water Stress	NDWI (Normalized Differentiated Water Index)	Daily observation with scale of **463 meters**	MODIS MCD43 A4 006 NDWI (Schaaf & Wang, 2015)
Short Wave Radiation	ALLSKY_SFC_SW_DWN	**25-km** grids	NASA's Power Project (NASA Prediction of Worldwide Energy Resources (POWER), 2023)
Temperature at 2 meters	T2M		
Precipitation	PRECTOTCORR		

3.2.3 Fitness for Use Criteria

Fitness for use is described by ESRI GIS Dictionary (2023) as "the degree to which a dataset is suitable for a particular application or purpose encompassing factors such as data quality, scale, interoperability, cost, data format, and so on."

The data products (section 3.1.1) are obtained from the Google Earth Engine Data Catalog (GEE Data Catalog, 2023). They do not contain raw imagery as they have been preprocessed according to the algorithms developed by their authors. This is why the data quality aspect of fitness of use of the source data is ensured. Another major advantage is that the data products are available free-of-cost making them attractive for a broad range of projects and stakeholders.

There are however several issues like problems of scale or interoperability. As data products have varying grid resolutions, certain transformations are needed to resample the data and bring it to a common scale. The data lacks interoperability because of different temporal resolutions. The frequency of observations ranges from daily to once over 16 days requiring specific methods to fill the gaps. Finally, the satellite imagery on its own is unfit for predicting GPP by means of machine learning or any other computational algorithms, and therefore it must be transformed into a numeric format.

To summarize, not all fitness for use criteria is satisfied by the original data, and the proposed methodology must take this circumstance into account.

3.3 Methodology Overview

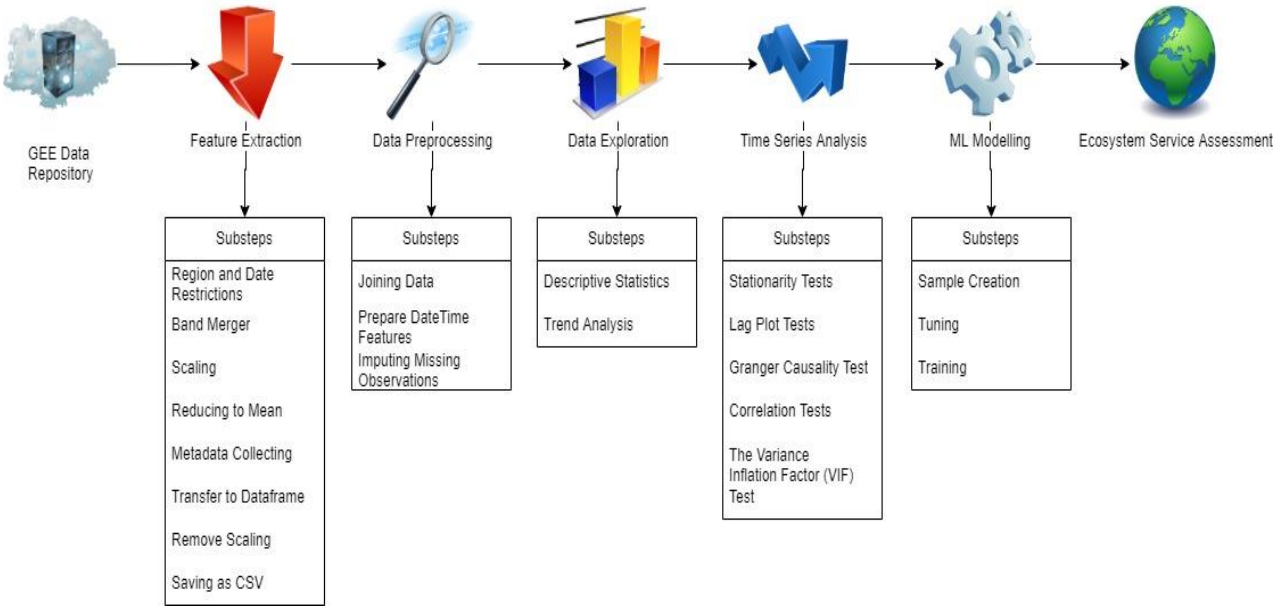


Figure 14 The Flowchart of Research Methodology

The overall logic of the proposed methodology includes six major phases: (1) feature extraction, (2) data processing, (3) data exploration, (4) time series analysis, (5) ML modeling, and (6) ecosystem service assessment (Figure 14). Subsequent sections present a detailed consideration of each phase.

3.4 Data Retrieval

3.4.1 The Google Earth Engine Data Catalog and its API for the Python Programming Language

The Google Earth Engine Data Catalog is an online data repository of remote-sensing imagery products (GEE Data Catalog, 2023). It was designed specifically for the Google Earth Engine (GEE), an online GIS maintained by the company. The GEE was designed with a graphic user interface supporting visualization features. Export of data is also included in the functionality of the system. The researcher would process the data by writing code in JavaScript. However, JavaScript is not a good option for working with data, unlike Python specifically equipped with a

large number of external modules for data wrangling, analysis and visualization. The instrument that can bridge the gap between Python and Earth Engine is the GEE Python API.

The technology of the application programming interface (API) allows to seamlessly link two programming environments. The interaction of the API and Python is presented in Figure 15. The GEE Python API enables to access the capabilities of GEE from the Jupyter Notebook environment which is a Python interpreter working on a local server. It communicates with the GEE data repository through the API that acts as an interface. The API is imported into Python as a library allowing it to retain its core capabilities while the syntax of the commands is adopted from JavaScript.

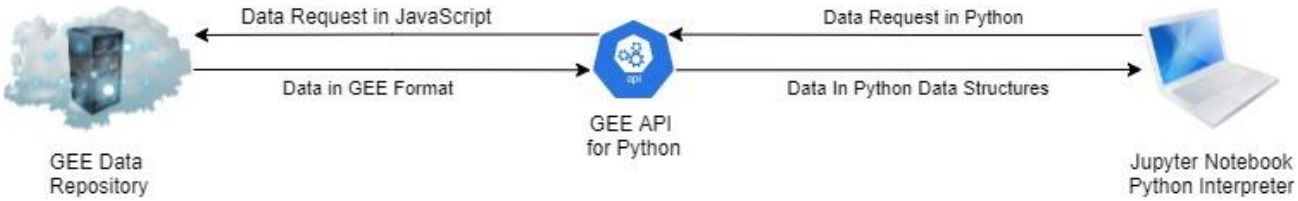


Figure 15 The Interaction of Python with the GEE Data Catalog

3.4.2 Feature Extraction

The product of satellite observations is imagery, however statistical tests, and ML modeling require data in the numeric format. This transformation constitutes the Feature Extraction stage of the methodology.

The Feature Extraction process is presented in Figure 16. After the data product is loaded into the Python interpreter, the first task is to restrict the imagery to the region of interest and the predefined time frame.

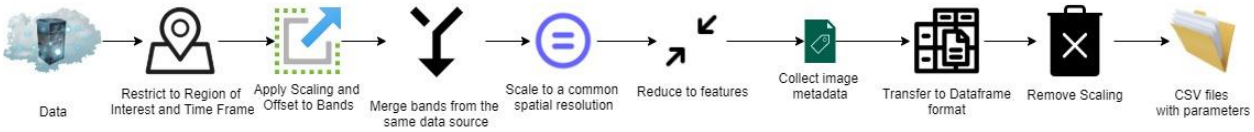


Figure 16 Feature Extraction Workflow

In order to set the correct projection for each image, it is multiplied by the scale and an offset is added to it. This ensures that an image is a more accurate depiction of the terrain. After scaling, all bands inside the same data product are combined into one band making all subsequent operations on it much easier.

The restricted imagery is then brought to the uniform scale. In remote sensing imagery fusion, the process of bringing all images to one scale is called *pansharpening*. As defined by Pohl & Genderen (2017), pansharpening is a downscaling technique according to which all images must be resampled to the lowest resolution. The lowest resolution of the available data is 500 meters per pixel and this number is chosen as the uniform scale. Given that the area of the region is 50 km^2, the total number of pixels in the study is 200.

The scaled images are then reduced to their features which involves collecting the value of each band of the image and its metadata such as the respective millisecond value and the date and time of observation. The collected features are first turned into a dictionary which is an inherently Python data structure and then into a data frame, a table-like data structure supported by the Pandas library. Pandas is a module that allows for easy data wrangling (McKinney, 2010), and the data frame is one of its core structures. Through this process, we are bridging the gap between imagery and numerical data that are better suitable for analysis. Each variable is now available in a form of data frame having each value paired with its observation date. In the last step, we save the data as CSV files for future use in data preprocessing and exploration. The code for the process of Feature Extraction is displayed in Appendix A.

3.5 Data Preprocessing

3.5.1 Merging Variable Data and Checking Data Quality

After the Feature Extraction stage is completed, each variable appears as an individual time series. However, in order to conduct statistical tests and split the variables into training and testing sets, the data must be combined in a single data object.

The Pandas library allows joining data from various sources into a single data frame on a common key or field. In our case, this field is the date and time of observation. This way, we attain a table of 11 columns and 1826 rows, one per each day in the study. We also programmatically extracted the year, month, day and day-of-year of the observation and added them to the data frame for analytical purposes. At this point, the issue of data interoperability needs to be addressed as all of the data came from sources with varying observation frequency.

We calculated the number of missing observations, and the result is presented in Table 5. To make the data interoperable, missing observations must be obtained from sources other than remote sensing products.

3.5.2 Data Interpolation for Missing Observations

Time Series is a collection of data obtained from repeated observations of a process at varying time points (Brockwell & Davis, 2002). A process can be considered stochastic if the contributing variables are random. Given that our variables were obtained from a satellite taking measurements of the same phenomenon at regular intervals and the observations depend on a multitude of factors and were not deterministic, they are stochastic time series which can be subjected to the time series analysis.

However, as mentioned in Section 3.2.1, the variables collected for this study originate from data products. In the process of their formation, various Remote Sensing Imagery Fusion

techniques are being applied to counteract issues inherent to RSD due to the construction features and performance of the sensors or atmospheric interference. To address these issues, original observations can be compressed into the corresponding average values or only the best quality observation over a certain period retained, thus making data intermittent. At the same time, this study requires continuous time series which calls for interpolation techniques to fill missing values.

Table 5 Number of Missing Observations per Variable

Variable Name	**Features Name**	**Number of Missing Values out of 1826**
Date	Date	0
Short Wave Radiation	ALLSKY_SFC_SW_DWN	0
Temperature	T2M	0
Precipitation	PRECTOTCORR	0
Solar Radiation	Fpar	1368
Leaf Area Density	Lai	1368
Gross Primary Productivity	Gpp	1596
Net Photosynthesis	PsnNet	1596
Canopy Thickness	EVI	1711
Tree and Forest Health	NDVI	1711
Plant Water Stress	NDWI	1

Interpolation is the process of deducing missing data points in between the known observations in a time series (Steffensen, 2006). The Pandas library supports several interpolation methods, such as linear, spline, polynomial, etc. The linear method treats observations as equally spaced which contradicts the definition of a stochastic process. The spline method fills the gaps with the values of a piece-wise polynomial function while the polynomial method passes the degree-n polynomial through the points of time series until there is no further improvement in the quality of interpolation from changing the value of n (Interpolation in NumPy, 2023). In our case, the second-degree polynomial interpolation provided a smooth enough interpolation. The distribution of features' values was checked before and after interpolation by means of histogram plots (see Figures 17 and 18). After using second-degree polynomial interpolation, no missing data points were left in any of the time series. Thus, the issue of data interoperability is resolved.

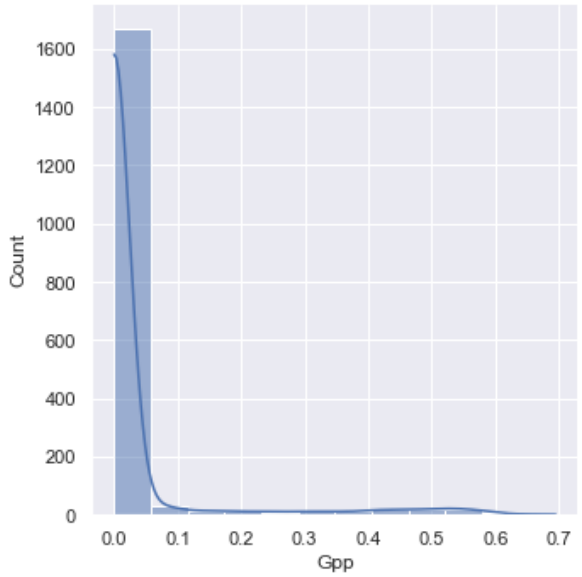


Figure 17 Value Distribution in GPP Time Series before Interpolation

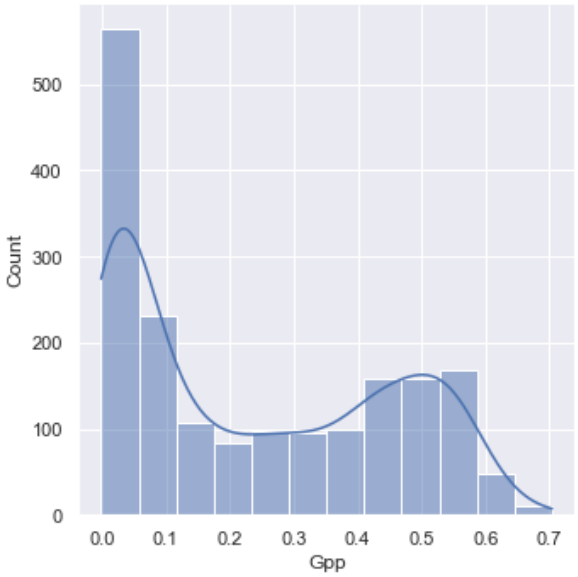


Figure 18 value Distribution in GPP Time Series after Interpolation

The workflow of the data preprocessing phase is presented in Figure 19 and the code is found in Appendix B. The next step in the methodology is deriving insights into the properties of the data.

Figure 19 Workflow of Data Preprocessing Phase

3.6 Using Descriptive Statistics to Identify Properties of the Data

The methods of descriptive statistics are used to quantitatively describe the properties of the data such as feature mean, standard deviation, minimum and maximum values as well as quantiles (Mann, 1995). The Five Number Summary method visualizes the descriptive statistical measures and allows their analysis and comparison. Two kinds of Five Number Summaries have been applied in the study to visualize data and its changes: (1) annual, and (2) monthly. This way, we were able to detect the behavior of the variables within the 5-year period as well as their seasonal changes.

3.7 Time Series Analysis

It has been established in sub-section 3.5.2 that the variables in consideration can be treated as stochastic time series and given their use in ML modelling, they will be referred to hereafter as features. Several statistical tests have been conducted to examine the time series' fitness for use in ML predictive modeling.

A stationary stochastic process is a process with a stable joint probability distribution; that is, a process with the same mean and variance (Brockwell & Davis, 2002). It is important to establish stationarity as it is one of the assumptions for various transformations as well as predictions. We used two tests to establish the stationarity of the features implemented in Python

code. The Augmented Dickey-Fuller Test (ADF) validates the null hypothesis that the time-series is non-stationary (Brockwell & Davis, 2002), with results shown in Figure 20.

```
ALLSKY_SFC_SW_DWN
ADF Statistic: -3.8835870679138114
p-value: 0.0021590131276925774
Critial Values:
   1%, -3.433966009459769
Critial Values:
   5%, -2.8631372667825503
Critial Values:
   10%, -2.567620331903232
Stationary
```

Figure 20 Results of the Augmented Dickey-Fuller Test (Short-Wave Radiation Feature)

The Kwiatkowski–Phillips–Schmidt–Shin Test (KPSS), on the contrary, deems the time series to be stationary and tries to disprove it (Kwiatkowski, 1992). By using two tests with contradicting null hypotheses, we were able to achieve a conclusive result on the stationarity of the majority of the time series, as demonstrated in Figure 21.

```
KPSS Statistic: 0.117315
p-value: 0.100000
Critial Values:
   10%, 0.347
Critial Values:
   5%, 0.463
Critial Values:
   2.5%, 0.574
Critial Values:
   1%, 0.739
Stationary
```

Figure 21 Result of the KPSS Test (Shortwave Radiation Feature)

Additionally, the Phillips-Perron test was used in an ambiguous situation (Phillips & Perron, 1988). The null hypothesis of this test also validates the non-stationarity of the time series (Figure 22).

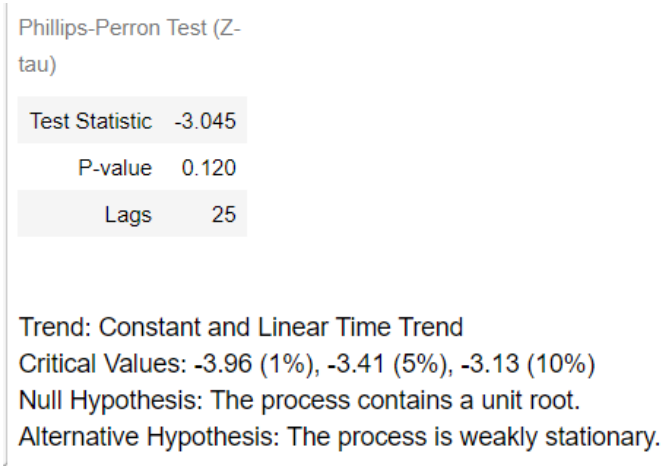


Figure 22 Result of the PP test (Temperature Feature)

The level of significance *p* was set to 0.05 in all tests. As a result, every time series proved to be stationary.

Lag Plots are commonly used in the cases where future changes in the time series can be explained by its past patterns (Brockwell & Davis, 2002). It is a plot of the Auto-Correlation Function (ACF) of the time series also used to establish the stochasticity of the processes. All features proved to be random. Determining randomness of data is important because otherwise, if there are certain patterns in the data (e.g., seasonality), additional manipulations are required to eliminate them. The presence of these trends leads to less reliable model predictions.

The Granger Causality Test is a statistical hypothesis test for determining whether one time series is useful in forecasting another, namely using the previous values of one time series to predict the future values of another time series (Granger, 1969). This test is applied in the study to determine features able to predict GPP in multifactorial prediction designs. It is demonstrated that

all variables are useful in predicting GPP; the example for the Shortwave radiation feature is presented in Figure 23.

```
ALLSKY_SFC_SW_DWN

Granger Causality
number of lags (no zero) 2
ssr based F test:         F=17.2002 , p=0.0000  , df_denom=1819, df_num=2
ssr based chi2 test:   chi2=34.4950 , p=0.0000  , df=2
likelihood ratio test: chi2=34.1728 , p=0.0000  , df=2
parameter F test:         F=17.2002 , p=0.0000  , df_denom=1819, df_num=2
```

Figure 23 Result of the Granger Causality Test (Shortwave Radiation Feature)

Now that we have established the stationarity of the features and fitness for prediction, we must investigate if any of the predictor variables may affect the outcome by introducing multicollinearity. *Multicollinearity* describes a situation whereby one predictor may be predicted from other predictors. It does not affect the overall model reliability or goodness-of-fit, but its presence may make it harder to explain the influence of each factor on the resulting prediction.

Firstly, we performed the correlation tests on the feature set. Our data does not satisfy the assumption of normality which is a mandatory condition for performing the parametric Pearson R correlation test. This is why we adopted the Spearman Rank correlation test (Spearman, 1904). The heatmap of the feature correlations is presented in Figure 24. It is evident that high multicollinearity is present, however in our case, it stems from the variables being related in the Light Use Efficiency Theory, as mentioned in the Motivation section 3.2.1. We will discuss the implications of multicollinearity later in the chapter.

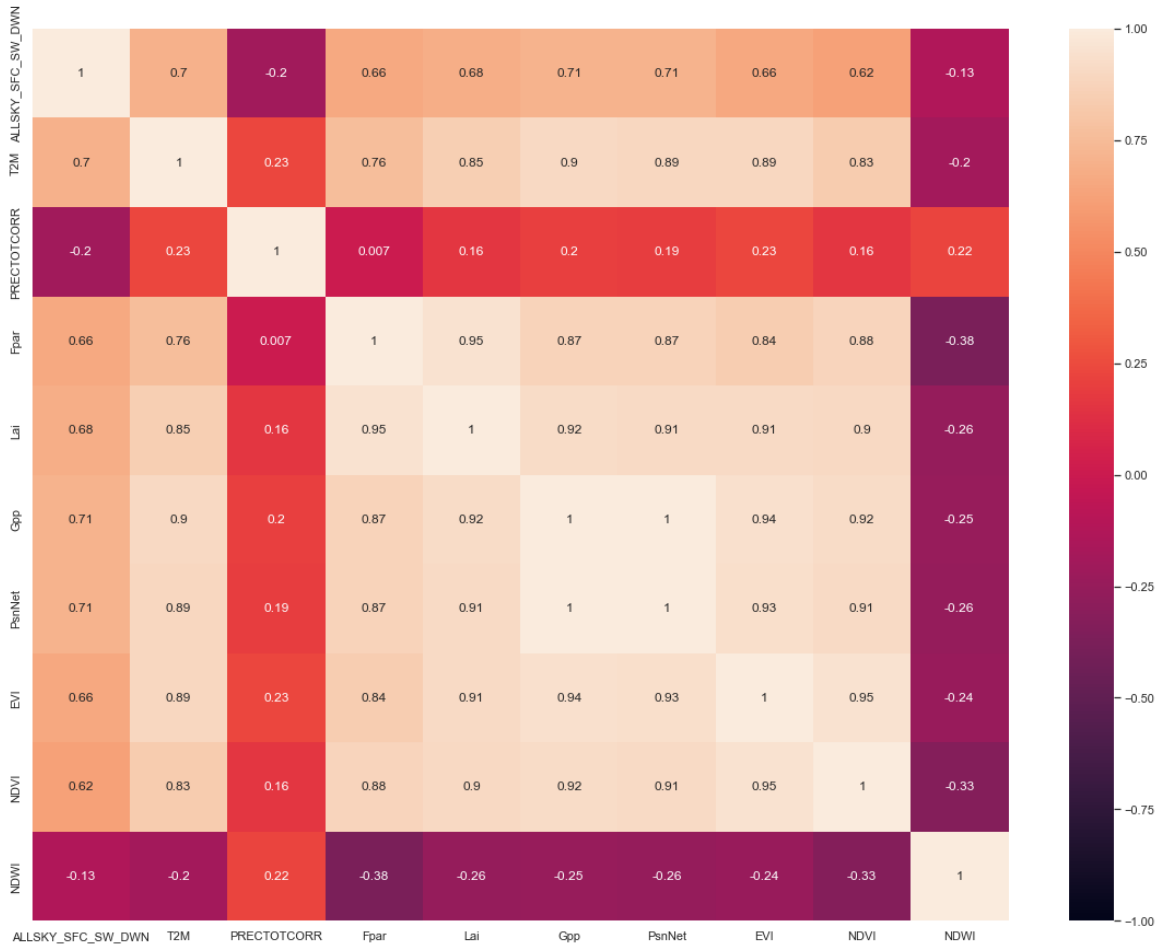


Figure 24 The Correlation Matrix of the Features Described in Table 5

Then, we conducted the Variance Inflation (VIF) test. The VIF test is used as a measure of multicollinearity between the features (James et al., 2017). It is a measure of how strong the multicollinearity is. The majority of features showed increasingly high VIF scores (Figure 25) indicating high multicollinearity and prompting to decrease it. To solve this problem, we performed standardization on the data which is the process of subtracting the sample mean from every observation and dividing the difference by the standard deviation. As a result of the standardization, the VIF scores have been significantly reduced. However, the Net Photosynthesis feature had a persistently high VIF score and was removed from participating in the analysis.

	feature	VIF
0	ALLSKY_SFC_SW_DWN	15.893141
1	T2M	13.500766
2	PRECTOTCORR	1.471874
3	Fpar	83.795036
4	Lai	71.169464
5	Gpp	1003.935443
6	PsnNet	906.941154
7	EVI	186.365918
8	NDVI	157.278571
9	NDWI	1.784349

Figure 25 Results of the Variance Inflation Test (all features)

Having established the suitability of the features for analysis, we have saved their values as a tab-delimited file for the next stage of the methodology which is machine learning experimentation.

3.8 Machine Learning Modeling and Experimentation

Historically, empirical statistical models and process-based models have been used in environmental management. There are limitations of traditional models described by Pei et al. (2022) and discussed in section 3.2.1. For this study, we selected 10 distinct variables as factors contributing to the formation of the target phenomenon (i.e., GPP), and specific mathematical techniques are needed to identify patterns and trends, handle large volumes of data, and consider the complex relationships between the predictor variables. Machine learning algorithms support these mechanisms making them an indispensable tool for modern data-driven projects.

3.8.1 The Choice of Machine Learning Algorithms

We have chosen three machine learning algorithms. The first two are the XGBoost Regressor (Chen & Guestrin, 2016) and the Random Forests Regressor (Breiman, 2001) already discussed in the previous chapter. The third algorithm is the Long Short-Term Memory (LSTM) artificial neural network. The LSTM has been first described by Hochreiter (1997). It is based on the feedback

principle that allows it to keep the model biases constant as well as to correct time lags. LSTM has been widely used for image, speech and sound recognition as well as time series predictions (Roy et al., 2023; Luo et al., 2023; Prasetyo, 2023). For time series forecasting, the training process is based on introducing the problem as a supervised learning problem with a moving time frame.

The implementation of Random Forests Regressor is available in the Scikit Learn library for Python (Pedregosa et al., 2012), while XGBoost was originally implemented in C++ and then has been adopted for Python by means of an API. The LSTM is available as part of the Tensorflow library (Abadi et al., 2015).

3.8.2 Tuning Algorithms

Hyperparameter optimization or *tuning* is a term that describes the process of selecting the combination of model hyperparameters that will lead to the best model performance possible (Feurer & Hutter, 2019). The majority of tuning practices are attributed to nonlinear optimization techniques.

Grid Search implemented in the Scikit Learn library provides an exhaustive search over the set of predefined hyperparameter values. The performance of each possible variant can be evaluated by one of the validation techniques, such as k-fold cross validation score. Grid Search is a heuristic algorithm which means it is designed to find a local optimum of the objective function but is not equipped to search for global optimum (Hsu et al., 2010). This is why the second technique we adopted was the Evolutionary Algorithm (Vikhar, 2016). This technique is based on the evolutionary principle of “the survival of the fittest” and involves producing hyperparameter sets from a predefined number of parent variants through a certain number of generations according to the rules of permutation. This algorithm is metaheuristic and is successful in finding the global optimum point.

3.8.3 Validation Techniques

Validation is a score measuring the goodness of fit of the model and the adequacy of its predictions. Cross-validation is a resampling method that splits the data into a certain number of subsamples or folds and sequentially feeds them into the model to assess the adequacy of predictions based on the cross-validation score (Geisser, 1993). The Cross Validation score is implemented in the Scikit Learn library.

According to Cerqueira et al. (2017), the cross-validation approach can provide better estimates of the model performance than the out-of-sample approach because it is less likely to overestimate the model's robustness. However, we also used the leave-one-out approach by splitting the data into training, testing and validation sets to perform experimentation.

3.8.4 Model Performance Metrics

The performance metrics for regression models usually demonstrate the error margin between the model predictions and real observations. We decided to use three metrics to see which one of them better captures the ability to predict the dynamics and trends of the time series.

Mean Square Error (MSE) is an indicator of model performance that measures the average of the squares of the errors; that is, the average squared difference between the estimated values and the actual value. R-Squared or the coefficient of determination is the proportion of the variation in the dependent variable that is predictable from the independent variables. The best possible score is 1.0, meaning perfect predictions. Mean Absolute Percentage Error (MAPE) is an indicator of regression loss indicated in percentage points from 0 to 100. All three metrics can be found in the Scikit Learn library.

In order to explore the influence of each factor on the predictions, we used Shapley Additive Explanation (SHAP) values (Lundberg & Lee, 2017). SHAP visualizes the distribution of the data and denotes the degree of influence by colour or by height of a column.

3.8.5 Experimentation Design

We have prepared 3 distinct experiment designs that were performed in Jupyter Notebook environment for Python.

The first experiment was designed to test the practical performance of the developed models in predicting GPP. The predictions are performed using a "leave-one-year-out" approach consisting in predicting the GPP value of one year using the data of all other years to train the model. For instance, GPP in 2018 is predicted based on a model trained on the 2014-2017 data.

The second experiment was conducted in order to assess the performance of the models as a function of the lead time of 12 months. For example, GPP (target variable) in 2015 will be predicted by predictor variables from 2014. The "leave-one-year-out" approach is also applied to obtain a validation of model's performance. Thus, GPP in 2018 is predicted by 2017 predictors.

In the first and second experiments, we used XGBoost and Random Forest regressors, sequentially tuning them with Grid Search and Evolutionary Algorithm techniques to compare them on the cross-validation scores (Cerqueira et al., 2017) and performance metrics.

The third experiment entailed predicting GPP from itself using LSTM. Although we used the same data set, it was not suitable for prediction with LSTM as it requires a transition to a different format of training data. An additional preprocessing step had to be performed to transform the time series into couples of training observations and their target prediction. The resulting dataset was fed into a sequential neural network model with a five-node input layer, a 256-layer LSTM hidden layer, an eight-node dense layer with Relu activation rule and a one-node dense

layer with a linear activation function. We used default values of hyperparameters of LSTM. Mean Squared Error was used as a measure of loss, the Adam optimizer was given a learning rate of 0.0001 and the Root Mean Squared Error was chosen as a performance metric. The model was trained over 10 epochs.

The flowchart of the machine learning model training and experimentation process is presented in Figure 26. A sample of code is listed in Appendix C.

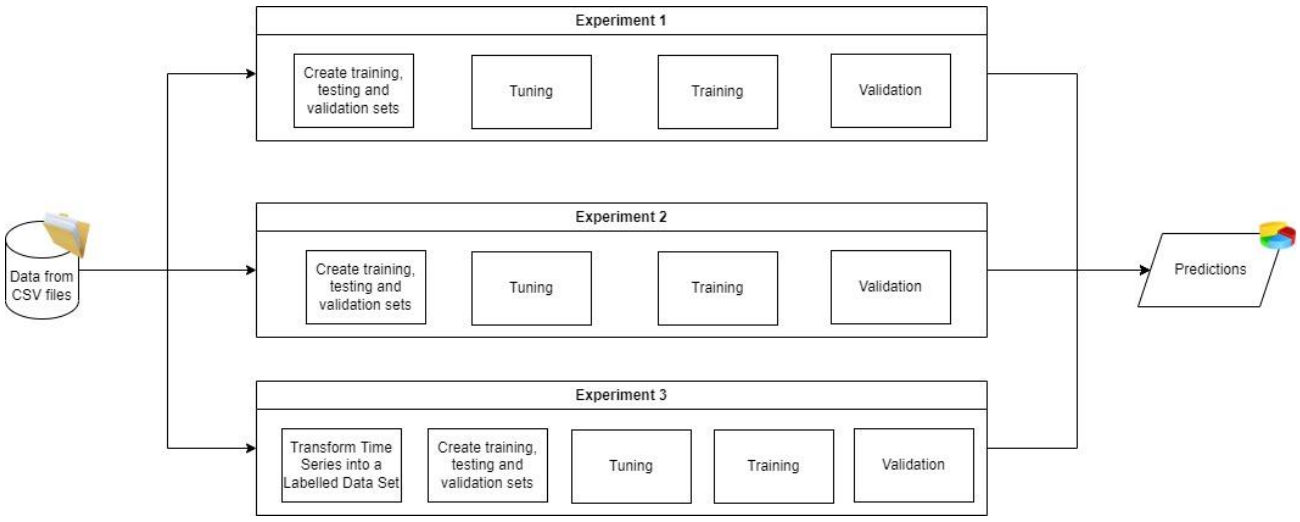


Figure 26 Flowchart of the Model Training and Experimentation Process

GPP is a target variable of prediction because it is a proxy measure of carbon sequestration. All tests and models in this study have been built in relation to GPP. Predicting GPP through ML modeling will equip the policy- and decision-makers with a tool to quantitatively assess probable changes in ecosystem's capacity to sequestrate carbon in the upcoming time periods, especially if the settings of the second experiment are adopted for practical sustainable environmental management. Additionally, predictions of GPP can be deployed in ecosystem service assessment through directly observed behaviour method and market pricing as discussed in the next section of the chapter.

3.9 Ecosystem Service Assessment

Gross Primary Productivity (GPP) is measured in mass of carbon per unit area per year (g C m^{-2} yr^{-1}). This is a physical measure but for assessing the carbon sequestration capacity as an economic good, we must transfer its value into the market price. When the researchers have a chance to observe the dynamics of ecosystem behavior, they can use *direct observed methods* to do so. Direct observed behavior methods measure the value of ES from the behavior of consumers by applying market prices. These methods are suitable for privately-owned, market-traded resources. Nowadays, the carbon emission has become a market-traded externality. An *externality* is an indirect benefit or cost that occurs as a side-effect from other economic activities (Pigou, 1920). The opportunity to cover the cost or claim the benefit of an externality can be sold on the market.

According to the World Bank's Carbon Pricing Dashboard (The World Bank, 2023), the two main financial instruments used as an economic representation of the externality are the carbon tax and the emission trading system (ETS). As of 2022, the Carbon tax mechanism has been implemented in countries like Argentina, Chile, Mexico, Canada, the European Union and Ukraine. The ETS is utilized in Canada, China, Kazakhstan, the European Union, as well as certain states of the USA. Several countries in the South-East Asia, such as Pakistan, Malaysia and Indonesia, are considering to adopt either of the two measures. These countries are major producers of emissions as they have large manufacturing sectors in their economy. As of now, India is not on the list of countries that consider introducing either action. That is why in conducting market evaluations in this study, we relied on foreign well-established systems.

The European Union ETS is the oldest trading system in the world originating in 2005. It is a cap-and-trade system, which is a type of system where the government caps the amount of annual

carbon emissions and then sells the rights to emit above that limit. This way, the emissions become a market asset and, by trading them, the government profits as an issuer. The markup price of the asset grows as the enterprises whose emissions cannot be cut, bargain to buy more emission rights. The sources of emissions can be power-generating and manufacturing sectors, aviation, *etc*. The cap is reduced by 2.2% each year. The EU ETS supports the climate goals of cutting emissions by 55% by 2030 and going carbon neutral by 2050. The profits from trading the emission limits are used to fund sustainable energy initiatives. For the current year, the price of a tonne of CO_2 emissions is $87 USD with the projected revenue totaling to $43,326 million in the European Union (The World Bank, 2023).

The China National ETS has been implemented in 2021. It covers 32.75% of the 26,000 tonnes of the annual CO_2 emissions from China's Power sector and is currently the largest in the world. Their price of one tonne of CO_2 is $9 USD (The World Bank, 2023).

The New Zealand ETS has been operating since 2008 covering emissions from fossil fuels combustion, power generation, industry and forestry. New Zealand has a significant forest cover and boasts several magnificent national parks that attract a significant number of tourists, so preserving the ecosystems is very important for the country. Their cap-and-trade system prices tonne of CO_2 at $53 USD. The 2022 revenues reached $1,648 million USD (The World Bank, 2023).

In order to assess the market price of GPP, we need to convert from grams of carbon (C) to tonnes of Carbon Dioxide (CO_2). As the units of measurement of GPP is g C m^{-2} yr^{-1} (Amthor & Baldocchi, 2001), we must divide the GPP by the square area of the region of study in meters. The area we selected is 50 km^2 or 5 million m^2. Then, we must convert grams into tonnes, there are one million grams in a tonne. Another issue is that C and CO_2 are different chemical compounds

with different molar masses. The molar mass of C is 12 and there are 12 grams of C in each mole of CO_2 which leads us to a rough estimate of 1 tonne of C in 3.66 tonnes of CO_2.

In conclusion, this chapter has presented the methodology that allows us to satisfy all research gaps outlined in chapter 2, such as fitness for use, data availability and quality concerns as well as the proper use of machine learning techniques.

The next chapter presents the results of the experimentation with machine learning algorithms as well as some case studies on GPP predictions for specific areas.

Chapter 4: Results and Case Studies

In this chapter we present the results of the experimentation that were conducted using the methodology described in chapter 3.

4.1 Performance of Machine Learning Models in Experiments

The first experiment concerned the possibility to predict GPP using ML ensemble methods as well as to assess the accuracy of these predictions. The models in this experiment were trained on the datasets from 2014 to 2017 while the predictions were obtained for 2018 and compared to actual data. The accuracy of the model predictions based on three specific performance indicators is presented in Table 6.

Table 6 Performance Indicators in Experiment 1

Regressor	**Base Model**	**Tuned with Grid Search**	**Tuned with Genetic Algorithm**
MSE			
XGBoost	0.0044	0.0049	0.0047
Random Forest	**0.0040**	0.0041	0.0043
R-Squared			
XGBoost	0.8738	0.8600	0.8635
Random Forest	0.8774	0.8794	**0.8834**
MAPE			
XGBoost	**0.2525**	0.2647	0.2689
Random Forest	0.2558	0.2529	0.2534

It is evident that both models compete closely in every possible tuning. While in terms of the Mean Squared Error (MSE), the Random Forest is the best performing model, the XGBoost leads for the remaining indicators. It is particularly interesting to note a high R-Squared score for the RF model tuned using the Genetic Algorithm which proves its efficiency as a tuning technique. A graphic representation of predictions made by the XGBoost and Random Forest models is presented in Figures 27 and 28.

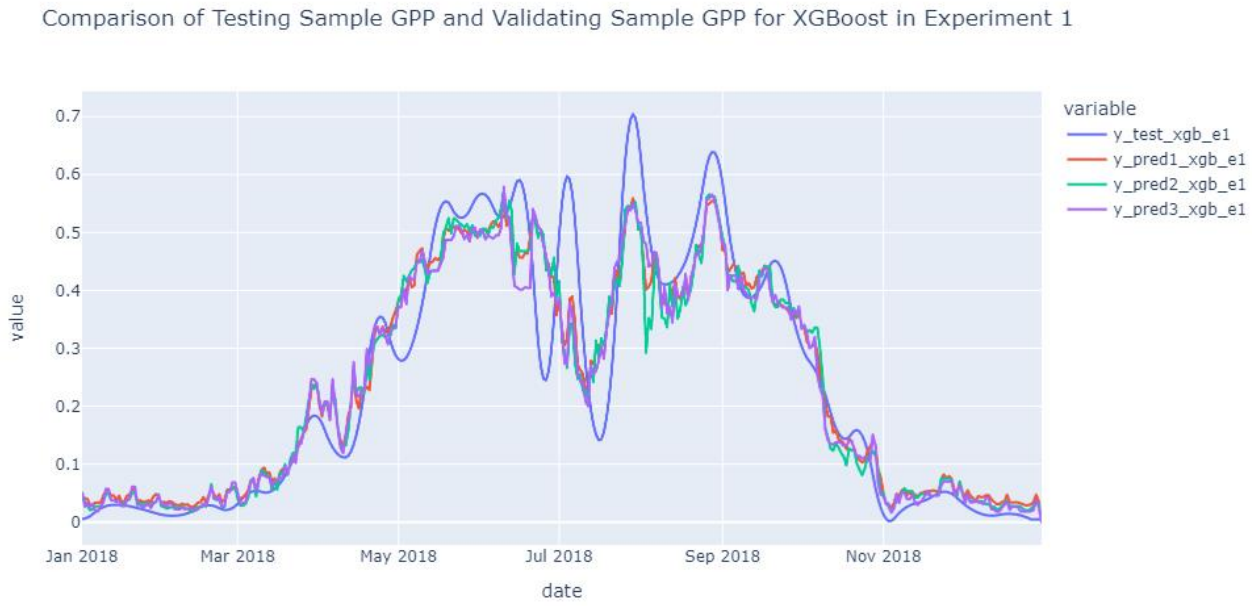


Figure 27 Comparison of Test GPP (blue) and Predicted GPP (red, green, violet) for XGBoost Regressor in Experiment 1

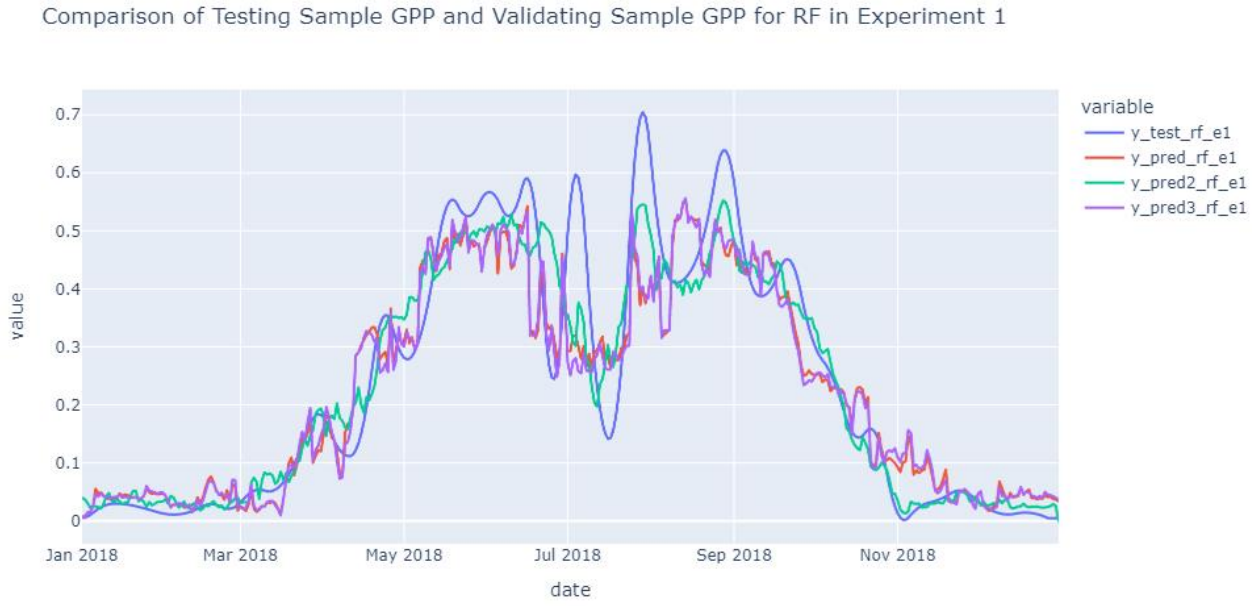


Figure 28 Comparison of Test GPP (blue) and Predicted GPP (red, green, violet) for Random Forest Regressor in Experiment 1

The second experiment predicted the GPP of the subsequent year based on predictors obtained from the previous years. The training sample included the data from 2014 to 2017 and validation conducted on 2017 features to predict GPP in 2018. Table 8 presents the resulting performance indicators for this experiment.

Table 7 Performance Indicators in Experiment 2

Regressor	**Base Model**	**Tuned with Grid Search**	**Tuned with Genetic Algorithm**
MSE			
XGBoost	0.0067	0.0078	0.0073
Random Forest	**0.0042**	0.0061	0.0061
R-Squared			
XGBoost	0.7690	0.7602	0.7639
Random Forest	**0.8158**	0.8086	0.8076
MAPE			
XGBoost	0.3843	0.3912	0.3912
Random Forest	**0.3388**	0.3425	0.3492

In this experiment, the Random Forest was an absolute leader according to the performance indicators. It is interesting to note that the random combination of parameters (i.e., the Base Model) was more successful than the combination of parameters selected intelligently through tuning. The comparison of testing and validating prediction for all models is shown in Figures 29 and 30.

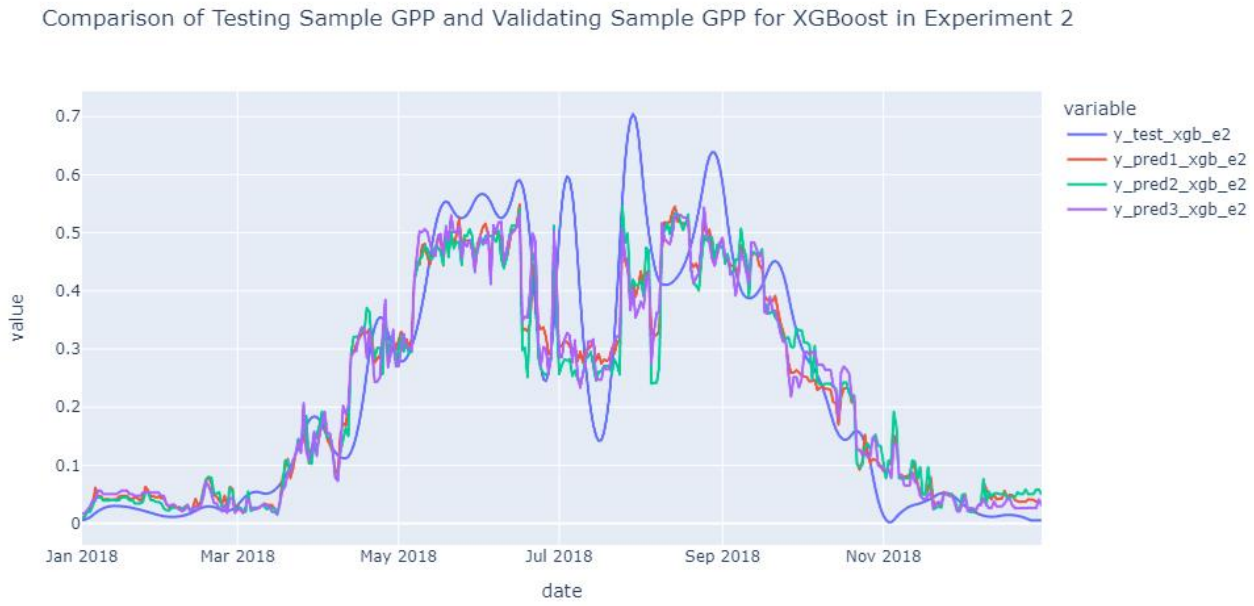


Figure 29 Comparison of Test GPP (blue) and Predicted GPP (red, green, violet) for XGBoost Regressor in Experiment 2

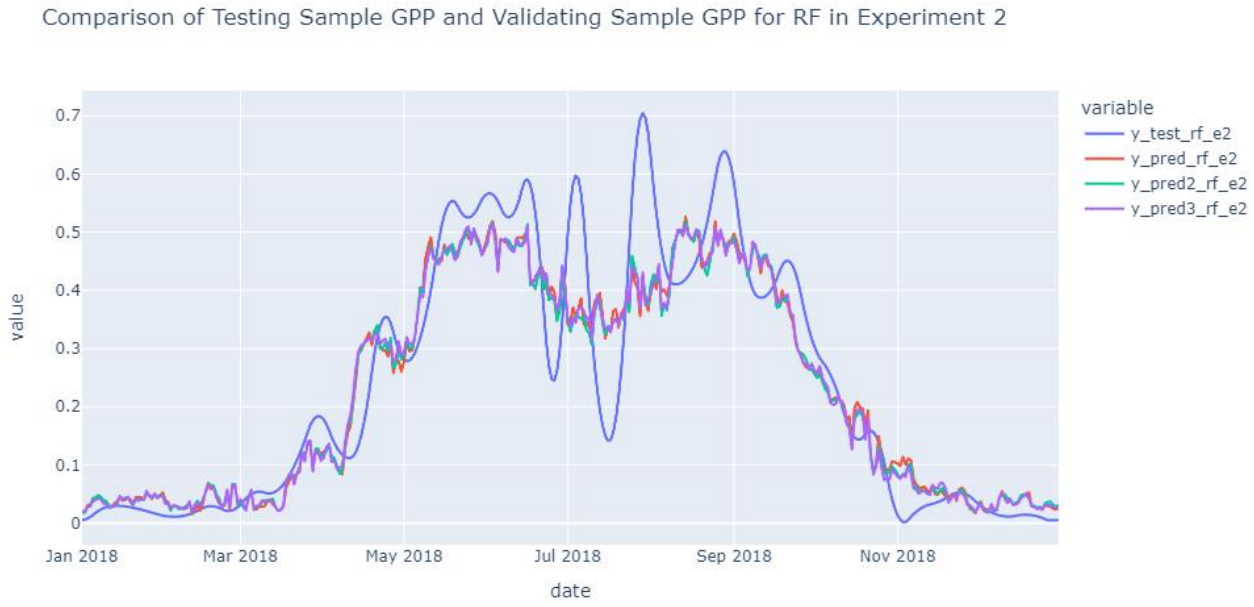


Figure 30 Comparison of Test GPP (blue) and Predicted GPP (red, green, violet) for Random Forest Regressor in Experiment 2

In the third experiment, we predicted GPP of the subsequent year from GPP in previous years by using an artificial neural network (ANN) with an LSTM layer. The performance of this model is measured in terms of the Root Mean Squared Error (RMSE), the Mean Absolute Percentage Error (MAPE) and its loss function calculated from the Mean Squared Error. The goal is to minimize the loss function (Hastie et al., 2001). The resulting indicators after the 10^{th} epoch are shown in Table 9.

Table 8 Performance Indicators in Experiment 3

Indicator of Performance	**Value**
Loss (Mean Squared Error)	0.0016
RMSE	0.0397
MAPE	0.0317

The errors are marginally small and significantly smaller than those for the models in the two previous experiments favouring a wider adoption of ANN for GPP predictions. Figure 31 shows that the test GPP and validation GPP are almost identical.

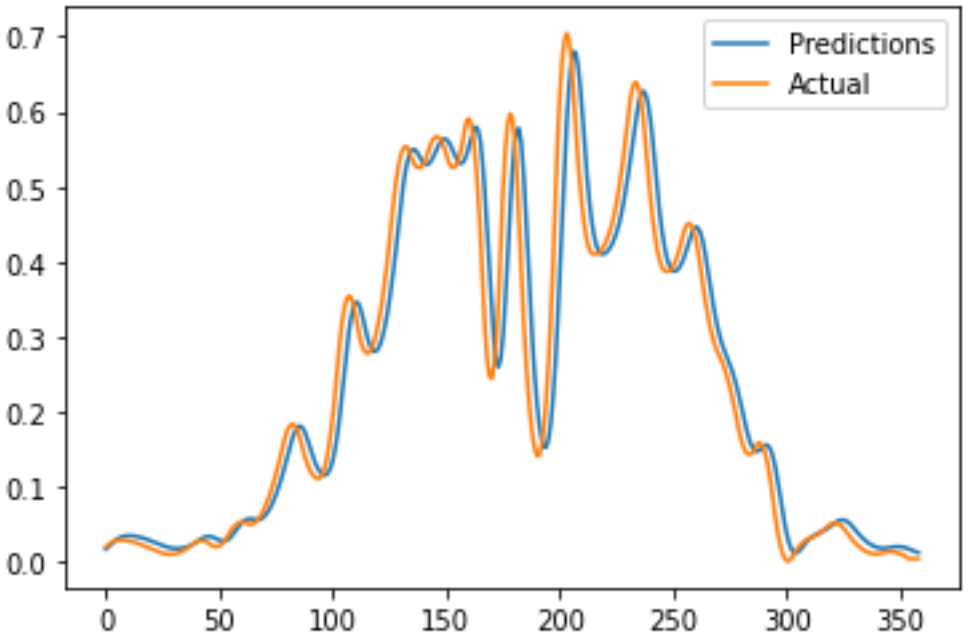


Figure 31 Comparison of Test GPP (orange) and Predicted GPP (blue) for ANN with LSTM Layer in Experiment 3

Now, we shall discuss the explainability of the modeling results and the influence of predictor variables in different experiments. Shapley Additive Explanation (SHAP) was used for the purpose (Shapley, 1951). As mentioned earlier (section 3.7), the multicollinearity found in the features cannot affect the reliability of the Random Forest, but it can impact explainability, while XGBoost is immune to multicollinearity. For this reason, only predictions from XGBoost will be considered in explainability tests. The LSTM will not be considered because only one parameter was used for prediction, and it is impossible to assess the contribution of other factors.

The bee swarm diagram (Figure 32) illustrates the contribution of each feature to the model prediction by plotting their distribution and encoding their importance in colour. The bar chart (Figure 33) ranges the features by the mean absolute Shapley value for a better visualization.

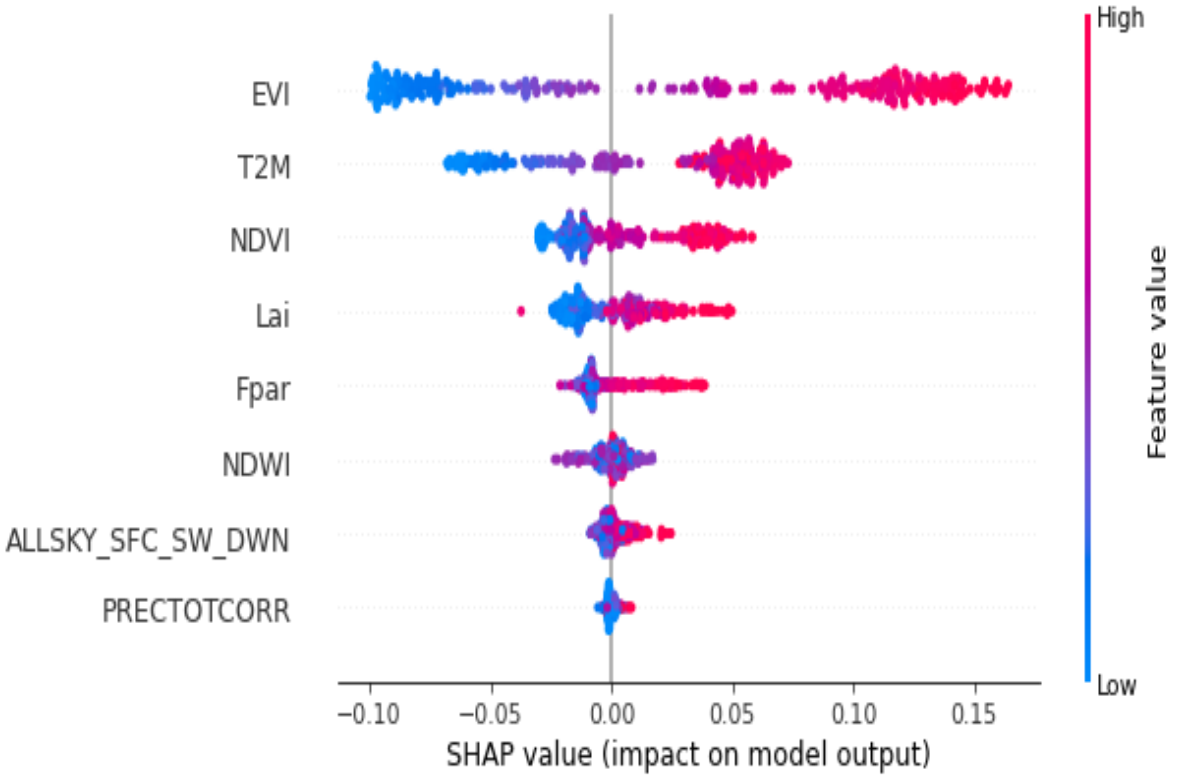


Figure 32 Bee Swarm Diagram of Shapley Values for XGBoost in Experiment 1

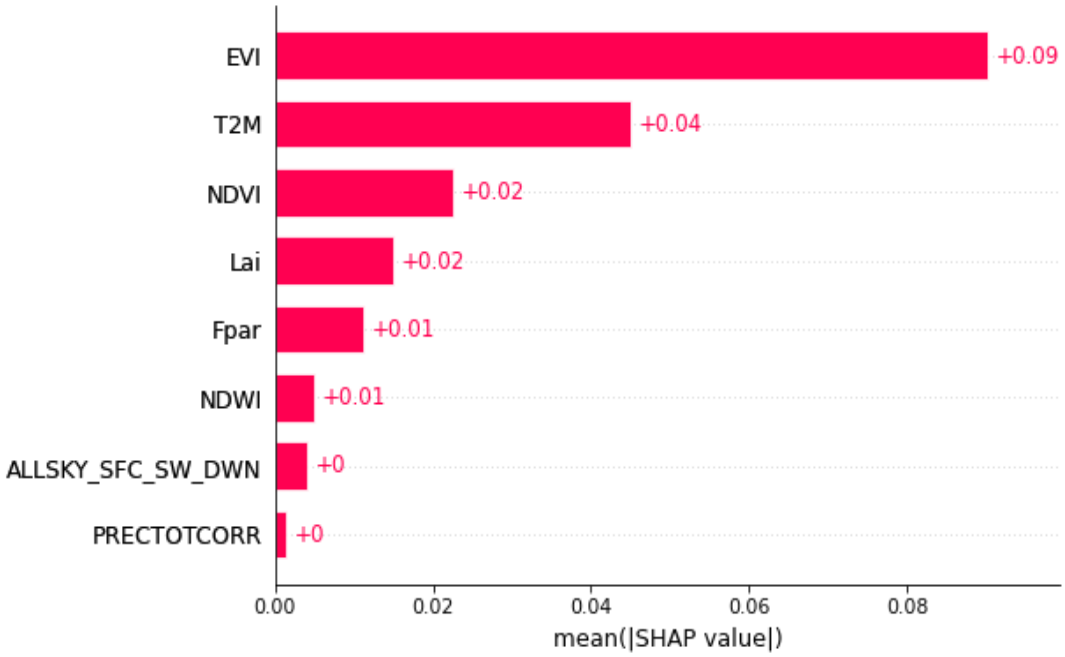


Figure 33 Bar Chart of Shapley Values for XGBoost in Experiment 1

According to Figures 32 and 33, the three most important predictors of GPP in the first experiment are the Enhanced Vegetation Index (EVI), average air temperature (T2M) and Normalized Differentiated Vegetation index (NDVI). The EVI is a proxy measure of vegetation health (Verstraete & Pinty, 1996), and its good performance in predicting GPP agrees with prior research by Zheng et al. (2018). The air temperature is an environmental stress factor playing a key role in regulating the photosynthesis, so its major influence on GPP is understandable. The NDVI is a measure of vegetation heath and particularly chlorophyll stock. It is one of the most popular indicators of biomass (NASA Earth Observatory, 2000). The Leaf Area Index (LAI) has similar influence on the prediction while additionally reflecting the seasonality patterns. The use of EVI and LAI indexes also relates the results of the first experiment to ecosystem structure and function assessment, as will be discussed later.

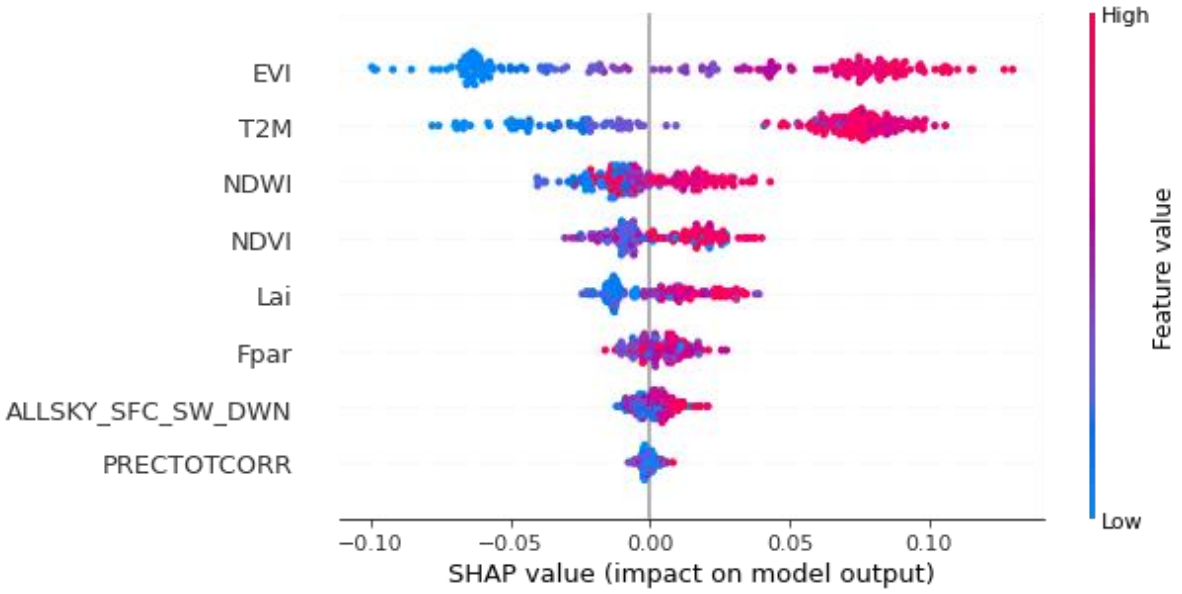


Figure 34 Bee Swarm Diagram of Shapley Values for XGBoost in Experiment 2

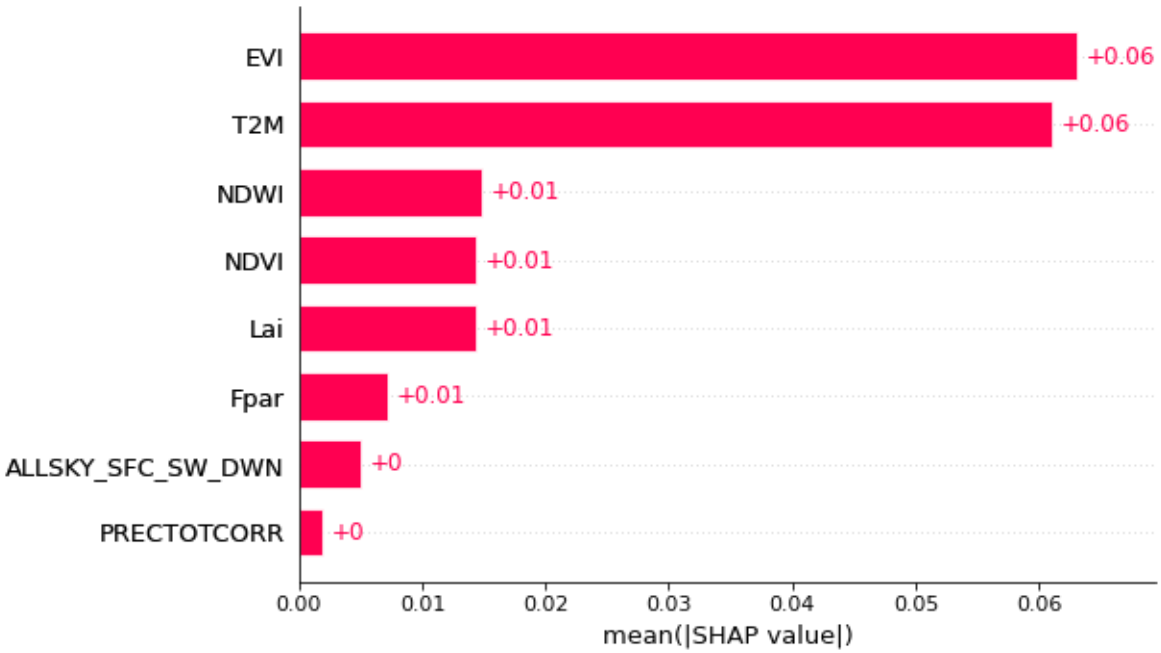


Figure 35 Bar Chart of Shapley Values for XGBoost in Experiment 2

Overall, the results of the second experimentation presented in Figures 34 and 35 are similar to those of the first one. However, the Normalized Differentiated Water index (NDWI) takes precedence over NDVI and LAI. The NDWI is a proxy measure of water content in leaves (Gao, 1996). It can serve for monitoring the risk of forest fires and be beneficial in the assessment of

drought risk. Given that in the experiments, the GPP is predicted for the subsequent period based on the predictors from preceding periods, it is possible to foresee how the current processes in the ecosystem may affect its future structure and functions.

4.2 Case Study 1: Dachigam National Park

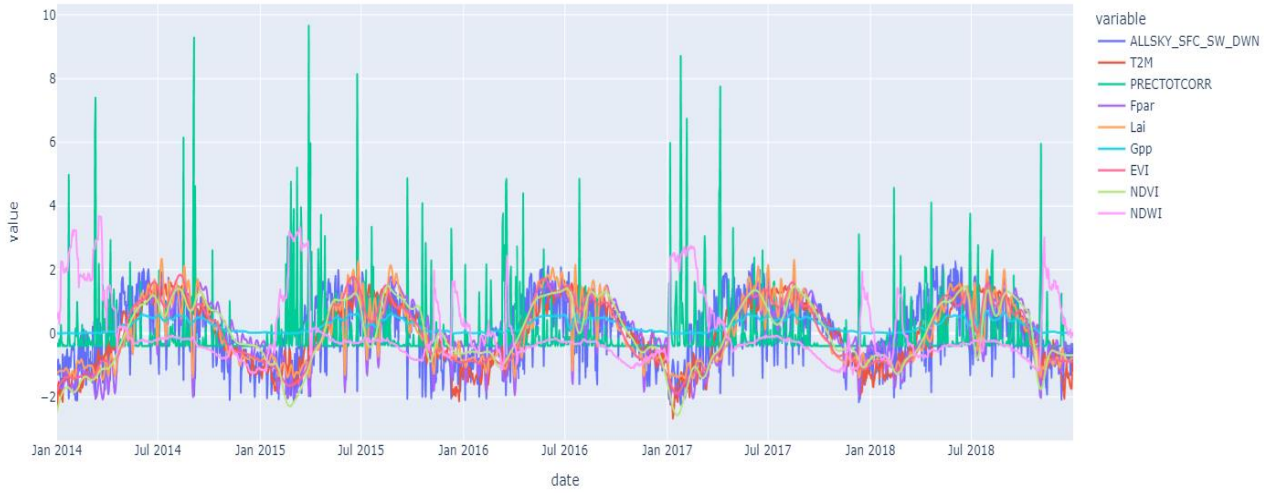
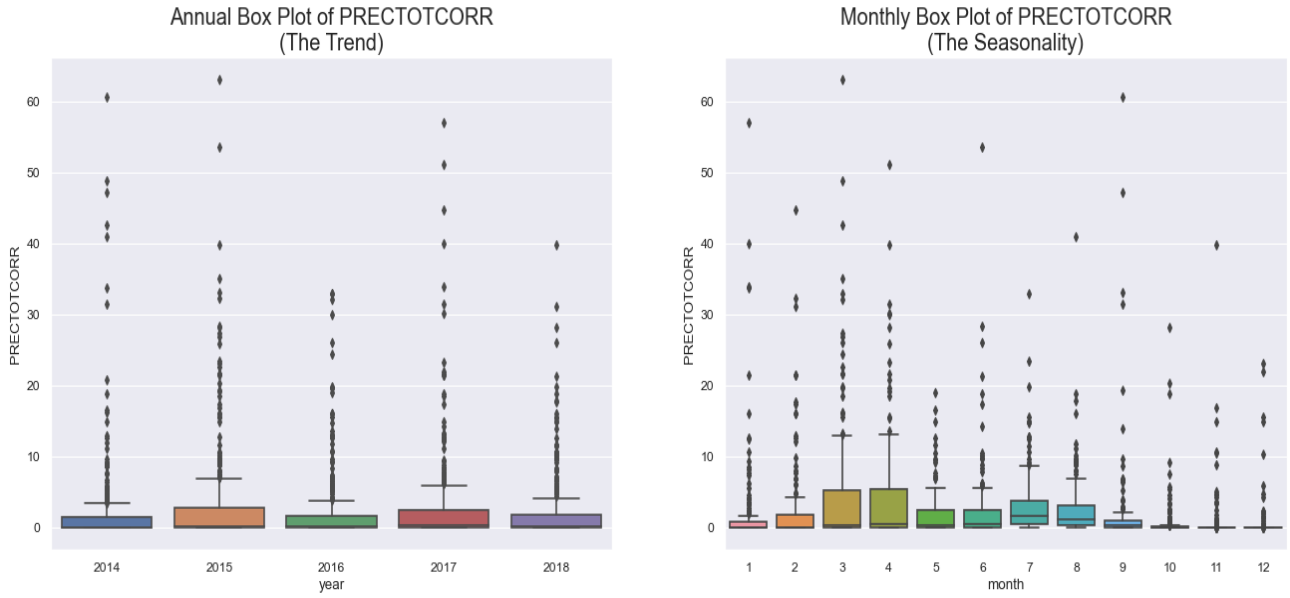


Figure 36 Research Parameters for Dachigam National Park

Figure 36 provides an insight into the research parameters obtained as a result of preprocessing of data for Dachigam National Park over the period from 2014 to 2018. The seasonal nature of the variables is evident which allows us to discuss the influence of seasonal events on the key variables including GPP. The bright green line on the chart denotes precipitations indicating that the heaviest precipitations happen in Winter. Given that the Kashmir Valley is a mountainous area surrounded by the 8000+ meter mountains, Winter precipitations can be amounted to snowfall.



(a) (b)

Figure 37 (a) Annual and (b) Monthly Trends in Precipitation in Dachigam National Park

Figure 37 illustrates precipitation patterns in the Dachigam. The trend chart shows that the amount of precipitation in the area of study is very high as can be seen through the distributions having only upper tails as well as a very significant number of outliers. Season-wise, March and April have long right tails and a large interquartile range which is consistent with the period of snow melting. Months 5 to 9 (May to September) correspond to the Summer Monsoon Season.

Malik & Hashmi (2021) estimated the damage caused by the September 2014 flood event to the economic infrastructure and crops but their analysis did not include the impact on vegetation. Another study by Ahmad et al. (2017) assessed the flood impact on the land-use and land-cover aspects in the area around the Wular Lake which is 50 km away from the Dachigam National Park. In any event, we were unable to find in the literature any study on the impact of floods on the GPP.

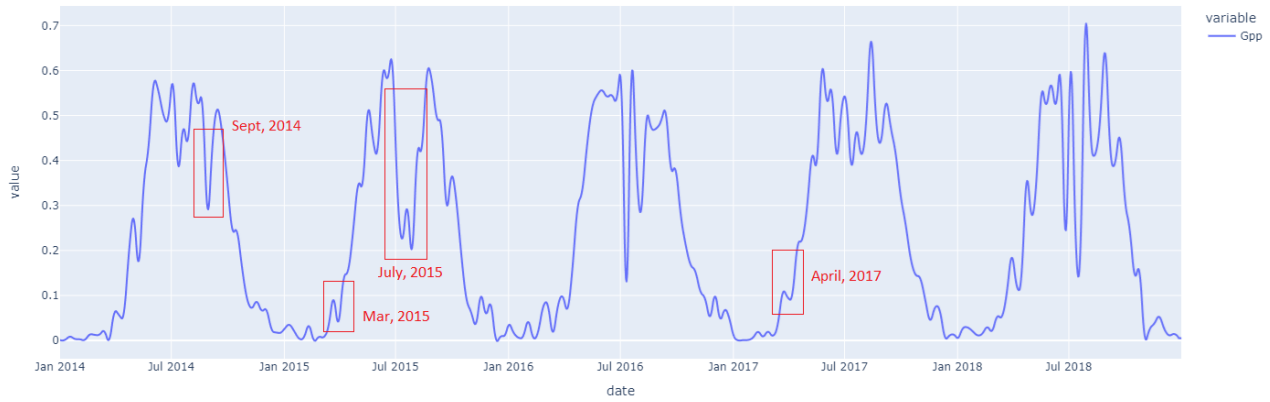


Figure 38 The Influence of Flood Events on GPP in Dachigam National Park

Figure 38 provides insight into the dynamics of the GPP over the five years (2014-2018) through plotting the preprocessed data with flood events marked by red rectangles. It is evident from the graph that the negative impact of floods on GPP can be roughly estimated as 0.25 units of GPP in September 2014, 0.5 points in March 2015, 0.41 points in July 2015 and 0.01 in April 2017.

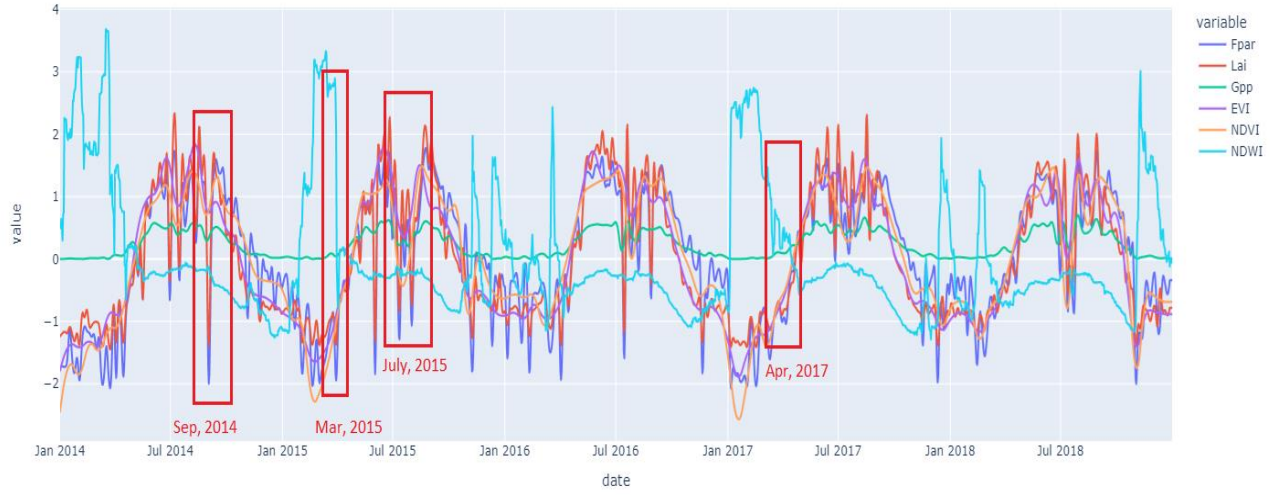


Figure 39 The Impact of Floods on Vegetation Indexes in Dachigam National Park

In general, the dynamics of vegetation indexes over 2014-2018 follows similar to GPP patterns (see Figure 39). LAI, EVI and NDVI as well as FPAR show sharp decline during the flood. It is also evident from the graph that the highest values of NDWI are reached in the winter, despite the LAI being the lowest in the winter months. The summer values of NDWI are significantly lower which is a sign of a dry, forest-fire dangerous situation. But even NDWI declines with the onset of floods as the cases of March 2015 and April 2017 events demonstrate.

Now that we have discussed the dendrological consequences of flood events, we shall move on to outlining the economic approaches to the corresponding estimations. As the carbon dioxide emissions are traded as externalities, we would suggest to adopt the opportunity cost approach to their pricing. The opportunity cost is defined as lost benefits from choosing one option over another (Oxford's Learner Dictionary, 2023). In Economics, the term is often used to describe financial and non-financial losses. An associated term is *sunk cost* describing a cost that has already been incurred and cannot be made up for. We propose to calculate the sunk cost of the floods in comparison what the estimation of GPP would have been if the floods never happened.

There are two possible ways to quantify the GPP loss during the flood. The first one is to calculate the difference between the maximum level of GPP before the flood and its lowest value preceding the recovery of the forest ecosystem. The second one is to estimate the cumulative GPP over the period between the start date of the flood and the first day that it recovered to its pre-flood levels. In our case study, we estimated the sunk costs for 2015 when two flood events occurred. Because we are using modern-day prices, we also corrected the price estimates for the inflation factor. One US dollar in 2015 is worth $1,26 in 2023 (Dollar Times, 2023).

Total Annual GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
4143,5	t/C	15165,21	tCO2
Gramms to Tones			
4143,5	tonne	4143500000	grams
Grams of C per sq meter per year into Grams			
82,87	gr C/m^2 yr	4143500000	gr
50000000	m2	area of region	
50	km2	area of region	
82,87	GPP units	equivalent to gr C/m^2 yr	

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$1 047 121,64
China National ETS	$9,00	$108 322,93
New Zealand ETS	$53,00	$637 901,69
Inflation correction	$1,26	

Flood Impact GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
45,5	t/C	166,53	tCO2
Gramms to Tones			
45,5	tonne	45500000	grams
Grams of C per sq meter per year into Grams			
0,91	gr C/m^2 yr	45500000	gr
50000000	m2	area of region	
50	km2	area of region	
0,91	GPP units	equivalent to gr C/m^2 yr	
the margin between the highest and lowest point in flood			

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$11 498,50
China National ETS	$9,00	$1 498,77
New Zealand ETS	$53,00	$7 004,83

Market Price Alternative Losses	
ETS	Price
EU ETC	$1 058 620,14
China National ETS	$109 821,70
New Zealand ETS	$644 906,52

Figure 40 The Estimation of Sunk Costs of the Floods for 2015 by Margin Method

Figure 40 presents the calculation by the marginal method using the pricing approaches of the three Emission Trading Systems (ETS) described in section 3.9. The Market Price Alternative Losses table presents a hypothetical estimate of GPP had the floods never occurred.

Total Annual GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
4143,5	t/C	15165,21	tCO2
Gramms to Tones			
4143,5	tonne	4143500000	grams
Grams of C per sq meter per year into Grams			
82,87	gr C/m^2 yr	4143500000	gr
50000000	m2	area of region	
50	km2	area of region	
82,87	GPP units	equivalent to gr C/m^2 yr	

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$1 047 121,64
China National ETS	$9,00	$108 322,93
New Zealand ETS	$53,00	$637 901,69
Inflation correction	$1,26	

Hurricane Impact GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
1244,5	t/C	4554,87	tCO2
Gramms to Tones			
1244,5	tonne	1244500000	grams
Grams of C per sq meter per year into Grams			
24,89	gr C/m^2 yr	1244500000	gr
50000000	m2	area of region	
50	km2	area of region	
24,89	GPP units	equivalent to gr C/m^2 yr	
the cumulative gpp from the start of decline to the end of recovery due to flood			

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$314 502,93
China National ETS	$9,00	$32 534,79
New Zealand ETS	$53,00	$191 593,74

Market Price Losses	
ETS	Price
EU ETC	$1 361 624,57
China National ETS	$140 857,71
New Zealand ETS	$829 495,43

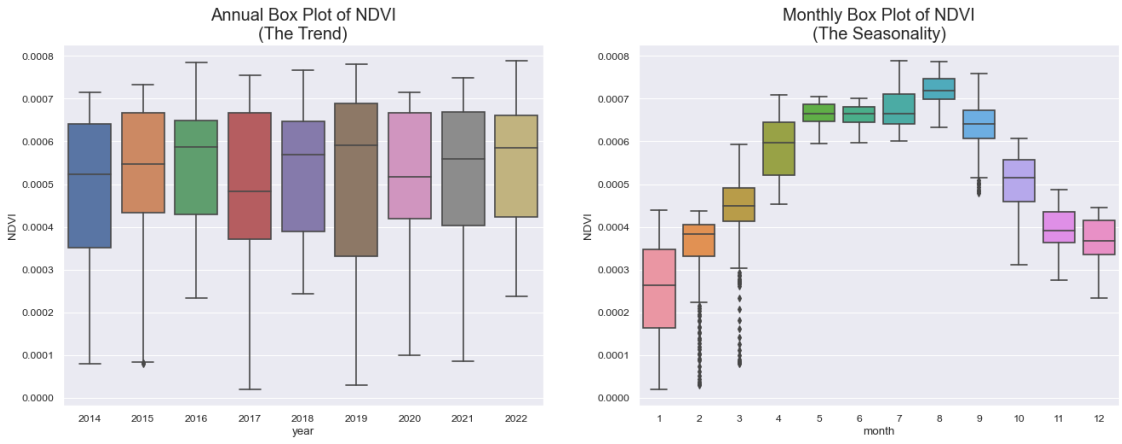
Figure 41 The Estimation of Sunk Costs of the Floods for 2015 by Cumulative Method

Figure 41 presents the same estimation but for the cumulative GPP between the start and the end of the flood. This approach leads to a more significant sunk cost of the flood ($314,502.93 compared to $11,498.50 based on the EU ETS pricing). It is worth noting that adopting one or the other approach depends on the overall economical situation in a given country, the budgeting concerns and the inflation considerations. We recommend implementing the ETS in India, so that pricing could be acquired more precisely. As we have seen in case of New Zealand's ETS (The World Bank, 2023), forestry is one of the industries that can be covered by such systems, and losses due to flood events can be equated to those cased by deforestation.

4.3 Case Study 2: Khrew, Jammu & Kashmir

In general, mining industry impacts air quality provision as an ecosystem service of the forest. In the town of Khrew, Jammu & Kashmir, mining industry is the major air pollutant. According to Mukhtar & Mukhtar (2020), the population of Khrew suffers from several types of air pollutants at once: external pollution from the transport used to haul the cement from the factories as well as smog and particle pollution from the mining. This unfavourable situation leads to respiratory and cardiac complications.

However, the same detrimental influence is observed with carbon sequestration by the surrounding forests. A study by Joshi & Swami (2009) demonstrated that the rate of photosynthesis can decrease by as much as 48% in certain tree species when air pollutants are present in the area. Figure 42 exhibits the NDVI, the main proxy value of chlorophyll storage, in the form of box plots. The annual trend in the left pane of the chart shows long left tails which translates into a large number of lower values or days when the photosynthesis was significantly reduced. Likewise, the right pane presents a large number of outliers in February and March.



(a) (b)

Figure 42 (a) Annual and (b) Monthly Trends in NDVI in the Khrew area

To determine the factors contributed to this reduction, it is important to investigate the dynamics of research parameters around Khrew. We decided to expand the time frame and include the years 2019-2022 as well. Figure 43 demonstrates the dynamics of research parameters. They exhibit seasonality just as in case study 1.

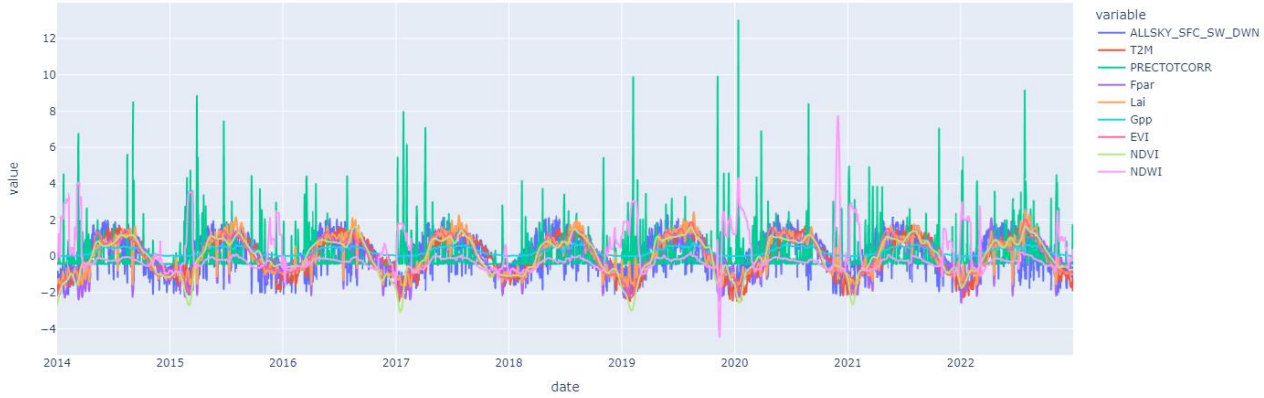


Figure 43 Research Parameters for Khrew, 2014-2022

The variation of GPP is presented in Figure 44. In 2019, the GPP has reached its maximum over the 8-year period, although subsequent years have not been as successful and showed a significant number of fluctuations over the summer month, especially in 2022.

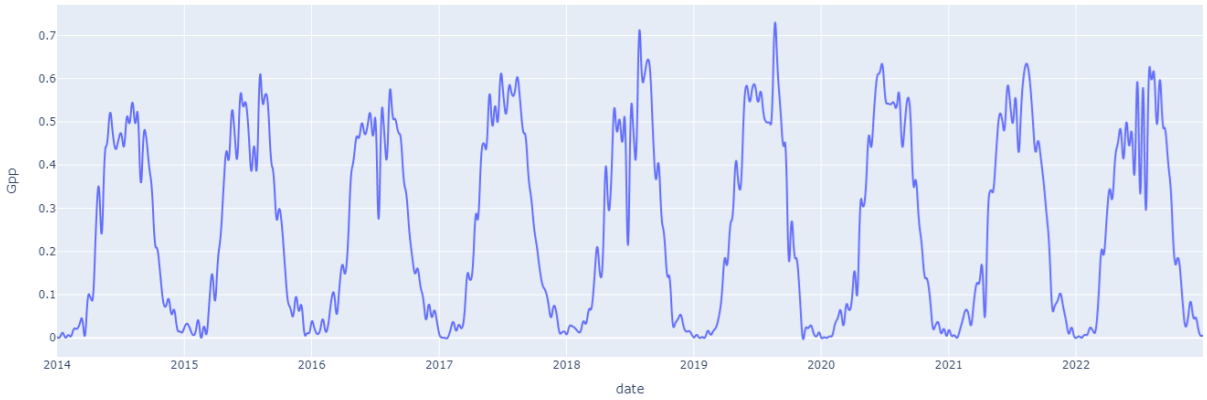


Figure 44 GPP Dynamics in Khrew

Figure 45 shows the comparison of GPP in Khrew and in Dachigam. Although the latter is a protected area, its GPP is not always higher than that in Khrew. While in the summer months, GPP tends to be higher, the fluctuations due to floods and other stressors affect the variable in Dachigam more. Due to a cumulative effect, the Fall months in Dachigam are more productive than in Khrew. The reverse is observed in Spring, especially during the time when the snow melts.

Figure 45 Comparison of GPP in Khrew and Dachigam National Park

Carbon offset is a type of measure used in sustainable management aiming at compensating for CO_2 emissions and air pollution caused by global economic processes. Comparing the actual GPP from a past period with predicted GPP for the upcoming period may be used as a signal sign for introduction of carbon offset measures, such as reforestation, avoided deforestation, or assessing efficiency of previously taken measures. Likewise, predicting GPP may assist in determining the potential negative impact of a planned new manufacturing or mining establishment or an expansion of an existing enterprise on the surrounding forest ecosystems in the affected area.

Figure 46 GPP for 2022 and Predicted GPP for 2023

We used the settings from Experiment 3 to predict GPP in Khrew for 2023. The R^2 for the ANN with LSTM layer was 0.87 which shows the robustness of predictions compared to testing data. As it follows from Figure 46, the predicted GPP is lower than that in 2022, which can be explained by cumulative effect if anthropogenic air pollution in the area. The algorithm also forecasted two declines in the Spring and one in the Summer which can be attributed to floods caused by snow-melting or summer monsoon.

The differences between predicted GPP for 2023 and actual GPP for 2022 are presented in Figure 47. It is evident that, for the majority of observations, the difference was negative which is a proof of a negative impact of the mining industry and air pollution on forest carbon sequestration capacity.

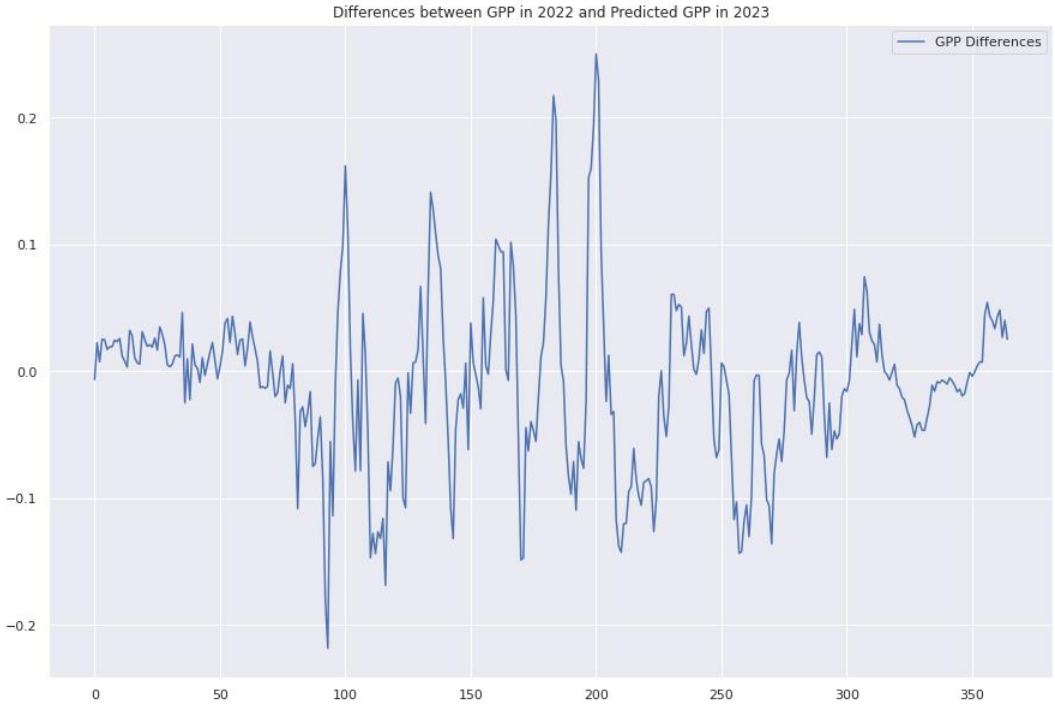


Figure 47 Difference between GPP in 2023 and 2022

Having compared the total annual GPP, we discovered that the 2023 GPP was 3.4 units less than that in 2022. From the governmental perspective, this difference can be viewed as the opportunity cost of operating mining enterprises in the area versus having no industrial activity at all. The estimation of the corresponding cost is presented in Figure 48. However, there is a way to turn these opportunity costs into an actual source of income by imposing a carbon tax. A carbon tax is a measure that covers the social costs of carbon emissions incurred by local communities through businesses operating on their territory, as the funds acquired through it can be redistributed for societal projects or redirected towards other carbon offset measures. The tax has been successfully implemented in several countries with vast forest ecosystems (e.g., Canada, Chile, Colombia and Mexico). As of 2022, India was not considering the adoption of carbon tax, but we recommend to do so as a wise measure for sustainable environmental management.

Total Annual GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
4790	t/C	17531,4	tCO2
Gramms to Tones			
4790	tonne	4790000000	grams
Grams of C per sq meter per year into Grams			
95,8	gr C/m^2 yr	4790000000	gr
50000000	m2	area of region	
50	km2	area of region	
95,8	GPP units	equivalent to gr C/m^2 yr	
GPP in 2023			

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$1 210 501,43
China National ETS	$9,00	$125 224,29
New Zealand ETS	$53,00	$737 431,90
Inflation correction	$1,26	

Mining Impact GPP Market Estimation			
Metric Tonn of C into Metric Tonne of Co2			
170	t/C	622,2	tCO2
Gramms to Tones			
170	tonne	170000000	grams
Grams of C per sq meter per year into Grams			
3,40	gr C/m^2 yr	170000000	gr
50000000	m2	area of region	
50	km2	area of region	
3,40	GPP units	equivalent to gr C/m^2 yr	
The difference between GPP in 2022 and 2023			

ETC pricing initiatives		
ETS	Price	Profit
EU ETC	$87,00	$42 961,43
China National ETS	$9,00	$4 444,29
New Zealand ETS	$53,00	$26 171,90

Market Price Alternative Losses	
ETS	Price
EU ETC	$1 253 462,86
China National ETS	$129 668,57
New Zealand ETS	$763 603,81

Figure 48 The Estimation of Opportunity Costs of the Mining Industry in Khrew

Important to note that such carbon offset measures as reforestation have a long discount period because newly planted trees must reach maturity stage before they are able to substitute the carbon sequestration capacity of the trees that have been cut down to expand mining operations. Additionally, reforestation may be conducted by planting tree species that benefit from absorbing certain pollutants (e. g. carbon dioxide, nitrogen dioxide or sulfur dioxide), thus reinforcing the measure's effect through the positive feedback. While reforestation is a crucial long-term solution, immediate measures in the meantime may include legally enforcing dust suppression or extraction practices in the mining industry.

In conclusion, this chapter summarized the results of machine learning experimentation in accordance with the designs described in Chapter 3. They demonstrate the robustness of all models in terms of the accuracy of predictions and in case of XGBoost also in terms of interpretability. We have also established an exceptional role of the Enhanced Vegetation Index (EVI) and air temperature in predictions of GPP. The impact of flood events on Dachigam National Park has been investigated by using the market prices from various Emission Trading Systems (ETS) to assess the sunk cost of GPP lost to the floods. Recommendations have been made regarding the establishment of a national ETS in India. An estimate of opportunity cost of the mining enterprises has been conducted for the city of Khrew and suggestions made as to imposing the carbon tax and enforcing the dust suppression practices.

Chapter 5: Conclusion and Future Work

The objective of this thesis is to propose a methodology for using Remote Sensing Data (RSD) for forest ecosystem service assessment in the context of Sustainable Management. The methodology is targeted at remotely-situated forest ecosystems with limited accessibility where *in situ* measurements are hard to organize. As the main characteristic of the forest ecosystems, we have chosen Gross Primary Productivity (GPP) that is the proxy measure of the carbon sequestration ecosystem services. Four Research Questions have been introduced. Below we summarize the main contributions of our research, listed in the order of respective Research Questions they answer.

The following RSD have been selected for the project: Gross Primary Productivity, Net Photosynthesis, Fraction of Absorbed Photosynthetically Active Radiations (FPAR), Leaf Area Index (LAI), Normalized Differentiated Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), Normalized Differentiated Water Index (NDWI), Short Wave Radiation, Air Temperature at 2 meters above the ground, and Precipitation. After considering their properties, we proposed two major stages for pre-processing. The first stage included RSD scaling, image pansharpening, extraction of metadata, and its transformation into a Pandas Data frame structure. The second stage involved a polynomial time series interpolation to obtain data for missing observations.

In order to ensure that the obtained variables can be used as features for machine learning (ML) experimentation, we applied several statistical tests. The first one is the five-number summary to learn about the properties of the data. Another large group of tests has been used to validate the time series' fit for prediction, explore the influence of multicollinearity on features and their relationships. The specific techniques applied included stationarity and lag plot tests,

Granger Causality tests and the Variance Inflation Factor test. It has been shown that the selected features satisfy the criteria of stationarity and causality and thus could be used for machine learning.

We investigated Random Forest Regressor, XGBoost Regressor and Artificial Neural Network with LSTM layer as candidate machine learning techniques to test their suitability for predicting RSD time series. Through experimentation, we have established that the Random Forest and XGBoost, both being ensemble learning techniques, performed very well with R^2 as high as 0.87. LSTM being a transformer algorithm performed even better with the Root Mean Square Error of 0.03. XGBoost was also used with SHAP Values Explainable AI technique to test its interpretability. On the basis of performance indicators and explainability considerations, both XGBoost and Random Forest Regressors were found to be the best match for predicting RSD time series depending on the experiment setting and purpose.

The obtained predictions were used to assess the negative impact of the 2015 flood events on the forest ecosystem of the Dachigam National Park in the Kashmir Valley, Jammu and Kashmir, India. The damages were expressed through the decline in carbon sequestration ability of the ecosystem and then estimated in monetary terms. The sunk cost was calculated based on price per tonne of carbon dioxide from three different Emission Trading Systems (ETS). In another case study, the influence of mining enterprises on carbon sequestration function of forests surrounding the village of Khrew, Jammu and Kashmir, India was estimated. The pricing technique applied here is based on an alternative cost of a carbon tax in the 2023 fiscal year. Recommendations are given to introduce the Indian National ETS as a part of sustainable management practices.

The developed methodology integrates remote sensing data fusion, methods of data analysis, such as time series interpolation and statistical tests with the use of machine learning algorithms. Nedorezov (2014) mentioned several problems of non-linear ecological models, such as finding optimal points and verifying their globality. Our model, being data-driven, accounts for the specificity of RSD in the best possible way. Unlike standard statistical models, it is built to satisfy as many constraints as possible and therefore is the best match for the problem in question.

There are several limitations to our methodology which may restrict its generalization to other ecosystems and geographical locations.

The first limitation stems from the spatial resolution of the NASA's Power Dataset which has a grid tile of twenty-five km^2. This is a large image resolution not matching the scale of other variables in the study. While the power of ML algorithms may still make this dataset applicable for use in areas larger than twenty-five km^2, in case of smaller areas, the predictions of the model may not reflect the real-life situation.

The second concern is the use of interpolation methods. The missing data so obtained may not necessarily reflect the actual fluctuations of the variables. One possible option of addressing this problem is to use datasets with a finer frequency of observations or applying different approximation techniques.

One major concern is the high multicollinearity of the features, which is present despite transformations applied to the data. Two out of three experiments involved the use of the Random Forest Regressor. While the accuracy of its predictions is unaffected by the multicollinearity of the features, it will affect the feature importance in interpretability tests. It is possible to consider in the future studies to substitute Random Forest Regressor for Lasso regression which uses a special penalization rule to avoid overtraining. As well, feature importance from Lasso will be unaffected

by multicollinearity. As to XGBoost, the performance and the interpretability will be absolutely unaffected by the multicollinearity because XGBoost is based on the principle of the boosted trees. The concern does not apply to the use of LSTM for GPP prediction, because it was univariate.

Finally, one concern stems from the time frame of up to five years chosen for the study. Additional transformations may be necessary to eliminate possible trends in longer time frames, given non-stationarity of certain features.

Future improvements to methodology can be achieved through using different data products that do not require interpolation. It is possible to use the same preprocessing steps to use the data in artificial neural networks to achieve better precision and investigate the influence of features on predictions. The methodology may be instrumental for policy- and decision-makers in sustainable forest management by integrating the predictions with economic estimates and indicators from the area of the emissions trading.

In conclusion, the proposed methodology bridges the gaps in research not addressed by prior contributions to the topic. It uses the state-of-the-art techniques of data science and machine learning and has several applications relating to ecological modelling, ecosystem service assessment and sustainable environmental management.

Refences

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. ArXiv:1603.04467: 1–19.

Ahmad, T., Pandey, A. C., & Kumar, A. (2017). Impact Of Flooding On Land Use/ Land Cover Transformation In Wular Lake And Its Environs, Kashmir Valley, India Using Geoinformatics. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, IV(4): 13–18

Alvarez, C., Teodoro, A., & Tierra, A. (2017). Evaluation of automatic cloud removal method for high elevation areas in Landsat 8 OLI images to improve environmental indexes computation. Proc. SPIE 10428, Earth Resources and Environmental Remote Sensing/GIS Applications VIII:1042809.

Amthor, J. S., & Baldocchi, D. (2001). Terrestrial higher plant respiration and net primary production. (Roy J, Saugier B, Mooney HA -Eds), Chapter 3:33–59.

Bandopadhyay, S., Pal, L., & Rahul, D. D. (2021). Predicting gross primary productivity and PsnNet over a mixed ecosystem under tropical seasonal variability: a comparative study between different machine learning models and correlation-based statistical approaches. Journal of Applied Remote Sensing, 15(01): 014523

Baniya, B., Tang, Q., Pokhrel, Y., & Xu, X. (2019). Vegetation dynamics and ecosystem service values changes at national and provincial scales in Nepal from 2000 to 2017, Environmental Development, 32: 100464.

Beer, C.; Reichstein, M.; Tomelleri, E.; Ciais, P.; Jung, M.; Carvalhais, N.; et al. (2010). "Terrestrial Gross Carbon Dioxide Uptake: Global Distribution and Covariation with Climate" (PDF). Science, 329 (5993): 834–838.

Breiman, L. (2001). Random Forests. Machine Learning 45: 5–32

Brockwell, P. J., & Davies, R. A. (2002). Introduction to Time Series and Forecasting, (2nd ed.). Springer New York.

Cerqueira, V., Torgo, L., Smailović, J., & Mozetič, I. (2017) A comparative study of performance estimation methods for time series forecasting. In: 2017 IEEE international conference on data science and advanced analytics (DSAA): 529–538.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 785 – 794.

Costanza R, d'Arge R, de Groot R, Farber S, Grasso M, Hannon B, Naeem S, Limburg K, Paruelo J, O'Neill RV, and others. 1997a. The value of the world's ecosystem services and natural capital. Nature, 387: 253-60.

CWC. (2023). Central Water Commission, Government of India. Retrieved February 22, 2023 from https://ffs.india-water.gov.in/

Davies, R. (2014). Deadly Flash Floods Strike in Jammu Kashmir. Floodlist.Com. Retrieved February 22, 2023 from https://floodlist.com/asia/deadly-flash-floods-jammu-kashmir

Davies, R. (2015). 27 Dead in Northern India and Pakistan After Week of Torrential Rain and Floods. Floodlist.Com. Retrieved February 22, 2023 from https://floodlist.com/asia/27-dead-northern-india-pakistan-torrential-rain-floods

Davies, R. (2017). India – Floods in Kashmir Valley After Jhelum River Overflows. Floodlist.Com. Retrieved February 22, 2023 from https://floodlist.com/asia/india-floods-kashmir-valley-jhelum-river-overflows

Didan, K. (2021). MODIS/Terra Vegetation Indices 16-Day L3 Global 250m SIN Grid V061 [Data set]. NASA EOSDIS Land Processes DAAC. Retrieved November 15, 2022 from https://doi.org/10.5067/MODIS/MOD13Q1.061

Dollar Times. (2023). Retrieved February 25, 2023 from https://www.dollartimes.com/inflation/inflation.php?amount=100&year=2015

ESRI. (2023). Band. In: GIS Dictionary. ESRI. Retrieved January 28, 2023 from https://support.esri.com/en-us/gis-dictionary/band

ESRI. (2023). Fitness for use. In: GIS Dictionary. ESRI. Retrieved January 28, 2023 from https://support.esri.com/en-us/gis-dictionary/fitness-for-use

ESRI. (2023). Resolution. In: GIS Dictionary. ESRI. Retrieved January 28, 2023 from https://support.esri.com/en-us/gis-dictionary/resolution

Floodlist News Desk. (2015). Deaths Feared After Flood and Landslides in Kashmir. Retrieved February 22, 2023 from https://floodlist.com/asia/deaths-feared-after-flood-and-landslides-in-kashmir.

Freeman, E. E., Moisten, G. G., Coulston, J. W., & Wilson, B. T. (2016). Random forests and stochastic gradient boosting for predicting tree canopy cover: comparing tuning processes and model performance. Canadian Journal of Forest Research, 46(3): 323-339

Friedman, J. H. (2022). Stochastic Gradient Boosting. Computational Statistics & Data Analysis 38(4): 367-378

Gao, B. (1996). "NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space" . Remote Sensing of Environment, 58 (3): 257–266.

GEE Data Catalog. (2023). Retrieved November 10, 2022 from https://developers.google.com/earth-engine/datasets/catalog

Geisser, S. (1975). The predictive sample reuse method with applications. Journal of the American Statistical Association, 70: 320–328.

Gorstko AB, Khaiter PA (1991) On a question of economic assessment of forest resources. Econ Math Meth 27(3): 522–526

Government of Jammu & Kashmir, Department of Mining. (2023). https://mining.jk.gov.in/

Granger, C. W. J. (1969). "Investigating Causal Relations by Econometric Models and Cross-spectral Methods". Econometrica, 37 (3): 424–438. doi:10.2307/1912791

Grigolato, S., Mologni, O., & Cavalli, R. (2017). GIS Applications in Forest Operations and Road Network Planning: an Overview over the Last Two Decades. Croatian Journal of Forest Engineering 38(2): 175-186

Hastie, T.; Tibshirani, R.; Friedman, J. H. (2001). The Elements of Statistical Learning. Springer Science & Business Media. p. 18.

He, H. S. (2008). Forest landscape models: definitions, characterization, and classification. Forest ecology and management, 254(3): 484-498.

Hsu C.; Chang C.; & Lin C. (2010). A practical guide to support vector classification. Technical Report, National Taiwan University.

Interpolation in Numpy. (2023). Retrieved December 5, 2022 fromhttps://docs.scipy.org/doc/scipy/tutorial/interpolate.html

James, G.; Witten, D.; Hastie, T.; & Tibshirani, R. (2017). An Introduction to Statistical Learning. 8th ed.. Springer Science+Business Media, New York

Joshi, P. C., & Swami, A. (2009). Air pollution induced changes in the photosynthetic pigments of selected plant species. Journal of Environmental Biology, 30(2): 295–298. https://www.researchgate.net/publication/41396478_Air_pollution_induced_changes_in_the_photosynthetic_pigments_of_selected_plant_species

Kankare, V., Vastaranta, M., Holopainen, M., Räty, M., Yu, X., Hyyppä, J., Alho P., & Viitala, R. (2013). Retrieval of forest aboveground biomass and stem volume with airborne scanning LiDAR. Remote Sensing, 5(5): 2257-2274.

Khaiter PA (1986) Problems of simulating the ecological-economic system of a forestry complex. In: Proceedings of the 10th Conference on mathematical modeling in the problems of rational environment use, Novorossijsk, 1986. RGU Press, Rostov-on-Don, USSR, 74–75

Khaiter PA (1989) Analyses of the anthropogenic impact on the hydrological regime in a 'forest-watershed' system. In: Proceedings of the 12th Conference on mathematical modeling in the

problems of rational environment use, Novorossijsk, 1989. RGU Press, Rostov-on-Don, USSR,: 129–130

Khaiter PA (1990) A study of the hydrological regime in a 'forest-watershed' system under anthropogenic impact. In: Proceedings of the Conference on problems in surface hydrology, Leningrad, 1990. GGI Press, Leningrad, USSR, 41–43

Khaiter PA (1991) Modeling of the anthropogenic dynamics of forest biogeocenoses. Znaniye, Kiev, Ukraine

Khaiter P.A. (1993a). Mathematical modeling in the study of the hydrological regime in 'Forest-Watershed' system. In: Proc. Int. Conf. Computational Methods in Water Resources. Boulder, Colorado, 1993. Southampton, UK: CMP: 789-794.

Khaiter PA (1993b) Decision support system 'Forest management'. In: Adey RA (ed) Proceedings of the 7th International Conference on Artificial Intelligence in Engineering. CMP, Southampton, 581–589

Khaiter PA (1996) Optimal control problem on the basis of a simulation system for environmental applications. Numerical Methods in Engineering Simulation, Merido, Venezuela. (M. Cerrolaza, C. Gajardo, C.A. Brebbia – Eds.). CMP, Southampton, UK, 297–302

Khaiter P.A (2005) "Valuing the ecological and socio-economic services in management of headwater ecosystems." In: 6th Int. Conf. Proc. Hydrology, ecology and water resources in headwaters. June 2005, Bergen, Norway: 1–10,

Khaiter, P. A., & Erechtchoukova, M. G. (2007). Environmental Assessment of Anthropogenic Impact through the Patterns of Ecosystem Stress Reactions. The International, Cultural, Economic Social Sustainability, 3(4): 181–190.

Khaiter, P.A., Erechtchoukova, M. G. (2010). A Model-Based Quantitative Assessment of Ecosystem Services in the Scenarios of Environmental Management. In: David A. Swayne, Wanhong Yang, A. A. Voinov, A. Rizzoli, T. Filatova (Eds.)/ 2010 International Congress on Environmental Modelling and Software Modelling for Environment's Sake, Fifth Biennial Meeting, 5–8 July 2010, Ottawa, Canada: 272-279.

Khaiter PA, Erechtchoukova MG (2012) Quantitative assessment of natural and anthropogenic factors in forest carbon sequestration. In: Seppelt R, Voinov AA, Lange S, Bankamp D (eds)Proceedings of the 6th International Congress on Environmental Modelling and Software (iEMSs 2012). International Environmental Modelling and Software Society, Leipzig, 2075–2082

Khaiter PA, Erechtchoukova MG (2013) Ecosystem services in environmental sustainability: a formalized approach using UML. In: Piantadosi J, Anderssen RS, Boland J (eds) Proceedings of the 20th International Congress on Modelling and Simulation (ModSim'13). Modelling and Simulation Society of Australia and New Zealand, Adelaide, SA, 1805–1811

Khaiter PA, Erechtchoukova MG (2017) Designing a software tool for environmental modelling and decision making in managing of biological invasion cases. In: Denzer R, Schimak G, Hřebíček J (eds) Environmental software systems. Springer-Verlag, Berlin, 209–222

Khaiter PA, Erechtchoukova MG (2020) Perspectives of sustainability: towards design and implementation. In: Sustainability perspectives: science, policy and practice. A global view of theories, policies and practice in sustainable development. Springer, 3–17. https://doi.org/10.870 1007/978-3-030-19550-2_1

Khaiter, P. A., & Erechtchoukova, M. G. (2022). Advanced Scientific Methods and Tools in Sustainable Forest Management: A Synergetic Perspective. In: Forest Dynamics and Conservation (Kumar M., Dhyani S., Kalra N. – Eds). Springer Singapore.: 279–312.

Kumar M, Singh MP, Singh H et al (2019a) Forest working plan for the sustainable management of forest and biodiversity in India. J Sustain For 1–22. https://doi.org/10.1080/10549811.2019.1632212

Kwiatkowski, D.; Phillips, P. C. B.; Schmidt, P.; & Shin, Y. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. Journal of Econometrics, 54 (1–3): 159–178. doi:10.1016/0304-4076(92)90104-Y

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. Advances in neural information processing systems, 4765-4774.

Luo, D., Zhang, H.K., Houborg, R., Ndekelu, L.M., Maimaitijiang, M., Tran, K.H., & McMaine, J. (2023). Utility of daily 3m Planet Fusion Surface Reflectance data for tillage practice mapping with deep learning. Science of Remote Sensing, 100085.

Malik, I. H., & Hashmi, S. N. I. (2021). The Great Flood and its Aftermath in Kashmir Valley: Impact, Consequences and Vulnerability Assessment. Journal of the Geological Society of India, 97(6): 661–669. https://www.researchgate.net/publication/352383249_The_Great_Flood_and_its_Aftermath_in_Kashmir_Valley_Impact_Consequences_and_Vulnerability_Assessment

Mann, Prem S. (1995). Introductory Statistics. 2nd ed. Wiley New York.

Matthias F.; & Frank H. Hyperparameter optimization. In: AutoML: Methods, Systems, Challenges, The Springer Series on Challenges in Machine Learning (Escalante H. J., Guyon I., Escalera S. – Eds): 3–38.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In: Proceedings of the 9th Python in Science Conference, 445.

McRoberts, R. E., Magnussen, S., Tomppo, E. O., & Chirici, G. (2011). Parametric, bootstrap, and jackknife variance estimators for the k-Nearest Neighbors technique with illustrations using forest inventory and satellite image data. Remote Sensing of Environment., 115(12): 3165-3174.

Millennium Ecosystem Assessment. (2003). Ecosystems and Human Well-being: A Framework for Assessment. Retrieved January 27, 2023 from https://www.millenniumassessment.org/en/Framework.html

Mishra, N. B., & Mainali, K. P. (2017). Greening and browning of the Himalaya: Spatial patterns and the role of climatic change and human drivers. Science of the Total Environment, 587-588: 326-339.

Mitchell, T. (1997). Machine Learning. McGraw-Hill Science/Engineering/Math.

Monteith, J.L. 1972. Solar radiation and productivity in tropical ecosystems. Journal of Applied Ecology, 9: 747—766

Montgomery, R.B. (1948) Vertical eddy flux of heat in the atmosphere. Journal of Meteorology, 5: 265–274

Mukhtar, A., & Mukhtar, F. (2020). Air Pollution a Major Threat to the People of Khrew (J&K). International Research Journal of Engineering and Technology (IRJET), 7(3): 16-20. Retrieved February 15, 2023 fromhttps://www.researchgate.net/publication/340209440_Air_Pollution_a_Major_Threat_to_the_People_of_Khrew_JK

NASA Earth Observatory. (2023). Retrieved November 10, 2022 from https://www.earthobservatory.nasa.gov/

NASA's Earth Observation System. (2022). NASA. Retrieved November 10, 2022 from https://eospso.gsfc.nasa.gov/

Nedorezov, L. V. (2014). Basic properties of ELP-model. Population Dynamics: Analysis, Modelling, Forecast, 3(4): 93-101.

Opportunity Cost. (2023). In: Oxford Learner's Dictionary. https://www.oxfordlearnersdictionaries.com/definition/english/opportunity-cost

P, Stergioudis A (2012) From desktop GIS to web-based cloud GIS: the globalization of geospatial data management. In: Proceedings Int. Symp. Modern Technologies, Education and Professional Practice in Geodesy and Related Fields Sofia, Bulgaria, 08–09 November

Pascual, A., Pukkala, T., Rodríguez, F., & De-Miguel, S. (2016). Using spatial optimization to create dynamic harvest blocks from LiDAR-based small interpretation units. Forests, 7(10): 220.Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., BrucherM., Perrot M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12: 2825-2830.

Pegov, S. (2008) Anthropospheric and Anthropogenic Impact on the Biosphere. In: Encyclopedia of Ecology, (Sven Erik Jørgensen, Brian D. Fath – Eds), Academic Press: 204-210. https://doi.org/10.1016/B978-008045405-4.00720-5.

Pei, Y., Dong, J., Zhang, Y., Yuan, W., Doughty, R., Yang, J., Zhou, D., Zhang, L., & Xiao, X. (2022). Evolution of light use efficiency models: Improvement, uncertainties, and implications. Agricultural and Forest Meteorology, 317: 108905.

Phillips, P. C. B.; & Perron, P. (1988). Testing for a Unit Root in Time Series Regression . Biometrika, 75 (2): 335–346. doi:10.1093/biomet/75.2.335

Pigou, A. C. (1920). The economics of welfare. Macmillan and Co., Limited, London.

Pohl, C., & van Genderen, J. (2017). Remote Sensing Image Fusion: A Practical Guide. Taylor & Francis, Boca Raton.

POWER (NASA Prediction of Worldwide Energy Resources). (2023). https://power.larc.nasa.gov/data-access-viewer/

Prasetyo, J. S. (2023). Stock Price Prediction Using Machine Learning with Long Short-Term Memory Method (LSTM). KILAT, 12(1).

Ramage, C. (1971). Monsoon Meteorology. International Geophysics Series, 15.: Academic Press, San Diego, CA

Ramirez-Reyes, C., Brauman, K. A., & Chaplin-Kramer, R. (2019). Reimagining the potential of Earth observations for ecosystem service assessments. Science of the Total Environment, 665: 1053-1063

Roy, T. S., Roy, J. K., & Mandal, N. (2023). Design of ear-contactless stethoscope and improvement in the performance of deep learning based on CNN to classify the heart sound. Medical & Biological Engineering & Computing, 1-23.

Running, D., Mu, Q., & Zhao, M. (2015). MOD17A2H MODIS/Terra Gross Primary Productivity 8-Day L4 Global 500m SIN Grid V006 [Data set]. NASA EOSDIS Land Processes DAAC. Retrieved November 15, 2022 from https://doi.org/10.5067/MODIS/MOD17A2H.006

Sandino, J., Pegg, G., Gonzalez, F., & Smith, G. (2018). Aerial mapping of forests affected by pathogens using UAVs, hyperspectral sensors, and artificial intelligence. Sensors, 18(4): 944.

Sarkar, D. P., Shankar, B. U., & Parida, B. R. (2022). Machine learning approach to predict terrestrial gross primary productivity using topographical and remote sensing data. Ecological Informatics, 70: 101697.

Schaaf, C., & Wang, Z. (2015). MCD43A4 MODIS/Terra+Aqua BRDF/Albedo Nadir BRDF Adjusted Ref Daily L3 Global - 500m V006 [Data set]. NASA EOSDIS Land Processes DAAC. Retrieved November 15, 2022 from https://doi.org/10.5067/MODIS/MCD43A4.006

Sepp H.; & Jürgen S. (1997). Long short-term memory. Neural Computation, 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735

Shapley, L. S. (1951). Notes on the n-Person Game -- II: The Value of an n-Person Game. RAND Corporation, Santa Monica, Calif.

Spearman C. (1904). The proof and measurement of association between two things. American Journal of Psychology, 15 (1): 72–101. doi:10.2307/1412159

Steffensen, J. F. (2006). Interpolation. (2nd ed.). Mineola, N.Y.

Stull, R. (2017). Practical Meteorology: An Algebra-based Survey of Atmospheric Science. University of British Columbia, Vancouver.

The World Bank. (2023). Carbon Pricing Dashboard. Retrieved February 17, 2023 from https://carbonpricingdashboard.worldbank.org/map_data

The World Bank. (2023). China National ETS. Retrieved February 17, 2023 from https://carbonpricingdashboard.worldbank.org/map_data

The World Bank. (2023). European Union ETS. Retrieved February 17, 2023 from https://carbonpricingdashboard.worldbank.org/map_data

The World Bank. (2023). New Zealand ETS. Retrieved February 17, 2023 from https://carbonpricingdashboard.worldbank.org/map_data

Tyagi, K., & Kumar, M. (2022). The resilience of Indian Western Himalayan forests to regime shift: Are they reaching towards no return point? Ecological Informatics, 69: 101644.

U.S. National Plant Germplasm System. (2023). Retrieved April 15, 2023 from https://npgsweb.ars-grin.gov/gringlobal/taxon/taxonomydetail?id=28538

UN FAO. (2000). Forest Resource Assessment Definitions. Retrieved January 20, 2023 from https://www.fao.org/3/y2328e/y2328e25.html

UN FAO. (2023). Sustainable Land Management. Retrieved January 20, 2023 from https://www.fao.org/land-water/land/sustainable-land-management/en/

UNGA (United Nations General Assembly) (2005) 2005 World Summit Outcome, Resolution A/60/1, adopted by the General Assembly on 15 September 2005. Retrieved April 20, 2023 from www.un.org/en/ development/desa/population/migration/generalassembly/docs/globalcompact/A_RES_60_1.pdf

Van Eck, N. J., & Waltman, L. (2011). Text mining and visualization using VOSviewer. arXiv preprint arXiv:1109.2058.

Verstraete, M. M., & Pinty, B. (1996). Designing optimal spectral indexes for remote sensing applications. IEEE Transactions on Geoscience and Remote Sensing, 34(5): 1254-1265.

Vikhar, P. A. (2016). Evolutionary algorithms: A critical review and its future prospects. In: Proceedings of the 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC). Jalgaon: 261–265.

Wang, M. H., Yu, T. C., & Ho, Y. S. (2010). A bibliometric analysis of the performance of Water Research. Scientometrics, 84(3): 813-820.

Xu, D., Zhang, Y., & Yang, Z. (2023). From Geospatial to Temporal Separation: A Review on Carbon Accounting Endogenizing Fixed Capital. Ecosystem Health and Sustainability, 9: 0002.

Zare, F., Elsawah, S., Iwanaga, T., Jakeman, A. J., & Pierce, S. A. (2017). Integrated water assessment and modelling: A bibliometric analysis of trends in the water resource sector. Journal of Hydrology, 552: 765-778.

Zhan, W., Yang, X., Ryu, Y., Dechant, B., Huang, Y., Goulas, Y., ... & Gentine, P. (2022). Two for one: Partitioning CO2 fluxes and understanding the relationship between solar-induced chlorophyll fluorescence and gross primary productivity using machine learning. Agricultural and Forest Meteorology, 321: 108980.

Zheng, Y., Zhang, L., Xiao, J., Yuan, W., Yan, M., Li, T., Zhang, Z., 2018. Sources of uncertainty in gross primary productivity simulated by light use efficiency models: Model structure, parameters, input data, and spatial resolution. Agricultural and Forest Meteorology,263: 242–257.

Zhu XJ, Yu GR, Chen Z, Zhang WK, Han L, Wang QF, Chen SP, Liu SM, Wang HM, Yan JH, Tan JL. (2023). Mapping Chinese annual gross primary productivity with eddy covariance measurements and machine learning. Science of The Total Environment, 857: 159390.

Appendices

Appendix A: Code for Feature Extraction

This appendix presents a sample of code for extracting features for the town of Khrew. Feature extraction for Dachigam National Park functions in the same way.

```
# import libs
import ee
import geemap
import pandas as pd
import altair as alt
import numpy as np
import folium
import datetime
import matplotlib as plt

# auth ee
ee.Authenticate()
ee.Initialize()
!jupyter nbextension enable --py --sys-prefix ipyleaflet
map = geemap.Map()
def applyScaleFactors(image, bands, scale, offset):
    opBands = image.select(bands).multiply(scale).add(offset)
    return image.addBands({'dstImg': opticalBands, 'overwrite': "true"})

# geometry reducer function
def create_reduce_region_function(geometry,
                                  reducer=ee.Reducer.mean(),
                                  scale=950,
                                  crs='EPSG:4326',
                                  bestEffort=True,
                                  maxPixels=1e13,
```

```
  def reduce_region_function(img):
    stat = img.reduceRegion(
        reducer=reducer,
        geometry=geometry,
        scale=scale,
        crs=crs,
        bestEffort=bestEffort,
        maxPixels=maxPixels,
        tileScale=tileScale)
    return ee.Feature(geometry, stat).set({'millis': img.date().millis()})
  return reduce_region_function

# Define a function to transfer feature properties to a dictionary. (it makes it faster to work with)
def fc_to_dict(fc):
  prop_names = fc.first().propertyNames()
  prop_lists = fc.reduceColumns(
      reducer=ee.Reducer.toList().repeat(prop_names.size()),
      selectors=prop_names).get('list')
  return ee.Dictionary.fromLists(prop_names, prop_lists)
# Function to add date variables to DataFrame.
def add_date_info(df):
  df['Timestamp'] = pd.to_datetime(df['millis'], unit='ms')
  df['Year'] = pd.DatetimeIndex(df['Timestamp']).year
  df['Month'] = pd.DatetimeIndex(df['Timestamp']).month
  df['Day'] = pd.DatetimeIndex(df['Timestamp']).day
  df['DOY'] = pd.DatetimeIndex(df['Timestamp']).dayofyear
  return df

# geographical point data
map.setCenter(75.0371223734549, 34.13767831857516, 2)
roi = ee.Geometry.Polygon(
    [[[74.995, 34.073], #top left
```

```
    [75.059, 34.069],#top right
    [75.069, 33.977],#bottom right
    [74.986, 33.984]]]) #bottom left

date_range = ee.DateRange('2014-01-01', '2022-12-31')
uni_scale = 950
map.addLayer(roi, {}, "Case 2")
map.centerObject(roi, 8)
map

# NDVI and EVI
veg_index_data = ee.ImageCollection("MODIS/061/MOD13Q1").filterDate(date_range).filterBounds(roi) #NDVI
and EVI

def applyScaleFactors(image):
   newBands = image.select(['NDVI', 'EVI']).multiply(0.0001)
   return image.addBands(newBands, None, True)

veg_data_scaled = veg_index_data.map(applyScaleFactors)
print(type(veg_data_scaled))

# visualization
ndvi = veg_data_scaled.select('NDVI')
ndvi_mean = ndvi.mean()
ndvi_viz = {'bands': 'NDVI', 'min': -2000, 'max': 10000,
        'palette': ['FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
   '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
   '012E01', '011D01', '011301']}
map.addLayer(ndvi_mean.clip(roi), ndvi_viz, 'NDVI')

evi = veg_data_scaled.select('EVI')
evi_mean = evi.mean()
evi_viz = {'bands': 'EVI', 'min': -2000, 'max': 10000,
        'palette': ['FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901',
```

```
    '66A000', '529400', '3E8601', '207401', '056201', '004C00', '023B01',
    '012E01', '011D01', '011301']}
map.addLayer(evi_mean.clip(roi), evi_viz, 'EVI')
map

# merge NDVI and EVI
ndvi_evi = ee.ImageCollection.combine(ndvi, evi)

# reduce the image using mean and scale it to 950 m
reduce_veg_ind = create_reduce_region_function(
    geometry=roi, reducer=ee.Reducer.mean(), scale=uni_scale, crs='EPSG:3310')

veg_ind_stat_fc = ee.FeatureCollection(ndvi_evi.map(reduce_veg_ind)).filter(
    ee.Filter.notNull(ndvi_evi.first().bandNames()))

# extract features to a df
# move data to a df
ndvi_evi_dict = fc_to_dict(veg_ind_stat_fc).getInfo()
ndvi_evi_df = pd.DataFrame(ndvi_evi_dict)
display(ndvi_evi_df)
print(ndvi_evi_df.dtypes)

# Remove scaling and add dates
ndvi_evi_df['NDVI'] = ndvi_evi_df['NDVI'] / uni_scale
ndvi_evi_df['EVI'] = ndvi_evi_df['EVI'] / uni_scale
ndvi_evi_df = add_date_info(ndvi_evi_df)
ndvi_evi_df.head(5)
ndvi_evi_df.to_csv(r'<path>)

# GPP - mean
gpp_data = ee.ImageCollection("MODIS/006/MOD17A2H").filterDate(date_range).filterBounds(roi)

def applyScaleFactors(image):
```

```
    newBands = image.select(['Gpp', 'PsnNet']).multiply(0.0001)
    return image.addBands(newBands, None, True)

gpp_data_scaled = gpp_data.map(applyScaleFactors)
#print(type(veg_data_scaled))

#vizualization layer
gpp = gpp_data.select('Gpp')
gpp_mean = gpp.mean()
gpp_viz = {'bands': 'Gpp', 'min': 0, 'max': 3000, 'palette': ['bbe029', '0a9501', '074b03']}
map.addLayer(gpp_mean.clip(roi), gpp_viz, 'Mean GPP')

psn_net = gpp_data.select('PsnNet')
psn_net_mean = psn_net.mean()
psn_net_viz = {'bands': 'PsnNet', 'min': -3000, 'max': 3000, 'palette': ['bbe029', '0a9501', '074b03']}
map.addLayer(psn_net_mean.clip(roi), psn_net_viz, 'Net Photosynthesis')
#map

# merge GPP and NetPsn
gpp_psnnet = ee.ImageCollection.combine(gpp, psn_net)

# reduce the image using mean and scale it to 950 m
reduce_gpp = create_reduce_region_function(
    geometry=roi, reducer=ee.Reducer.mean(), scale=uni_scale, crs='EPSG:3310')
gpp_psnnet_stat_fc = ee.FeatureCollection(gpp_psnnet.map(reduce_gpp)).filter(
    ee.Filter.notNull(gpp_psnnet.first().bandNames()))

# extract features to a df
# move data to a df
gpp_psnnet_dict = fc_to_dict(gpp_psnnet_stat_fc).getInfo()
gpp_psnnet_df = pd.DataFrame(gpp_psnnet_dict)
display(gpp_psnnet_df)
print(gpp_psnnet_df.dtypes)
```

```
# remove scaling and add dates
gpp_psnnet_df['Gpp'] = gpp_psnnet_df['Gpp'] / uni_scale
gpp_psnnet_df['PsnNet'] = gpp_psnnet_df['PsnNet'] / uni_scale
gpp_psnnet_df = add_date_info(gpp_psnnet_df)
gpp_psnnet_df.head(5)
gpp_psnnet_df.to_csv(r'<path>')

# FPAR and LAI
light_data = ee.ImageCollection("MODIS/061/MCD15A3H").filterDate(date_range).filterBounds(roi)

def applyScaleFactors(image):
    newFpar = image.select(['Fpar']).multiply(0.01)
    return image.addBands(newFpar, None, True)

light_data = light_data.map(applyScaleFactors)

def applyScaleFactors(image):
    newLai = image.select(['Lai']).multiply(0.1)
    return image.addBands(newLai, None, True)

light_data = light_data.map(applyScaleFactors)
print(type(light_data))

# visualization layers
fpar = light_data.select('Fpar')
fpar_mean = fpar.mean()
fpar_viz = {'bands': 'Fpar', 'min': 0, 'max': 100, 'palette': ['e1e4b4', '999d60', '2ec409', '0a4b06']}
map.addLayer(fpar_mean.clip(roi), fpar_viz, 'FPAR')

lai = light_data.select('Lai')
lai_mean = lai.mean()
lai_viz = {'bands': 'Lai', 'min': 0, 'max': 100, 'palette': ['e1e4b4', '999d60', '2ec409', '0a4b06']}
```

```
map.addLayer(lai_mean.clip(roi), lai_viz, 'LAI')
#map

# merge LAI and FPAR
lai_fpar = ee.ImageCollection.combine(fpar, lai)
print(type(lai_fpar))

# reduce the image using mean and scale it to 950 m
reduce_light_data = create_reduce_region_function(
    geometry=roi, reducer=ee.Reducer.mean(), scale=uni_scale, crs='EPSG:3310')

light_data_stat_fc = ee.FeatureCollection(lai_fpar.map(reduce_light_data)).filter(
    ee.Filter.notNull(lai_fpar.first().bandNames()))
print(type(light_data_stat_fc))

# extract features to a df
# move data to a df
fpar_lai_dict = fc_to_dict(light_data_stat_fc).getInfo()
fpar_lai_df = pd.DataFrame(fpar_lai_dict)
display(fpar_lai_df)
print(fpar_lai_df.dtypes)

# remove scaling and add dates
fpar_lai_df['Fpar'] = fpar_lai_df['Fpar'] / uni_scale
fpar_lai_df['Lai'] = fpar_lai_df['Lai'] / uni_scale
fpar_lai_df = add_date_info(fpar_lai_df)
fpar_lai_df.head(5)
fpar_lai_df.to_csv(r'<path>')

# NDWI
ndwi_data = ee.ImageCollection("MODIS/MCD43A4_006_NDWI").filterBounds(roi).filterDate(date_range)

#visualization layer
```

```
ndwi = ndwi_data.select('NDWI')
print(type(ndwi))
ndwi_mean =ndwi.mean()
print(type(ndwi_mean))
ndwi_viz = {'bands': 'NDWI', 'min': -1, 'max': 1, 'palette': [
    "ffffa0","f7fcb9","d9f0a3","addd8e","78c679","41ab5d",
    "238443","005b29","004b29","012b13","00120b",
  ]}
map.addLayer(ndwi_mean.clip(roi), ndwi_viz, 'NDWI')

reduce_ndwi = create_reduce_region_function(
    geometry=roi, reducer=ee.Reducer.mean(), scale=uni_scale, crs='EPSG:3310')

ndwi_stat_fc = ee.FeatureCollection(ndwi.map(reduce_ndwi)).filter(
    ee.Filter.notNull(ndwi.first().bandNames()))
print(type(ndwi_stat_fc))

# extract features to a df
# move data to a df
ndwi_dict = fc_to_dict(ndwi_stat_fc).getInfo()
ndwi_df = pd.DataFrame(ndwi_dict)
display(ndwi_df)
print(ndwi_df.dtypes)

# remove scaling and add dates
ndwi_df['NDWI'] = ndwi_df['NDWI'] / uni_scale
ndwi_df = add_date_info(ndwi_df)
ndwi_df.head(5)
ndwi_df.to_csv(r'<path>')
```

Appendix B: Code for Data Preprocessing, Data Exploration and Time Series Analysis

This code shows the preprocessing of data obtained at the previous step (see section 3.4.2 and Appendix A) and the POWER dataset, its exploration (section 3.5) and time series analysis (section 3.7).

```
# import libs
import pandas as pd
from pandas.plotting import autocorrelation_plot
from pandas.plotting import lag_plot
import altair as alt
import numpy as np
from scipy import stats
import statsmodels.api as sm
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.stattools import grangercausalitytests
from statsmodels.stats.outliers_influence import variance_inflation_factor
from joblib import Parallel, delayed
from arch.unitroot import *
from dateutil.parser import parse
import folium
import datetime
import matplotlib.pyplot as plt
import seaborn as sns


#POWER dataset loading
daily_climate_2014_df = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\01_raw\POWER_Regional_Daily_2014.csv')
daily_climate_2015_df = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\01_raw\POWER_Regional_Daily_2015.csv')
daily_climate_2016_df = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\01_raw\POWER_Regional_Daily_2016.csv')
```

```
daily_climate_2017_df =
pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\01_raw\POWER_Regional_Daily_2017.csv')
daily_climate_2018_df =
pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\01_raw\POWER_Regional_Daily_2018.csv')


#coordinates for the data and limiting the dataset to these coordinates
top_right_lat = 34.152
top_right_lon = 75.076
bottom_left_lat = 34.081
bottom_left_lon = 74.868
min_lat = 33.75
max_lat = 34.25
min_lon = 74.75
max_lon = 75.25
def trim_to_roi(df):
    roi_df = df.loc[(df['LAT'] >= 33.75)]
    roi_df = roi_df.loc[(roi_df['LAT'] <= 34.25)]
    roi_df = roi_df.loc[(roi_df['LON'] >= 74.75)]
    roi_df = roi_df.loc[(roi_df['LON'] <= 75.25)]
    return roi_df
roi_monthly_climate_df = trim_to_roi(monthly_climate_df)


#set all datasets to one dataset
daily_climate_df = pd.concat([daily_climate_2014_df,daily_climate_2015_df, daily_climate_2016_df,
                              daily_climate_2017_df, daily_climate_2018_df], axis=0)
daily_climate_df = trim_to_roi(daily_climate_df)
daily_climate_df = daily_climate_df.rename(columns={'YEAR': 'year',"MO": "month", "DY": "day"})
print(daily_climate_df.head(5))


#extract date and time
daily_climate_df['date'] = pd.to_datetime(daily_climate_df[["year", "month", "day"]])
print(daily_climate_df.head(5))


#calculate mean temperature, precipitation and shortwave radiation
```

```
sw_mean =daily_climate_df.groupby(['date'])['ALLSKY_SFC_SW_DWN'].mean()
temp_mean =daily_climate_df.groupby(['date'])['T2M'].mean()
prec_mean =daily_climate_df.groupby(['date'])['PRECTOTCORR'].mean()
mean_climate_df = pd.concat([sw_mean, temp_mean, prec_mean], axis=1)
print(mean_climate_df.head(15))
mean_climate_df.to_csv(r'<path>')

# function to create time series by year
def line_chart_by_year(df, parameter:str):
    years = df['year'].unique()
    pos = df.columns.get_loc(parameter)
    # Prep Colors
    #np.random.seed(100)
    mycolors = ['g', 'b', 'r', 'c', 'm']
    # Draw Plot
    plt.figure(figsize=(16,12), dpi= 80)
    for i, y in enumerate(years):
        if i >= 0:
            plt.plot('doy', str(parameter), data=df.loc[df['year'] == y], color=mycolors[i], label=y)
            #plt.text(df[parameter].loc[df.year==y].shape[0]-.9, df.loc[df.year==y, 'value'][-1:].values[0], y, fontsize=12,
color=mycolors[i])
    # Decoration
    par_min = df[parameter].min()
    par_max = df[parameter].max()
    plt.gca().set(xlim=(0, 366), ylim=(par_min, par_max), ylabel=parameter, xlabel='$DOY$')
    plt.yticks(fontsize=12, alpha=.7)
    plt.title("Seasonal Plot of "+parameter, fontsize=20)
    plt.show()

# function to create time series by year
def box_plot_by_year(df, parameter:str):
    years = df['year'].unique()
    # Draw Plot
    fig, axes = plt.subplots(1, 2, figsize=(20,7), dpi= 80)
```

```
    sns.boxplot(data=df, x='year', y = str(parameter), ax = axes[0])
    sns.boxplot(data=df.loc[df.year.isin(years), :], x='month', y = str(parameter), ax = axes[1])

    # Set Title
    axes[0].set_title('Annual Box Plot of '+ parameter +'\n(The Trend)', fontsize=18);
    axes[1].set_title('Monthly Box Plot of '+ parameter +'\n(The Seasonality)', fontsize=18)
    plt.show()
    # save the image
    #plt.savefig(str(parameter)+'_box_plot.png')



def decomposition(df, parameter:str):
    # Multiplicative Decomposition
    result_mul = seasonal_decompose(df[parameter], model='multiplicative', extrapolate_trend='freq')
    # Additive Decomposition
    result_add = seasonal_decompose(df[parameter], model='additive', extrapolate_trend='freq')
    # Plot
    plt.rcParams.update({'figure.figsize': (10,10)})
    result_mul.plot().suptitle('Multiplicative Decompose', fontsize=22)
    result_add.plot().suptitle('Additive Decompose', fontsize=22)
    plt.show()

def gaussian_kde(x1):
    kde1 = stats.gaussian_kde(x1)
    kde2 = stats.gaussian_kde(x1, bw_method='silverman')
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.plot(x1, np.zeros(x1.shape), 'b+', ms=20)  # rug plot
    ax.plot(x1, kde1(x1), 'k-', label="Scott's Rule")
    ax.plot(x1, kde2(x1), 'r-', label="Silverman's Rule")
    plt.show()

def acf_season(df, parameter:str):
```

```
    # detrend the data
    data = df[parameter].diff()

    #plot the detrended data
    sns.lineplot(data)
    #plot the ACF
    fig, axes = plt.subplots(1,2,figsize=(16,3), dpi= 100)
    plot_acf(data.tolist(), lags=50, ax=axes[0])
    plot_pacf(data.tolist(), lags=50, ax=axes[1])

def lag_plots(df, parameter:str):
    # Plot
    fig, axes = plt.subplots(1, 4, figsize=(10,3), sharex=True, sharey=True, dpi=100)
    for i, ax in enumerate(axes.flatten()[:4]):
        lag_plot(df[parameter], lag=i+1, ax=ax, c='firebrick')
        ax.set_title('Lag ' + str(i+1))
    plt.show()

def hist_compare(df, parameter:str, method:str, order:int):
    interpolation = df[parameter].interpolate(method=method, order=order)
    sns.displot(df[parameter].fillna(0), kde=True) # bimodal
    sns.displot(df[parameter].fillna(0), kind='kde', bw_adjust=.5)
    sns.displot(interpolation, kde=True) # bimodal
    sns.displot(interpolation, kind='kde', bw_adjust=.5)

def validation (df, parameter:str, method:str, order:int):
    interpolation = df[parameter].interpolate(method=method, order=order)
    targets = df[parameter].fillna(0)
    rmse = np.sqrt(((interpolation - targets) ** 2).mean())
    return rmse

def trend_analysis(df, parameter:str):
    # Multiplicative Decomposition
```

```
    result_mul = seasonal_decompose(df[parameter], model='multiplicative', period = 5)

    # Additive Decomposition
    result_add = seasonal_decompose(df[parameter], model='additive', period = 5)
    # Plot
    plt.rcParams.update({'figure.figsize': (10,10)})
    result_mul.plot().suptitle('Multiplicative Decompose', fontsize=22)
    result_add.plot().suptitle('Additive Decompose', fontsize=22)
    plt.show()

def adf_test(df, parameter:str):
    # ADF Test
    # H0: the time series is non-stationary
    # if the p-value is < 0.05, reject H0
    result = adfuller(df[parameter].values, autolag='AIC')
    print(f'ADF Statistic: {result[0]}')
    print(f'p-value: {result[1]}')
    for key, value in result[4].items():
        print('Critial Values:')
        print(f'   {key}, {value}')
    if result[1] < 0.05:
        print('Stationary')
    else:
        print('Non-Stationary')

def kpss_test(df, parameter:str):
    # KPSS Test
    # H0: the time series is stationary
    # if the p-value is < 0.05, reject H0
    result = kpss(df[parameter].values, regression='c')
    print('\nKPSS Statistic: %f' % result[0])
    print('p-value: %f' % result[1])
    for key, value in result[3].items():
```

```
        print('Critial Values:')
        print(f'   {key}, {value}')

    if result[1] < 0.05:
        print('Non-stationary')
    else:
        print('Stationary')

def pp_test(df, parameter:str):
    # H0: the time series is non-stationary
    # if the p-value is < 0.05, reject H0
    php_ct = PhillipsPerron(df[parameter], trend = 'ct')
    return php_ct.summary()

def granger_causality(df, parameter:str):
    # first run the AIC test to determine the optimal number of lags
    '''aic_test = adfuller(df[parameter], autolag='AIC')
    opt_lag = list(aic_test)[5]
    print(opt_lag)'''
    # granger causality test
    grangercausalitytests(df[['Gpp', parameter]], maxlag=[2])
    # H0: the second series doesn't cause the first
    # if p-value is < 0.05, reject H0

def multicollinearity_check(X, thresh=5.0):
    data_type = X.dtypes
    # print(type(data_type))
    int_cols = \
    X.select_dtypes(include=['int', 'int16', 'int32', 'int64', 'float', 'float16', 'float32', 'float64']).shape[1]
    total_cols = X.shape[1]
    try:
        if int_cols != total_cols:
            raise Exception('All the columns should be integer or float, for multicollinearity test.')
```

```
        else:
            variables = list(range(X.shape[1]))
            dropped = True
            print('"\n\nThe VIF calculator will now iterate through the features and calculate their respective values.
            It shall continue dropping the highest VIF features until all the features have VIF less than the threshold of
5.\n\n"')
            while dropped:
                dropped = False
                vif = [variance_inflation_factor(X.iloc[:, variables].values, ix) for ix in variables]
                print('\n\nvif is: ', vif)
                maxloc = vif.index(max(vif))
                if max(vif) > thresh:
                    print('dropping \'' + X.iloc[:, variables].columns[maxloc] + '\' at index: ' + str(maxloc))
                    # del variables[maxloc]
                    X.drop(X.columns[variables[maxloc]], 1, inplace=True)
                    variables = list(range(X.shape[1]))
                    dropped = True
            print('\n\nRemaining variables:\n')
            print(X.columns[variables])
            # return X.iloc[:,variables]
            return X
    except Exception as e:
        print('Error caught: ', e)


def standardize(df, parameter:str):
    df[parameter] = (df[parameter]-df[parameter].mean())/df[parameter].std()


def features_set(df, f_year: int, t_year: int):
    drop_list = ['Gpp', 'carbon_0', 'carbon_10', 'carbon_30', 'year', 'month', 'day', 'doy']
    # make a feature set for expriment 2
    f = df.loc[df['year']==f_year].reset_index(drop=True)
    f = f.drop(drop_list, axis=1)
    t = df['Gpp'].loc[df['year']==t_year].reset_index(drop=True)
    r = pd.concat([f, t], axis=1)
```

```
    return r

######## Viz Params ##########
sns.set(rc={'figure.figsize':(20,15)})

######## Load Data ###########
carbon = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\carbon.csv',
                     usecols=['b0', 'b10', 'b30', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                     dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 1 for all

climate_sw = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\climate_sw.csv') #
daily

elevation = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\elevation.csv',
                        usecols=['elevation', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                        dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 1 for all

fpar_lai = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\fpar_lai.csv',
                       usecols=['Fpar', 'Lai', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                       dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 4 days

gpp_psnnet = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\gpp_psnnet.csv',
                         usecols=['Gpp', 'PsnNet', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                         dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 8 days

ndvi_evi = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\ndvi_evi.csv',
                       usecols=['EVI', 'NDVI', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                       dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 16 days

ndwi = pd.read_csv(r'C:\Users\polsy\OneDrive\Документы\YU\Thesis\02_intermediate\ndwi.csv',
                   usecols=['NDWI', 'millis', 'Timestamp', 'Year', 'Month', 'Day', 'DOY'],
                   dtype={'Year':int, 'Month':int, 'Day':int, 'DOY':int}) # 16 days

###################### Joins ####################
```

```
features = climate_sw.merge(fpar_lai[['Timestamp', 'Fpar', 'Lai']], how='left', left_on='date', right_on='Timestamp')
features = features.merge(gpp_psnnet[['Timestamp', 'Gpp', 'PsnNet']], how='left', left_on='date',
right_on='Timestamp',
                          suffixes=('_x', '_y'))
features = features.merge(ndvi_evi[['Timestamp', 'EVI', 'NDVI']], how='left', left_on='date', right_on='Timestamp',
                          suffixes=('_a', '_b'))
features = features.merge(ndwi[['Timestamp', 'NDWI']], how='left', left_on='date', right_on='Timestamp',
                          suffixes=('_m', '_n'))
features['carbon_0'] = carbon['b0']
features['carbon_10'] = carbon['b10']
features['carbon_30'] = carbon['b30']
features = features.drop(labels=['Timestamp_x', 'Timestamp_y', 'Timestamp_m', 'Timestamp_n'],axis=1)
features['date'] =  pd.to_datetime(features['date'], yearfirst=True)
print(features.columns)
print(features.dtypes)
features['carbon_0'] = features['carbon_0'].fillna(method='ffill')
features['carbon_10'] = features['carbon_10'].fillna(method='ffill')
features['carbon_30'] = features['carbon_30'].fillna(method='ffill')



# Prepare data
features['year'] = [d.year for d in features['date']]
features['month'] = [d.month for d in features['date']]
features['day'] = [d.day for d in features['date']]
features['doy'] = [d.timetuple().tm_yday for d in features['date']]
print(features['month'].nunique())
print(features.shape)
print(features.head(20))
features.isna().sum()



########## Interpolations ###############
hist_compare(features, 'Gpp', 'polynomial', 2)
validation(features, 'Gpp', 'polynomial', 2)
```

```
# pass the interpolated values into df
features['Gpp'] = features['Gpp'].interpolate(method='polynomial', order=2)
#pad the last missing values
features['Gpp'] = features['Gpp'].fillna(method="pad")

hist_compare(features, 'Lai', 'polynomial', 2)
validation(features, 'Lai', 'polynomial', 2)
# pass the interpolated values into df
features['Lai'] = features['Lai'].interpolate(method='polynomial', order=2)
#pad the last missing values
features['Lai'] = features['Lai'].fillna(method="pad")

hist_compare(features, 'Fpar', 'polynomial', 2)
validation(features, 'Fpar', 'polynomial', 2)
# pass the interpolated values into df
features['Fpar'] = features['Fpar'].interpolate(method='polynomial', order=2)
#pad the last missing values
features['Fpar'] = features['Fpar'].fillna(method="pad")

hist_compare(features, 'PsnNet', 'polynomial', 2)
validation(features, 'PsnNet', 'polynomial', 2)
# pass the interpolated values into df
features['PsnNet'] = features['PsnNet'].interpolate(method='polynomial', order=2)
#pad the last missing values
features['PsnNet'] = features['PsnNet'].fillna(method="pad")

hist_compare(features, 'NDVI', 'polynomial', 2)
validation(features, 'NDVI','polynomial', 2)
# pass the interpolated values into df
features['NDVI'] = features['NDVI'].interpolate(method='polynomial', order=2)
# fill in the last missing values
features['NDVI'] = features['NDVI'].fillna(method='ffill')
```

```
hist_compare(features, 'EVI', 'polynomial', 2)
validation(features, 'EVI', 'polynomial', 2)
# pass the interpolated values into df
features['EVI'] = features['EVI'].interpolate(method='polynomial', order=2)
# fill in the last missing values
features['EVI'] = features['EVI'].fillna(method='ffill')

features['NDWI'] = features['NDWI'].fillna(method="pad")

features.isna().sum()

########## Yearly Visualizations ######
names = features.columns[1:11]
#for name in names:
sns.lineplot(features, x='date', y='ALLSKY_SFC_SW_DWN')

# five number summary
names = features.columns[1:11]
for name in names:
    box_plot_by_year(features, name)

########### Stationarity Checks ###################
names = features.columns[1:11]
for name in names:
    print(name)
    adf_test(features, str(name))
    kpss_test(features, str(name))

# additional testing for T2M
pp_test(features, 'T2M')

###### Test for Seasonality with ACF #############
```

```
names = features.columns[1:11]
#acf_season(features, 'T2M')
for name in names:
    print(name)
    acf_season(features, str(name))
########### Lag Plots ######################
names = features.columns[1:11]
#acf_season(features, 'T2M')
for name in names:
    print(name)
    lag_plots(features, str(name))

########### Granger Causality #################
y = 'Gpp'
x = features.columns[1:11]
x = x.drop('Gpp')
for var in x:
    print("  ")
    print(var)
    granger_causality(features, str(var))

# descriptive statistics
for var in x:
    print("  ")
    print(var)
    granger_causality(features, str(var))

#spearman rank correlation
my_r = correlate.corr(method="spearman")
print(my_r)
heatmap = sns.heatmap(my_r, vmin=-1,
                      vmax=1, annot=True)
# VIF test
```

```
vif_data = pd.DataFrame()
vif_data["feature"] = features.columns[1:11]
preds = features.iloc[:, 1:11]
factors = preds.columns
print(preds)

# calculating VIF for each feature
vif_data["VIF"] = [variance_inflation_factor(preds, factor)
                   for factor in range(len(factors))]
print(vif_data)

#standartization
names = features.columns[1:11]
names = names.drop('Gpp')
print(names)
for name in names:
    standardize(features, str(name))
print(features.head())

#droppping net photosynthesis
features = features.drop('PsnNet', axis=1)
print(features.head())

features.to_csv(r"<path>', columns=features.columns[0:10], index=0)
```

Appendix C: Code for Machine Learning Experimentation

This appendix contains a sample of code using XGBoost with the setting of the first experiment (section 3.8.5).

```
# import libraries
import pandas as pd
import time
import numpy as np
from xgboost import XGBClassifier
from sklearn.pipeline import Pipeline
from sklearn.model_selection import GridSearchCV, TimeSeriesSplit
from sklearn.model_selection import KFold
from sklearn_genetic import GASearchCV
from sklearn_genetic import ExponentialAdapter
from sklearn_genetic.space import Continuous, Categorical, Integer
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.metrics import mean_squared_error, r2_score, mean_absolute_percentage_error
import pandas as pd
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pathlib import Path
import os
from tqdm import tqdm
import seaborn as sns
import math
import warnings
warnings.filterwarnings('ignore')
import xgboost as xgb
import gc
import pickle
```

```
import shap

#reduce memory usage function
def reduce_mem_usage(df):
    """ iterate through all the columns of a dataframe and modify the data type
        to reduce memory usage.
    """
    start_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage of dataframe is {:.2f} MB'.format(start_mem))

    for col in df.columns:
        col_type = df[col].dtype

        if col_type != object:
            c_min = df[col].min()
            c_max = df[col].max()
            if str(col_type)[:3] == 'int':
                if c_min > np.iinfo(np.int8).min and c_max < np.iinfo(np.int8).max:
                    df[col] = df[col].astype(np.int8)
                elif c_min > np.iinfo(np.int16).min and c_max < np.iinfo(np.int16).max:
                    df[col] = df[col].astype(np.int16)
                elif c_min > np.iinfo(np.int32).min and c_max < np.iinfo(np.int32).max:
                    df[col] = df[col].astype(np.int32)
                elif c_min > np.iinfo(np.int64).min and c_max < np.iinfo(np.int64).max:
                    df[col] = df[col].astype(np.int64)
            else:
                if c_min > np.finfo(np.float16).min and c_max < np.finfo(np.float16).max:
                    df[col] = df[col].astype(np.float16)
                elif c_min > np.finfo(np.float32).min and c_max < np.finfo(np.float32).max:
                    df[col] = df[col].astype(np.float32)
                else:
                    df[col] = df[col].astype(np.float64)
        else:
```

```
            df[col] = df[col].astype('category')

    end_mem = df.memory_usage().sum() / 1024**2
    print('Memory usage after optimization is: {:.2f} MB'.format(end_mem))
    print('Decreased by {:.1f}%'.format(100 * (start_mem - end_mem) / start_mem))

    return df
# load feature data

features = pd.read_csv(r'<path>')
print(features.head())

#### Create X and Y training data here.....
# cut off 2018 for validation
split_date = '2018-01-01'
features_train = features.loc[features['date'] <= split_date].copy()
features_test = features.loc[features['date']> split_date].copy()
print(type(features_train))
features_train = reduce_mem_usage(features_train)
# from training dataset create X and y
X = features_train.drop(['date', 'Gpp'], axis=1)
y = features_train["Gpp"]
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42, shuffle=False)
# drop Xs and ys for memory management
del X
del y

# create a testing X and y
x_pred = features_test.drop(['date', 'Gpp'], axis=1)
y_real = features_test['Gpp']

# initialize 10-fold validation
kfold = KFold(n_splits=10, shuffle=True, random_state=10)
```

```
############## XGBoost without any Tuning ####################
# initialize the pipeline
p1 = Pipeline(steps = [('XGboost', xgb.XGBRegressor(
    objective='reg:squarederror',
    n_estimators=1000,
    learning_rate=0.03,
    max_depth=6,
    subsample=0.9,
    colsample_bytree=0.7,
    random_state=1,
    eval_metric='mae'))])

p1.fit(X_train, y_train)

# performance metrics
# performance metrics
results = cross_val_score(p1, X_train, y_train, cv=kfold)
y_test_pred = p1.predict(x_pred)

mse = mean_squared_error(y_test_pred, y_real)
r2 = r2_score(y_test_pred, y_real)
mape = mean_absolute_percentage_error(y_test_pred, y_real)
print('Cross Validation Score: ', results)
print('MSE: ', mse)
print('R2 : ', r2)
print('MAPE : ', mape)

# Shapley
shap.initjs()
explainer = shap.Explainer(p1[0], X_train)
shap_values = explainer(X_val)
shap.plots.beeswarm(shap_values)
shap.plots.bar(shap_values)
```

```
########### XGBoost with Grid Search Tuning #########
# time model performance
start_time=time.time()

# grid search
param_grid = {
      'XGboost__n_estimators':[1000, 2000],
      'XGboost__min_child_weight': [0.5, 0.6],
      'XGboost__gamma': [0, 1],
      'XGboost__colsample_bytree': [0.8, 0.9, 1], # add 1 here
      'XGboost__subsample': [0.8, 0.9, 1]
      }

'''model = xgb.XGBRegressor(
   objective='reg:squarederror',
   n_estimators=1000,
   learning_rate=0.03,
   max_depth=12,
   subsample=0.9,
   colsample_bytree=0.7,
   eval_metric='mae')'''

#kfold = KFold(n_splits=10, shuffle=True, random_state=10)
grid_search = GridSearchCV(p1,
                           param_grid=param_grid,
                           scoring="neg_mean_absolute_error",
                           cv=kfold)
grid_result = grid_search.fit(X_train, y_train)
# pickle the model
with open("xgb_grid_e1.pkl", "wb") as f:
   pickle.dump(grid_result, f)

# summarize results
```

```
#print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
print('Best Score: %s' % grid_result.best_score_)
print('Best Hyperparameters: %s' % grid_result.best_params_)
means = grid_result.cv_results_[ 'mean_test_score' ]
stds = grid_result.cv_results_[ 'std_test_score' ]
params = grid_result.cv_results_[ 'params' ]
print(time.time()-start_time)
# performance metrics
#results2 = cross_val_score(grid_search, X_train, y_train, cv=kfold)
y_test_pred_gs = grid_search.predict(x_pred)

# pickle the y_pred
with open("y_pred_xgb2_e1_case_2.pkl", "wb") as f:
   pickle.dump(y_test_pred_gs, f)
mse_gs = mean_squared_error(y_test_pred_gs, y_real)
r2_gs = r2_score(y_test_pred_gs, y_real)
mape_gs = mean_absolute_percentage_error(y_test_pred_gs, y_real)
#print('Cross Validation Score: ', results2)
print('MSE: ', mse_gs)
print('R2 : ', r2_gs)
print('MAPE : ', mape_gs)

############ XGBoost with Genetic Algorith Tuning ################
# define the params of GA
mutation_adapter = ExponentialAdapter(initial_value=0.8, end_value=0.2, adaptive_rate=0.1)
crossover_adapter = ExponentialAdapter(initial_value=0.2, end_value=0.8, adaptive_rate=0.1)

param_grid = {
      'XGboost__n_estimators': Integer(1000, 2000),
      'XGboost__min_child_weight': Continuous(0.5, 0.6, distribution='uniform'),
      'XGboost__gamma': Integer(0, 1),
      'XGboost__colsample_bytree': Continuous(0.8, 0.9, distribution='uniform'), # add 1 here
      'XGboost__subsample': Continuous(0.8, 0.9, distribution='uniform')
```

```
    }

evolved_estimator = GASearchCV(estimator=p1,
                        cv=kfold,
                        scoring="neg_mean_absolute_error",
                        population_size=10,
                        generations=20,
                        mutation_probability=mutation_adapter,
                        crossover_probability=crossover_adapter,
                        param_grid=param_grid,
                        n_jobs=-1)

# Train and optimize the estimator
evolved_estimator.fit(X_train, y_train)

# pickle the model
'"with open("ga_xgb_e1.pkl", "wb") as f:
    pickle.dump(evolved_estimator, f)'"

# summarize results
#print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
print('Best Score: %s' % evolved_estimator.best_score_)
print('Best Hyperparameters: %s' % evolved_estimator.best_params_)
means = evolved_estimator.cv_results_[ 'mean_test_score' ]
stds = evolved_estimator.cv_results_[ 'std_test_score' ]
params = evolved_estimator.cv_results_[ 'params' ]

#print(time.time()-start_time)

#pass the parameters into the model
regressor3=xgb.XGBRegressor(booster='gbtree',
                        objective='reg:squarederror',
                        eval_metric='mae',
```

```
                predictor='auto',
                gamma = evolved_estimator.best_params_['XGboost__gamma'],
                #learning_rate = grid_result.best_params_["learning_rate"],
                n_estimators  = evolved_estimator.best_params_["XGboost__n_estimators"],
                #max_depth     = grid_result.best_params_["max_depth"],
                min_child_weight = evolved_estimator.best_params_["XGboost__min_child_weight"],
                colsample_bytree = evolved_estimator.best_params_["XGboost__colsample_bytree"],
                subsample = evolved_estimator.best_params_["XGboost__subsample"])

# train the 3rd model
regressor3.fit(X_train, y_train, early_stopping_rounds=10, eval_set=[(X_val, y_val)], verbose=1)

# performance metrics
# performance metrics
results3 = cross_val_score(regressor3, X_train, y_train, cv=kfold)
y_test_pred_ga = evolved_estimator.predict(x_pred)

# pickle the y pred
with open("y_pred_xgb3_e1_case_2.pkl", "wb") as f:
    pickle.dump(y_test_pred_ga, f)

mse_ga = mean_squared_error(y_test_pred_ga, y_real)
r2_ga = r2_score(y_test_pred_ga, y_real)
mape_ga = mean_absolute_percentage_error(y_test_pred_ga, y_real)
print('Cross Validation Score: ', results3)
print('MSE: ', mse_ga)
print('R2 : ', r2_ga)
print('MAPE : ', mape_ga)
```