Enhancing High-Resolution Forest Stand Mean Height Mapping in China through an Individual Tree-Based Approach with Close-Range LiDAR Data

Yuling Chen[1,⋆], Haitao Yang[1,⋆], Zekun Yang[1], Qiuli Yang[2,3], Weiyan Liu[4], Guoran Huang[5], Yu Ren[1], Kai Cheng[1], Tianyu Xiang[6], Mengxi Chen[1], Danyang Lin[4], Zhiyong Qi[1], Jiachen Xu[1], Yixuan Zhang[1], Guangcai Xu[7], Qinghua Guo[1,8*]

[1]Institute of Remote Sensing and Geographic Information System, School of Earth and Space Sciences, Peking University, Beijing 100871, China

[2]College of Geography and Remote Sensing Science, Xinjiang University, Urumqi 800017, China

[3]Xinjiang Key Laboratory of Oasis Ecology, Xinjiang University, Urumqi 830017, China

[4]State Forestry and Grassland Administration Key Laboratory of Forest Resources & Environmental Management, Beijing Forestry University, Beijing 100083, China

[5]College of Forestry, Southwest Forestry University, Kunming 650224, China

[6]College of Earth Sciences, Chengdu University of Technology, Chengdu 610059, China

[7]Institute of Ecology, College of Urban and Environmental Sciences, Peking University, Beijing 100871, China

⋆These authors contributed equally to this work.

Correspondence to: Qinghua Guo (guo.qinghua@pku.edu.cn)

Supplementary data

Table S1. The mathematic formulas of the stand mean height growth equations.

NO.	Model	Equation
1	Logistic,1838	$H = \frac{A}{1 + m \times e^{-rt}}$
2	Mitscherlich,1850	$H = A \times (1 - e^{-rt})$
3	Gompertz,1825	$H = A \times e^{-m \times e^{-rt}}$
4	Korf,1939	$H = A \times e^{-m \times t^{-r}}$
5	Richards,1959	$H = A \times (1 - e^{-rt})^{m}$
6	Schumacher，1939	$H = A \times e^{-r/t}$

Notes: H is the forest stand height (m), t is the forest age (years), and A, m, and r ($r > 0$) are the model parameters. A is maximum growth parameter,m is the parameter related to the initial value,r is the growth rate parameter.

Table S2. The fitting results of stand weighted mean height growth model

NO.	Parameter estimation			RMSE	R2
	A	m	r		
1	15.937	6.057	0.101	4.4	0.48
2	16.762	---	0.043	4.5	0.46
3	16.182	2.320	0.071	4.5	0.47
4	1258.080	6.656	0.096	4.9	0.37
5	16.346	1.294	0.054	4.5	0.46
6	18.748	---	12.734	4.6	0.43

Table S3. Test results for stand weighted mean height growth model

NO.	ME	MAE	RMSE
1	**0.1**	**3.5**	**4.4**
2	0.0	3.6	4.5
3	0.1	3.6	4.4
4	0.8	3.8	4.9
5	0.1	3.6	4.5
6	0.2	3.7	4.6

Table S4. Comparison of deviations between weighted mean heights with different weights (w_1 and w_2) and Lorey's mean height (national forest inventory data)

Weight	ME	MAE	RMSE
$\boldsymbol{w_1}$	-0.0197	1.8069	2.4178
$\boldsymbol{w_2}$	-0.0072	1.8056	2.4174

Table S5. Model selection results based on *PyCaret*

Model	ID	Algorithm	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
$\boldsymbol{h_a}$	et	Extra Trees Regressor	1.6919	5.6881	2.3849	0.8456	0.2304	0.2094	36.987
	rf	Random Forest Regressor	1.8013	6.2508	2.5001	0.8303	0.2426	0.2256	149.23
	xgboost	Extreme Gradient Boosting	2.4366	10.4521	3.2329	0.7162	0.3096	0.3106	2.3
	catboost	CatBoost Regressor	2.5239	11.2227	3.35	0.6953	0.318	0.3214	5.143
	lightgbm	Light Gradient Boosting Machine	2.6471	12.1383	3.484	0.6705	0.3304	0.3395	2.491
	dt	Decision Tree Regressor	2.3943	12.1877	3.491	0.6691	0.331	0.2861	14.082
	knn	K Neighbors Regressor	2.6074	12.6528	3.557	0.6565	0.3361	0.3308	8.528
	gbr	Gradient Boosting Regressor	2.9095	14.6332	3.8253	0.6027	0.3552	0.3736	434.882
	ridge	Ridge Regression	3.2646	18.7622	4.3315	0.4907	0.3881	0.4112	0.851
	lr	Linear Regression	3.2648	18.7634	4.3316	0.4906	0.388	0.4112	1.018
	en	Elastic Net	3.5316	21.9934	4.6897	0.403	0.4076	0.444	6.587
	lasso	Lasso Regression	3.5516	22.2246	4.7142	0.3967	0.4101	0.4479	3.508
	llar	Lasso Least Angle Regression	3.5515	22.2239	4.7142	0.3967	0.4101	0.4478	0.483
	ada	AdaBoost Regressor	3.7989	22.4426	4.7318	0.3906	0.4513	0.5552	58.039
	huber	Huber Regressor	3.5113	23.2	4.8164	0.3702	0.4037	0.4099	19.96
	omp	Orthogonal Matching Pursuit	3.7264	24.1636	4.9156	0.344	0.4306	0.4726	0.478
	dummy	Dummy Regressor	4.8126	36.8399	6.0695	-0.0001	0.5225	0.6262	0.579
	par	Passive Aggressive Regressor	7.7406	101.2993	8.9423	-1.7571	0.6511	1.0584	2.015
$\boldsymbol{h_w}$	et	Extra Trees Regressor	1.9065	7.3298	2.7073	0.8136	0.2392	0.2079	37.832
	rf	Random Forest Regressor	2.0165	7.9122	2.8128	0.7988	0.249	0.2221	235.223
	xgboost	Extreme Gradient Boosting	2.7142	12.7939	3.5768	0.6747	0.3115	0.3031	1.389
	catboost	CatBoost Regressor	2.8081	13.6734	3.6977	0.6523	0.3194	0.3135	3.786
	lightgbm	Light Gradient Boosting Machine	2.9343	14.6817	3.8316	0.6267	0.3303	0.3294	1.303
	knn	K Neighbors Regressor	2.9068	15.4093	3.9254	0.6082	0.3364	0.3222	9.02
	dt	Decision Tree Regressor	2.6828	15.4519	3.9308	0.6071	0.3402	0.2784	14.538
	gbr	Gradient Boosting Regressor	3.2151	17.4789	4.1807	0.5555	0.3528	0.3609	305.293
	lr	Linear Regression	3.5761	21.9306	4.683	0.4424	0.3806	0.3924	1.011
	ridge	Ridge Regression	3.5758	21.9286	4.6827	0.4424	0.3807	0.3924	0.844
	en	Elastic Net	3.8366	25.5036	5.05	0.3516	0.3972	0.4198	8.725
	ada	AdaBoost Regressor	3.9779	25.6071	5.0553	0.3483	0.4245	0.4867	55.947
	lasso	Lasso Regression	3.85	25.7064	5.07	0.3464	0.3991	0.4223	2.785
	llar	Lasso Least Angle Regression	3.85	25.7057	5.07	0.3464	0.3991	0.4223	0.748
	huber	Huber Regressor	3.826	26.2451	5.1229	0.3327	0.396	0.4062	27.058
	omp	Orthogonal Matching Pursuit	3.9208	26.6611	5.163	0.3222	0.4078	0.4322	0.768
	dummy	Dummy Regressor	4.8249	39.3345	6.2715	-0.0001	0.4776	0.5369	0.38
	par	Passive Aggressive Regressor	8.2102	103.7467	9.5691	-1.6382	0.6415	0.903	2.384

Notes: RMSLE is root mean squared logarithmic error, MAPE is mean absolute percentage error, TT (Sec) is model training time, measured in seconds. Extra Trees Regressor (et) exhibits overfitting; therefore, it was not selected in this study.

Table S6. The hyperparameter range for four ML algorithms.

Algorithm	Python Package	Hyperparameter Range
RF	sklearn.ensemble	max_depth:[3,30]
		n_estimators:[5000, 8000]
		max_features:['sqrt', 'log2']
		min_samples_split:[2, 10]
		min_samples_leaf:[2, 10]
		random_state: 2024
XGBoost	xgboost	max_depth:[3,10]
		learning_rate: [0.005, 0.01]
		n_estimators:[2000, 8000]
		min_child_weight:[1, 300]
		gamma:(0.0001,1,log=True)
		reg_alpha: (0.0001,10,log=True)
		reg_lambda: (0.0001,10,log=True)
		colsample_bytree:[0.1, 0.8]
		subsample:[0.6, 0.8]
		random_state: 2024
LightGBM	lightgbm	reg_alpha: [0.001, 10.0]
		reg_lambda: [0.001, 10.0]
		num_leaves: [11, 333]
		min_child_samples: [5, 100]
		max_depth: [3, 20]
		learning_rate: [0.001,0.005,0.01,0.05,0.1]
		colsample_bytree: [0.1, 0.5]
		n_estimators: [7000, 8000]
		cat_smooth: [10, 100]
		cat_l2: [1, 20]
		min_data_per_group: [50, 200]
		cat_feature: [10, 60]
		n_jobs: -1
		random_state: 2024
CatBoost	catboost	depth: [3, 10]
		learning_rate: [0.001,0.005,0.01,0.05,0.1]
		iterations: [5000, 9000]
		max_bin: [200, 400]
		min_data_in_leaf: [1, 30]
		l2_leaf_reg: [0.0001, 1.0, log=True]
		subsample: [0.6, 0.9]
		random_state: 2024

Table S7. The optimal hyperparameter parameter values for different MLAs.

Algorithm	Hyperparameter values (h_a)	Hyperparameter values (h_w)
RF	{'max_depth': 22,	{'max_depth': 16,
	'n_estimators': 100,	'n_estimators': 100,
	'max_features': 'sqrt',	'max_features': 'sqrt',
	'min_samples_split': 10,	'min_samples_split': 10,
	'min_samples_leaf': 1}	'min_samples_leaf': 1}

Algorithm	Hyperparameter values (h_a)	Hyperparameter values (h_w)
	'random_state': 2024}	'random_state': 2024}
XGBoost	{'max_depth': 10,	{'max_depth': 10,
	'learning_rate': 0.01,	'learning_rate': 0.01,
	'n_estimators': 6265,	'n_estimators': 6718,
	'min_child_weight': 15,	'min_child_weight': 4,
	'gamma': 0.002099983206704338,	'gamma': 0.013363217961312412,
	'alpha': 0.1497537570416905,	'alpha': 1.9018751797973954,
	'lambda': 0.002539897498632027,	'lambda': 0.009243058986896945,
	'colsample_bytree': .676096315705994,	'colsample_bytree': .5708992971238507,
	'subsample': 0.6382245679230509,	'subsample': 0.769691304776139,
	'random_state': 2024}	'random_state': 2024}
LGBM	{'reg_alpha': 2.2098573262841947,	{ 'reg_alpha': 1.9287086325195282,
	'reg_lambda': 4.777845816164959,	'reg_lambda': 5.462281985773448,
	'num_leaves': 312,	'num_leaves': 298,
	'min_child_samples': 5,	'min_child_samples': 5,
	'max_depth': 17,	'max_depth': 19,
	'learning_rate': 0.1,	'learning_rate': 0.05,
	'colsample_bytree': .4790174793827581,	'colsample_bytree': .4671945299775325,
	'n_estimators': 7042,	'n_estimators': 7759,
	'cat_smooth': 61,	'cat_smooth': 14,
	'cat_l2': 17,	'cat_l2': 16,
	'min_data_per_group': 117,	'min_data_per_group': 122,
	'cat_feature': 13,	'cat_feature': 20
	'random_state': 2024}	'random_state': 2024}
CatBoost	{'depth': 10,	{'depth': 10,
	'learning_rate': 0.05,	'learning_rate': 0.05,
	'iterations': 7556,	'iterations': 7936,
	'max_bin': 366,	'max_bin': 366,
	'min_data_in_leaf': 14,	'min_data_in_leaf': 12,
	'l2_leaf_reg': 0.004530685052947802,	'l2_leaf_reg': 0.048609449544924604,
	'subsample': 0.6243190198912542,	'subsample': 0.6316663275897422
	'random_state': 2024}	'random_state': 2024}

Table S8. The training results of different MLAs

Model	Algorithm	Train					Test			
		R2	MSE	RMSE	MAE	Times(s)	R2	MSE	RMSE	MAE
h_a	RF	0.9107	3.2902	1.8139	1.3638	126.52	0.8138	6.8901	2.6249	1.9405
	XGBoost	0.8670	4.8885	2.2110	1.6840	78.23	0.8000	7.4147	2.7230	2.0220
	LGBM	0.9860	0.5170	0.7190	0.5320	42.37	0.8210	6.6255	2.5740	1.8580
	CatBoost	0.8800	4.4310	2.1050	1.6110	427.16	0.8010	7.3495	2.7110	2.0120
h_w	RF	0.7696	9.0625	3.0104	2.3070	98.40	0.7169	11.1984	3.3464	2.5307
	XGBoost	0.8770	4.8180	2.1950	1.6780	98.03	0.7750	8.9162	2.9860	2.2060
	LGBM	0.9560	1.7424	1.3200	0.9790	96.71	0.7890	8.3405	2.8880	2.0940
	CatBoost	0.8590	5.5413	2.3540	1.7970	672.64	0.7670	9.2112	3.0350	2.2510

Notes: h_a is arithmetic mean height; h_w is weighted mean height

Table S9. The fitting results of ML-based mixed-effects models

Model	Equation	Npar	AIC	BIC	logLik	Chisq	Df	Pr(>Chisq)	Signif. codes
$\boldsymbol{h_a}$	$h_a \sim fixed\_h_a + (1$ | zone)	4	962124	962165	-481058				
	$h_a \sim fixed\_h_a + (fixed\_h_a$ | zone)	6	962107	962168	-481048	20.594	2	3.370E-05	***
$\boldsymbol{h_w}$	$h_w \sim fixed\_h_w + (1$ | zone)	4	1008669	1008710	-504331				
	$h_w \sim fixed\_h_w + (fixed\_h_a$ | zone)	6	1008650	1008711	-504319	23.173	2	9.293E-06	***

Notes: h_a is arithmetic mean height, corresponding to the response variable in the mixed model structure; h_w is weighted mean height , corresponding to the response variable in the mixed model structure; fixed_h_a represents the predicted values of arithmetic mean height in machine learning, corresponding to the covariate in the mixed model structure; fixed_h_w represents the predicted values of weighted mean height in machine learning, corresponding to the covariate in the mixed model structure; (1 | zone) indicates fitting random intercepts at the level of the grouping variable " zone "; (x | zone) indicates fitting a random slope related to the covariate at each level of the grouping variable " zone "; Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Note S1

The relative uncertainty for each pixel is calculated using Eq (1)

$$\varepsilon_i = \frac{RMSE}{\hat{y}_i} \tag{1}$$

where $RMSE$ is the root mean squared error of validation set, $\hat{y}_i$ represents the predicted value of i^{th} observed value.

The mean of relative uncertainty for all pixels is calculated using Eq (2)

$$\bar{\varepsilon} = \frac{\sum_{i=1}^{N} \frac{RMSE}{\hat{y}_i}}{N} \times 100\% = RMSE \frac{\sum_{i=1}^{N} \frac{1}{\hat{y}_i}}{N} \times 100\% \tag{2}$$

We know that the harmonic mean is less than or equal to the arithmetic mean, which is expressed as Eq (3), then mathematically Eq (2) is expressed as Eq (4):

$$\frac{N}{\sum_{i=1}^{N} \frac{1}{\hat{y}_i}} \leq \frac{\sum_{i=1}^{N} \hat{y}_i}{N} = \bar{y} \tag{3}$$

$$\bar{\varepsilon} = \frac{\sum_{i=1}^{N} \frac{RMSE}{\hat{y}_i}}{N} \times 100\% = RMSE \frac{\sum_{i=1}^{N} \frac{1}{\hat{y}_i}}{N} \times 100\% \leq \frac{RMSE}{\bar{y}} \times 100\% \tag{4}$$

So, the maximum of mean $\varepsilon_{\mathrm{product}}$ is calculated by Eq (5)

$$\bar{\varepsilon}_{max} = \frac{RMSE}{\bar{y}} \times 100\% \tag{5}$$