Automated Machine Learning Driven Stacked Ensemble Modeling for Forest Aboveground Biomass Prediction Using Multitemporal Sentinel-2 Data

Parth Naik [ID], Michele Dalponte [ID], *Senior Member, IEEE*, and Lorenzo Bruzzone [ID], *Fellow, IEEE*

Abstract—**Modeling and large-scale mapping of forest above-ground biomass (AGB) is a complicated, challenging, and expensive task. There are considerable variations in forest characteristics that create functional disparity for different models and needs comprehensive evaluation. Moreover, the human-bias involved in the process of modeling and evaluation affects the generalization of models at larger scales. In this article, we present an automated machine learning framework for modeling, evaluation, and stacking of multiple base models for AGB prediction. We incorporate a hyperparameter optimization procedure for automatic extraction of targeted features from multitemporal Sentinel-2 data that minimizes human-bias in the proposed modeling pipeline. We integrate the two independent frameworks for automatic feature extraction and automatic model ensembling and evaluation. The results suggest that the extracted target-oriented features have an excessive contribution of red-edge and short-wave infrared spectrum. The feature importance scale indicates a dominant role of summer-based features as compared to other seasons. The automated ensembling and evaluation framework produced a stacked ensemble of base models that outperformed individual base models in accurately predicting forest AGB. The stacked ensemble model delivered the best scores of $R^2_{cv} = 0.71$ and RMSE = 74.44 Mgha^{-1}. The other base models delivered R^2_{cv} and RMSE ranging between 0.38–0.66 and 81.27–109.44 Mg ha^{-1}, respectively. The model evaluation metrics indicated that the stacked ensemble model was more resistant to outliers and achieved a better generalization. Thus, the proposed study demonstrated an effective automated modeling pipeline for predicting AGB by minimizing human-bias and deployable over large and diverse forest areas.**

Index Terms—**Aboveground biomass (AGB), automated features, automated machine learning (AutoML), ensemble modeling, hyperparameter optimization.**

I. INTRODUCTION

THE selection of suitable machine learning (ML) algorithms to solve the forest aboveground biomass (AGB) problem

requires domain expertise for improving and regulating model performances [1], [2], [3], [4], [5]. There are different ML-based models developed in the literature for the prediction of forest AGB. The most fundamental modeling approach to AGB prediction is generalized linear regression that allows to relate model parameters to the response variable via a link function [73], [77], [78]. Furthermore, kernel-based learners that include methods such as support vector machine [79], [80], [81] and Gaussian process regression [82], [83] have proven to be effective for the development of AGB estimation models. Additionally, the tree-based models such as Random forest (bagging), XGBoost, and CatBoost (boosting) were also found to be very efficient for the prediction of forest AGB [58], [84]. The latest developments in AGB estimation include artificial neural network based learners that use backpropagation algorithm for training the network to reliably predict AGB. These approaches include the use of some basic multilayer perceptron models [85], sparse autoencoders [86], and deep neural networks with generative learning strategy [87], [88]. However, these various ML models are trained and optimized with different strategies that make it difficult to determine whether a given technique is genuinely better or simply better tuned. Automated machine learning (AutoML) approaches can support a wide range of ML algorithms and automate algorithm selection, feature generation, hyperparameter tuning, iterative modeling, and model assessment to develop an optimized ML pipeline.

The task of creating an optimized ML pipeline for AGB prediction is complex and time-consuming due to the diversity in forest type and species. The traditional approaches proposed for the development of ML pipelines use a trial and error mechanism for stacking and selection of models for AGB prediction [6], [7], [8]. In practice, the human element cannot be completely eliminated from the process due to the need of performing model checks and model explanation w.r.t target parameter, i.e., AGB. However, the human intervention can be reduced and replaced with efficient search and hyperparameter importance learning algorithms that minimize the number of user-regulated parameters and achieve faster and efficient modeling with reduced human-bias [9], [10], [11], [12]. In this regard, the concept of AutoML can be very instrumental in gaining better machine learning (ML) performance with the available computational budget and reduced human assistance to model forest AGB. AutoML automates knowledge intensive tasks such as configuring

learning tools for feature engineering, neural architecture search (NAS) and algorithm selection using an optimization-evaluation mechanism [13]. In a general architecture, the AutoML controller consists of an evaluator and an optimizer. The evaluator measures the performance of learning tools and provides feedback to the optimizer in order to update configurations for better performance. The optimizer generates configurations based on a search space that is determined by a process of targeted learning [14]. For example, if the learning process is "feature engineering," the learning tools to be configured are the "classifiers" and the search space would consist of "feature sets, feature enhancing methods (dimension reduction, feature generation, feature encoding) and related hyperparameters."

There are a wide-ranging applications of AutoML such as medical image recognition [15], [16], object detection [17], [18], [19], super resolution [20], [21], [22], language modeling [23], text classification [24], semantic segmentation [25], etc. AutoML has delivered a quality performance for various tasks but scarcely experimented with developing an ML pipeline for AGB prediction. An ML pipeline is a directed graph of learning elements that can be automated by using a search of estimators/predictors, search of learning algorithms, and search of ensemble models [26], [27], [28]. In literature, there are various algorithms developed for automation of these learning elements of an ML pipeline [29]. The evolutionary algorithms such as particle swarm model selection (PSMS) use particle swarm optimization to automate the full model selection problem [30]. Such evolutionary algorithms inspired ensemble PSMS [31] which is a precursor to the development of the latest AutoML systems [28]. Recent AutoML systems are capable of building regression/classification pipelines, full model selection, multiobjective optimization and best architecture/hyperparameter search for deep learning models [32]. Auto-WEKA [33] system uses the sequential model-based algorithm configuration method [34] which is a robust stochastic optimization framework under noisy function evaluations for the lowest cross-validation misclassification. AutoSklearn [35], which is a system similar to sequential model-based optimization (SMBO), has a distinctive search process initialization that exploits a meta-learner delivered the best performance in many AutoML challenges. The tree-based pipeline optimization tool [37] is an open-source genetic AutoML system that optimizes a series of ML models for high-accuracy and compact pipelines for supervised classification. The recent surge in the application of deep neural networks led to the development of AutoML systems like Auto-Net [38] and NAS [39], [40] which are built on combination of Bayesian optimization and Hyperband to automatically tune deep neural networks without human intervention.

The concept of meta-learning is integral to AutoML systems and enables them to learn from the meta-data of the learning elements [41]. AutoML systems perform faster and more efficiently on a new task with meta-learning techniques that replace human-engineered ML pipelines with data-driven pipelines [42]. Meta-learning techniques have been successfully implemented in literature to automate various learning elements

(feature engineering, architecture search, hyperparameter optimization) of the AutoML systems [26], [43], [44], [45], [46], [47]. In this context, there are many studies conducted using meta-learning techniques to execute various ML tasks in remote sensing applications. A trivial problem in deploying advanced ML algorithms in the RS domain is few-shot (low training samples) learning for which meta-learning techniques have provided some significant solutions [48], [49], [50], [51]. Meta-learning techniques can efficiently handle multiscale RS data and provide efficient solutions to inversion modeling problems [52], [53]. These techniques have been used to solve more specific RS problems such as the study in [54] specifically focused on deep neural network model for semantic segmentation of circular objects in satellite images. Apart from such specific problems, meta-learning techniques are capable of dealing with combination of such problems to solve larger problems in RS. For example, a recent study [55] aimed at generating captions for RS images using meta-learning—a task that requires dealing with a combination of problems based on visual and textual features to generate RS image captions. Thus, meta-learning approaches have a significant contribution in solving complex RS problems and hold a great potential for improving AGB modeling in diverse scenarios such as mixed tree species, forest types, and varying density of forests.

The possibility is unlikely for a single modeling algorithm to perform effectively on most types of AGB modeling scenarios. Ensemble models are a combination of base models built on the hypothesis that the combination of multiple (weak) models can produce a more reliable and accurate model as compared to the individual base models [56]. Ensemble models are developed based on techniques such as bagging, boosting, and stacking. The bagging and boosting models both involve homogeneous weak learners that are combined by a deterministic strategy. The difference is bagging algorithms follow a parallel learning strategy (e.g., Random Forest) and boosting algorithms follow a sequential learning strategy (e.g., AdaBoost, XGBoost). Differently, stacking involves heterogeneous weak learners that are combined using a meta-model with a parallel learning strategy. Ensemble models have been used for AGB modeling and prediction by various studies in literature [8], [57], and [58]. However, developing a stacked ensemble with manual trial and error approaches to model AGB is an inefficient and time-consuming task. Meta-learning-driven AutoML systems can be adequately used for developing optimal stacked ensemble models (SEM) for AGB prediction [59]. AutoML systems use a collection of base models with hyperparameter tuning algorithms for producing SEMs [60]. The AutoGluon system [61] uses a multilayer stack ensembling with K-fold bagging that stacks models in multiple layers and trains in a layer-wise manner. The AutoSklearn system [28] employs a Bayesian optimization algorithm for searching through the hyperparameter space and meta-learning for warm-starting of the search procedure. The AutoSklearn 2.0 [35], which is an improvement upon Autosklearn, uses portfolio learning after performing Bayesian optimization. Among the most recent AutoML systems, "H2O" AutoML [62] performs stacked ensembling with random forest, gradient boosting

machines, linear models, and deep learning models using a super learning algorithm.

The primary objective of this article is the prediction of AGB using multitemporal multispectral (MT-MS) satellite remote sensing data. Prediction of AGB using satellite remote sensing requires a robust ML pipeline to deal with outliers in the training data and increase the generalization ability of the model. Moreover, satellite remote sensing is generally used for large-scale AGB mapping and may require testing multiple models due to the spatially varying characteristics of the forest. Thus, it is required to have a comprehensive evaluation framework for the multiple models that are considered. Therefore, we propose an AutoML system for the prediction of forest AGB that enables training and evaluation of models within a single framework. In particular, we use a meta-learning-driven AutoML system that automates the selection and stacking of candidate base learners for modeling AGB. Additionally, we propose to incorporate a SMBO procedure to automatically extract features from the MT-MS satellite data to minimize human-bias in the proposed ML pipeline.

Fig. 1. Study area with reference AGB plot locations (red dots).

II. STUDY AREA AND DATASETS

A. Study Area Description and Field Data

The study area is the province of Trento (6216 km^2) situated in north-eastern Italy in the southern part of the Alps. The area is mountainous and has a 60% of forest cover mostly owned by public institutions that are subject to broad goals of forest management (i.e., forest protection, species biodiversity, carbon storage etc.). The area has mountains with high elevations and landlocked valleys suitable for species such as Norway spruce (*Picea abies* (L.) Karst.) and Swiss pine (*Pinus cembra L.*). The area has mountains with lower elevations of an average 1000 m above sea level that are suitable for species such as Silver fir (*Abies alba* Mill.) and European beech (*Fagus sylvatica* L.). The areas that are below 1000 m are mainly characterized by broadleaves species such as *Ostrya carpinifolia, Carpinus betulus, Fraxinus ornus, Quercus pubescens, Quercus petrae,* etc. The geographical location of the study area and the distribution of the field plots in the study area are shown in Fig. 1.

The field data consists of 315 circular plots with a fixed radius of 15 m (see Fig. 1). These data include 98 broadleaf plots, 152 coniferous plots, and 65 mixed plots. The total 315 plots were divided into three categories (broadleaf, coniferous, and mixed plot) such that more than 80% of the plot AGB should be derived from one particular tree type (broadleaf or coniferous), otherwise the plot was considered as a mixed plot. All trees having a diameter at breast height (DBH) above 7 cm inside a plot were geolocated and their species, DBH and heights were measured. The AGB of each tree was computed using the allometric equations stated in [63], and the plot level AGB was computed as the sum of the AGB of all measured trees inside the plot. The field-estimated plot level AGB values ranged from 1.07 to 711.41 Mg ha^{-1}. The plot coordinates were recorded using a survey-grade GPS unit.
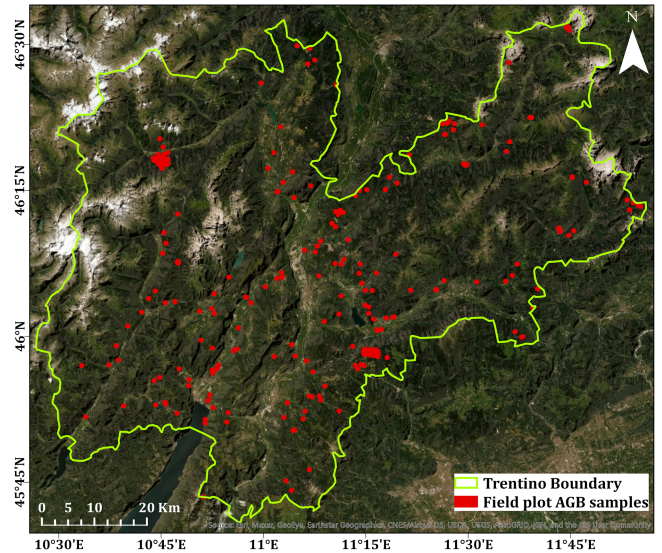
TABLE I
ACQUISITION DATES (YYYY-MM-DD) OF SENTINEL-2 IMAGES

Seasons	Dates of Acquisition (Sentinel-2)
Spring	2016-03-20
Summer	2016-07-18
Autumn	2016-10-16
Winter	2016-12-15

B. Remote Sensing Data

The study was performed using MT-MS images acquired by ESA's Sentinel-2A, 2B satellites. The acquisition dates of the Sentinel-2 images are stated in Table I. Sentinel-2 images are characterized by 13 spectral bands at 60, 20, and 10 m spatial resolution. We used ten spectral bands for our experiments, i.e., four spectral bands (B, G, R, and NIR) at 10 m spatial resolution and six spectral bands (3-Red Edge, Narrow NIR, and 2-SWIR) at 20 m spatial resolution. The central wavelengths of these ten considered spectral bands ranged from 490 to 2190 nm.

III. DESCRIPTION OF ALGORITHMS

The developed mechanism for achieving the proposed objectives was based on coupling two algorithms: 1) Tree-structured Parzen Estimator (TPE) [64] which is a hyperparameter optimization algorithm based on an iterative search process on a dynamic search space and 2) the Super Learner (SL) [65] which is a V-fold cross-validation based meta-learning algorithm that selects weights for combining candidate models. The following subsections provide a description of the two algorithms.

A. TPE for Automatic Feature Extraction

The TPE algorithm is based on a SMBO approach that overcomes the limitations of computationally expensive and inefficient random search and grid search algorithms. The inputs required by the TPE algorithm are parameters (x) and loss (y) based on the prior search history to deduce hyperparameters for the next trial. The input pair (parameters and loss) is split into two densities $(\ell(x)$ and $g(x))$ based on the loss of the historical data by a γ quantile (2) of the search result. The TPE defines this split $p(x|y)$ using the two densities according to the following equations:

$$p\ (x|y) = \begin{cases} \ell\left(x\right) & \text{if } y < y^* \\ g\left(x\right) & \text{if } y \geq y^* \end{cases} \quad (1)$$

and

$$\gamma = p(y > y^*) = \int_{-\infty}^{y^*} p(y)dy \quad (2)$$

where $\ell(x)$ should be maximized as it represents the density formed from the observations $x^{(i)}$ that correspond to the loss (y) smaller than the target performance (y^*), whereas $g(x)$ should be minimized as it represents the density formed by using the remaining observations. Therefore, the output of the TPE algorithm can be simply characterized as the ratio between $g(x)$ and $\ell(x)$.

The idea of the TPE algorithm is based on maximizing the expected improvement $(\text{EI}_{y^*}(x))$ computed by (3) based on convergence of loss (y) and target performance (y^*) in subsequent trials

$$\begin{aligned} EI_{y^*}\ (x) &= \int_{-\infty}^{y^*} \max\left(y^* - y, 0\right) p\left(y|x\right) dy \\ &= \int_{-\infty}^{y^*} (y^* - y) \frac{p\left(x|y\right) p\left(y\right)}{p\left(x\right)} dy \end{aligned} \quad (3)$$

where $p(x)$ can be written as

$$p\left(x\right) = \int_{\mathbb{R}} p\left(x|y\right)\ p\left(y\right) dy \ = \ \gamma\ell(x) + (1 - \gamma)\,g\left(x\right). \quad (4)$$

Therefore, we can deduce

$$\begin{aligned} EI_{y^*}(x) &= \int_{-\infty}^{y^*} (y^* - y)p(x|y)p(y)dy \\ &= \ell(x) \int_{-\infty}^{y^*} (y^* - y)p(y)dy \\ &= \gamma y^* \ell(x) - \ell(x) \int_{-\infty}^{y^*} p(y)dy. \end{aligned} \quad (5)$$

Finally combining (4) and (5), we obtain

$$\begin{aligned} EI_{y^*}(x) &= \left(\gamma + \frac{g(x)}{\ell(x)}(1 - \gamma)\right)^{-1} \cdot \left(\gamma y^* - \int_{-\infty}^{y^*} p(y)y dy\right) \\ &\propto \left(\gamma \frac{g(x)}{\ell(x)}(1 - \gamma)\right)^{-1}. \end{aligned} \quad (6)$$

The expression in (6) is a product of two terms of which the second term is independent of "x" and hence the expression is directly proportional to the first term and implies that to maximize improvement points "x" with high probability under $\ell(x)$ and low probability under $g(x)$ are desirable. Also, this expression shows that the ratio $\frac{g(x)}{\ell(x)}$ should be minimized in order to increase the expected improvement.

B. SL Algorithm for Automated Stacked Ensemble Modeling

The SL algorithm can be used for ensemble modeling by considering a set of diverse modeling algorithms as base learners and selecting an optimal ensemble through V-fold cross-validation. We first define a library "$\mathcal{L}$" of "K" modeling algorithms trained to solve a regression problem of estimating the expectation of target $'A'$ given the observed variable $'A'$. The optimal value of the parameter of interest $'\psi_0(A)'$ that is required to be estimated and the related loss function $'L(O, \psi)'$ that indicates the difference between the observed and the predicted value (squared error loss) can be written as

$$\psi_0\ (A) = E\left(B|A\right) \quad (7)$$

$$L\left(O, \psi\right) = (B - \psi\left(A\right))^2. \quad (8)$$

Thus, the optimal values of the parameter of interest for the "K" individual algorithms included in the library "$\mathcal{L}$" can be written as $'\hat{\psi}_k(A)'$ where $k = 1, 2, \ldots., K$. The number of algorithms "K" to be considered in the library is dependent on the sample size of the data. Note that all the considered algorithms estimate the same parameter $'\hat{\psi}_k(A)'$ but may use different subsets of $'A'$, different basis function, estimation procedures, and range of tuning parameters. The identification of the best predicting algorithm from the library over the data distribution (P_o) is determined by minimizing the expected risk difference $'d_n(\hat{\psi}_k, \psi_0)'$, i.e.,

$$d_n\left(\hat{\psi}_k, \psi_0\right) = \int \left\{L\left(B, \hat{\psi}_k\right) - L\left(B, \psi_0\right)\right\} dP_o \quad (9)$$

$$\widetilde{k}_n = \arg\min_{k}\{d_n(\widehat{\psi}_k, \psi_0)\}. \quad (10)$$

The use of the same data to estimate $\hat{\psi}_k(A)$ and $d_n(\hat{\psi}_k, \psi_0)$ generates bias in the estimation of the true risk to determine the best algorithm. Thus a V-fold cross-validation selector is used for unbiased estimation of the risk. Given the empirical distributions for the training and validation sets for each V-fold, the cross-validation selector can be defined as

$$\widehat{k} = \arg\min_{k} \left\{ E_{Cn} \sum_{i:C_n(i)=1} L(B_i \widehat{\psi}_k(A_i|P^0_{n,C_n})) \right\} \quad (11)$$

where P^0_{n,C_n} and P^1_{n,C_n} are the distributions of training set $(C_n(i) = 0)$ and validation set $(C_n(i) = 1)$, respectively.

IV. PROPOSED APPROACH

The flowchart of the proposed approach given in Fig. 2 shows the process of integrated implementation of TPE and SL algorithms to create an ML pipeline for AGB prediction. In the following subsections, the pre-processing of Sentinel-2
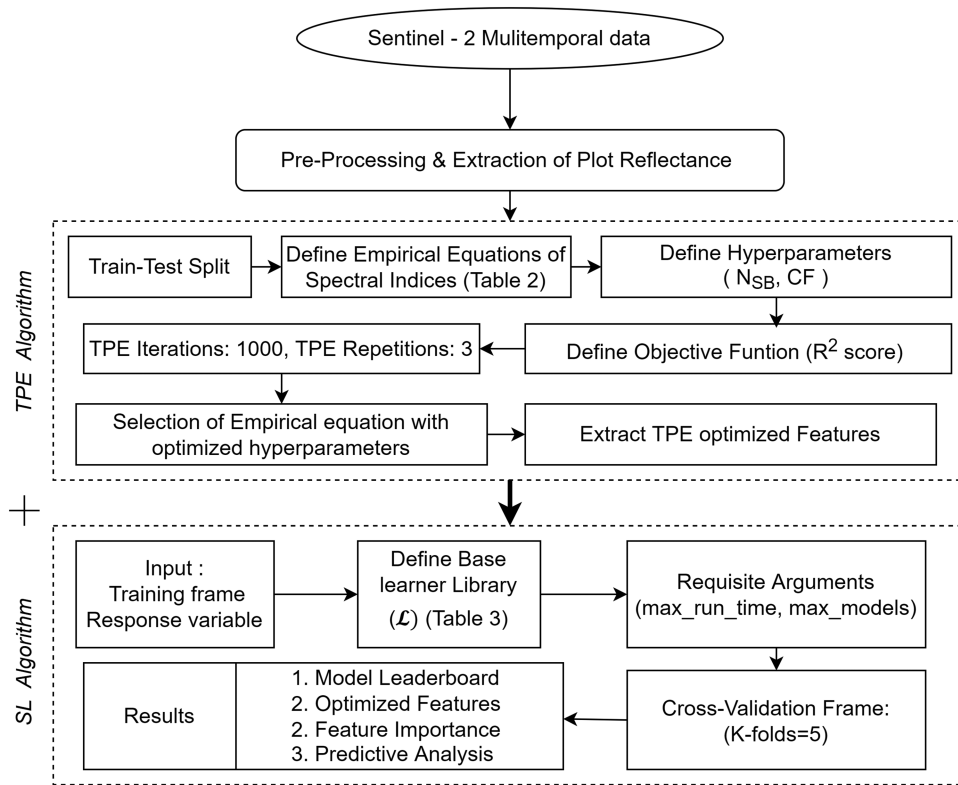
Fig. 2.   Proposed approach for AGB prediction.

data, implementation of TPE and SL algorithms, and model explanation parameters are described in detail.

The extraction of targeted spectral features from Sentinel-2 data for forest AGB prediction requires a comprehensive understanding of the dynamic changes of optical properties with respect to AGB. Vegetation indices are standardized yet retain bias attributed to their pre-defined equations. The TPE algorithm provides flexibility to the spectral features and fits spectral bands to empirical equations based on dynamics with AGB. The TPE algorithm supports spectral bands in parameter search space and composes the spectral features in computationally efficient manner. This makes TPE algorithm an optimal choice for producing target-oriented spectral features from empirical equations. This process was followed by the deployment of the SL algorithm that uses K-fold crossvalidation to estimate the performance of multiple ML models (base learners) and the same model with different hyperparameter settings. This creates an optimal weighted average of base learners (i.e., an ensemble model) that improves prediction accuracy, avoids overfitting, and minimizes parametric assumptions. Thus, the SL algorithm becomes a suitable choice to achieve the objective of stacked ensemble modeling for prediction of forest AGB.

A. Data Preprocessing

The Sentinel-2 images were acquired in Level-1C (top of the atmosphere reflectance) format and converted to Level-2A (bottom of the atmosphere reflectance) format including atmospheric and terrain correction using the Sen2cor processor [66]. The

spectral bands at 20 m spatial resolution were resampled at 10 m using the nearest neighbor sampling method for spatial consistency and performing computations. The preprocessed data were used for extracting plot reflectance and preparing season-wise analysis-ready data frames using "*rgeos*" and "*rgdal*" packages of R software. The analysis-ready data frames of each season consisted of plot reflectance values for each spectral band for all sample plots.

B. Implementation of Algorithms

The two core frameworks of the proposed approach are based on TPE and SL algorithms implemented sequentially to produce a robust ML pipeline for AGB prediction. First, the TPE algorithm generates automatically optimized features from the preprocessed analysis-ready data frames. Later, these optimized features and the response variable are used for training the SL algorithm that automates the process of training a large number of base models and performs stacked ensembling to produce a leaderboard of models. The TPE algorithm was implemented using "Optuna" framework that was developed in [67] and the SL algorithm was implemented using "H2O-3" framework that was developed in [62]. The sequential implementation of the two frameworks is explained in detail in the following paragraphs.

The implementation of the TPE algorithm to extract features from the input spectral bands requires a model, i.e., an empirical equation for which it can generate a set of parameters and select a combination of optimal spectral bands to accurately predict the

TABLE II
EMPIRICAL EQUATIONS OF SPECTRAL INDICES FOR THE TPE ALGORITHM

No. of Spectral Bands (N_{SB} = 2-6)	Index No. (I_n = 1-33)	Empirical Equations
Two	1	(B1-B2)/(B1+B2)
	2	(B1/B2)
	3	B1-B2
	4	α + β (B1-B2)/(B1+B2)
	5	(B1/B2) - α
	6	log(1/B1)-log(1/B2)
	7	log(1/B1)+log(1/B2)
	8	α [(B1- αB2- β)/(αB1+B2-α+ ρ)]
	9	B1- α [B2/(B1+ β*B2)]
	10	{(B1/B2)-1}/sqrt{(B1/B2)+1}
Three	11	(B1-B2)/(B3-B2)
	12	α (β(B1+B2) – B3)
	13	B1(B2/B3^2)
	14	2*B1-B2-B3)/ (2*B1+B2+B3)
	15	[(B1-B2)/(B1+B2)]/[(B1-B3)/(B1+B3)]
	16	(B1-B2-α(B2-B3))/ (B1+B2-α(B2-B3))
	17	(B1-B2)/B3
	18	{B1−(B2+B3)}/{B1+(B2+B3)}
	19	B1/(B1+B2+B3)
	20	arctan({(α*B1-B2-B3)/β}{B2-B3})
	21	α [β(B1-B2)- ρ(B3-B2)]/sqrt[(2*B1+1)^2-(γ*B1- σ*sqrt(B3))- α/3]
Four	22	(B1/B2)/[(B3-B4)/(B3+B4)]
	23	B1/(B2+B3+B4)
	24	{(B1+B2)-B3}/(B4-B3)
	25	[(B1-B2)- α(B1-B3)(B1/B2)]/[(1+α)(B4-B2)/(B4+B2+ α)]
Five	26	[(B1-B2)-α(B1-B3)(B1/B2)]/[α{β(B4-B3)-ρ(B2-B3)/sqrt((B4+1)^2-(B5-ρ*2*sqrt(B2))-0.5)}]
	27	B1/(B2+B3+B4+B5)
	28	(B1-B2)/(B2+B3+B4+B5)
	29	{(B1+B2)-B3}/(B4-B5)
Six	30	(B1*B2*B3)/ (B4*B5*B6)
	31	B1/(B2+B3+B4+B5+B6)
	32	(B1-B2)/(B2+B3+B4+B5+B6)
	33	[(B1-B2)-α(B1-B3)(B1/B2)]/[(1+α)(B4-B2)/(B4+B2+B5+B6+α)]

target variable. The considered empirical equations are given in Table II. The library of 33 empirical equations (indexed as I_n) framed from an exhaustive database of 500+ spectral indices [68] was available from [69]. The TPE randomly selects an empirical equation and generates parameters depending on the number of spectral bands (N_{SB}) and the coefficients (CF = α, β, γ, ρ, and σ) in the equation. A generalized linear regression model is fitted to select the optimal spectral bands and coefficient values to maximize the coefficient of determination (objective function). The coefficient of determination enables quantifying the proportion of the variance in extracted optimized feature w.r.t the target AGB values. The coefficient of determination should be maximized to determine the best fit in terms of spectral bands and coefficient values for the empirical equations. The TPE initialization provides a pair of

parameters and loss as stated in Section III-A that are split into two densities ($\ell(x)$ and $g(x)$) as per (1) and (2). The ratio of the two densities should be minimized (thus maximizing the objective function (R^2) in our case) through an iterative process and the best empirical equation is determined with the related optimum set of parameters (spectral bands, coefficient value). However, the random initialization creates a selection bias in the TPE algorithm that favors the empirical equations with less number of parameters, i.e., empirical equations involving less spectral bands. This selection bias problem was resolved using a meta-learning strategy by performing optimization process in groups. In total, the empirical equations were divided into five groups based on the number of spectral bands (N_{SB} = 2, 3, 4, 5, 6) and 1000 iterations were performed for each group. The best empirical equation in each group was identified and

TABLE III
LIST OF BASE MODELS FOR SL ALGORITHM

Models	Count	Type
eXtreme Gradient Boosting (XGBoost-1)	3	Pre-specified
Generalized Linear Model (GLM)	1	Fixed Grid
Distributed Random Forest (DRF)	1	Default
Gradient Boosting Machines (GBM-1)	5	Pre-specified
Deep Neural Net (DNN-1)	1	Near-default
eXtreme Gradient Boosting (XGBoost-2)	1	Random Grid
Gradient Boosting Machines (GBM-2)	1	Random Grid
Deep Neural Net (DNN-2)	1	Random Grid

after performing three repetitions of this process, the overall best empirical equation was identified. The selected empirical equation, optimal spectral bands, and coefficient values were used to compute the features and directed to the subsequent framework.

The SL framework receives the computed TPE-optimized automatic features and the target response variable as the input training frame. As previously mentioned, we used the H2O-3 platform to deploy the SL algorithm which consists of a library of base models stated in Table III. Additional details regarding the specifications of the base models and model parameters can be accessed from H2O.ai documentation (https://docs.h2o.ai/). The SL algorithm exploits the base models and derives an optimal SEM that minimizes the expected risk difference as per (10) and (11). In order to achieve the defined objective of this study, we used all the available base models in the library (see Table III). All base models were trained until their convergence with five-fold cross-validation on a 128 GB NVIDIA GeForce GTX 3090 GPU and Linux-based operating system. However, the maximum number of models and maximum runtime for models can be specified before the training process depending on the available computational budget and time restrictions.

C. Model Explanations

The process of sequential deployment of TPE and SL frameworks followed in the proposed approach results in TPE-optimized features, model rankings, model predictions, and prediction assessment metrics. Model explanations such as model rankings, evaluation statistics, regression scatterplots, and feature importance chart are provided as results. The performance of the base models and the SEM were evaluated on the basis of different evaluation metrics and ranked according to the model agreement score (coefficient of determination). All models were cross-validated using five-fold cross-validation method. The metrics used to evaluate these models are coefficient of determination (R^2_{cv}), root mean squared error (RMSE), root mean squared log error (RMSLE), mean absolute error (MAE), mean absolute percentage error (MAPE), and relative absolute error (RAE). The R^2_{cv} shows the goodness-of-fit of the regression

model, the RMSE shows the standard deviation of the residuals, RMSLE shows the log-transformed standard deviation of residuals, MAE shows the mean of absolute values of residuals, MAPE shows the accuracy of prediction as a percentage, and RAE shows accuracy of measurements relative to the range of the observed variable.

V. EXPERIMENTAL RESULTS

A. TPE Optimized Features and Feature Importance

The features for the SL framework were extracted based on the hyperparameter optimization results of the TPE algorithm. The empirical equations with respective spectral bands and coefficient values for each season selected post-TPE optimization procedure are given in Table IV. The selected empirical equations have been referenced using index numbers (I_n) mentioned in Table II. The spectral bands selected for the respective equations have been referenced w.r.t their central wavelengths (in nm). The corresponding Sentinel-2 spectral band for each central wavelength can be identified from Copernicus website (https://sentinels.copernicus.eu/). The empirical equations that had multiple selected combinations of spectral bands were referenced as "a," "b," and "c."

In total, 24 extracted features were selected for training the base models for AGB prediction following the TPE-based optimization procedure. Particularly, there were six features from autumn, seven from spring, six from summer, and five from winter seasons. Interestingly, all extracted features consisted of spectral bands from the vegetation red-edge spectrum (VRE) and short-wave infrared (SWIR) spectrum of the Sentinel-2 data. The TPE algorithm that was conditional on the target variable for the optimization process suggested an important role of the VRE and SWIR spectral bands for modeling forest AGB. Precisely, the SWIR band at $\lambda_c = 1610$ were selected for 21 out of the 24 extracted features and the contribution of VRE was distributed at $\lambda_c = 705, 740,$ and 865. The season-wise analysis indicated that VRE at $\lambda_c = 740$ and SWIR at $\lambda_c = 1610$ was selected for all autumn features. Also, SWIR at $\lambda_c = 1610$ was selected for five features and $\lambda_c = 2190$ for the remaining two features. For the summer season, VRE at $\lambda_c = 705$ and 865 were selected for five out of the total six features and SWIR at $\lambda_c = 1610$ for all features. Thus, the clear observable pattern in the specific spectral contributions of the extracted features accounts for their target-oriented properties.

The stacked bar chart in Fig. 3 shows the computed feature importance for all the considered base models. The features have been referenced as "index_season" and the scaled feature importance values of the base models have been color coded for analyzing the percentage contributions. The highest average feature importance was observed for "12a_sum" (79%) that was extracted with $\lambda_c = 705, 490,$ and 665. The feature achieved specific feature importance of 74% for DNN, 24% for DRF, and 100% for GBM, XGBoost, and GLM. This indicated that feature "12_sum" contributed significantly for most of the base models and has a dominant role in modeling AGB. A total of five features for DNN (26_spr, 12a_sum, 8b_sum, 8a_sum, 12b_sum), two features for DRF (28c_aut, 33_sum), five features for GBM

TABLE IV
TPE OPTIMIZED FEATURES FOR ALL SEASONS

Index (I_n)	Spectral Band (Central Wavelength (λ_c) in *nm*)	Season (aut, spr, sum, win)
17	842, 1610, 740	autumn (aut)
23	842, 1610 740, 2190	autumn
28	a = 1610, 842, 665, 740, 740 b = 1610, 842, 705, 740, 740 c = 1610, 842, 740, 2190, 665	autumn
32	842, 1610, 665, 740, 783, 740	autumn
16	a = 705, 842, 2190 (α=1) b= 2190, 842, 705 (α=1)	spring (spr)
23	a = 842, 705, 665, 1610 b = 842, 665, 1610, 740	spring
27	842, 665, 1610, 705, 740	spring
26	842, 1610, 665, 705, 705	spring
32	842, 705, 2190, 1610, 665, 1610	spring
7	1610, 705	summer (sum)
8	a = 705, 1610 (α=0.4, β=0.7, ρ=0.9) b = 665, 1610 (α=0.6, β=0.6, ρ=0.9)	summer
12	a = 705, 490, 665 (α=0.8, β=1) b = 865, 665, 1610 (α=0.03, β=0.1)	summer
33	1610, 665, 490, 865, 560, 560 (α = 0.01)	summer
23	842, 705, 1610, 1610	winter (win)
27	842, 1610, 1610, 1610, 705	winter
28	a = 842, 1610, 2190, 665, 1610 b = 842, 1610, 705, 665, 1610	winter
32	842, 1610, 665, 1610, 705, 1610	winter

(12a_sum, 32_aut, 23_aut, 28b_aut, 7_sum), three features for XGBoost (12a_sum, 7_sum, 12b_sum), and five features for GLM (12a_sum, 7_sum, 33_sum, 23_aut, 17_aut) were characterized by feature importance score greater than 50%. The feature "12a_sum" achieved the highest feature importance for three (GBM, XGBoost, and GLM) out of the five base models and the features '26_spr' and '28b_aut' achieved highest feature importance for DNN, and DRF, respectively. All these features were associated with either summer, autumn, or spring season indicating comparatively low importance of winter features in predicting AGB. Overall, the summer features were the most dominant with a total of 12 out of 20 features with a feature importance greater than 50%.

B. Model Leaderboard and Predictive Analysis

The H2O-3-based AutoML framework used to implement the SL algorithm trains all the base models (see Table III) and also produces an optimal SEM for predicting AGB. The total computational time for executing the SL algorithm to produce the modeling results was 1215 s.

The model ranking results and computed assessment metrics shown in Table V indicated that SEM achieved the best overall performance for predicting forest AGB. The SEM model achieved the highest agreement of $R^2_{cv} = 0.71$ and the best prediction precision RMSE = 74.44 Mg ha^{-1}. The RMSLE = 0.56 was equal for the top three ranked models despite a significant difference in the RMSE indicating the existence of outliers in the data. This explained the slightly higher MAE of the SEM despite achieving better model agreement as compared to DNN and DRF. The same explanation is applicable to the DNN and DRF which have a significant difference in the MAE and MAPE despite an identical RMSLE score. This suggests that SEM is more robust to outliers and the meta-learning process enabled the model to achieve better model fitting. The DNN model achieved the second-best performance on the leaderboard and the overall results suggested DNN to be more prone to outliers as compared to SEM.

The bagging-based DRF model achieved third position on the leaderboard with $R^2_{cv} = 0.65$ and RMSE = 82.01 Mg ha^{-1}. The next two positions were achieved by boosting-based GBM and XGBoost models. Both GBM and XGBoost had an identical performance and GBM outperformed XGBoost by a small margin. The GBM and XGBoost models achieved R^2_{cv} = 0.5, 0.48 and RMSE = 98.49, 99.95 Mg ha^{-1}, respectively. The GLM model achieved the last position on leaderboard with $R^2_{cv} = 0.38$ and RMSE = 109.44 Mg ha^{-1}.

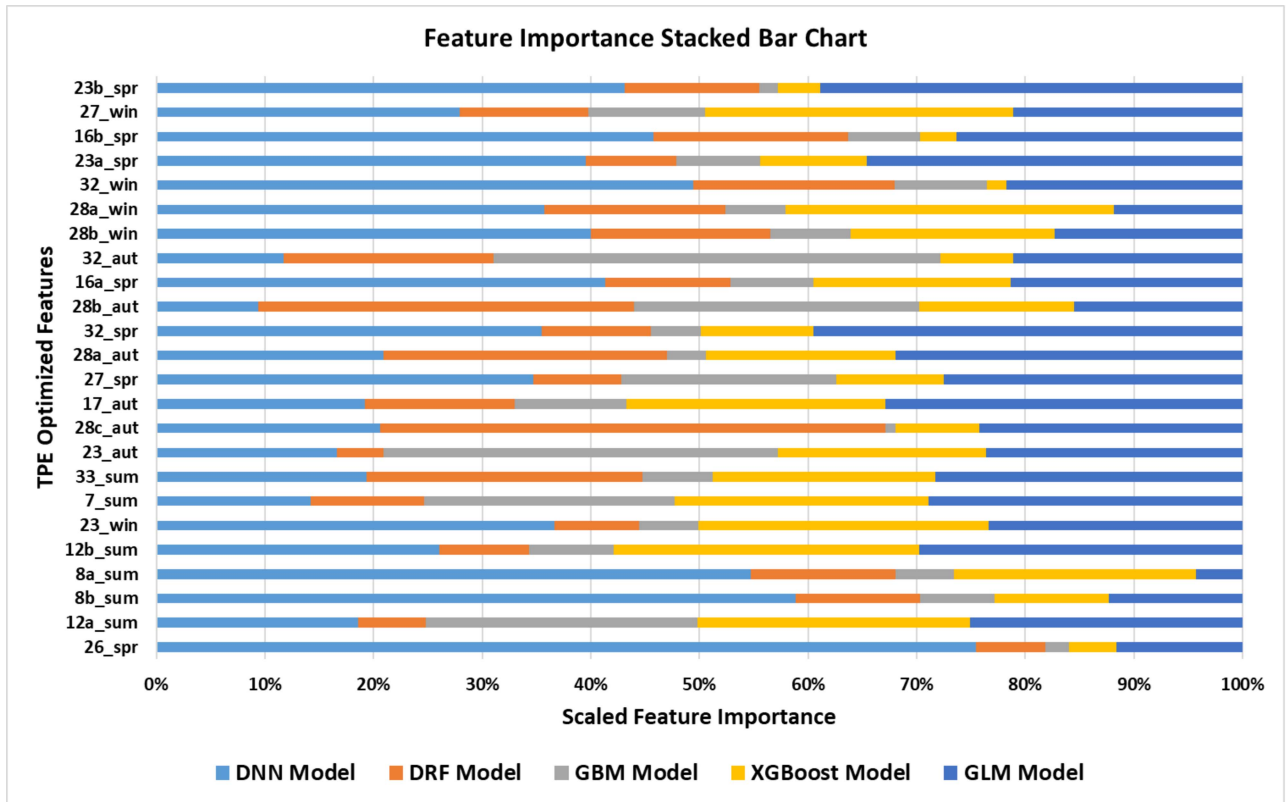Fig. 3. Feature importance stacked bar chart.

TABLE V
MODEL LEADERBOARD AND ASSESSMENT METRICS

Models	Rank	R^2_{cv}	RMSE (Mg ha^{-1})	RMSLE	MAE (Mg ha^{-1})	MAPE	RAE
SEM	1	0.71	74.44	0.56	48.58	0.30	0.43
DNN	2	0.66	81.27	0.56	48.48	0.30	0.43
DRF	3	0.65	82.01	0.56	51.17	0.36	0.45
GBM	4	0.50	98.49	0.65	74.28	0.38	0.66
XGBoost	5	0.48	99.95	0.64	73.70	0.39	0.65
GLM	6	0.38	109.44	0.72	84.94	0.42	0.75

The RAE score for the SEM model was 0.43 and successively increased to 0.75 for other models on the leaderboard. This indicated the lowest saturation from SEM and successive increment in saturation for other models to predict AGB. The low RAE score for SEM and DNN models shows that these models can predict greater AGB values with better accuracy as compared to the other models. This is also evident from the scatterplots of the regression models shown in Fig. 4. The scatterplots clearly indicate that SEM, DNN, and DRF models efficiently predicted large values of AGB as compared to GBM, XGBoost, and GLM models. Thus, the proposed automated ML pipeline yielded SEM that outperformed single base models and efficiently modeled AGB. The results also pointed out the

crucial role of the meta-learning process in modeling AGB for eliminating uncertainties associated with the data and producing robust models.

The forest plot-type-based analysis of model performance was carried out by independently computing RMSE for broadleaves, coniferous, and mixed plots. The results (shown in Table VI) indicate that the SEM model (which showed the best overall performance) was sensitive to the type of forest plot. Overall, the least RMSE errors were observed for the mixed-type plots and the highest RMSE errors were recorded for broadleaf plots. This type of behavior was consistent with the results from previous studies that obtained more accurate AGB estimations results for conifers as compared to the broadleaves using spectral data
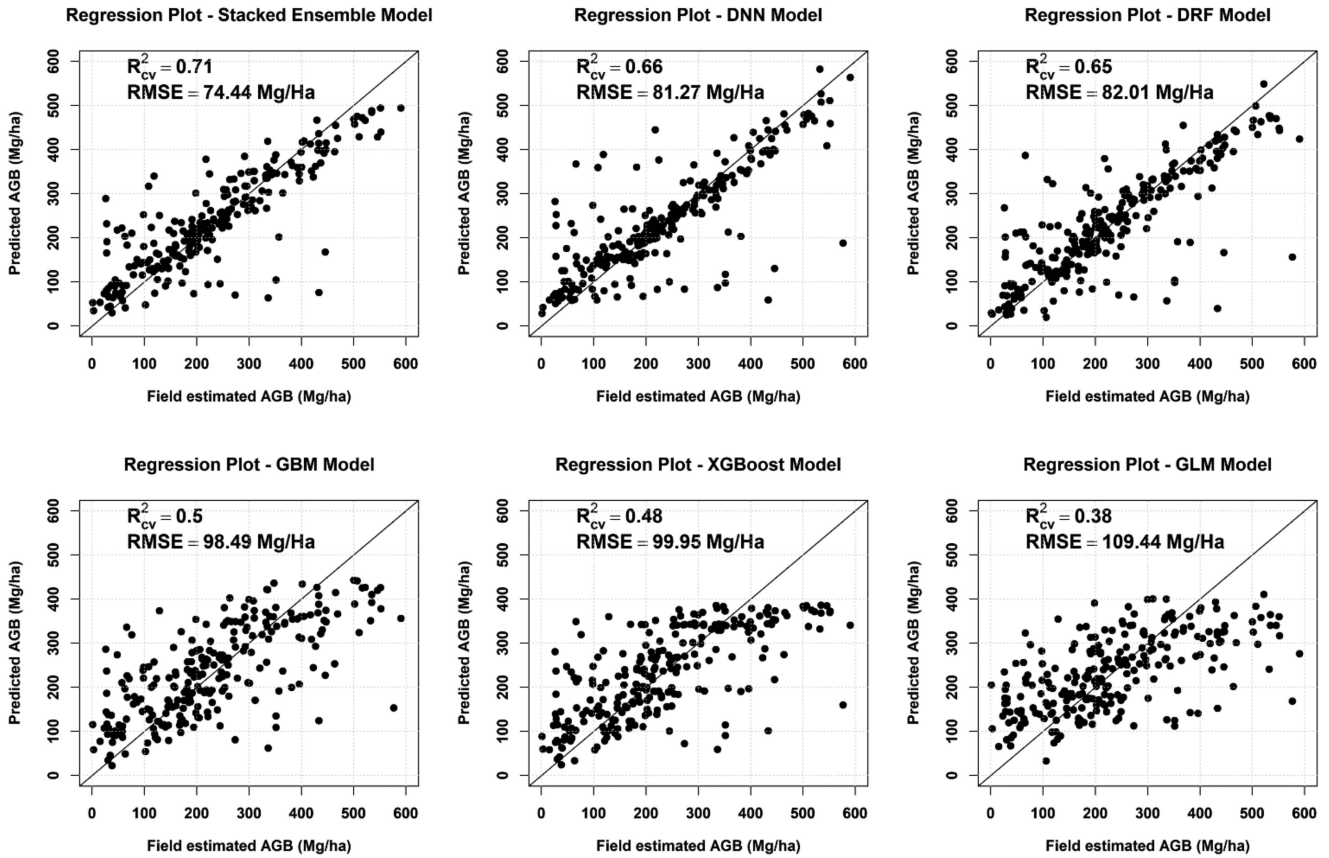
Fig. 4.   Regression scatter plots of the considered models.

TABLE VI
FOREST PLOT-TYPE (BROADLEAVES, CONIFEROUS, AND MIXED) BASED
PREDICTION RMSE FOR DEVELOPED MODELS

Models	Broadleaves RMSE (Mg ha^{-1})	Coniferous RMSE (Mg ha^{-1})	Mixed RMSE (Mg ha^{-1})
SEM	112.55	58.59	48.38
DNN	121.21	63.26	57.22
DRF	118.34	69.71	47.75
GBM	111.41	101.04	66.11
XGBoost	113.00	103.77	61.19
GLM	105.72	119.85	76.25

[89], [90]. Broadleaves have multiple canopy layers and a more complex structure that possess a challenge at a fundamental level with passive spectral data. Therefore, the SEM model showed the least RMSE error for coniferous type plots. It also delivered a comparable performance for the mixed-type plots (RMSE = 48.38 Mg ha^{-1}), where the DRF model achieved an RMSE score of 47.75 Mg ha^{-1}. Although all base models secured better RMSE scores as compared to SEM for broadleaf type. But, it is important to note that all the models were not trained independently on each distinct type of plot separately. This showed that the SEM model is capable of learning critical features and generalizing them for training data with multiple types of plots. This enabled SEM to achieve the best overall performance in spite of underperforming for a particular type of plot (i.e., broadleaf).

VI. DISCUSSION

In this article, we have proposed an automated ML pipeline for developing an SEM for the prediction of forest AGB using multitemporal Sentinel-2 data. The key elements of the ML pipeline were the TPE and SL algorithms that automated the process of feature extraction from the data and training a library of base models leading to the development of a stacked ensemble for modeling AGB. The automated ML pipeline was proposed for dealing with various issues identified in the literature related to the human-bias in AGB modeling and systematic evaluation of models. In this section, we extensively analyze our results with respect to the contemporary literature and precisely identify the advancements delivered by the study for AGB modeling using satellite remote sensing data.

The choice of features is very crucial in any modeling process and the performance of the models is highly dependent on their quality. The use of MS data for AGB modeling has led to the development and testing of various features. Practically, there are numerous combinations of spectral bands that could be used for extracting a feature from MS data for modeling AGB. Studies in literature use a few standard vegetation indices as features for modeling and mapping forest AGB [70], [71], [72], [73].

A comparative analysis of these studies indicates an unstable response of vegetation indices depending upon the spatial resolution, sensor specifications, and available spectral bands of the data. To state simply, a vegetation index identified as effective for a particular study area or data with a certain radiometric specification may not be as effective for other study areas or radiometric specifications. Thus, the human-intelligence-based selection of these vegetation indices can be highly inefficient and ambiguous for AGB modeling. In order to overcome this issue, our study proposed an automated mechanism for extracting such features. The TPE algorithm-based optimization procedure extracts features that are highly target-oriented and involves less human intervention. It is capable of overcoming the shortcomings identified in the literature with regard to extraction and selection of effective features from satellite MS data for AGB modeling. The reduction of human-bias from the process enables the development of more robust and reliable features. Moreover, it performs necessary changes in composing the features based on the specifications of the data such as spatial, spectral, and radiometric resolution. Thus, our study demonstrated the success of the proposed automated approach that led to consistent and accurate modeling results.

The second component of the modeling process after developing robust features is the choice of a modeling algorithm. The type of modeling algorithm substantially affects the precision and accuracy of the predictions. As per literature, there are several studies that use different ML models for predicting forest AGB [7], [57], [58] and provide a comparative assessment of the models in order to identify the best modeling algorithm for AGB prediction. Apart from the range of ML models that can be used, each model has associated hyperparameters that require tuning. A faulty hyperparameter tuning can adversely affect the performance of an ML model and restrict its generalization capability. Moreover, studies that use the same ML models but with a different combination of hyperparameters or architecture sometimes lead to contrasting performance on an identical task. For example, the study in [58] identified XGBoost to deliver the best performance as compared to the other competing models. However, the XGBoost model has multiple associated hyperparameters that were defined manually for finding the best combination of hyperparameters using a grid search. This introduces a human-bias in the process and reduces the chances of reproducing the results for other data or scenarios. To deal with this problem, our study proposed the use of an AutoML approach that effectively automates the iterative tasks associated with the development of a model. Precisely, it automates the process of hyperparameter selection and a range of ML models are trained in the same pipeline for effective comparison and reproducible results.

Studies have identified that a combination of models (stacked ensemble) can produce more efficient results as compared to a single model [74], [75]. There are only a few studies that focused on using SEMs for remote-sensing-based forest applications [6], [76]. These studies used a manual or semi-automatic approach for identifying an optimal combination of models for AGB prediction. An optimal SEM requires a library of diverse models and a systematic algorithm to evaluate the combination of models with optimized hyperparameters and generate a meaningful explanation of models. The solution for this complex problem was provided by the proposed use of SL algorithm implemented using H2O-3 framework that produced a stacked ensemble of base models. The developed SEM model in this study delivered the best performance as compared to the individual base models. Moreover, it limited the required number of user-defined parameters reducing human-bias in the ensemble selection. Thus, a reliable and automated pipeline for robust stacked ensembling and model training was established in this study.

VII. CONCLUSION AND FUTURE WORK

This article proposed an end-to-end ML pipeline for modeling forest AGB using MT-MS satellite remote sensing data. The results demonstrated that reducing human-bias from the modeling process and deploying a comprehensive model evaluation strategy under a single framework can provide better model explanations. The derived model explanations can be instrumental to frame effective schemes for accurately mapping AGB on large areas with diverse forest characteristics. Moreover, instead of using predefined features for modeling, an automated optimization procedure can produce more effective features by weighing more on the spectrum of the data that holds greater importance in explaining the target AGB. The future developments of this work can possibly aim at improving the robustness of the proposed pipeline by the addition of optimization elements and by replacing or improving the deployed meta-learning strategies. The advances in the latest AutoML systems can be possibly incorporated to derive additional model explanations for framing better modeling schemes for large-scale AGB mapping. These advanced AutoML systems could be possibly deployed at a global scale to develop models that can handle a greater diversity of forest types, species, and geographical conditions.

REFERENCES

[1] T. Dube and O. Mutanga, "Evaluating the utility of the medium-spatial resolution Landsat 8 multispectral sensor in quantifying aboveground biomass in uMgeni catchment, South Africa," *ISPRS J. Photogramm. Remote Sens.*, vol. 101, pp. 36–46, 2015.

[2] G. M. Foody, D. S. Boyd, and M. E. J. Cutler, "Predictive relations of tropical forest biomass from Landsat TM data and their transferability between regions," *Remote Sens. Environ.*, vol. 85, no. 4, pp. 463–474, 2003.

[3] N. Ghasemi, M. R. Sahebi, and A. Mohammadzadeh, "A review on biomass estimation methods using synthetic aperture radar data," *Int. J. Geomatics Geosci.*, vol. 1, no. 4, pp. 776–788, 2011.

[4] M. L. R. Sarker, J. Nichol, B. Ahmad, I. Busu, and A. A. Rahman, "Potential of texture measurements of two-date dual polarization PALSAR data for the improvement of forest biomass estimation," *ISPRS J. Photogramm. Remote Sens.*, vol. 69, pp. 146–166, 2012.

[5] M. Zhang, H. Lin, S. Zeng, J. Li, J. Shi, and G. Wang, "Impacts of plot location errors on accuracy of mapping and scaling up aboveground forest carbon using sample plot and landsat tm data," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1483–1487, Nov. 2013.

[6] Y. Zhang, J. Ma, S. Liang, X. Li, and J. Liu, "A stacking ensemble algorithm for improving the biases of forest aboveground biomass estimations from multiple remotely sensed datasets," *GIScience Remote Sens.*, vol. 59, no. 1, pp. 234–249, 2022.

[7] L. Chen, Y. Wang, C. Ren, B. Zhang, and Z. Wang, "Optimal combination of predictors and algorithms for forest above-ground biomass mapping from Sentinel and SRTM data," *Remote Sens.*, vol. 11, no. 4, 2019, Art. no. 414.

[8] R. C. Sharma, "Ensemble learning of Multi-source satellite sensors dataset for estimating forest biomass in New England region," MDPI AG, Feb. 16, 2021, doi: 10.20944/preprints202102.0338.v1.

[9] C. Liu et al., "Progressive neural architecture search," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 19–35.

[10] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Computation*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[11] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] D. Maclaurin, D. Duvenaud, and R. P. Adams, "Gradient-based hyperparameter optimization through reversible learning," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, vol. 37, pp. 1–10.

[13] X. He, K. Zhao, and X. Chu, "AutoML: A survey of the state-of-the-art," *Knowl.-Based Syst.*, vol. 212, 2021, Art. no. 106622.

[14] Q. Yao et al., "Taking human out of learning applications: A survey on automated machine learning," pp. 1–20, 2018, *arXiv:abs/1810.13306*.

[15] X. He et al., "Automated model design and benchmarking of 3D deep learning models for COVID-19 detection with chest CT scans," pp. 1–13, 2021, *arXiv:abs/2101.05442*.

[16] L. Faes et al., "Automated deep learning design for medical image classification by health-care professionals with no coding experience: A feasibility study," *Lancet Digit. Heal.*, vol. 1, no. 5, pp. e232–e242, 2019.

[17] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, "DetNAS: Backbone search for object detection," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[18] H. Xu, L. Yao, Z. Li, X. Liang, and W. Zhang, "Auto-FPN: Automatic network architecture adaptation for object detection beyond classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 6648–6657.

[19] J. Guo et al., "Hit-detector: Hierarchical Trinity architecture search for object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11402–11411.

[20] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li, "Fast, accurate and lightweight super-resolution with neural architecture search," in *Proc. Int. Conf. Pattern Recognit.*, 2020, pp. 9015–9022.

[21] D. Song, C. Xu, X. Jia, Y. Chen, C. Xu, and Y. Wang, "Efficient residual dense block search for image super-resolution," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 12007–12014.

[22] Y. Guo, Y. Luo, Z. He, J. Huang, and J. Chen, "Hierarchical neural architecture search for single image super-resolution," *IEEE Signal Process. Lett.*, vol. 27, pp. 1255–1259, 2020.

[23] Y. Jiang, C. Hu, T. Xiao, C. Zhang, and J. Zhu, "Improved differentiable architecture search for language modeling and named entity recognition," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2020, pp. 3585–3590.

[24] J. Chen, K. Chen, X. Chen, X. Qiu, and X. Huang, "Exploring shared structures and hierarchies for multiple NLP tasks," 2018.

[25] C. Liu et al., "Auto-deeplab: Hierarchical neural architecture search for semantic image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 82–92.

[26] C. Yang, J. Fan, Z. Wu, and M. Udell, "AutoML pipeline selection: Efficiently navigating the combinatorial space," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1446–1456.

[27] H. J. Escalante et al., "AutoML @ NeurIPS 2018 challenge: Design and results," in *The NeurIPS '18 Competition*. Berlin, Germany: Springer, 2020, pp. 209–229.

[28] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2962–2970.

[29] N. Pillay and R. Qu, *Automated Design of Machine Learning and Search Algorithms*. Berlin, Germany: Springer, 2021.

[30] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle swarm model selection," *J. Mach. Learn. Res.*, vol. 10, pp. 405–440, 2009.

[31] H. J. Escalante, M. Montes, and E. Sucar, "Ensemble particle swarm model selection," in *Proc. Int. Joint Conf. Neural Netw.*, 2010, pp. 1–8.

[32] D. A. Masood, *Automated Machine Learning Hyperparameter Optimization, Neural Architecture Search, and Algorithm Selection With Cloud Platforms*. Birmingham, U.K.: Packt Publishing Ltd, 2021.

[33] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 847–855.

[34] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Proc. Int. Conf. Learn. Intell. Optim.*, 2011, pp. 507–523.

[35] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-Sklearn 2.0: Hands-free AutoML via meta-learning," pp. 1–56, 2020.

[36] F. Hutter, H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration lecture notes in computer science," in *Proc. Int. Conf. Learn. Intell. Optim.*, 2011, pp. 507–523.

[37] R. S. Olson, O. Edu, and J. H. Moore, "TPOT: A tree-based pipeline optimization tool for automating machine learning," *J. Mach. Learn. Res.*, vol. 64, pp. 66–74, 2016.

[38] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, and F. Hutter, "Towards automatically-tuned neural networks," in *Proc. Workshop Autom. Mach. Learn.*, 2016, vol. 1, pp. 1–8.

[39] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[40] J. Vreeken and K. Yamanishi, "Auto-Keras: An efficient neural architecture search system," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1946–1956.

[41] J. Vanschoren, *Automated Machine Learning*. Berlin, Germany: Springer, 2018.

[42] J. P. Monteiro, D. Ramos, D. Carneiro, F. Duarte, J. M. Fernandes, and P. Novais, "Meta-learning and the new challenges of machine learning," *Int. J. Intell. Syst.*, vol. 36, no. 11, pp. 6240–6272, 2021.

[43] K. A. Smith-Miles, "Cross-disciplinary perspectives on meta-learning for algorithm selection," *ACM Comput. Surv.*, vol. 41, no. 1, pp. 1–25, 2008.

[44] M. Maher and S. Sakr, "SmartML: A meta learning-based framework for automated selection and hyperparameter tuning for machine learning algorithms," in *Proc. 22nd Int. Conf. Extending Database Technol.*, 2019, pp. 554–557.

[45] C. Yang, Y. Akimoto, D. W. Kim, and M. Udell, "OBoe: Collaborative filtering for automl model selection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1173–1183.

[46] K. Li and J. Malik, "Learning to optimize," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–9.

[47] F. Nargesian, H. Samulowitz, U. Khurana, E. B. Khalil, and D. Turaga, "Learning feature engineering for classification," in *Proc. Int. Jt. Conf. Artifical Intell.*, 2017, pp. 2529–2535.

[48] H. Li et al., "RS-MetaNet: Deep metametric learning for few-shot remote sensing scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6983–6994, Aug. 2021.

[49] M. El-Dairi and R. J. House, "Optic nerve hypoplasia," in *Handbook of Pediatric Retinal OCT and the Eye-Brain Connection*. Amsterdam, Netherlands: Elsevier, 2019, pp. 285–287.

[50] P. Zhang, Y. Bai, D. Wang, B. Bai, and Y. Li, "Few-shot classification of aerial scene images via meta-learning," *Remote Sens.*, vol. 13, no. 1, pp. 1–21, 2021.

[51] M. Ruswurm, S. Wang, M. Korner, and D. Lobell, "Meta-learning for few-shot land cover classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 788–796.

[52] M. Werther et al., "Meta-classification of remote sensing reflectance to estimate trophic status of inland and nearshore waters," *ISPRS J. Photogramm. Remote Sens.*, vol. 176, pp. 109–126, 2021.

[53] X. Tang et al., "Multi-scale meta-learning based networks for high-resolution remote sensing scene classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Brussels, Belgium, 2021, pp. 4928–4931, doi: 10.1109/IGARSS47720.2021.9555134.

[54] G. Sarmonikas and A. Mendez, "Meta-learning for instance segmentation on satellite imagery," *ResearchGate*, pp. 1–8, 2020, doi: 10.13140/RG.2.2.26490.41929.

[55] Q. Yang, Z. Ni, and P. Ren, "Meta captioning: A meta learning based remote sensing image captioning framework," *ISPRS J. Photogramm. Remote Sens.*, vol. 186, pp. 190–200, 2022.

[56] L. Rokach, *Pattern Classification Using Ensemble Methods*. Singapore: World Scientific, vol. 75, 2018.

[57] H. Su, W. Shen, J. Wang, A. Ali, and M. Li, "Machine learning and geostatistical approaches for estimating aboveground biomass in Chinese subtropical forests," *Forest Ecosystems*, vol. 7, 2020, Art. no. 64.

[58] Y. Li, M. Li, C. Li, and Z. Liu, "Forest aboveground biomass estimation using Landsat 8 and Sentinel-1A data with machine learning algorithms," *Sci. Rep.*, vol. 10, no. 1, pp. 1–12, 2020.

[59] K. T. Ngo, "Stacking ensemble for auto_ml," Doctoral Dissertations, Virginia Tech, Blacksburg, VA, USA, 2018.

[60] J. Yoo, T. Joseph, D. Yung, S. A. Nasseri, and F. Wood, *Ensemble Squared: A Meta AutoML System*. New York, NY, USA: Association for Computing Machinery, 2020, vol. 1, no. 1.

[61] N. Erickson et al., "AutoGluon-tabular: Robust and accurate AutoML for structured data," in *Proc. 7th ICML Workshop Automated Mach. Learn.*, 2020, pp. 1–16.

[62] E. Ledell and S. Poirier, "H2O AutoML: Scalable automatic machine learning," in *Proc. 7th ICML Workshop Automated Mach. Learn.*, 2020, pp. 1–16.

[63] M. Dalponte and D. A. Coomes, "Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data," *Methods Ecol. Evol.*, vol. 7, no. 10, pp. 1236–1245, 2016.

[64] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2011, vol. 24, pp. 2546–2554.

[65] M. J. Van Der Laan, E. C. Polley, and A. E. Hubbard, "Super learner," *Stat. Appl. Genet. Mol. Biol.*, vol. 6, no. 1, 2007, Art. no. 25.

[66] J. Louis et al., "Sentinel-2 SEN2COR: L2A processor for users," *Eur. Space Agency, (Special Publication ESA SP)*, vol. SP-740, pp. 9–13, 2016.

[67] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2623–2631.

[68] C. Henrich, V. Krauss, G. Götze, and C. Sandow, "IDB - index-database; development of a database for remote sensing indices," Crop Science Colloquium, Univ. of Bonn, 2012.

[69] J. C. O. Koh, B. P. Banerjee, G. Spangenberg, and S. Kant, "Automated hyperspectral vegetation index derivation using a hyperparameter optimisation framework for high-throughput plant phenotyping," *New Phytologist*, vol. 233, no. 6, pp. 2659–2670, 2022.

[70] N. Jha et al., "The real potential of current passive satellite data to map aboveground biomass in tropical forests," *Remote Sens. Ecol. Conservation*, vol. 7, no. 3, pp. 504–520, 2021.

[71] Y. Wang, X. Zhang, and Z. Guo, "Estimation of tree height and aboveground biomass of coniferous forests in North China using stereo ZY-3, multispectral Sentinel-2, and DEM data," *Ecological Indicators*, vol. 126, 2021, Art. no. 107645.

[72] A. B. Baloloy et al., "Estimation of Mangrove forest aboveground biomass using multispectral bands, vegetation indices and biophysical variables derived from optical satellite imageries: Rapideye, planetscope and sentinel-2," *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, vol. 4, no. 3, pp. 29–36, 2018.

[73] P. Naik, M. Dalponte, and L. Bruzzone, "Prediction of forest aboveground biomass using multitemporal multispectral remote sensing data," *Remote Sens.*, vol. 13, no. 7, Mar. 2021, Art. no. 1282, doi: 10.3390/rs13071282.

[74] F. Gunes, R. Wolfinger, and P.-Y. Tan, "Stacked ensemble models for improved prediction accuracy," in *Proc. Sas*, 2017, pp. 1–19.

[75] I. Kandel, M. Castelli, and A. Popovič, "Comparing stacking ensemble techniques to improve musculoskeletal fracture image classification," *J. Imag.*, vol. 7, no. 6, pp. 1–24, 2021.

[76] S. P. Healey et al., "Mapping forest change using stacked generalization: An ensemble approach," *Remote Sens. Environ.*, vol. 204, pp. 717–728, 2018.

[77] P. Naik, M. Dalponte, and L. Bruzzone, "A comparison on the use of different satellite multispectral data for the prediction of aboveground biomass," in *Proc. Image Signal Process. Remote Sens.*, 2020, Paper 1153315.

[78] M. A. Salinas-Melgoza, M. Skutsch, and J. C. Lovett, "Predicting aboveground forest biomass with topographic variables in human-impacted tropical dry forest landscapes," *Ecosphere*, vol. 9, no. 1, 2018, Art. no. e02063.

[79] G. Ying, Z. Li, E. Chen, and Q. He, "Estimation of forest biomass using Support Vector machines from comprehensive remote sensing data," in *Proc. IEEE Int. Conf. Remote Sens., Environ. Transp. Eng.*, 2011, pp. 2146–2150.

[80] P. Wang, S. Tan, G. Zhang, S. Wang, and X. Wu, "Remote sensing estimation of forest aboveground biomass based on Lasso-SVR," *Forests*, vol. 13, no. 10, 2022, Art. no. 1597.

[81] Y. Guo et al., "Optimal support vector machines for forest above-ground biomass estimation from multisource remote sensing data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2012, pp. 6388–6391.

[82] P. Varvia, T. Lahivaara, M. Maltamo, P. Packalen, and A. Seppanen, "Gaussian process regression for forest attribute estimation from airborne laser scanning data," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 6, pp. 3361–3369, Jun. 2019.

[83] R. Xie et al., "Mapping leaf area index in a mixed temperate forest using Fenix airborne hyperspectral data and Gaussian processes regression," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 95, 2021, Art. no. 102242.

[84] M. Luo et al., "Combination of feature selection and CatBoost for prediction: The first application to the estimation of aboveground biomass," *Forests*, vol. 12, no. 2, 2021, Art. no. 216.

[85] S. Nandy et al., "Neural network-based modelling for forest biomass assessment," *Carbon Manage.*, vol. 8, no. 4, pp. 305–317, 2017.

[86] L. Zhang, Z. Shao, J. Liu, and Q. Cheng, "Deep learning based retrieval of forest aboveground biomass from combined LiDAR and landsat 8 data," *Remote Sens.*, vol. 11, no. 12, 2019, Art. no. 1459.

[87] P. Naik, M. Dalponte, and L. Bruzzone, "Generative feature extraction from sentinel 1 and 2 data for prediction of forest aboveground biomass in the Italian Alps," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4755–4771, 2022.

[88] P. Naik, M. Dalponte, and L. Bruzzone, "A disentangled variational autoencoder for prediction of above ground biomass from hyperspectral data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 2991–2994.

[89] C. Li, M. Li, J. Liu, Y. Li, and Q. Dai, "Comparative analysis of seasonal landsat 8 images for forest aboveground biomass estimation in a subtropical forest," *Forests*, vol. 11, no. 1, 2019, Art. no. 45.

[90] C. Li, M. Li, and Y. Li, "Improving estimation of forest aboveground biomass using Landsat 8 imagery by incorporating forest crown density as a dummy variable," *Can. J. Forest Res.*, vol. 50, no. 4, pp. 390–398, 2020.