Contents lists available at ScienceDirect

Ecological Informatics

journal homepage: www.elsevier.com/locate/ecolinf

Assessing ensemble models for carbon sequestration and storage estimation in forests using remote sensing data

Mehdi Fasihi [a,*], Beatrice Portelli [a,b], Luca Cadez [c,d], Antonio Tomao [c], Alex Falcon [a], Giorgio Alberti [c], Giuseppe Serra [a]

[a] Department of Mathematics, Computer Science and Physics, University of Udine, Udine, Italy
[b] Department of Biology, University of Napoli Federico II, Napoli, Italy
[c] Department of Agricultural, Food, Environmental and Animal Sciences, University of Udine, Udine, Italy
[d] Department of Life Sciences, University of Trieste, Trieste, Italy

ARTICLE INFO

Keywords:
Carbon storage estimation
Carbon sequestration estimation
Ensemble models
Remote sensing
Canopy height model
National Forest inventory

ABSTRACT

Forests play a crucial role in storing much of the world's carbon (C). Accurately estimating C sequestration is essential for addressing and mitigating the impacts of global warming. While many studies have used machine learning models to estimate carbon storage (CS) in forests based on remote sensing data, this research further examines C sequestration (i.e., the annual carbon uptake by trees; CSE). The objectives of this study are two-fold: firstly, to identify the best models for estimating CSE and CS by testing various methods, and secondly, to examine the effect of climatic data and the canopy height model (CHM) on the estimation of CSE. To achieve the first objective, we will compare the performance of fourteen models, including twelve machine learning models, one deep learning model, and an ensemble model that combines the top four independent models. For the second objective, we study the effect of four input configurations: the first is a baseline configuration based solely on attributes extracted from satellite images (Sentinel-2) and geomorphology; the second combines satellite features with climatic data; the third uses a CHM derived from LiDAR instead of climatic data; and the fourth combines all available features: satellite images, climatic data, and CHM. The results show that adding climatic data does not improve the estimation of CSE and CS. However, adding CHM features significantly improves the models' performance for both targets. The implemented ensemble model demonstrated the best performance across all configurations.

1. Introduction

Forests play a fundamental role in the global carbon (C) cycle, storing 861 ± 66 Pg C, of which 363 ± 28 Pg C (42 %) in live biomass (above and below ground) (Pan et al., 2011). Forests are key natural C sinks, playing a vital role in diminishing atmospheric C levels (Rehman and Lal, 2023) (Carbon Storage by Urban Forests (U.S. National Park Service), 2024) even though their C storage has been estimated to be under the natural potential (Mo et al., 2023). Annually, over 65 % of terrestrial C sequestration (CSE) takes place in these ecosystems (Post et al., 1982). Therefore, accurate estimation of C storage (CS) and CSE in forests can provide policymakers with valuable insights into the best strategies to address greenhouse gas emissions, contributing to current and future climate change mitigation (Dai et al., 2021). According to the Intergovernmental Panel on Climate Change (IPCC), various C pools have been recognized (Pan et al., 2011). Notably, plant biomass, encompassing both above-ground and below-ground components, serves as the primary means for removing CO_2 from the atmosphere (Pan et al., 2011).

Biomass derived from forests can be quantified through field measurements, utilizing both destructive methods (which involve tree harvesting, as seen in references (Goetz et al., 2007; Konda et al., 2017)) and non-destructive approaches. The latter relies on allometric equations to convert diameter and height to volume, biomass, and subsequently to C. The C sequestration effect resulting from tree growth can be gauged through periodic field surveys (i.e. repeated forest inventories) or through tree rings analysis (i.e. dendro-chronology), where the annual volume increment is measured and converted into C

[*] Corresponding author.
E-mail addresses: Fasihi.mehdi@spes.uniud.it (M. Fasihi), portelli.beatrice@spes.uniud.it (B. Portelli), luca.cadez@uniud.it (L. Cadez), antonio.tomao@uniud.it (A. Tomao), falcon.alex@spes.uniud.it (A. Falcon), giorgio.alberti@uniud.it (G. Alberti), giuseppe.serra@uniud.it (G. Serra).

https://doi.org/10.1016/j.ecoinf.2024.102828
Received 23 February 2024; Received in revised form 10 August 2024; Accepted 15 September 2024
Available online 19 September 2024

sequestered by tree growth (Krug et al., n.d.). However, the methods currently employed by National Forestry Inventories, though accurate, are not practical for extensive assessments of large areas as they allow estimation of growth at forest category level, only. Consequently, remote sensing has emerged as a crucial tool for estimating and mapping forest biomass, rapidly becoming a widely recognized tool for monitoring sustainability in forestry (Fardusi et al., 2017; Estoque, 2020).

Remote sensing techniques use satellites, aircraft, and unmanned vehicles to observe and analyze qualitative and quantitative characteristics from a distance (*What is Remote Sensing and What is it Used for? | U. S. Geological Survey*, 2024), providing data over large areas and enabling access to inaccessible places. Optical sensors, radar, and LiDAR systems are among the technologies employed for this purpose (Zhao et al., 2016). In this regard, the estimation of biomass C with LiDAR is considered more accurate when compared to passive sensors (Stelmaszczuk-Górska et al., 2015). However, LiDAR data availability is still limited in spatial and temporal coverage due to the high acquisition costs, data volumes, and high data pre-processing requirements (Zhu and Liu, 2015). Therefore, it is crucial to investigate the CSE and CS estimation capabilities of cheaper yet accurate systems, such as optical sensors, and systematically compare them to LiDAR.

The usage of remote sensing data to estimate forest attributes, such as CS, has sparked ample research interest in recent years thanks to machine learning techniques (Zhang et al., 2022) (Dai et al., 2021). Indeed, machine learning methods can better capture non-linear relationships between biomass and multiple environmental covariates, if compared to parametric approaches (e.g., logistic regression and perceptron) (Gao and Hailu, 2012). However, there is still a lack of extensive research on applying machine learning methods to estimate CSE, despite some relevant studies being conducted in agricultural or urban contexts (e.g., (Wang et al., 2022); (Uniyal et al., 2022)). In this regard, a major issue is also the choice of the prediction method for CS or CSE estimation (Safari et al., 2017a). Indeed, although some algorithms such as Random Forest demonstrated to be very promising (Chirici et al., 2020), there is no definitive consensus on the most suitable method, as algorithm performance can vary based on sample size, location, and validation procedures (Safari et al., 2017a; Zhu and Liu, 2015).

Bearing in mind all these considerations, this paper aims at (i) identifying the most suitable model for estimating CS and CSE at a large scale in the Friuli Venezia Giulia region (Italy), and (ii) examining the effect of different combinations of input features on the estimation of the target variables.

For the first objective, we selected a total of fourteen algorithms. Among these algorithms, there are twelve machine learning models, specifically: Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Multilayer Perceptron (MLP), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), extreme Gradient Boosting (XGBoost), Categorical Boosting (CatBoost), Bayesian Neural Network (BayesianNN), Stack Ensemble (StackEns), Light Gradient Boosting (LightGBM), Adaptive Boosting (AdaBoost), and Bagged Decision Trees (BaggedDT). In addition to experimenting with machine learning models, we also tested several standard deep learning models for image processing. Among these, the VGG16 model delivered the best performance, and we refer to this model as DeepCNN in our presentation of the results. Finally, we developed a model called “Ensemble,” which combines the best four singular models: RF, CatBoost, AdaBoost, and StackEns.

To address the second objective, we created four different configurations of input features. The first input configuration serves as a baseline and consists of features extracted from geomorphology data and satellite images only. The second configuration combines the satellite features with the ones derived from climatic data, under the hypothesis that climatic conditions influence both CS and CSE. The third configuration uses satellite features and CHM derived from high-resolution LiDAR data, to understand to what extent LiDAR data can improve results of CS and CSE estimations over the whole region. Finally, the fourth configuration employs all the previously mentioned features, to study their overall interaction.

This study makes four primary contributions. Firstly, it offers a comprehensive comparison of fourteen competitive models for CS and CSE estimation. Secondly, it introduces a new ensemble method for CS and CSE estimation that surpasses the performance of most of the models. Thirdly, it underscores the necessity of tailored model selection based on specific site characteristics and data inputs, emphasizing the context-dependent nature of model performance. Lastly, the research highlights the significance of integrating diverse input configurations, such as satellite features, climatic data, and CHM derived from LiDAR, providing valuable insights for enhancing the precision of carbon estimation for operational purposes.

2. Materials and methods

2.1. *Study area*

The research was conducted in the Autonomous Region of Friuli Venezia Giulia in North-East Italy (Fig. 1). The study area spans a total area of 3273 km^2, accounting for approximately 41 % of the region’s surface. Within this region, most of the forests are situated in hilly and mountainous areas. However, since 1861, the forested area has doubled due to depopulation and the subsequent abandonment of traditional agro-forestry practices. In contrast, the presence of woods in the plains area is relatively lower, primarily due to the intensification of agricultural activities over time. Land reorganization and reclamation efforts have significantly reduced the semi-natural areas in the plain, which are now predominantly found along rivers or in the Karst region, sharing similarities with the hilly area.

2.2. *Data*

2.2.1. *Dependent variables*

The objective of this study is to estimate CS and CSE. The reference data for these two variables were obtained from the Third Italian National Forest Inventory (INFC) conducted in 2015, which employed a three-phase systematic sampling design, and the data was collected locally between 2017 and 2019 (Gasparini and Papitto, 2022). The INFC consists of three phases. The first phase regards the preliminary classification of land use and land cover through the photointerpretation of orthophotos at over 301,000 points, one for each mesh of the 1 km × 1 km grid in which the national territory has been divided. The second sampling phase involves a subsample of the first phase points, selected according to a sampling stratified by region and class of land use and land cover. In the third phase, strata are identified by the forest type assigned in the second phase together with the land use and cover class, and the region. For each forest stratum, a subsample of points is extracted to carry out field measurements. For the Third INFC, these field data were collected between 2017 and 2019. Dendrometric data are collected in two concentric plots (4 and 13 m-radius depending on tree diameter) to derive quantitative parameters, including, among others, annual volume increment (m^3 ha^{-1} y^{-1}), CS in both living and dead biomass (tC ha^{-1}), and CSE by living trees (tC ha^{-1} yr^{-1}). In particular, CSE was derived from annual volume increment, which was measured on a sub-sample of trees in each survey plot. Such trees were cored with an increment borer 1.30 m from the ground. Only one core is taken per tree and the diameter increment in the last five annual rings (excluding the current year ring) was measured with a ruler and then converted into volume increment using allometric Eqs. (Gasparini and Papitto, 2022). Finally, the increment of volume was converted into C uptake according to IPCC guidelines (Krug et al., n.d.). Table 1 provides the statistical distribution of the CSE and CS variables in the 279 inventory plots that comprise our dataset.

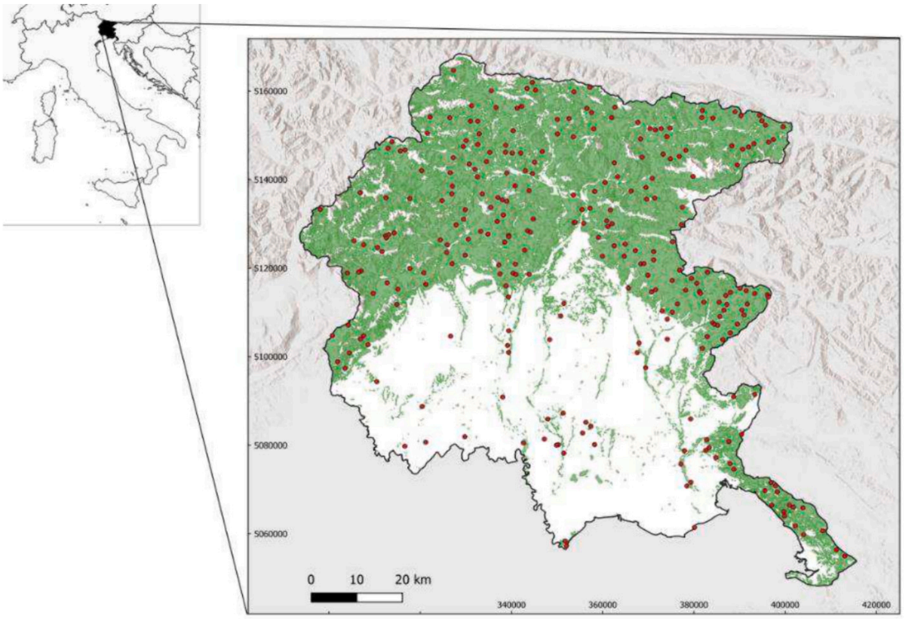

Fig. 1. The Friuli Venezia Giulia region is in the northeast of Italy, and the forest surface covers over 327.000 ha. The original 279 sampling plots of the National Forest inventory used are represented with red dots. Reference system RDN2008 / UTM zone 33 N (EPSG 6708). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Statistics related to the distribution of target variables in the data. The table reports the average value (avg), standard deviation (std), minimum (min), maximum (max), 25th, 50th, and 75th percentiles (Q1, Q2, Q3).

Statistic	CSE (tC ha^{-1} yr^{-1})	CS (tC ha^{-1})
Avg	1.68	76.80
Std	1.21	61.63
Min	0.01	0.20
Max	6.50	294.04
Q1	0.74	30.54
Q2	1.44	60.33
Q3	2.37	107.57

2.2.2. *Predictors*

We considered twelve predictors (Table 2) for the estimation of the target variables. Fig. 2 shows the flowchart for how the predictors were extracted from the input data. The spatial indexes have been produced in Google Earth Engine, while the other data are managed through the cartographic software QGIS. Particularly, the input data have been sampled on the National inventory plot of 13 m of radius. So, the dataset contains both the ground truth of forestry inventory and the values of each predictor. In detail, the twelve variables can be categorized into three groups.

The first group, called “Satellite” comprises four spectral indices commonly used in forestry research, namely Vegetation Indices. These are derived from remote sensing data and provide effective and

Table 2
Names and descriptions of the twelve input features. The table reports the average value (avg), standard deviation (std), median (med), and maximum (max).

Type	Name	Meaning	Statistics
Satellite -Sentinel-2	NDVI	Normalized Difference Vegetation Index $(NIR - Red)/(NIR + Red)$	max, med, avg., std.
	NDII	Normalized Difference Infrared Index $(NIR - SWIR)/(NIR + SWIR)$	max, med, avg., std.
	GNDVI	Green-NDVI $(NIR - Green)/(NIR + Green)$	max, med, avg., std.
	EVI	Enhanced Vegetation Index (*EVI (Enhanced Vegetation Index) | Sentinel Hub Custom Scripts*, 2024) $2.5^{*}((NIR - Read)/(NIR + 6^{*}Red - 7.5^{*}Blue + 1))$	max, med, avg., std.
DEM and derivative	ELE	Digital Terrain Model at 10 m	max, med, avg., std.
	SLO	Slope (percentage)	max, med, avg., std.
	ASP	Aspect (degree)	max, med, avg., std.
	CHM	Canopy Height Model (DSM first - DTM)	max, med, avg., std.
Climatic - EURO-CORDEX project. Data 2019–2015	TEMP summer	Average temperature at soil during summer months (June, July, August)	max, min, avg.
	TEMP spring	Average temperature at soil during spring months (March, April, May)	max, min, avg.
	PREC summer	Average precipitation during summer months (June, July, August)	avg
	PREC spring	Average precipitation during spring months (March, April, May)	avg

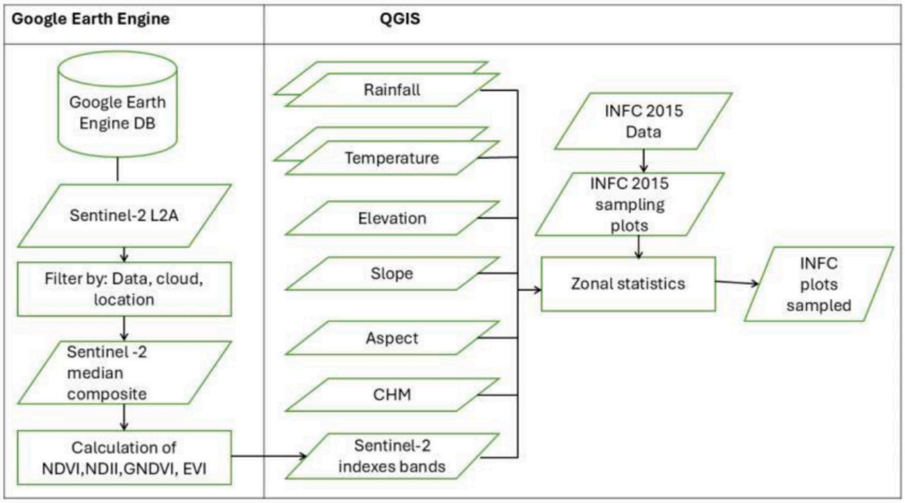

Fig. 2. Flowchart for input data production, and the software involved.

straightforward algorithms for assessing vegetation cover, vigor, and growth dynamics (Xue and Su, 2017). To obtain these indices, we utilized the Google Earth Engine (Gorelick et al., 2017) platform to create a median dataset using spectral bands from June, July, and August within the 2019–2021 timeframe, with a resolution of 10 m. While this timeframe extends slightly beyond the one covered by the dependent variables in the forest inventory, it was selected to address gaps resulting from cloud coverage. NDVI is the most widespread spectral index used in the biological field and can be used to estimate various vegetation properties, including biomass and plant productivity (Huang et al., 2021). NDII is a water-sensitive index designed to detect changes in the water content of plant canopies. As the water content increases, the index values also rise accordingly (*Landsat Enhanced Vegetation Index | U. S. Geological Survey*, 2024). GNDVI is a modified version of NDVI, more sensitive to variation in chlorophyll content than NDVI (*NDII (Normalized Difference 819/1600) | Sentinel Hub Custom Scripts*, 2024). EVI is similar to NDVI and can be used to quantify vegetation greenness; it is more sensitive in areas with dense vegetation. (*Landsat Enhanced Vegetation Index | U.S. Geological Survey*, 2024).

The second group of predictors pertains to three geomorphological parameters: elevation (ELE), obtained from a Digital Terrain Model (DTM) (Chirici et al., 2020) with a resolution of 10 m, and its two derivatives, percentage slope (SLO) and aspect expressed in degrees (ASP). Additionally, we utilized a Canopy Height Model (CHM) with a resolution of 0.5 m, which represents the vegetation height above the ground. In this case, the CHM was calculated by subtracting the Digital Surface Model (DSM) derived from the first LiDAR pulse that records the top of objects from the DTM (Guth et al., 2021) with data collected between 2017 and 2020 through an Aerial Laser Scanning (ALS) survey. This calculation was performed only for wooded areas - defined by the regional map of forest categories (Friuli Venezia Giulia Autonomous Region, 2024) - as a convenient method to obtain the information without processing all the original LiDAR data.

The third group of predictors focuses on climatic parameters, specifically temperature, and rainfall, generated through the EURO-CORDEX project. These parameters represent the latest advancements in regional climate models (RCM) at a European scale, offering high spatial resolution. For our study, we utilized the RCP4.5 scenario for the 2015–2019 timeframe, which was provided by the Regional Environmental Protection Agency of Friuli Venezia Giulia. The data was provided in the form of NetCDF files, with temperature represented at a grid resolution of 500 m and rainfall at 5000 m.

To obtain the predictors statistics outlined in Table 2, we utilized the Geographic Information System (GIS) software QGIS. Using a buffer with a radius of 13 m, we sampled the statistics starting from the coordinates of the INFC inventory points. This buffer ensured that the sampled surface matched the inventory surface, which was 530 m^2. The initial dataset consisted of 284 inventory points. We then removed areas where any of the variables listed in Table 2 were unavailable or could potentially provide misleading results. Additionally, we excluded areas affected by the 2018 storm Vaia (Chirici et al., 2019), as the sample within those areas would not align with the remote sensing data, particularly the CHM derived from LiDAR data. After these exclusions, the final dataset, therefore, comprised a total of 279 points distributed across the entire region, as shown in Fig. 1.

In addition to the tabular dataset, we also created an image-based dataset to train the deep learning models. The original data for all predictors described above have been upscaled or downscaled to a resolution of 10 m, and their grids have been aligned. Then, for each of the 279 inventory points, we extracted an input patch of 32 × 32 pixels centered on the point itself to be used as input for the model. The resolution and input size were chosen to provide enough contextual information to enhance the model's ability to capture and learn spatial patterns, while also maintaining focus on the inventory area.

According to the twelve predictors listed above, four input configurations were investigated to evaluate the impact of various input features on the estimation of CSE and CS:

- Configuration 1 (Conf1): Includes NDVI, NDII, GNDVI, EVI, ELE, SLO, and ASP.
- Configuration 2 (Conf2): Contains all the features from Conf1 along with additional climatic features.
- Configuration 3 (Conf3): Encompasses all the features from Conf1 along with the CHM (Canopy Height Model) feature.
- Configuration 4 (Conf4): Utilizes all the available features without any specific constraints.

2.3. *Methodology*

In this research, we aim to provide a comprehensive comparison of machine learning and deep learning models for the estimation of CSE and CS. To achieve this, we implemented twelve different algorithms from the literature, including ensemble models, and developed a new ensemble model. Additionally, we utilized a DeepCNN model on an image-based version of the dataset. This thorough examination of the models' performance allows us to accurately select the most suitable one for this study's final analysis and conclusion.

2.3.1. Support vector regression (SVR)

SVR is a machine learning model based on Vapnik–Chervonenkis dimension theory, which aims to strike a balance between the complexity of the model and its learning ability to achieve optimal performance. Nevertheless, selecting the appropriate kernel function poses a significant hurdle when aiming to accurately estimate CSE. To tackle this challenge, we employed a grid-search technique to determine the most suitable kernel function, along with other essential parameters like SVR type and penalty parameters.

2.3.2. K-nearest neighbors (KNN)

KNN determines the output by averaging the values of its k "nearest neighbors" for the given input object (*k-nearest neighbors algorithm - Wikipedia*, 2024). These "nearest neighbors" are k objects with the most similar features to the input object. The value of k is a crucial parameter in achieving an accurate estimation of CSE. Greater values of k decrease the impact of noise on estimation, yet they can also result in less well-defined boundaries between classes, ultimately causing suboptimal performance (Xue and Su, 2017). To determine the optimal value of k, we employed the grid-search technique. This involved evaluating different values of k to identify the one that yields the highest performance.

2.3.3. Multilayer perceptron (MLP)

We selected the MLP model from the family of artificial neural networks as the most representative one. The MLP is a feedforward neural network that includes an input layer, an output layer, and one or more hidden layers. In this model, we applied the Rectified Linear Activation Function (ReLU) as a standard approach to introduce nonlinearity into the network, allowing it to learn more complex features. During the training stage, the model tries to output a value that is close to the target value and then updates the weight matrix accordingly. The main objective of this model is to find a relationship between the input variable and the output variable (Gao et al., 2018).

2.3.4. Random forest (RF)

RF stands as an ensemble learning technique pivotal for classification and regression tasks. This method coordinates a multitude of decision trees during training, each constructed on a random subset of features. Employing a technique called bootstrap aggregating (bagging), RF generates diverse training datasets by repeatedly sampling from the original data (Breiman, 2001). The core strength of RF lies in its ability to blend the predictions of these individual trees through either majority voting (for classification) or averaging (for regression), thus enhancing overall predictive accuracy and resilience against overfitting (Breiman, 2001). Moreover, RF offers tunable parameters such as the number of trees in the forest and the maximum depth of each tree. Skillful parameter tuning is vital to balancing model complexity and performance (Feurer and Hutter, 2019). RF's capacity to handle high-dimensional data, mitigate overfitting, and provide valuable insights into feature importance renders it indispensable in predictive modeling endeavors (Obata et al., 2021).

2.3.5. Gradient boosting decision tree (GBDT)

GBDT is a popular machine learning technique used for both regression and classification tasks. It is an ensemble learning method that combines the predictions of multiple weak learners, typically decision trees, to create a strong learner. In GBDT, the weak learners are typically shallow decision trees, also called decision stumps, which are simple decision trees with a single split (Friedman, 2001). The algorithm works by sequentially adding these weak learners to the ensemble, with each subsequent learner focusing on the mistakes made by the previous ones (Friedman, 2001). The term "gradient boosting" refers to the optimization algorithm used in GBDT to minimize the loss function of the model. It works by iteratively fitting new models to the residuals or errors made by the previous models in the ensemble (Friedman, 2002). This iterative process continues until a specified number of weak learners have been added or until a certain level of performance is achieved (Friedman, 2002).

2.3.6. eXtreme gradient boosting (XGBoost)

XGBoost is an extension of GBDT that uses distributed multiple decision trees for solving classification or regression problems (Chen and Guestrin, 2016). One of the key advantages of this algorithm over traditional GBDT is the introduction of a regularization term in the objective function. This term applies a generalization performance constraint, which helps to reduce overfitting, thereby improving the model's ability to generalize to new data (Huang et al., 2022). As a result, XGBoost has become a popular choice for various applications, including image and speech recognition, natural language processing, and financial modeling.

2.3.7. Categorical boosting (CatBoost)

CatBoost is a popular member of the boosting algorithm family and is considered an alternative to XGBoost. One of the main features of CatBoost is its ability to handle categorical features by using a permutation-driven approach, which can lead to better accuracy than traditional algorithms (Prokhorenkova et al., 2017). CatBoost also offers several other improvements over previous boosting models, including simpler hyperparameter tuning and faster processing times than XGBoost (Huang et al., 2022). These advantages make CatBoost also another useful tool for a variety of applications.

2.3.8. Bayesian neural network (BayesianNN)

BayesianNN is a type of neural network that uses Bayesian inference to estimate the probability distributions of the network's weights instead of fixed values (Izmailov et al., 2021). This approach allows the network to measure the uncertainty in its predictions, making it more robust and reliable, especially with limited or noisy data (Izmailov et al., 2021). BayesianNNs are particularly useful in applications where understanding prediction confidence is crucial, such as medical diagnosis and autonomous driving (Sagi and Rokach, 2018).

2.3.9. Stack Ensemble (StackEns)

StackEns is an advanced machine-learning technique that combines multiple base models to improve predictive performance. In StackEns, the predictions of individual models are used as inputs for a meta-model, which learns to make the final prediction (Faska et al., 2023). This method leverages the strengths of different models and mitigates their weaknesses, leading to more accurate and robust results (Faska et al., 2023). StackEns is particularly useful in competitions and complex predictive tasks where maximizing model performance is crucial.

2.3.10. Light gradient boosting machine (LightGBM)

LightGBM is a highly efficient and scalable gradient-boosting framework designed by Microsoft. It is particularly well-suited for handling large-scale data and high-dimensional features due to its innovative techniques like Gradient-based one-sided sampling (GOSS) and Exclusive Feature Bundling (EFB) (Ke et al., n.d.). It supports various advanced functionalities such as categorical feature handling and parallel learning, making it a popular choice for competitive machine learning tasks and real-world applications (Ke et al., n.d.).

2.3.11. Adaptive boosting (AdaBoost)

AdaBoost is a powerful ensemble learning algorithm developed by Yoav Freund and Robert Schapire (Schapire, 2003). It enhances weak classifiers by combining them sequentially to form a strong classifier. AdaBoost adjusts the weights of training examples, focusing on those misclassified in previous rounds, and iteratively improves the model's accuracy (Schapire, 2003). This method is particularly effective in boosting the performance of binary classifiers.

2.3.12. Bagged decision trees (BaggedDT)

BaggedDT is an ensemble learning method that improves decision tree performance and robustness by reducing variance and preventing overfitting (Bbeiman, 1996). Through bagging (Bootstrap Aggregating), multiple decision trees are trained on random subsets of the training data. The final prediction is an average of the trees' outputs (regression) or a majority vote (classification), resulting in more accurate and stable predictions than a single decision tree (Bbeiman, 1996).

2.3.13. Deep convolutional neural network (DeepCNN)

The literature offers a wide range of deep learning models based on convolutional neural networks (DeepCNN), trained on image inputs. Since the dataset contains a limited number of samples, we focused on pre-trained models. Preliminary experiments were performed using VGG11 (Simonyan and Zisserman, 2014), VGG16 (Simonyan and Zisserman, 2014), ResNet18 (He et al., 2015), ResNet50 (He et al., 2015), MobileNet_V3_Small (Howard et al., 2019), and EfficientNet_V2_S (Tan and Le, 2021). VGG16 showed the best performance and was therefore selected as the representative DeepCNN for the following experiments.

VGG is a CNN-based architecture available in various depths, including 11 and 16 layers. The initial part of the architecture comprises a stack of convolutional layers (thirteen for VGG16) designed to identify spatial patterns in images. These convolutional layers are followed by three fully connected layers that learn a mapping function to solve the prediction problem. VGG was originally designed for classification tasks, so the last fully connected layer has one neuron for each output class. Since VGG was developed for RGB images (three input channels), the first convolutional layer was modified to accommodate a larger number of inputs depending on the input configuration (e.g., seven input channels for Conf1 and twelve input channels for Conf4). Additionally, the last fully connected layer was modified to output a single value (CS or CSE) to adapt the model for regression tasks.

The VGG16 model can be initialized either with random weights or with pre-trained weights, tuned on the large-scale image dataset ImageNet. The pre-trained weights provide a more stable initial model for further training on small-sized datasets. Furthermore, the pre-trained weights can either be frozen, preserving them and preventing further updates, or left unfrozen to allow further training on the new data.

2.3.14. Ensemble model

Averaging predictions is a widely accepted method for creating ensemble models, where the predictions of multiple individual models are combined to produce a single, robust prediction. This technique leverages the strengths of each model, mitigating individual weaknesses and reducing overall prediction variance. The ensemble prediction for a given data point is calculated by taking the mean of the predictions from all constituent models. Mathematically, this can be expressed as follows (Eq. (1)):

$$\widetilde{y}_{ensemble,i} = \frac{1}{N}\sum_{j=1}^{N}\widetilde{y}_{j,i} \tag{1}$$

where $\widetilde{y}_{ensemble,i}$ is the ensemble prediction for the $i-th$ data point, N is the number of models, and $\widetilde{y}_{j,i}$ is the prediction of $j-th$ the model for the $i-th$ data point. By averaging these predictions, the ensemble model aims to provide more accurate and reliable predictions compared to any single model, effectively capturing the underlying patterns in the data more comprehensively. Based on the Friedman test (Section 3.1.2), this research selected the following top four models for the ensemble: RF, CatBoost, AdaBoost, and StackEns. These models showed superior individual performance, and their combined predictions create a robust ensemble model.

2.4. Experimental setup

We employed a spatial block hold-out strategy (Roberts et al., 2017) to address the challenge of spatial autocorrelation in our predictive modeling. This method involves partitioning the study area into ten distinct spatial blocks, which were selected using the K-means clustering algorithm. Fig. 3 shows the resulting spatial clusters. After generating these ten blocks, they are divided into five folds, with each fold containing two blocks. We use 5-fold cross-validation to evaluate all models, where each iteration involves training the model on four folds (comprising eight blocks) and testing on the remaining fold (comprising two blocks). This operation is performed five times, with different blocks designated for training and testing in each iteration. By averaging the performance metrics across these five iterations, we obtained a robust and reliable estimate of the model's accuracy and generalization capabilities. This approach minimizes the risk of overfitting and provides a more realistic estimate of how the model will perform on new, unseen spatial data (Roberts et al., 2017). The block hold-out strategy thus enhances the model's ability to generalize and ensures that its predictive accuracy is not biased by spatial dependencies within the dataset.

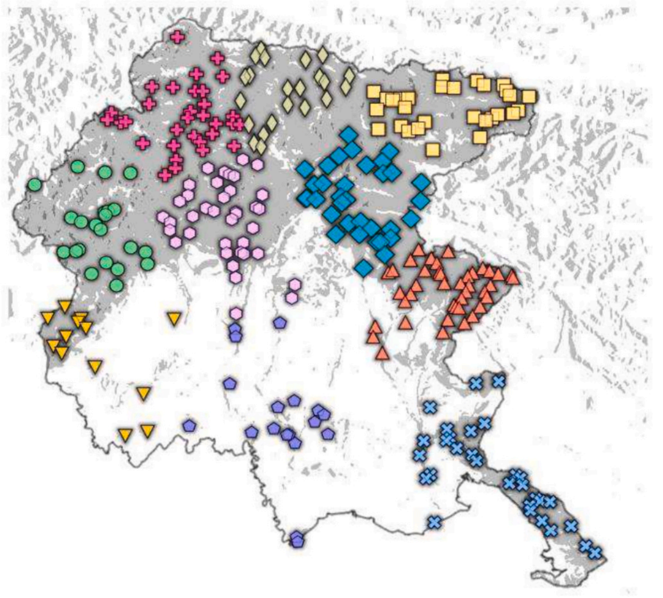

Fig. 3. The ten clusters used for the spatial block hold-out training strategy. Each block contains between 13 and 41 inventory points.

Furthermore, we employ the grid search method (Feurer and Hutter, 2019) to maximize the performance of our models and enhance our estimates for CS and CSE. Grid search involves creating a grid of hyperparameter values and fitting the models according to every possible combination (Feurer and Hutter, 2019). This approach is beneficial as it systematically explores various hyperparameter combinations, allowing us to identify the optimal configuration for our models. Grid search was also performed employing a separate 5-fold cross-validation on the spatial blocks to identify the overall best hyperparameters. Appendix A contains details regarding the range of hyperparameter values explored, as well as the optimal hyperparameters for all models.

2.5. Evaluation metrics

We consider three error measurements for the evaluation of the models, including coefficient of determination (R^2), root-mean-square error (RMSE), and root-mean-square error percentage (%RMSE). Generally, the lower values of RMSE and %RMSE show a better performance while a better estimation performance happens for a higher R^2. Eqs. (2)–(4) show R^2, RMSE, and %RMSE respectively. In these Equations, $\hat{y}$ refers to the predicted value, y_i is the measured observed value, $\bar{y}$ represents the mean of the observed values, and n is the test

Table 3
Performance of the considered models for the target variable CS, over all input configurations. The reported metrics are R^2 (higher is better), RMSE (lower is better), and %RMSE (lower is better). The best result is bolded. Average is the average performance of 13 models for each metric.

Input Features	Model	R^2	RMSE	%RMSE
Conf1 (Sentinel-2)	AdaBoost	0.32 ± 0.09	49.82 ± 11.06	65.08 ± 15.61
	BaggedDT	0.31 ± 0.08	50.11 ± 9.59	65.44 ± 13.82
	BayesianNN	0.11 ± 0.10	57.06 ± 12.68	74.58 ± 18.20
	CatBoost	0.32 ± 0.07	49.64 ± 9.95	64.86 ± 14.44
	DeepCNN	0.23 ± 0.11	53.04 ± 12.64	69.23 ± 18.18
	GBDT	0.29 ± 0.14	50.49 ± 10.45	66.00 ± 15.46
	KNN	0.16 ± 0.21	53.86 ± 6.93	70.33 ± 10.98
	LightGBM	0.31 ± 0.11	49.70 ± 9.38	64.87 ± 13.34
	MLP	0.07 ± 0.01	58.36 ± 14.99	76.36 ± 21.34
	RF	0.34 ± 0.07	49.24 ± 10.61	64.34 ± 15.10
	SVR	0.08 ± 0.05	56.50 ± 6.91	73.76 ± 11.13
	StackEns	0.33 ± 0.08	49.42 ± 9.83	64.53 ± 14.03
	XGBoost	0.30 ± 0.10	50.03 ± 9.24	65.33 ± 13.41
	▶ Average	0.24 ± 0.13	52.02 ± 10.14	67.96 ± 14.74
	▶ Ensemble	0.34 ± 0.08	48.91 ± 10.43	63.90 ± 14.83
Conf2 (Sentinel-2 + Climatic)	AdaBoost	0.30 ± 0.06	50.36 ± 10.53	65.80 ± 15.14
	BaggedDT	0.31 ± 0.09	49.84 ± 8.97	65.07 ± 13.01
	BayesianNN	0.12 ± 0.10	56.91 ± 13.32	74.41 ± 19.04
	CatBoost	0.28 ± 0.14	50.87 ± 9.38	66.44 ± 13.90
	DeepCNN	0.27 ± 0.15	51.96 ± 14.93	68.31 ± 20.91
	GBDT	0.28 ± 0.12	50.91 ± 9.97	66.51 ± 14.56
	KNN	0.17 ± 0.23	53.46 ± 5.99	69.79 ± 9.93
	LightGBM	0.33 ± 0.06	49.37 ± 9.44	64.47 ± 13.61
	MLP	0.06 ± 0.30	58.38 ± 15.62	76.39 ± 21.88
	RF	0.32 ± 0.05	49.86 ± 10.48	65.16 ± 15.10
	SVR	0.13 ± 0.19	55.34 ± 7.81	72.31 ± 12.45
	StackEns	0.30 ± 0.06	50.29 ± 9.73	65.68 ± 13.99
	XGBoost	0.28 ± 0.11	50.88 ± 8.53	66.46 ± 12.78
	▶ Average	0.24 ± 0.13	52.21 ± 9.98	68.21 ± 14.62
	▶ Ensemble	0.33 ± 0.05	49.36 ± 9.86	64.49 ± 14.24
Conf3 (Sentinel-2 + CHM)	AdaBoost	0.71 ± 0.09	32.72 ± 10.67	42.78 ± 14.66
	BaggedDT	0.71 ± 0.08	32.49 ± 9.22	42.40 ± 12.44
	BayesianNN	0.68 ± 0.08	34.34 ± 10.00	44.86 ± 13.71
	CatBoost	0.70 ± 0.07	33.48 ± 9.72	43.79 ± 13.72
	DeepCNN	0.22 ± 0.12	53.59 ± 13.87	70.46 ± 20.03
	GBDT	0.68 ± 0.07	34.01 ± 8.77	44.42 ± 12.21
	KNN	0.18 ± 0.18	53.50 ± 7.21	69.84 ± 11.22
	LightGBM	0.69 ± 0.05	33.77 ± 8.28	44.12 ± 11.63
	MLP	0.63 ± 0.08	36.81 ± 9.95	48.16 ± 14.06
	RF	0.71 ± 0.06	32.44 ± 8.94	42.38 ± 12.29
	SVR	0.34 ± 0.10	49.21 ± 10.67	64.29 ± 15.16
	StackEns	**0.72 ± 0.09**	**31.91 ± 9.21**	**41.61 ± 12.24**
	XGBoost	0.68 ± 0.11	33.98 ± 11.14	44.41 ± 15.16
	▶ Average	0.59 ± 0.09	37.86 ± 9.81	49.50 ± 13.73
	▶ Ensemble	**0.73 ± 0.07**	**31.55 ± 9.35**	**41.22 ± 12.82**
Conf4 (Sentinel-2 + Climatic+ CHM)	AdaBoost	0.70 ± 0.10	33.42 ± 11.35	43.69 ± 15.41
	BaggedDT	0.69 ± 0.08	33.64 ± 9.71	43.93 ± 13.27
	BayesianNN	0.67 ± 0.06	34.98 ± 9.85	45.74 ± 13.79
	CatBoost	0.71 ± 0.07	32.58 ± 9.62	42.58 ± 13.30
	DeepCNN	0.30 ± 0.11	50.87 ± 12.84	66.45 ± 18.49
	GBDT	0.67 ± 0.06	35.06 ± 9.20	45.83 ± 12.99
	KNN	0.14 ± 0.26	54.17 ± 5.09	70.68 ± 8.61
	LightGBM	0.69 ± 0.05	33.88 ± 8.69	44.27 ± 12.15
	MLP	0.53 ± 0.07	41.53 ± 10.01	54.21 ± 13.72
	RF	0.69 ± 0.05	33.70 ± 8.98	44.03 ± 12.43
	SVR	0.17 ± 0.17	53.85 ± 7.88	70.37 ± 12.47
	StackEns	0.70 ± 0.08	32.83 ± 9.25	42.84 ± 12.42
	XGBoost	0.66 ± 0.11	35.21 ± 11.77	46.08 ± 16.35
	▶ Average	0.56 ± 0.10	30.98 ± 9.56	50.82 ± 13.49
	▶ Ensemble	0.72 ± 0.07	31.89 ± 9.81	41.66 ± 13.35

sample size.

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2} \tag{3}$$

$$\%RMSE = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y})^2}}{\bar{y}} \times 100 \tag{4}$$

3. Results

In this section, we present a detailed analysis of the model's performance by comparing four configurations for the target variables, CSE and CS. Table 3 outlines the performance related to the target variable

Table 4
Performance of the considered models for the target variable CSE, over all input configurations. The reported metrics are R^2 (higher is better), RMSE (lower is better), and %RMSE (lower is better). The best result is bolded. Average is the average performance of 13 models for each metric.

Input Features	Model	R^2	RMSE	%RMSE
Conf1 (Sentinel-2)	AdaBoost	0.21 ± 0.13	1.04 ± 0.07	62.06 ± 5.25
	BaggedDT	0.23 ± 0.16	1.03 ± 0.05	61.15 ± 4.61
	BayesianNN	−0.06 ± 0.11	1.21 ± 0.11	72.24 ± 8.11
	CatBoost	0.23 ± 0.16	1.02 ± 0.07	61.04 ± 5.80
	DeepCNN	0.28 ± 0.11	0.99 ± 0.06	59.01 ± 5.14
	GBDT	0.16 ± 0.11	1.08 ± 0.07	64.14 ± 5.35
	KNN	−0.09 ± 0.21	1.22 ± 0.10	72.91 ± 7.85
	LightGBM	0.25 ± 0.19	1.01 ± 0.05	60.12 ± 4.37
	MLP	−0.03 ± 0.03	1.20 ± 0.14	71.67 ± 9.62
	RF	0.25 ± 0.15	1.01 ± 0.05	60.18 ± 4.06
	SVR	−0.02 ± 0.02	1.20 ± 0.15	71.32 ± 9.77
	StackEns	0.20 ± 0.10	1.05 ± 0.07	62.44 ± 5.45
	XGBoost	0.17 ± 0.21	1.06 ± 0.04	63.12 ± 3.39
	▶ Average	0.14 ± 0.13	1.09 ± 0.08	64.72 ± 6.06
	▶ Ensemble	0.25 ± 0.13	1.02 ± 0.06	60.46 ± 5.01
Conf2 (Sentinel-2 + Climatic)	AdaBoost	0.24 ± 0.17	1.02 ± 0.05	60.46 ± 4.42
	BaggedDT	0.22 ± 0.19	1.03 ± 0.06	61.09 ± 5.07
	BayesianNN	−0.03 ± 0.20	1.19 ± 0.08	70.74 ± 5.37
	CatBoost	0.27 ± 0.17	1.00 ± 0.05	59.45 ± 4.01
	DeepCNN	0.26 ± 0.16	1.00 ± 0.07	59.60 ± 4.57
	GBDT	0.24 ± 0.10	1.02 ± 0.08	61.00 ± 5.72
	KNN	−0.05 ± 0.28	1.19 ± 0.07	70.84 ± 5.99
	LightGBM	0.25 ± 0.16	1.01 ± 0.04	60.11 ± 3.73
	MLP	−0.03 ± 0.03	1.20 ± 0.14	71.64 ± 9.71
	RF	0.26 ± 0.16	1.00 ± 0.05	59.67 ± 3.67
	SVR	−0.05 ± 0.20	1.20 ± 0.08	71.53 ± 6.93
	StackEns	0.22 ± 0.08	1.04 ± 0.08	61.80 ± 6.14
	XGBoost	0.18 ± 0.20	1.05 ± 0.05	62.80 ± 4.94
	▶ Average	0.15 ± 0.16	1.07 ± 0.07	63.90 ± 5.41
	▶ Ensemble	0.26 ± 0.14	1.00 ± 0.05	59.79 ± 4.36
Conf3 (Sentinel-2 + CHM)	AdaBoost	0.39 ± 0.10	0.92 ± 0.11	54.89 ± 7.99
	BaggedDT	0.37 ± 0.11	0.93 ± 0.10	55.36 ± 7.22
	BayesianNN	0.32 ± 0.11	0.97 ± 0.13	58.00 ± 9.32
	CatBoost	0.37 ± 0.11	0.94 ± 0.11	55.78 ± 7.89
	DeepCNN	0.30 ± 0.11	0.98 ± 0.09	57.99 ± 5.35
	GBDT	0.39 ± 0.07	0.93 ± 0.12	55.21 ± 8.52
	KNN	−0.09 ± 0.21	1.22 ± 0.10	72.86 ± 7.87
	LightGBM	0.40 ± 0.08	0.91 ± 0.09	54.30 ± 6.83
	MLP	−0.03 ± 0.03	1.20 ± 0.15	71.56 ± 9.84
	RF	0.41 ± 0.08	0.91 ± 0.08	53.94 ± 6.29
	SVR	−0.01 ± 0.02	1.19 ± 0.14	70.99 ± 9.68
	StackEns	0.39 ± 0.08	0.92 ± 0.11	54.97 ± 7.88
	XGBoost	0.39 ± 0.11	0.92 ± 0.12	54.81 ± 8.00
	▶ Average	0.28 ± 0.09	1.00 ± 0.11	59.28 ± 7.90
	▶ Ensemble	**0.41 ± 0.08**	**0.90 ± 0.10**	**53.85 ± 7.59**
Conf4 (Sentinel-2 + Climatic+ CHM)	AdaBoost	0.37 ± 0.11	0.93 ± 0.09	55.44 ± 7.11
	BaggedDT	0.38 ± 0.10	0.93 ± 0.09	55.28 ± 7.00
	BayesianNN	0.25 ± 0.18	1.01 ± 0.13	60.37 ± 8.49
	CatBoost	**0.42 ± 0.08**	**0.90 ± 0.08**	**53.40 ± 6.14**
	DeepCNN	0.29 ± 0.11	0.99 ± 0.09	58.99 ± 5.98
	GBDT	0.40 ± 0.06	0.92 ± 0.09	54.60 ± 6.44
	KNN	−0.08 ± 0.33	1.20 ± 0.07	71.66 ± 6.07
	LightGBM	0.41 ± 0.09	0.90 ± 0.11	53.84 ± 8.05
	MLP	−0.03 ± 0.04	1.20 ± 0.15	71.70 ± 10.35
	RF	0.41 ± 0.08	0.90 ± 0.07	53.89 ± 5.57
	SVR	−0.04 ± 0.20	1.19 ± 0.08	71.04 ± 6.95
	StackEns	0.37 ± 0.08	0.93 ± 0.10	55.74 ± 7.44
	XGBoost	0.39 ± 0.13	0.91 ± 0.11	54.33 ± 7.38
	▶ Average	0.27 ± 0.12	0.99 ± 0.10	59.27 ± 7.25
	▶ Ensemble	**0.41 ± 0.08**	**0.90 ± 0.09**	**53.93 ± 6.54**

CS, while Table 4 focuses on the performance related to the target variable CSE. The results for both variables are reported on the test set in Tables 3 and 4. For simplicity, the training results for CS are included in Appendix B for further reference. Additionally, all results are available via the link mentioned in the Data Availability section of the paper.

In Table 3, our analysis reveals that the models which consistently deliver robust performance across all configurations are: RF, StackEns, LightGBM, and AdaBoost. They outperform other algorithms such as GBDT, KNN, MLP, and SVR. The inclusion of CHM data in Conf3 results in the highest R^2 values observed, with StackEns achieving an R^2 of 0.72, RMSE of 31.91, and %RMSE of 41.61, closely followed by AdaBoost, RF, and BaggedDT. In Conf4, which combines all input features, CatBoost achieves the best R^2 of 0.71.

The Ensemble model consistently outperforms individual models across various configurations, demonstrating superior performance. Specifically, in Conf3, the Ensemble model achieves an R^2 of 0.73, RMSE of 31.55, and %RMSE of 41.22. In Conf4, which includes all available data, the model maintains high performance with an R^2 of 0.72, RMSE of 31.89, and %RMSE of 41.66.

In Table 4, we observe CatBoost, LightGBM and RF consistently

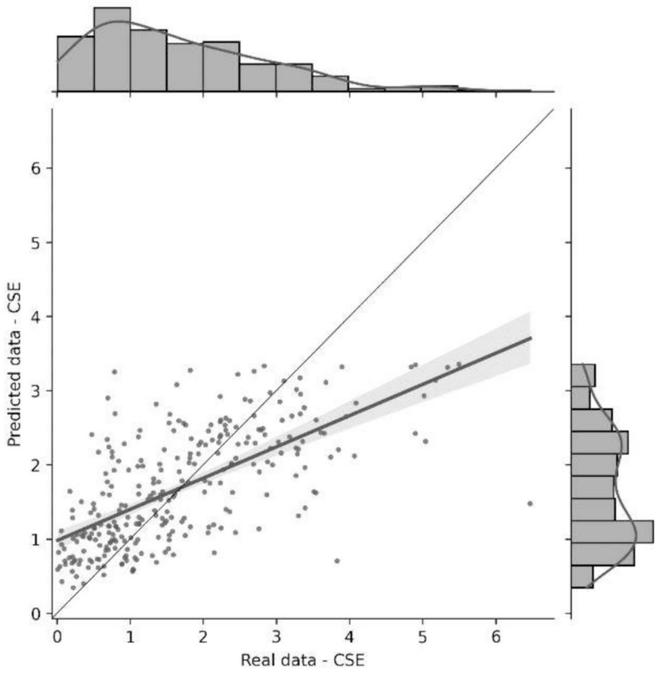

Fig. 4. Real data versus predicted values for the target variable CSE, for the best model: Ensemble with input configuration Conf3.

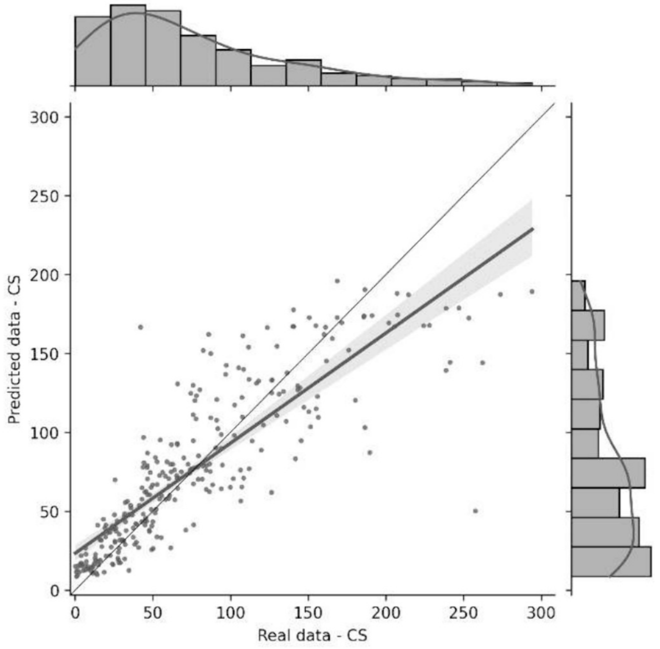

Fig. 5. Real data versus predicted values for the target variable CS, for the best model: Ensemble with input configuration Conf3.

emerge among the top-performing models, demonstrating robust performance across various configurations for CSE estimation. Interestingly, DeepCNN is among the top-performing models for Conf1 and Conf2, while it is surpassed by the other machine learning models for Conf3 and Conf4. In Conf1, RF is the best machine learning model, closely followed by LightGBM and BaggedDT. When Sentinel-2 data is combined with climatic variables (Conf2), both RF and LightGBM continue to show robust results, together with CatBoost and AdaBoost. The same is true for Conf3, where other models also showcase a boost in performance. In Conf4, which combines all input features, CatBoost achieves the highest R^2 value. Notably, the Ensemble model demonstrates again a robust performance across all configurations, matching or surpassing all the single machine learning models.

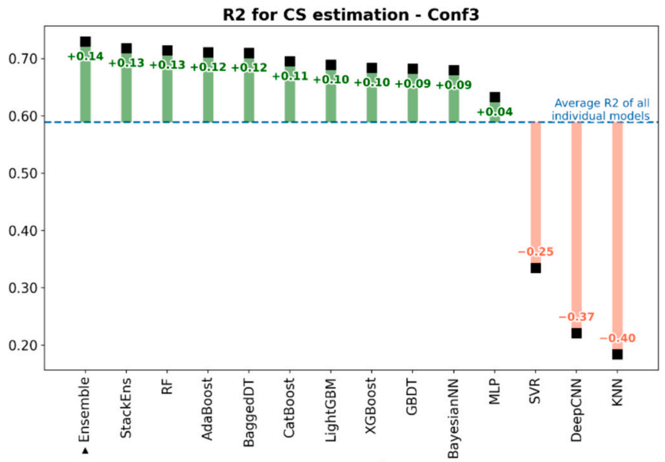

Fig. 6. Difference in performance of all models for R^2 in Conf3 for CS.

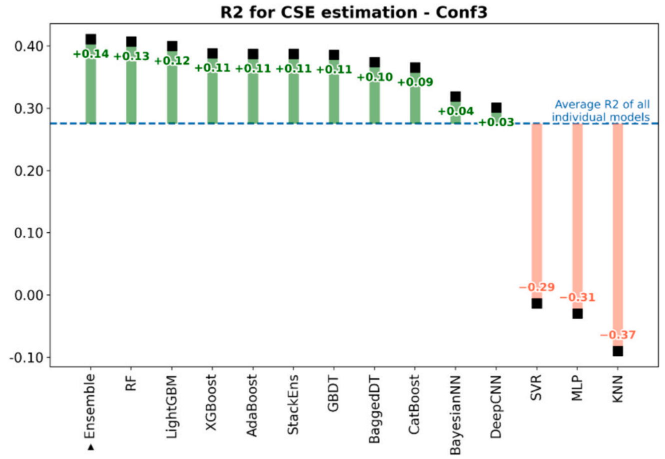

Fig. 7. Difference in performance of all models for R^2 in Conf3 for CSE.

In summary, the Ensemble model achieves the highest predictive accuracy overall, demonstrating the value of integrating multiple models. The addition of climatic data shows minimal impact on estimation accuracy, while the inclusion of CHM data significantly enhances performance, underscoring its importance in achieving accurate estimations.

We evaluate the predictions of the most effective method and input configuration for the two tasks, concentrating on the Ensemble model trained with Conf3. Figs. 4 and 5 illustrate the relationship between actual data and model predictions for the target variables, featuring marginal distribution bar charts. The thin diagonal line indicates the perfect fit, while the thicker line shows the regression line from the model.

3.1. Statistical comparison of model performance

3.1.1. Averaging performance diversity

To assess the performance and diversity of each model, we calculated the average performance of the individual models (reported in Table 3 and Table 4 as "Average") and compared each model's performance to this average. Additionally, we compared the performance of the Ensemble model with the average performance of the individual models. For simplicity, here we only discuss the results for R^2 in Conf3. Further

Table 5
Average ranks of models across all configurations for CS.

Model	Average R^2 Rank	Average RMSE Rank	Average % RMSE Rank	Overall Average Rank
StackEns	3.25	2.5	2.5	2.75
RF	3.5	3.5	3.5	3.5
AdaBoost	5.0	4.25	4.5	4.5
CatBoost	5.25	4.5	4.5	4.7
BaggedDT	5.0	4.75	4.75	4.8
LightGBM	5.5	5.0	5.0	5.1
XGBoost	7.75	6.0	6.0	6.5
GBDT	7.0	7.25	7.25	7.1
SVR	10.5	10.25	10.75	10.5
MLP	10.5	10.75	10.75	10.6
BayesianNN	11.0	11.5	11.5	11.3
DeepCNN	11.0	11.75	11.5	11.4
KNN	11.75	11.75	11.5	11.6

plots can be accessed in the additional materials. As depicted in Figs. 6 and 7, for both targets (CS and CSE), the MLP, SVR, KNN, and DeepCNN models performed close to or below the average, while the Ensemble model consistently ranked among the top two. RF also consistently performed above average. This pattern of performance is consistent across all configurations.

3.1.2. *Performance evaluation across all configuration*

We perform further analyses to identify the most reliable models across all input configurations. First, we calculate the average performance of each model across the four configurations for all metrics (R^2, RMSE, and %RMSE). For simplicity, we focused on the target variable CS. The following analysis combines the performance ranks across all configurations to comprehensively compare the models. Table 5 presents the average ranks of models across all configurations for CS.

The Friedman test was conducted on these average ranks to determine if there were significant differences in the performance of the models. The test yielded a Friedman statistic of 38.60, resulting in a *p*-value of 0.00012. Since the p-value is significantly less than 0.05, we reject the null hypothesis and conclude that there are significant differences in model performance.

To further explore these differences, we look at the average ranks. The StackEns and RF models consistently ranked high across all configurations and metrics. AdaBoost and CatBoost also showed strong performance, closely following the previous two models. BaggedDT exhibited moderate performance, while other models were consistently outperformed such as BayesianNN, DeepCNN, and KNN.

In conclusion, the StackEns and RF models demonstrated the best overall performance, followed closely by AdaBoost and CatBoost. These models are recommended for further applications and analysis due to their superior performance across various configurations and evaluation metrics.

3.2. *Assessing the significance of input features using SHAP values*

We further explored the significance of various input features across different configurations using SHAP values for RF. In the context of CSE estimation, as shown in Fig. 8, NDII and GNDVI emerge as crucial features when using Conf1, consistently occupying top positions in the plots. This is particularly evident in the bar plot (left), which highlights their significant impact on the model's output magnitude. NDII and GNDVI are especially responsive to canopy moisture and vegetation water stress, making them essential for accurately predicting CSE.

Additionally, the inclusion of climatic data in Conf2 maintains the significance of Sentinel-2 features, while climatic factors, such as maximum summer and spring temperatures, emerge among the top 15 contributors. Specifically, lower temperatures (represented by orange and yellow in the central violin plot) are associated with higher CSE values. When Sentinel-2 and CHM features are incorporated in Conf3, CHM-related features become dominant, surpassing the influence of vegetation indices. In Conf4, where all features are utilized, CHM remains the most impactful, followed by GNDVI and NDII, while climatic features, particularly average spring precipitation, and maximum summer temperature, show relatively low significance.

Fig. 9 extends the analysis of influential features impacting the target variable CS across different configurations. In Conf1, the most significant features include spectral indices such as GNDVI and NDII, alongside topographical factors like slope (SLO) and elevation (ELE). Notably, high values of average, maximum, and median elevation strongly correlate with elevated CS values, as illustrated in the violin and summary plots for Conf1. In Conf2, the focus shifts to climatic features, where lower maximum temperatures during Spring and Summer are linked to higher CS values. Conf3 reveals that CHM-related features become the most influential, significantly surpassing the impact of satellite-derived features, with CHM max being three times more

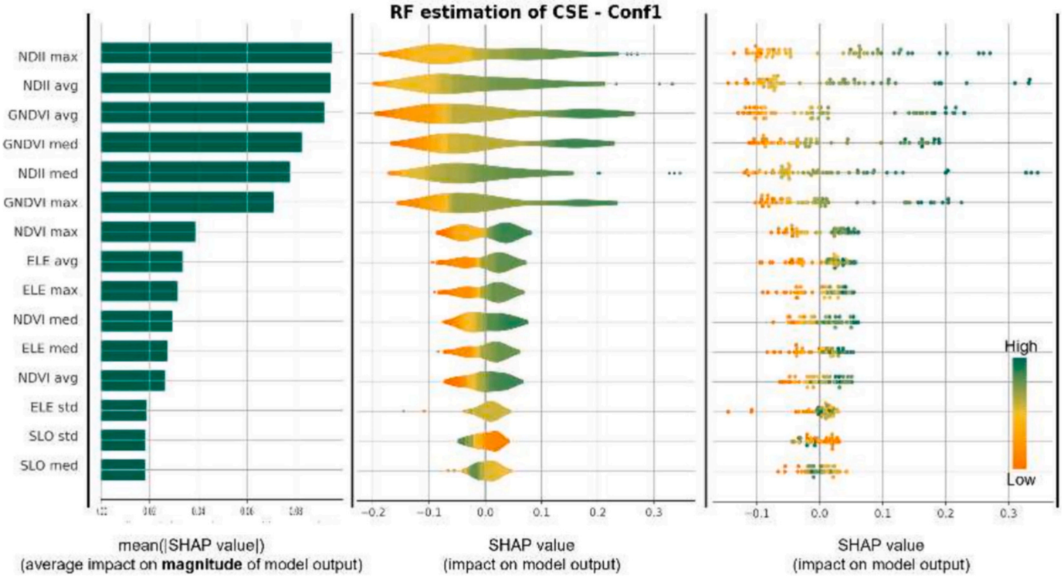

Fig. 8. Plots of the SHAP values for the 15 most influential features of the RF Model on CSE, for all input configurations. From left to right: bar plot showing the average impact of the features on the magnitude of the output, violin plot, and summary plot showing the impact of the features on the output.

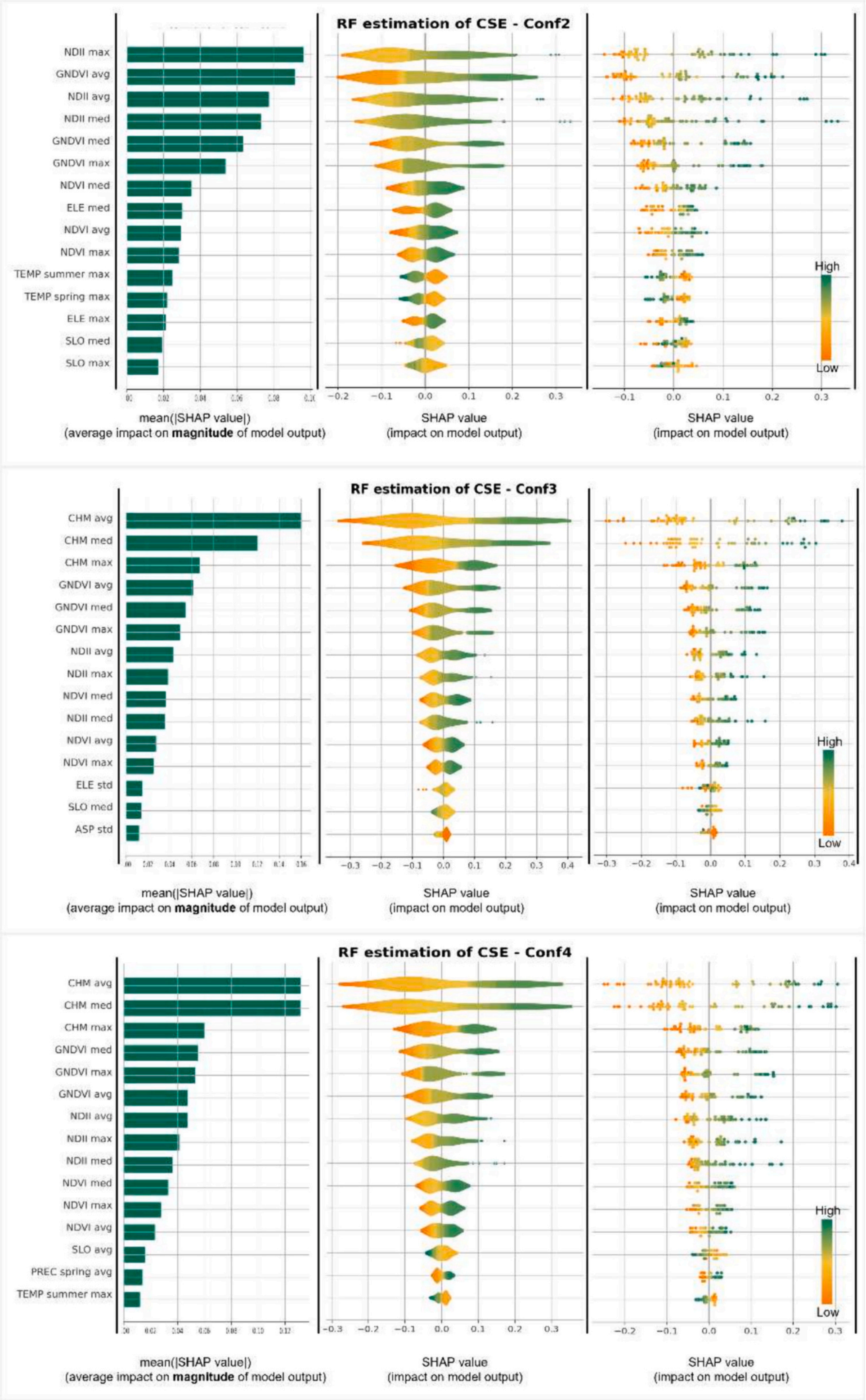

Fig. 8. (*continued*).

significant than elevation max. This trend continues in Conf4, where CHM remains the most critical factor, followed by climatic features and elevation, reflecting a consistent pattern across the different configurations.

Fig. 10 showcases a force plot for RF, Conf3 on CSE (top) and CS (bottom), to further visualize the effect that the input features have on the final prediction. The top half of the figure displays an example of a correctly predicted sample on CSE. The high values of the CHM-related variables (CHM avg. = 23.54, CHM med = 24.92, CHM max = 35.63) push the model to predict a high CSE value, while the low values of the GNDVI-related variables (GNDVI avg. = 0.71, GNDVI max = 0.73, GNDVI med = 0.71) push towards a lower output value. This confirms the trend seen in the violin and summary plots in Fig. 8, where high values of vegetation indices were correlated with high outputs and vice

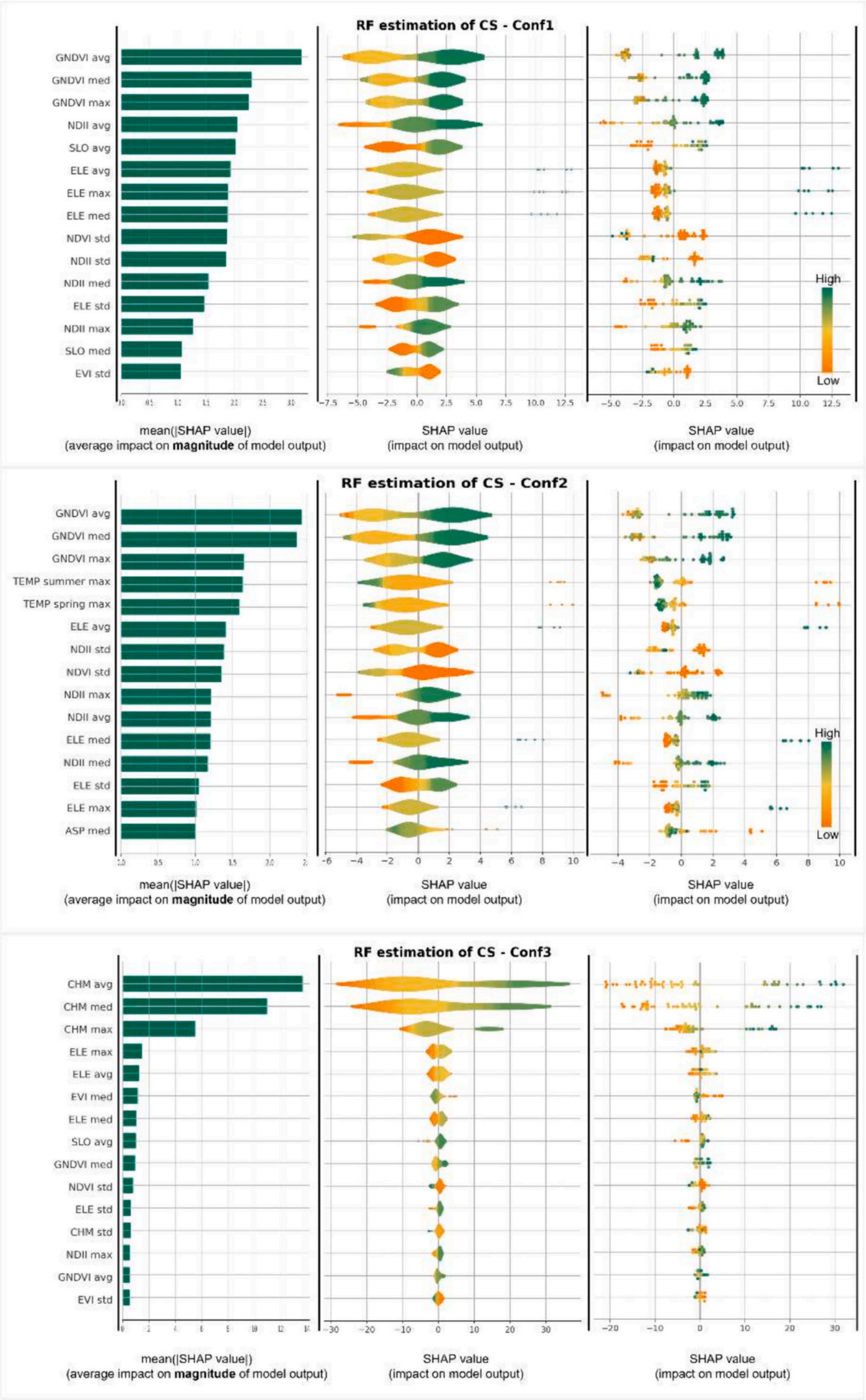

Fig. 9. Plots of the SHAP values for the 15 most influential features of the RF Model on CS, for all input configurations. From left to right: bar plot showing the average impact of the features on the magnitude of the output, violin plot, and summary plot showing the impact of the features on the output.

versa. In the end, the RF predicted a CSE value of 2.10, which is very close to the measured 2.12.

The lower half of Fig. 10 shows a similar plot for CS. The high values of the vegetation indices (e.g. GNDVI med = 0.80) push the predicted CS towards higher values, but extremely low CHM-related values (e.g., CHM avg. = 9.78, CHM max = 18.74) push it down to 60.57, which is close to the real 60.25 value. Since CHM is the most important feature for CS prediction, this is also aligned with the patterns observed in Fig. 9.

4. Discussion

In this study, we focused on estimating C storage and sequestration in Friuli Venezia Giulia region (Italy) using various data combinations, including remote sensing and geomorphologic data. We specifically investigated the contributions of climatic data and the canopy height model (CHM) to these estimations. Our analysis involved implementing and evaluating fourteen different models across four configurations

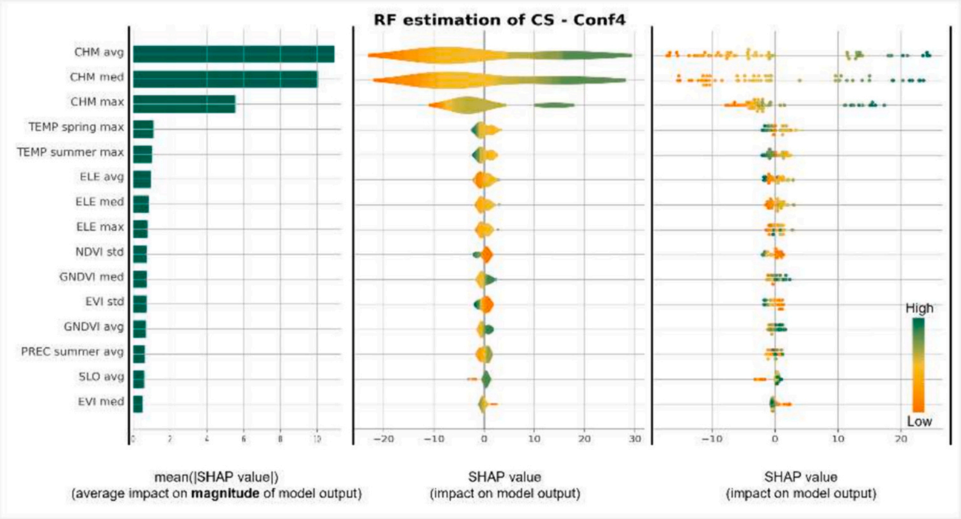

Fig. 9. (*continued*).

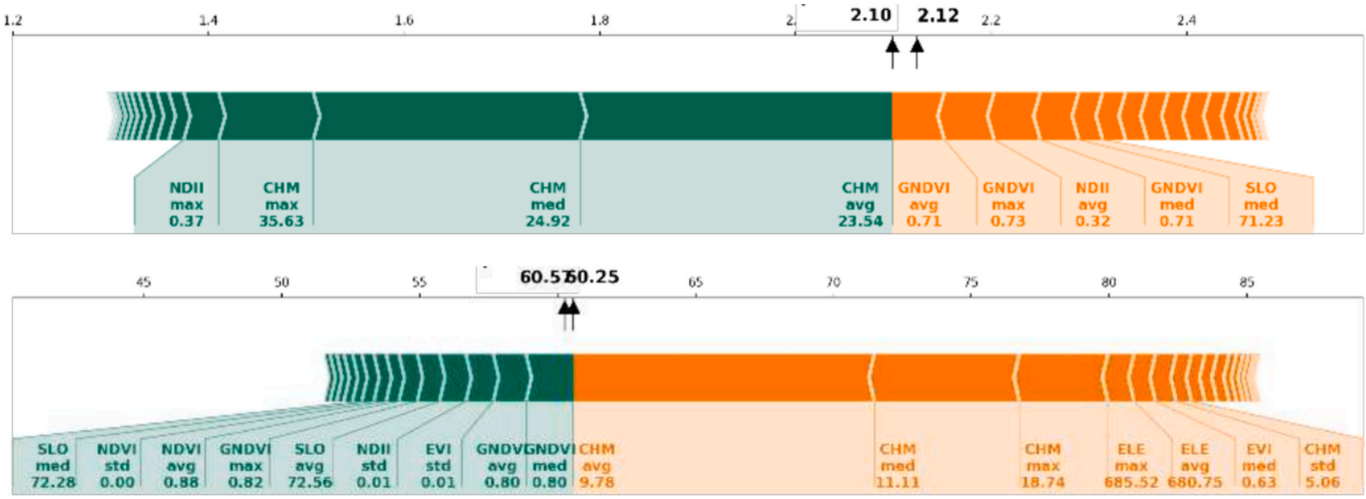

Fig. 10. SHAP force plot for two predictions of the RF model, Conf3, for CSE (top) and CS (bottom).

based on combinations of Sentinel-2 satellite images, climatic data, and CHM.

Among the tested models, the newly implemented Ensemble model demonstrated the best performance regarding R^2, RMSE, and %RMSE. Our findings indicate that including climatic data did not improve estimation accuracy in either regression task. However, combining satellite images with CHM significantly improved both CS and CSE estimations. Interestingly, using all available features did not significantly enhance model performance for either target.

In this study, the Ensemble method demonstrated superior performance in estimating C storage (CS) for Conf3, achieving an R^2 value of 0.73, a RMSE of 31.55, and a %RMSE of 41.20 %. Our findings indicate significant advancements in CS estimation compared to previous research. For instance, Safari et al. (Safari et al., 2017) explored methods to enhance the estimation of aboveground C using Landsat 8 OLI data, employing various machine learning algorithms. Among these, Random Forest (RF) showed superior performance, with an RMSE (%) of approximately 35 % and an R^2 value of 0.65. Furthermore, Uniyal et al. (Uniyal et al., 2022) highlighted the XGBoost algorithm's superior prediction accuracy in quantifying carbon storage in urban forests, emphasizing the context-dependent nature of model performance. When compared to other studies using machine learning approaches for estimating forest parameters, particularly total aboveground biomass (Du et al., 2010; Frazier et al., 2014; Labrecque et al., 2006), our results show better performance, evidenced by a lower RMSE and a higher R^2. However, direct comparisons between studies with different forest conditions, sampling methods, and modeling approaches are challenging (Safari et al., 2017; Zandler et al., 2015). This underscores the importance of selecting models tailored to specific site characteristics and data inputs.

Fig. 4 illustrates a comparable plot for the CS target variable, where the trend line closely aligns with the diagonal, indicating a good model fit. However, examining the marginal distributions reveals challenges in predicting high CS values, potentially attributed to saturation effects in input values, similar to those observed in the analysis of CSE. This issue of underestimation for larger values of forest stand attributes like CSE (and to a lesser extent, CS) is consistent with previous studies. Chirici et al. (Chirici et al., 2020) highlighted this problem when estimating stand volumes using RF models, attributing it to the low sensitivity of spectral reflectance, especially in multi-layer canopy forests or dense forests (Giannetti et al., 2018). Moreover, areas with complex topographic features, such as those in our case study ranging from flat terrain to mountains up to 2000 m above sea level, can affect the spectral signature and the data saturation values of forest aboveground biomass

(Lu et al., 2014). This saturation effect has also been reported in studies using LiDAR data (Giannetti et al., 2018; Nilsson et al., 2017).

In addition, this research offers a notable advantage through its incorporation of diverse input configurations. By integrating satellite features, climatic data, and the Canopy Height Model (CHM), the study employs a meticulous methodology. A key component of this integration is the thorough investigation into how LiDAR data can enhance the precision of CSE and CS estimation, particularly when compared with optical sensors. This exploration is especially critical for operational purposes, where balancing accuracy with practical constraints is essential.

5. Conclusions

This study demonstrates that the newly implemented model called Ensemble is the most effective for estimating C sequestration (CSE) and C storage (CS) in forests, utilizing machine learning and remote sensing data. Integrating canopy height model (CHM) data from LiDAR significantly enhances the accuracy of CS and CSE estimation. The combination of satellite features, climatic data, and CHM data offers valuable insights into forest carbon dynamics. These findings provide practical guidance on selecting the best ensemble models and input configurations for accurate carbon estimation and creating high-detail maps of CS and CSE. These maps are essential tools for identifying C sequestration hotspots and coldspots, thus informing evidence-based forest policies, planning, and management. However, the research is limited by the lack of detailed soil data, which is crucial for a comprehensive understanding of C sequestration dynamics. Incorporating soil data in future research could improve the accuracy and reliability of estimates.

Funding

The project was funded by the European rural development program of the Autonomous Region Friuli Venezia Giulia, Measure 16.1.1, Decree No. 422/AGFOR of 22/01/2020. Additionally, this study was carried out within the Agritech National Research Center and received funding from the European Union's Next-Generation EU (National Recovery and Resilience Plan (PNRR) – Mission 4, Component 2, Investment 1.4 – Decree No. 1032 of 17/06/2022, CN00000022). This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

CRediT authorship contribution statement

Mehdi Fasihi: Conceptualization, Methodology, Software, Validation, Writing – original draft. **Beatrice Portelli:** Methodology, Software, Writing – original draft, Validation, Visualization. **Luca Cadez:** Methodology, Data curation, Writing – original draft, Writing – review & editing. **Antonio Tomao:** Conceptualization, Resources, Data curation, Writing – review & editing. **Alex Falcon:** Writing – review & editing. **Giorgio Alberti:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Giuseppe Serra:** Conceptualization, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code, dataset, and ISO-compliant metadata associated with our work are accessible through the following link [https://zenodo.org/doi/10.5281/zenodo.10932817].

Acknowledgments

Information on CS and CSE from the original plot data of Carabinieri, Council for Agricultural Research and Analysis of the Agricultural Economy - National Inventory of Forests and forest carbon reservoirs – INFC www.inventoryforest.org released under the Creative Commons Attribution 4.0 license. DEM and derivative data from Autonomous Region Friuli Venezia Giulia released under the Italian Open Data License v2.0 (IODL 2.0). Contains modified Copernicus Sentinel data 2019, 2020, 2021. Climate data from the Climate projections platform for the north-east produced by the Regional Agency for Environmental Prevention and Protection of Veneto and by the Regional Agency for Environmental Prevention and Protection of Friuli Venezia Giulia, on data provided by the Royal Netherlands Meteorological Institute, Swedish Meteorological and Hydrological Institute, Institute for Max-Planck Meteorology, CLM-Community, ARPAV, ARPA FVG, ARPAE Emilia-Romagna, Autonomous Province of Trento, Autonomous Province of Bolzano, Italian Air Force; and released under the Creative Commons Attribution 3.0 Italy license (CC BY-SA 3.0 IT) https://clima.arpa.veneto.it.

Appendix A. Grid search method

Table A-1 reports the hyperparameters taken into consideration for each model, their meaning, and the range of search. The DeepCNN model was trained for 50 epochs using early stopping to determine the best epoch at which to halt the training. Tables A-2 and A-3 report all the best hyperparameters selected for all the models and input configurations. Table A-2 refers to the CSE target variable and Table A-3 to the CS target variable.

Table A-1
List of the hyperparameters of the algorithms, their meaning, and the range explored during grid search.

Model	Hyperparameter	Meaning	Range
AdaBoost	n_estimators	The number of estimators in the ensemble	[50, 100, 200]
	learning_rate	Weight applied to each regressor at each boosting iteration	[0.01, 0.1, 1.0]
BaggedDT	n_estimators	The number of estimators in the ensemble	[50, 100, 200]
	base_estimator__max_depth	The maximum depth of the tree	[4, 5, 10]
BayesianNN	n_iter	Number of training iterations	[100,200,300]
	lambda_2	Inverse scale parameter (rate parameter) for the Gamma distribution prior over the lambda parameter	[1e-6, 1e-5, 1e-4]
	lambda_1	Shape parameter for the Gamma distribution prior over the lambda parameter	[1e-6, 1e-5, 1e-4]

(continued on next page)

Table A-1 (*continued*)

Model	Hyperparameter	Meaning	Range
	alpha_2	Inverse scale parameter (rate parameter) for the Gamma distribution prior over the alpha parameter	[1e-6, 1e-5, 1e-4]
	alpha_1	Shape parameter for the Gamma distribution prior over the alpha parameter	[1e-6, 1e-5, 1e-4]
CatBoost	learning_rate	Boosting learning rate	[0.03, 0.1]
	l2_leaf_reg	Coefficient at the L2 regularization term of the cost function	[0.2, 0.5, 1, 3, 4]
	iterations	Max count of trees	[100, 150, 200, 250]
	depth	Depth of a tree	[2, 4, 6, 8, 10]
DeepCNN	pretrained	Whether to load the pretrained weights of the model (ImageNet) or initialize the model with random weights	[False, True]
	freeze	Whether the weights of convolutional layers are frozen or allowed to be further trained	[False, True]
	batch_size	Batch size used for the training procedure	[4, 8, 16]
	learning_rate	Learning rate for the Stochastic Gradient Descent Optimizer	[1e-4, 1e-3, 1e-2]
GBDT	subsample	The fraction of samples to be used for fitting the individual base learners	[0.5, 0.75, 1]
	n_estimators	The number of boosting stages to perform	[1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000]
	max_leaf_nodes	Grow trees with max_leaf_nodes in the best-first fashion.	[2, 5, 10, 20, 50, 100]
	max_depth	Maximum depth of the individual regression estimators	[1, 2, 4]
	learning_rate	Boosting learning rate	[1, 0.1, 0.01,0.001]
KNN	n_neighbors	The number of neighbors	[1, 2, …, 30]
LightGBM	n_estimators	The number of estimators in the ensemble	[50, 100, 200]
	max_depth	The maximum depth of the tree	[4, 5, 10]
	learning_rate	Learning rate	[0.05, 0.1, 0.2]
MLP	max_iter	Number of training epochs	[10, 20, 50]
	learning_rate_init	Initial learning rate	[0.01, 0.10]
	hidden_layer_sizes	Number of neurons in the hidden layer	[2, 4, 8, 64, 128]
RF	n_estimators	The number of trees in the forest	[200, 400, 600, 1000, 1200]
	min_samples_split	The minimum number of samples required to split an internal node	[10, 12, 14, 16]
	min_samples_leaf	The minimum number of samples required to be at a leaf node	[4–6, 8]
	max_features	The number of features to consider when looking for the best split	[4–6]
	max_depth	The maximum depth of the tree	[100, 110, 120, 130, 140, 150]
SVR	gamma	Kernel coefficient	[1e-5, 1e-4, 1e-3, 1e-2]
	C	Regularization parameter	[1000, 100, 10, 1, 0.01]
StackEns	final_estimator__alpha	Regularization strength for Ridge meta-model	[0.1, 1.0, 10.0]
XGBoost	n_estimators	The number of boosting stages to perform	[500, 600, 700, 800]
	max_depth	Maximum tree depth for base learners	[4, 5, 10]
	learning_rate	Boosting learning rate	[0.01, 0.015, 0.02]

Table A-2
Best hyperparameters found for all the models, for all configurations, for the target variable CSE.

Model	Hyperparameter	Conf1	Conf2	Conf3	Conf4
AdaBoost	n_estimators	100	200	200	200
	learning_rate	0.1	1.0	1.0	1.0
BaggedDT	n_estimators	100	100	100	100
	base_estimator__max_depth	5	5	10	5
BayesianNN	n_iter	100	300	100	100
	lambda_2	1e-4	1e-4	1e-4	1e-4
	lambda_1	1e-6	1e-4	1e-6	1e-6
	alpha_2	1e-6	1e-6	1e-6	1e-6
	alpha_1	1e-4	1e-4	1e-4	1e-4
CatBoost	learning_rate	0.1	0.1	0.03	0.1
	l2_leaf_reg	3	3	1	0.2
	iterations	100	100	250	200
	depth	2	2	4	4
DeepCNN	pretrained	True	True	True	True
	freeze	False	False	False	False
	batch_size	8	8	8	8
	learning_rate	1e-3	1e-3	1e-3	1e-3
GBDT	subsample	1	1	0.5	0.5
	n_estimators	50	50	200	200
	max_leaf_nodes	100	100	100	100
	max_depth	2	2	4	4
	learning_rate	0.1	0.1	0.01	0.01
KNN	n_neighbors	8	10	9	10
LightGBM	n_estimators	50	50	50	50
	max_depth	4	10	4	10
	learning_rate	0.05	0.05	0.1	0.1
MLP	max_iter	20	50	20	50
	learning_rate_init	0.01	0.01	0.01	0.01
	hidden_layer_sizes	[64]	[64]	[128]	[4]
RF	n_estimators	600	200	600	600
	min_samples_split	10	10	12	10

(*continued on next page*)

Table A-2 (*continued*)

Model	Hyperparameter	Conf1	Conf2	Conf3	Conf4
	min_samples_leaf	5	5	4	5
	max_features	6	6	6	6
	max_depth	100	150	140	100
SVR	gamma	1e-5	1e-5	1e-5	1e-5
	C	10	10	100	100
StackEns	final_estimator__alpha	10.0	10.0	0.1	0.1
XGBoost	n_estimators	500	500	500	700
	max_depth	4	4	5	4
	learning_rate	0.015	0.02	0.02	0.015

Table A-3
Best hyperparameters found for all the models, for all configurations, for the target variable CS.

Model	Hyperparameter	Conf1	Conf2	Conf3	Conf4
AdaBoost	n_estimators	100	200	200	200
	learning_rate	0.1	1.0	1.0	1.0
BaggedDT	n_estimators	100	100	100	100
	base_estimator__max_depth	5	5	10	5
BayesianNN	n_iter	100	300	100	100
	lambda_2	1e-4	1e-4	1e-4	1e-4
	lambda_1	1e-6	0.0001	1e-6	1e-6
	alpha_2	1e-6	1e-6	1e-6	1e-6
	alpha_1	0.0001	0.0001	0.0001	0.0001
CatBoost	learning_rate	0.1	0.1	0.03	0.1
	l2_leaf_reg	3	3	1	0.2
	iterations	100	100	250	200
	depth	2	2	4	4
DeepCNN	pretrained	True	True	True	True
	freeze	False	False	False	False
	batch_size	8	8	8	8
	learning_rate	1e-3	1e-3	1e-3	1e-3
GBDT	subsample	1	1	0.5	0.5
	n_estimators	50	50	200	200
	max_leaf_nodes	100	100	100	100
	max_depth	2	2	4	4
	learning_rate	0.1	0.1	0.01	0.01
KNN	n_neighbors	8	10	9	10
LightGBM	n_estimators	50	50	50	50
	max_depth	4	10	4	10
	learning_rate	0.05	0.05	0.1	0.1
MLP	max_iter	20	50	20	50
	learning_rate_init	0.01	0.01	0.01	0.01
	hidden_layer_sizes	[64]	[64]	[128]	[4]
RF	n_estimators	600	200	600	600
	min_samples_split	10	10	12	10
	min_samples_leaf	5	5	4	5
	max_features	6	6	6	6
	max_depth	100	150	140	100
SVR	gamma	1e-5	1e-5	1e-5	1e-5
	C	10	10	100	100
StackEns	final_estimator__alpha	10.0	10.0	0.1	0.1
XGBoost	n_estimators	500	500	500	700
	max_depth	4	4	5	4
	learning_rate	0.015	0.02	0.02	0.015

Appendix B. Results of the training set

R^2 (R-squared): Higher is better. R^2 tends to be higher in training data than in test data, indicating that the model fits the training data more closely. A higher R^2 means that a larger proportion of the variance in the dependent variable is predictable from the independent variables.

RMSE (Root Mean Square Error): Lower is better. RMSE might be lower in the training data and higher in the test data, reflecting a tighter fit to the training data and potentially larger errors on unseen data.

%RMSE (Percentage Root Mean Square Error): Lower is better. %RMSE might be lower in the training data and higher in the test data, indicating that the relative prediction error is smaller in the training data compared to unseen data.

Table B-1
Training results for CS estimation.

Input Features	Model	R^2	RMSE	%RMSE
Conf1 (Sentinel-2)	AdaBoost	0.91 ± 0.02	18.52 ± 1.32	24.10 ± 1.76
	BaggedDT	0.90 ± 0.01	19.73 ± 0.76	25.69 ± 1.29
	BayesianNN	0.19 ± 0.04	55.36 ± 3.64	72.01 ± 3.75
	CatBoost	0.82 ± 0.01	26.26 ± 1.27	34.18 ± 1.66
	DeepCNN	0.38 ± 0.30	47.72 ± 14.51	61.59 ± 17.37
	GBDT	0.93 ± 0.00	16.07 ± 0.50	20.92 ± 0.64
	KNN	0.41 ± 0.05	47.01 ± 1.27	61.17 ± 0.89
	LightGBM	0.83 ± 0.01	25.18 ± 1.33	32.78 ± 1.56
	MLP	0.18 ± 0.03	55.66 ± 3.31	72.42 ± 3.71
	RF	0.69 ± 0.01	33.95 ± 1.29	44.19 ± 1.59
	SVR	0.48 ± 0.04	44.19 ± 1.39	57.51 ± 1.30
	StackEns	0.76 ± 0.03	30.29 ± 2.63	39.38 ± 2.56
	XGBoost	1.00 ± 0.00	3.15 ± 0.56	4.11 ± 0.86
	▸ Average	0.65 ± 0.04	32.54 ± 2.60	42.31 ± 2.99
	▸ Ensemble	0.82 ± 0.01	25.94 ± 1.39	33.75 ± 1.46
Conf2 (Sentinel-2 + Climatic)	AdaBoost	0.93 ± 0.01	16.24 ± 0.82	21.15 ± 1.13
	BaggedDT	0.90 ± 0.01	19.55 ± 0.62	25.45 ± 1.01
	BayesianNN	0.20 ± 0.05	54.99 ± 3.85	71.52 ± 3.89
	CatBoost	0.90 ± 0.00	19.03 ± 0.57	24.77 ± 0.85
	DeepCNN	0.42 ± 0.18	46.36 ± 8.78	60.53 ± 10.33
	GBDT	0.93 ± 0.00	15.76 ± 0.83	20.50 ± 0.90
	KNN	0.44 ± 0.04	45.99 ± 1.17	59.85 ± 0.56
	LightGBM	0.85 ± 0.00	24.13 ± 1.05	31.40 ± 0.90
	MLP	0.19 ± 0.08	55.38 ± 5.39	71.99 ± 5.78
	RF	0.68 ± 0.01	34.65 ± 1.31	45.10 ± 1.42
	SVR	0.63 ± 0.02	37.55 ± 1.44	48.86 ± 1.00
	StackEns	0.76 ± 0.03	29.81 ± 3.29	38.73 ± 3.33
	XGBoost	0.99 ± 0.00	4.42 ± 0.70	5.77 ± 1.06
	▸ Average	0.68 ± 0.03	31.07 ± 2.29	40.43 ± 2.47
	▸ Ensemble	0.85 ± 0.01	23.67 ± 1.34	30.80 ± 1.24
Conf3 (Sentinel-2 + CHM)	AdaBoost	0.98 ± 0.00	8.95 ± 0.81	11.65 ± 1.14
	BaggedDT	0.96 ± 0.01	12.33 ± 0.98	16.06 ± 1.51
	BayesianNN	0.74 ± 0.02	31.57 ± 2.68	41.10 ± 3.49
	CatBoost	0.97 ± 0.00	11.23 ± 1.02	14.61 ± 1.18
	DeepCNN	0.55 ± 0.36	38.34 ± 19.05	50.16 ± 24.94
	GBDT	1.00 ± 0.00	0.01 ± 0.00	0.02 ± 0.00
	KNN	0.39 ± 0.05	47.94 ± 1.44	62.39 ± 1.49
	LightGBM	0.91 ± 0.01	18.22 ± 1.96	23.72 ± 2.50
	MLP	0.69 ± 0.06	33.93 ± 4.14	44.17 ± 5.43
	RF	0.86 ± 0.01	22.72 ± 1.67	29.59 ± 2.29
	SVR	0.80 ± 0.02	27.31 ± 2.54	35.49 ± 2.58
	StackEns	0.92 ± 0.02	17.22 ± 1.87	22.45 ± 2.89
	XGBoost	0.99 ± 0.00	5.85 ± 0.50	7.63 ± 0.83
	▸ Average	0.83 ± 0.04	21.20 ± 2.97	27.62 ± 3.87
	▸ Ensemble	0.95 ± 0.01	13.58 ± 1.11	17.69 ± 1.59
Conf4 (Sentinel-2 + Climatic+ CHM)	AdaBoost	0.98 ± 0.00	9.09 ± 0.80	11.84 ± 1.09
	BaggedDT	0.96 ± 0.01	12.83 ± 1.16	16.71 ± 1.63
	BayesianNN	0.74 ± 0.02	31.10 ± 2.63	40.49 ± 3.43
	CatBoost	1.00 ± 0.00	3.60 ± 0.23	4.69 ± 0.35
	DeepCNN	0.32 ± 0.11	50.48 ± 4.22	65.63 ± 6.24
	GBDT	1.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00
	KNN	0.55 ± 0.01	41.05 ± 2.04	53.41 ± 1.93
	LightGBM	0.92 ± 0.01	17.51 ± 1.93	22.79 ± 2.43
	MLP	0.62 ± 0.09	37.61 ± 4.24	48.89 ± 4.77
	RF	0.87 ± 0.01	22.40 ± 1.68	29.16 ± 2.17
	SVR	0.66 ± 0.01	36.03 ± 1.51	46.88 ± 1.16
	StackEns	0.92 ± 0.02	17.43 ± 2.12	22.68 ± 2.63
	XGBoost	1.00 ± 0.00	2.21 ± 0.36	2.88 ± 0.48
	▸ Average	0.81 ± 0.02	21.64 ± 1.76	28.16 ± 2.18
	▸ Ensemble	0.96 ± 0.00	11.82 ± 1.00	15.38 ± 1.28

References

Bbeiman, L., 1996. Bagging Predictors.

Breiman, L., 2001. Random Forests.

Carbon Storage by Urban Forests (U.S. National Park Service), 2024. Accessed: Apr. 20, 2023. [Online]. Available. https://www.nps.gov/articles/000/uerla-trees-carbon-storage.htm.

Chen, T., Guestrin, C., Aug. 2016. XGBoost: A scalable tree boosting system. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, pp. 785–794. https://doi.org/10.1145/2939672.2939785.

Chirici, G., et al., Feb. 2019. Forest damage inventory after the 'Vaia' storm in Italy. Forest@ - Rivista di Selvicoltura ed Ecologia Forestale 16 (1), 3–9. https://doi.org/10.3832/efor3070-016.

Chirici, G., et al., Feb. 2020. Wall-to-wall spatial prediction of growing stock volume based on Italian National Forest Inventory plots and remotely sensed data. Int. J. Appl. Earth Obs. Geoinf. 84, 101959. https://doi.org/10.1016/J.JAG.2019.101959.

Dai, L., Zhang, Y., Wang, L., Zheng, S., Xu, W., Feb. 2021. Assessment of carbon density in natural mountain forest ecosystems at Northwest China. Int. J. Environ. Res. Public Health 18 (4), 1–12. https://doi.org/10.3390/ijerph18042098.

Du, H., et al., Oct. 2010. The responses of Moso bamboo (*Phyllostachys heterocycla* var. pubescens) forest aboveground biomass to Landsat TM spectral reflectance and

NDVI. Acta Ecol. Sin. 30 (5), 257–263. https://doi.org/10.1016/J.CHNAES.2010.08.005.

Estoque, R.C., Jun 01, 2020. A review of the sustainability concept and the state of SDG monitoring using remote sensing. MDPI AG. https://doi.org/10.3390/rs12111770.

EVI (Enhanced Vegetation Index) | Sentinel Hub Custom Scripts. Accessed: Jan. 30, 2024. [Online]. Available. https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/evi/.

Fardusi, M.J., Chianucci, F., Barbati, A., May 2017. Concept to practice of geospatial-information tools to assist forest management and planning under precision forestry framework: a review. Ann. Silvicult. Res. 41 (1), 3–14. https://doi.org/10.12899/ASR-1354.

Faska, Z., Khrissi, L., Haddouch, K., El Akkad, N., Dec. 2023. A robust and consistent stack generalized ensemble-learning framework for image segmentation. J. Eng. Appl. Sci. 70 (1), 1–20. https://doi.org/10.1186/S44147-023-00226-4/FIGURES/10.

Feurer, M., Hutter, F., 2019. Hyperparameter Optimization, pp. 3–33. https://doi.org/10.1007/978-3-030-05318-5_1.

Frazier, R.J., Coops, N.C., Wulder, M.A., Kennedy, R., Jun. 2014. Characterization of aboveground biomass in an unmanaged boreal forest using Landsat temporal segmentation metrics. ISPRS J. Photogramm. Remote Sens. 92, 137–146. https://doi.org/10.1016/J.ISPRSJPRS.2014.03.003.

Friedman, J.H., 2001. Greedy Function Approximation: A Gradient Boosting Machine.

Friedman, J.H., 2002. Stochastic Gradient Boosting [Online]. Available: www.elsevier.com/locate/csda.

Friuli Venezia Giulia Autonomous Region, 2024. Enviromental and territorial data catalogue - IRDATfvg - dettaglio-diretto. Accessed: Jan. 31, 2024. [Online]. Available. http://irdat.regione.fvg.it/consultatore-dati-ambientali-territoriali/detail/irdat/dataset/10191.

Gao, L., Hailu, A., May 2012. Ranking management strategies with complex outcomes: an AHP-fuzzy evaluation of recreational fishing using an integrated agent-based model of a coral reef ecosystem. Environ. Model Softw. 31, 3–18. https://doi.org/10.1016/J.ENVSOFT.2011.12.002.

Gao, Y., et al., 2018. Comparative analysis of modeling algorithms for forest aboveground biomass estimation in a subtropical region. Remote Sens (Basel) 10 (4). https://doi.org/10.3390/rs10040627.

Gasparini, P., Papitto, G., 2022. The Italian Forest Inventory in Brief: L'inventario Forestale Nazionale Italiano in Breve. Springer Tracts in Civil Engineering, pp. 1–15. https://doi.org/10.1007/978-3-030-98678-0_1/TABLES/1.

Giannetti, F., Chirici, G., Gobakken, T., Næsset, E., Travaglini, D., Puliti, S., Aug. 2018. A new approach with DTM-independent metrics for forest growing stock prediction using UAV photogrammetric data. Remote Sens. Environ. 213, 195–205. https://doi.org/10.1016/J.RSE.2018.05.016.

Goetz, S.J., et al., Dec. 2007. Monitoring and estimating tropical forest carbon stocks: making REDD a reality. Environ. Res. Lett. 2 (4), 045023. https://doi.org/10.1088/1748-9326/2/4/045023.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., Moore, R., Dec. 2017. Google earth engine: planetary-scale geospatial analysis for everyone. Remote Sens. Environ. 202, 18–27. https://doi.org/10.1016/J.RSE.2017.06.031.

Guth, P.L., et al., 2021. Digital elevation models: terminology and definitions. Remote Sensing 13, 3581. https://doi.org/10.3390/RS13183581.

He, K., Zhang, X., Ren, S., Sun, J., Dec. 2015. Deep Residual Learning for Image Recognition [Online]. Available: http://arxiv.org/abs/1512.03385.

Howard, A., et al., May 2019. Searching for MobileNetV3 [Online]. Available: http://arxiv.org/abs/1905.02244.

Huang, S., Tang, L., Hupy, J.P., Wang, Y., Shao, G., Feb. 2021. A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing. J. For. Res. (Harbin.) 32 (1), 1–6. https://doi.org/10.1007/S11676-020-01155-1/FIGURES/2.

Huang, H., Wu, D., Fang, L., Zheng, X., 2022. Comparison of multiple machine learning models for estimating the forest growing stock in large-scale forests using multi-source data. Forests 13 (9). https://doi.org/10.3390/f13091471.

Izmailov, P., Vikram, S., Hoffman, M.D., Wilson, A.G., 2021. What Are Bayesian Neural Network Posteriors Really like?.

Ke, G., et al.. LightGBM: A Highly Efficient Gradient Boosting Decision Tree [Online]. Available: https://github.com/Microsoft/LightGBM.

k-nearest neighbors algorithm - Wikipedia. Accessed: Jul. 24, 2023. [Online]. Available. https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm.

Konda, R.K., Giri, V., Mandla, V.R., 2017. Study and evaluation of carbon sequestration using remote sensing and gis: a review on various techniques. Int. J. Civ. Eng. Technol. (IJCIET) 8 (4), 8–12 [Online]. Available: http://www.iaeme.com/IJCIET/index.asp287http://www.iaeme.com/IJCIET/issues.asp?JType=IJCIET&VType=8&IType=4.

Krug, T., Kurz, W.A., Lasco, R.D., Martino, D.L., McConkey, B.G., 2024. Volume 4: Agriculture, Forestry and Other Land Use 2.2 2006 IPCC Guidelines for National Greenhouse Gas Inventories.

Labrecque, S., Fournier, R.A., Luther, J.E., Piercey, D., May 2006. A comparison of four methods to map biomass from Landsat-TM and inventory data in western Newfoundland. For. Ecol. Manag. 226 (1–3), 129–144. https://doi.org/10.1016/J.FORECO.2006.01.030.

Landsat Enhanced Vegetation Index | U.S. Geological Survey. Accessed: Jul. 10, 2023. [Online]. Available. https://www.usgs.gov/landsat-missions/landsat-enhanced-vegetation-index.

Lu, D., Chen, Q., Wang, G., Liu, L., Li, G., Moran, E., Jan. 2014. A survey of remote sensing-based aboveground biomass estimation methods in forest ecosystems, 9 (1), 63–105. https://doi.org/10.1080/17538947.2014.990526.

Mo, L., et al., Nov. 2023. Integrated global assessment of the natural forest carbon potential. Nature 624 (7990), 92–101. https://doi.org/10.1038/s41586-023-06723-z.

NDII (Normalized Difference 819/1600) | Sentinel Hub Custom Scripts. Accessed: Jul. 10, 2023. [Online]. Available. https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/ndii/.

Nilsson, M., et al., Jun. 2017. A nationwide forest attribute map of Sweden predicted using airborne laser scanning data and field data from the national forest inventory. Remote Sens. Environ. 194, 447–454. https://doi.org/10.1016/J.RSE.2016.10.022.

Obata, S., Cieszewski, C.J., Lowe, R.C., Bettinger, P., 2021. Random forest regression model for estimation of the growing stock volumes in georgia, USA, using dense landsat time series and fia dataset. Remote Sens. 13 (2), 1–18. https://doi.org/10.3390/rs13020218.

Pan, Y., et al., Aug. 2011. A large and persistent carbon sink in the world's forests. Science (1979) 333 (6045), 988–993. https://doi.org/10.1126/SCIENCE.1201609/SUPPL_FILE/PAPV2.PDF.

Post, W.M., Emanuel, W.R., Zinke, P.J., Stangenberger, A.G., 1982. Soil carbon pools and world life zones. Nature 298 (5870), 156–159. https://doi.org/10.1038/298156a0.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., Jun. 2017. CatBoost: unbiased boosting with categorical features [Online]. Available: http://arxiv.org/abs/1706.09516.

Rehman, A.N., Lal, B., 2023. Machine learning in CO2 sequestration. In: Machine Learning and Flow Assurance in oil and gas Production. Springer Nature Switzerland, Cham, pp. 119–140. https://doi.org/10.1007/978-3-031-24231-1_7.

Roberts, D.R., et al., Aug. 01, 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Blackwell Publishing Ltd. https://doi.org/10.1111/ecog.02881.

Safari, A., Sohrabi, H., Powell, S., Shataee, S., 2017a. A comparative assessment of multi-temporal Landsat 8 and machine learning algorithms for estimating aboveground carbon stock in coppice oak forests. Int. J. Remote Sens. 38 (22), 6407–6432. https://doi.org/10.1080/01431161.2017.1356488.

Safari, A., Sohrabi, H., Powell, S., Shataee, S., Nov. 2017. A comparative assessment of multi-temporal Landsat 8 and machine learning algorithms for estimating aboveground carbon stock in coppice oak forests. Int. J. Remote Sens. 38 (22), 6407–6432. https://doi.org/10.1080/01431161.2017.1356488.

Sagi, O., Rokach, L., Jul. 2018. Ensemble learning: A survey. In: Wiley Interdiscip Rev Data Min Knowl Discov, 8, p. e1249. https://doi.org/10.1002/WIDM.1249 no. 4.

Schapire, R.E., 2003. The Boosting Approach to Machine Learning An Overview. Springer [Online]. Available: www.research.att.com/.

Simonyan, K., Zisserman, A., Sep. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition [Online]. Available: http://arxiv.org/abs/1409.1556.

Stelmaszczuk-Górska, M.A., Rodriguez-Veiga, P., Ackermann, N., Thiel, C., Balzter, H., Schmullius, C., Dec. 2015. Non-parametric retrieval of aboveground biomass in siberian boreal forests with ALOS PALSAR interferometric coherence and backscatter intensity. J. Imag. 2 (1), 1. https://doi.org/10.3390/JIMAGING2010001.

Tan, M., Le, Q.V., Apr. 2021. EfficientNetV2: Smaller Models and Faster Training [Online]. Available: http://arxiv.org/abs/2104.00298.

Uniyal, S., Purohit, S., Chaurasia, K., Rao, S.S., Amminedu, E., 2022. Quantification of carbon sequestration by urban forest using Landsat 8 OLI and machine learning algorithms in Jodhpur, India. Urban For. Urban Green. 67 (December 2021), 127445. https://doi.org/10.1016/j.ufug.2021.127445.

Wang, L., et al., 2022. A transferable learning classification model and carbon sequestration estimation of crops in farmland ecosystem. Remote Sens. 14 (20), 1–19. https://doi.org/10.3390/rs14205216.

What is Remote Sensing and What is it Used for? | U.S. Geological Survey. Accessed: Apr. 21, 2023. [Online]. Available. https://www.usgs.gov/faqs/what-remote-sensing-and-what-it-used.

Xue, J., Su, B., 2017. Significant remote sensing vegetation indices: a review of developments and applications. J. Sens. 2017. https://doi.org/10.1155/2017/1353691.

Zandler, H., Brenning, A., Samimi, C., Mar. 2015. Quantifying dwarf shrub biomass in an arid environment: comparing empirical methods in a high dimensional setting. Remote Sens. Environ. 158, 140–155. https://doi.org/10.1016/J.RSE.2014.11.007.

Zhang, F., Tian, X., Zhang, H., Jiang, M., Jul. 2022. Estimation of aboveground carbon density of forests using deep learning and multisource remote sensing. Remote Sens. 14 (13). https://doi.org/10.3390/rs14133022.

Zhao, P., Lu, D., Wang, G., Wu, C., Huang, Y., Yu, S., 2016. Examining spectral reflectance saturation in landsat imagery and corresponding solutions to improve forest aboveground biomass estimation. Remote Sens. 8 (6). https://doi.org/10.3390/rs8060469.

Zhu, X., Liu, D., Apr. 2015. Improving forest aboveground biomass estimation using seasonal Landsat NDVI time-series. ISPRS J. Photogramm. Remote Sens. 102, 222–231. https://doi.org/10.1016/J.ISPRSJPRS.2014.08.014.