Supporting Information for

Accurate prediction of arsenic and lead contamination in karst groundwater using explainable ensemble machine learning

Jiayi Deng [a], Min Wu [a], Qiang Tang [c], Shengyan Pu [a,b,*]

a.State Key Laboratory of Geohazard Prevention and Geoenvironment Protection (Chengdu University of Technology), 1#, Dongsanlu, Erxianqiao, Chengdu, 610059, China

b.State Key Laboratory of Environmental Criteria and Risk Assessment, Chinese Research Academy of Environmental Sciences, Beijing 100012, China

c.Scientific Research Academy of Guangxi Environmental Protection, Nanning 530022, China

[*]Corresponding author at: State Key Laboratory of Geohazard Prevention and Geoenvironment Protection (Chengdu University of Technology), Chengdu 610059, P. R. China. Tel./fax: +86(028)-84073864; E-mail addresses: pushengyan@gmail.com; pushengyan13@cdut.edu.cn (S.Y.Pu)

Notes: The authors declare that there is no conflict of interests regarding the publication of this paper.

Supporting Information file includes:

Text S1. Machine Learning Models

1. Decision Tree (DT)

Decision Tree is a non-parametric supervised learning method that can be used for both

classification and regression tasks(Song and Lu, 2015). The model recursively partitions the feature space by selecting optimal splitting criteria at each node, constructing a tree-like structure where each internal node represents a feature test, each branch represents the outcome of the test, and each leaf node represents a prediction value. For regression tasks, Decision Trees minimize the mean squared error at each split by evaluating different feature thresholds. The prediction for a sample is determined by:

$$\hat{y} = \frac{1}{|N_m|} \sum_{i \in N_m} y_i$$

where N_m represents the set of samples in leaf node m, and y_i is the actual value of the i-th sample. Common splitting criteria include Gini impurity for classification and variance reduction for regression. While Decision Trees are interpretable and easy to visualize, they are prone to overfitting, especially with deep trees.

2. Random Forest (RF)

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mean prediction for regression or majority vote for classification(Salman et al., 2024). It employs two key randomization techniques: Bootstrap aggregating (bagging) for sampling training data and random feature selection at each split. This dual randomness reduces correlation among trees and enhances model generalization. The final prediction is computed as:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(x)$$

where $h_t(x)$ represents the prediction of the t-th tree, and T is the total number of trees. Random Forest effectively handles high-dimensional data, requires minimal hyperparameter tuning, and provides feature importance rankings. The out-of-bag (OOB) error serves as an unbiased estimate of the models performance without requiring a separate validation set.

3. Extreme Gradient Boosting (XGBoost)

XGBoost is an optimized implementation of gradient boosting that builds an ensemble of weak learners sequentially, with each new tree correcting the errors of previous ones(Fatima et al., 2023). It incorporates regularization terms in the objective function to control model complexity and prevent overfitting. The objective function at iteration t is:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t)$$

where l is the loss function, $\hat{y}_i^{(t-1)}$ is the prediction from previous iterations, f_t is the new tree being added, and $\Omega(f_t) = \gamma T + \frac{1}{2}\lambda||\omega||^2$ is the regularization term with T being the number of leaves and ω the leaf weights. XGBoost also features parallel processing, tree pruning, and built-in handling of missing values, making it highly efficient and accurate for various tasks.

4. Light Gradient Boosting Machine (LightGBM)

LightGBM is a fast, distributed, high-performance gradient boosting framework that uses histogram-based algorithms and tree-growing strategies(Ke et al., 2017; Shi et al., 2022). Unlike traditional level-wise tree growth, LightGBM employs a leaf-wise strategy that splits the leaf with the maximum delta loss, leading to better accuracy with fewer iterations. It uses Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to improve efficiency. The split gain is calculated as:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] - \gamma$$

where G_L and G_R are the sum of gradients for left and right child nodes, H_L and H_R are the sum of hessians, λ is the L2 regularization parameter, and γ is the minimum loss reduction. LightGBM excels in handling large-scale datasets with high efficiency and lower memory consumption.

5. Support Vector Machine (SVM)

Support Vector Machine is a powerful supervised learning algorithm that finds the optimal

hyperplane to maximize the margin between different classes in classification or fit data in regression(Montesinos L ó pez et al., 2022). For regression tasks (SVR), the model seeks to find a function that deviates from the actual values by at most ϵ while being as flat as possible. The optimization problem is formulated as:

$$\min_{\omega,b} \frac{1}{2}||\omega||^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*)$$

subject to $y_i - (\omega^T x_i + b) \leq \epsilon + \xi_i$ and $(\omega^T x_i + b) - y_i \leq \epsilon + \xi_i^*$, where ω is the weight vector, b is the bias, C is the regularization parameter, and ξ_i, ξ_i^* are slack variables. SVM can handle non-linear relationships through kernel functions (e.g., RBF, polynomial), mapping input features to higher-dimensional spaces where linear separation becomes possible.

6. Gradient Boosting Decision Tree (GBDT)

Gradient Boosting Decision Tree is an ensemble method that sequentially builds decision trees, where each new tree is trained to predict the residuals (errors) of the previous ensemble(Rao et al., 2019). Unlike bagging methods, GBDT uses gradient descent to minimize the loss function by adding trees that follow the negative gradient direction. The model update at iteration m is:

$$F_m(x) = F_{m-1}(x) + \nu \cdot h_m(x)$$

where $F_{m-1}(x)$ is the ensemble prediction from previous iterations, $h_m(x)$ is the new tree fitted to the negative gradient $-\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$, and ν is the learning rate controlling the contribution of each tree. GBDT typically uses shallow trees to maintain low variance while reducing bias through boosting. The method is highly effective for structured data and provides excellent predictive performance with proper regularization.

7. Artificial Neural Network (ANN)

Artificial Neural Network is a computational model inspired by biological neural networks, consisting of interconnected layers of neurons that can learn complex non-linear relationships(Liu et al., 2022; Shahmansouri et al., 2021). A typical feedforward ANN contains an input layer, one or

more hidden layers, and an output layer. Each neuron computes a weighted sum of its inputs and applies an activation function. For a neuron j in layer l, the output is:

$$a_j^{(l)} = f\left(\sum_i w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$$

where $w_{ji}^{(l)}$ represents the weight connecting neuron i in layer $l-1$ to neuron j in layer l, $b_j^{(l)}$ is the bias term, and f is the activation function (e.g., ReLU, sigmoid, tanh). The network is trained using backpropagation algorithm, which computes gradients of the loss function with respect to weights and updates them through optimization methods like stochastic gradient descent (SGD) or Adam. ANNs excel at capturing intricate patterns in high-dimensional data and are widely used in various domains including image recognition, natural language processing, and time series forecasting.

Table S1 Multicollinearity Diagnostics of the Predictors (VIF)

Variable	VIF	R^2
P	1.376598716	0.273571892
AT	1.011102589	0.010980675
PD	1.071979425	0.067146275
DEM	1.572719152	0.364158566
K	3.924522212	0.745191912
GWD	1.105022415	0.095040982
EDI	2.644303989	0.621828654
KNDI	2.58583099	0.613277123
C_i	4.52344902	0.778929751
SPT	1.32361662	0.244494225
CEC	1.37449399	0.272459533
LULC	1.025517421	0.024882484

Table S2 Predictor variables for machine learning models

Category	Variable	Description	Data source
Climate conditions	P	Annual precipitation(2024)	Data Center for Resources and Environmental Sciences, Chinese Academy of Sciences (RESDC) (https://www.resdc.cn)
	AT	Mean annual temperature (℃)	
Human activities	PD	Population density(2023)	
	LULC	Land use and land cover(2023), including: Arable land, forest land, grassland, water bodies, urban and industrial/mining land, unutilized land	
Topography	DEM	30m resolution digital elevation model (DEM) (m)	National Earth System Science Data Center(https://www.geodata.cn)
Hydrogeological conditions	K	Hydraulic conductivity	National Geological Archives(https://www.ngac.cn)
	GWD	Groundwater depth (m)	
	EDI	Epikarst development intensity	
	KNDI	Karst network development intensity	
	C_i	Rainfall infiltration coefficient	
Soil Physical and Chemical Properties	SPT	Soil profile thickness	China soil science database(http://vdb3.soil.csdb.cn/)
	CEC	Soil cation exchange capacity	National Tibetan Plateau Data CenterThird Pole Environment Data Center(https://data.tpdc.ac.cn/)

Table S3 Hyperparameter tuning details of different models for arsenic (As) prediction on the training set

Model	Parameters	Range	Optimum value
DT	criterion	['squared_error', 'friedman_mse']	friedman_mse
	max_depth	[None, 10, 20, 30, 40]	10
	min_samples_split	[2, 5, 10]	10
	min_samples_leaf	[1, 2, 4]	1
	max_features	[None, 'sqrt', 'log2']	None
	n_estimators	[50, 100, 200]	200
RF	criterion	['squared_error', 'friedman_mse']	friedman_mse
	max_depth	[None, 10, 20, 30, 40]	20
	min_samples_split	[2, 5, 10]	10
	min_samples_leaf	[1, 2, 4]	1
	max_features	[None, 'sqrt', 'log2']	None
	n_estimators	[50, 100, 200]	100
XGB	learning_rate	[0.01, 0.2]	0.2
	max_depth	[3, 5]	3
	min_child_weight	[1, 5]	1
	subsample	[0.8, 1.0]	1.0
	colsample_bytree	[0.8, 1.0]	0.8
	gamma	[0, 0.2]	0
LGBM	n_estimators	[50, 200]	200
	learning_rate	[0.01, 0.2]	0.2
	max_depth	[3, 7]	3
	num_leaves	[31, 100]	31
SVM	C	[0.1, 1, 10]	1
	gamma	[0, 0.2]	0
GBDT	n_estimators	[50, 100, 200]	200
	learning_rate	[0.01, 0.1, 0.2]	0.1
	max_depth	[3, 5, 7]	3
	subsample	[0.8, 1.0]	1.0
	min_samples_split	[2, 5, 10]	10
	min_samples_leaf	[1, 2, 4]	1
ANN	hidden_layer_sizes	[(50,), (100,), (100, 50)]	100
	learning_rate_init	[0.001, 0.01, 0.1]	0.001
	alpha	[0.0001, 0.001, 0.01]	0.001

Table S4 Hyperparameter tuning details of different models for lead (Pb) prediction on the training set

Model	Parameters	Range	Optimum value
DT	criterion	['squared_error', 'friedman_mse']	squared_error
	max_depth	[None, 10, 20, 30, 40]	10
	min_samples_split	[2, 5, 10]	10
	min_samples_leaf	[1, 2, 4]	2
	max_features	[None, 'sqrt', 'log2']	None
	n_estimators	[50, 100, 200]	200
RF	criterion	['squared_error', 'friedman_mse']	friedman_mse
	max_depth	[None, 10, 20, 30, 40]	10
	min_samples_split	[2, 5, 10]	5
	min_samples_leaf	[1, 2, 4]	1
	max_features	[None, 'sqrt', 'log2']	None
	n_estimators	[50, 100, 200]	100
XGB	learning_rate	[0.01, 0.2]	0.2
	max_depth	[3, 5]	3
	min_child_weight	[1, 5]	1
	subsample	[0.8, 1.0]	1.0
	colsample_bytree	[0.8, 1.0]	0.8
	gamma	[0, 0.2]	0.2
LGBM	n_estimators	[50, 200]	50
	learning_rate	[0.01, 0.2]	0.2
	max_depth	[3, 7]	3
	num_leaves	[31, 100]	31
SVM	C	[0.1, 1, 10]	1
	gamma	[0, 0.2]	0
GBDT	n_estimators	[50, 100, 200]	100
	learning_rate	[0.01, 0.1, 0.2]	0.1
	max_depth	[3, 5, 7]	3
	subsample	[0.8, 1.0]	1.0
	min_samples_split	[2, 5, 10]	2
	min_samples_leaf	[1, 2, 4]	1
ANN	hidden_layer_sizes	[(50,), (100,), (100, 50)]	100
	learning_rate_init	[0.001, 0.01, 0.1]	0.01
	alpha	[0.0001, 0.001, 0.01]	0.001

Table S5 Performance evaluation table for arsenic (As) prediction models

Mode	MSE		RMSE		MAE		R^2		Pearson	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
TRE	0.0270	0.0425	0.1644	0.2062	0.1315	0.1650	0.9324	0.8937	0.9656	0.9454
DT	0.0386	0.0918	0.1966	0.3030	0.1573	0.2424	0.9034	0.7704	0.9505	0.8777
RF	0.0908	0.0731	0.3013	0.2704	0.1918	0.2163	0.9080	0.8172	0.9529	0.9040
XGB	0.0368	0.0662	0.1918	0.2574	0.1534	0.2059	0.9207	0.8344	0.9595	0.9135
LGBM	0.0284	0.0694	0.1686	0.2635	0.1349	0.2108	0.9289	0.8264	0.9638	0.9091
SVM	0.0362	0.0630	0.1903	0.2511	0.1522	0.2009	0.9094	0.8424	0.9537	0.9178
GBDT	0.0389	0.0706	0.1973	0.2657	0.1578	0.2126	0.9027	0.8234	0.9501	0.9074
ANN	0.0481	0.0681	0.2194	0.2610	0.1755	0.2088	0.8797	0.8297	0.9379	0.9109

Table S6 Performance evaluation table for lead (Pb) prediction models

Mode	MSE		RMSE		MAE		R^2		Pearson	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
TRE	0.1496	0.1797	0.3868	0.4239	0.3094	0.3391	0.9065	0.8877	0.9521	0.9422
DT	0.2203	0.4934	0.4694	0.7024	0.3698	0.5619	0.8623	0.6916	0.9286	0.8316
RF	0.1666	0.3408	0.4081	0.5838	0.3265	0.4670	0.8959	0.7870	0.9465	0.8872
XGB	0.2219	0.3363	04711	0.5800	0.3769	0.4640	0.8613	0.7898	0.9281	0.8911
LGBM	0.2589	0.3294	0.5088	0.5740	0.4070	0.4592	0.8382	0.7941	0.9155	0.8911
SVM	0.2453	0.3272	0.4953	0.5720	0.3962	0.4576	0.8467	0.7955	0.9202	0.8919
GBDT	0.2379	0.3314	0.4878	0.5756	0.3902	0.4605	0.8513	0.7929	0.9227	0.8904
ANN	0.2304	0.3182	0.4800	0.5641	0.3840	0.4513	0.8560	0.8011	0.9252	0.8950

Table S7 The entropy weight method is used to calculate the index weights

Index weight	Groundwater As prediction		Groundwater Pb prediction	
	Coefficient of difference	Weight (%)	Coefficient of difference	Weight (%)
MSE	0.0276	45.38	0.0418	50.86
RMSE	0.0119	19.58	0.0147	17.88
MAE	0.0122	20.07	0.0150	18.24
R^2	0.0058	9.53	0.0069	8.37
Pearson	0.0062	10.24	0.0072	8.70

Table S8 TOPSIS calculates the closeness and weight of the model

Model weight	Groundwater As prediction		Groundwater Pb prediction	
	Relative proximity	Weight (%)	Relative proximity	Weight (%)
DT	0.0411	0.22	0.0290	0.58
RF	0.6605	14.90	0.8709	17.48
XGB	0.8487	19.14	0.8910	17.91
LGBM	0.7786	17.62	0.9365	18.85
SVM	1.0000	22.53	0.9453	19.03
GBDT	0.7474	16.97	0.9268	18.66
ANN	0.8247	18.62	1.0000	20.17

Table S9 Bootstrap 95% confidence intervals of R^2 for the TRE and best single models

Model	**As R^2 (95% CI)**	**Pb R^2 (95% CI)**
Best single model	0.8424 (0.8187-0.8649)	0.8011 (0.7726-0.8281)

TRE	0.8937 (0.8761-0.9098)	0.8877 (0.8654-0.9072)

Fatima, S., Hussain, A., Amir, S.B., Ahmed, S.H. and Aslam, S.M.H. 2023. Xgboost and random forest algorithms: an in depth analysis. Pakistan Journal of Scientific Research (PJOSR) 3(1), 26-31.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems 30.

Liu, Y.R., Wang, L., Gu, K.X. and Li, M. 2022. Artificial Neural Network (ANN)- Bayesian Probability Framework (BPF) based method of dynamic force reconstruction under multi-source uncertainties. KNOWLEDGE-BASED SYSTEMS 237.

Montesinos López, O.A., Montesinos López, A. and Crossa, J. (2022) Multivariate statistical machine learning methods for genomic prediction, pp. 337-378, Springer.

Rao, H., Shi, X.Z., Rodrigue, A.K., Feng, J.J., Xia, Y.C., Elhoseny, M., Yuan, X.H. and Gu, L.C. 2019. Feature selection based on artificial bee colony and gradient boosting decision tree. APPLIED SOFT COMPUTING 74, 634-642.

Salman, H.A., Kalakech, A. and Steiti, A. 2024. Random forest algorithm overview. Babylonian Journal of Machine Learning 2024, 69-79.

Shahmansouri, A.A., Yazdani, M., Ghanbari, S., Bengar, H.A., Jafari, A. and Ghatte, H.F. 2021. Artificial neural network model to predict the compressive strength of eco-friendly geopolymer concrete incorporating silica fume and natural zeolite. JOURNAL OF CLEANER PRODUCTION 279.

Shi, Y., Ke, G., Chen, Z., Zheng, S. and Liu, T.-Y. 2022. Quantized training of gradient boosting decision trees. Advances in neural information processing systems 35, 18822-18833.

Song, Y.-Y. and Lu, Y. 2015. Decision tree methods: applications for classification and prediction. Shanghai archives of psychiatry.

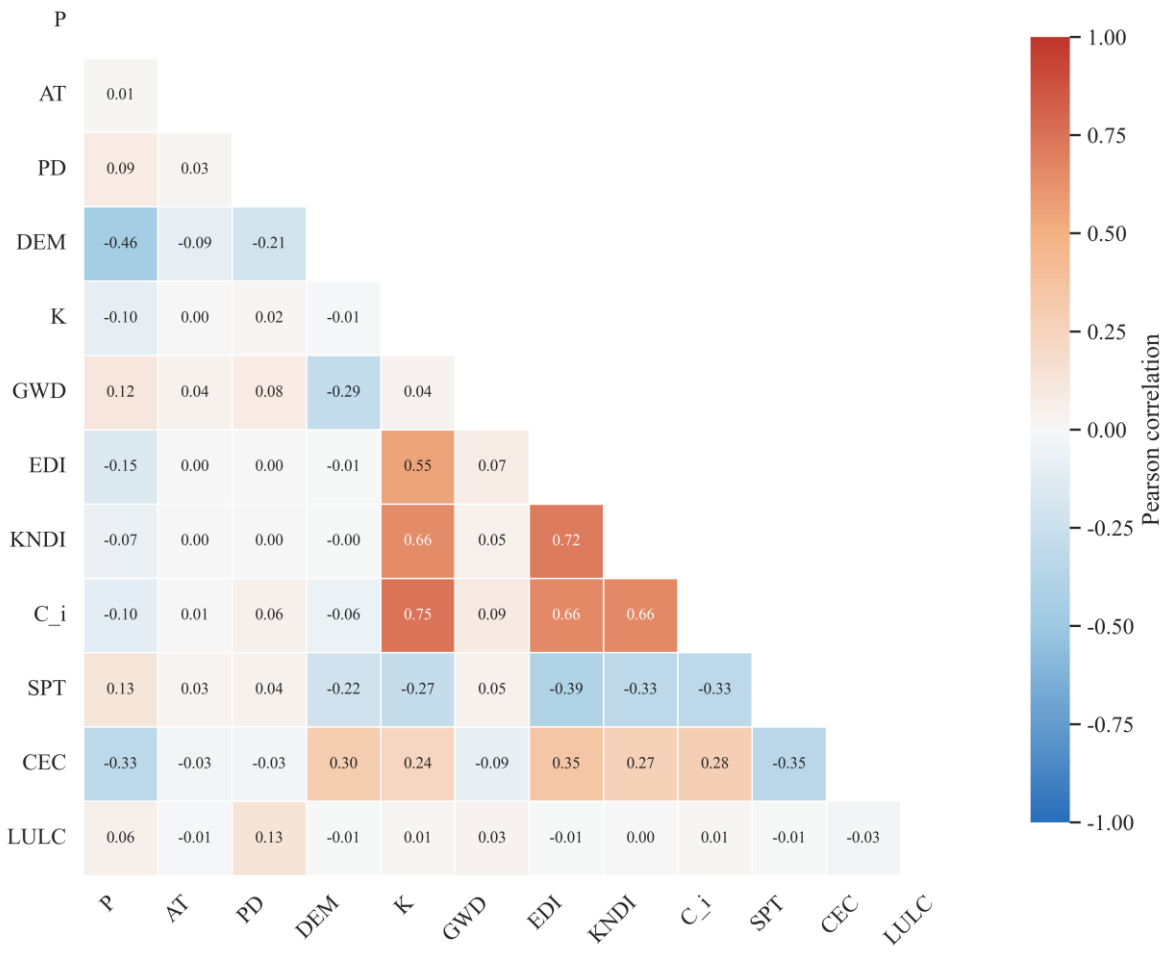

Figure S1 Pearson correlation matrix of the predictor variables used in the machine learning models

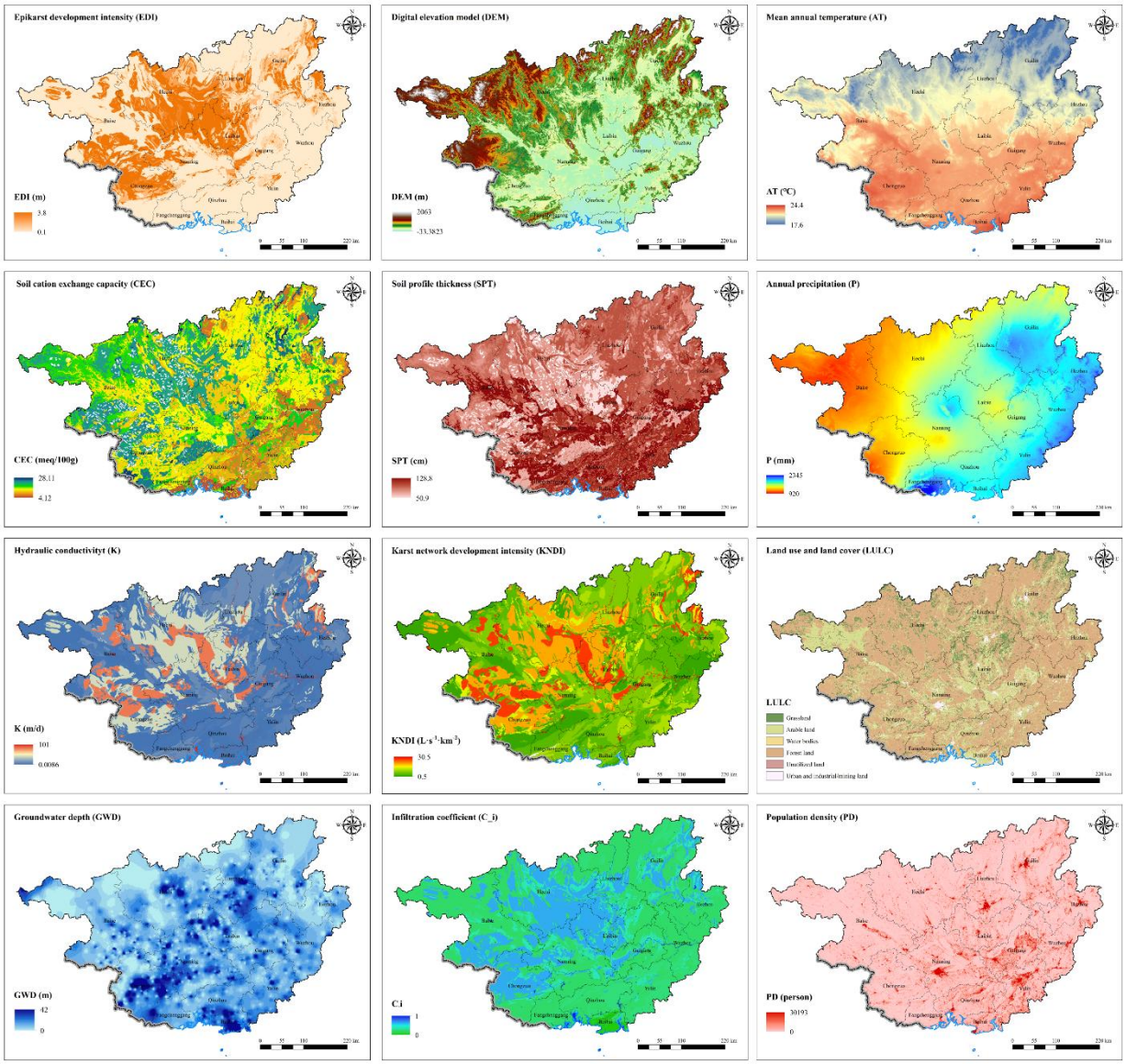

Figure S2　Spatial data distribution of predictor variables in the Guangxi.
Map Review Number GS 京（2026) 2069 号.

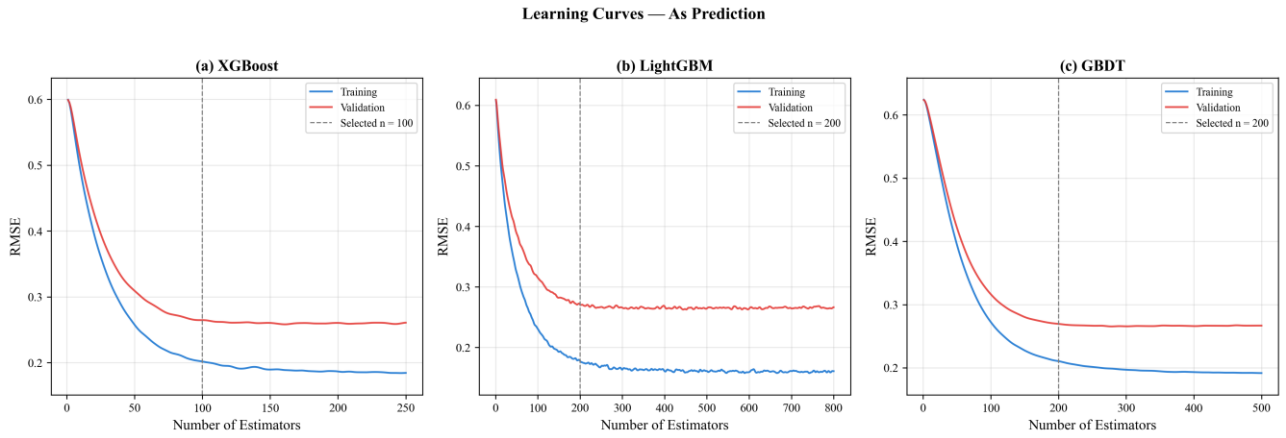

Figure S3. Learning curves of boosting tree models showing training and validation RMSE as a function of the number of estimators: (a) XGBoost, (b) LightGBM, and (c) GBDT in As prediction

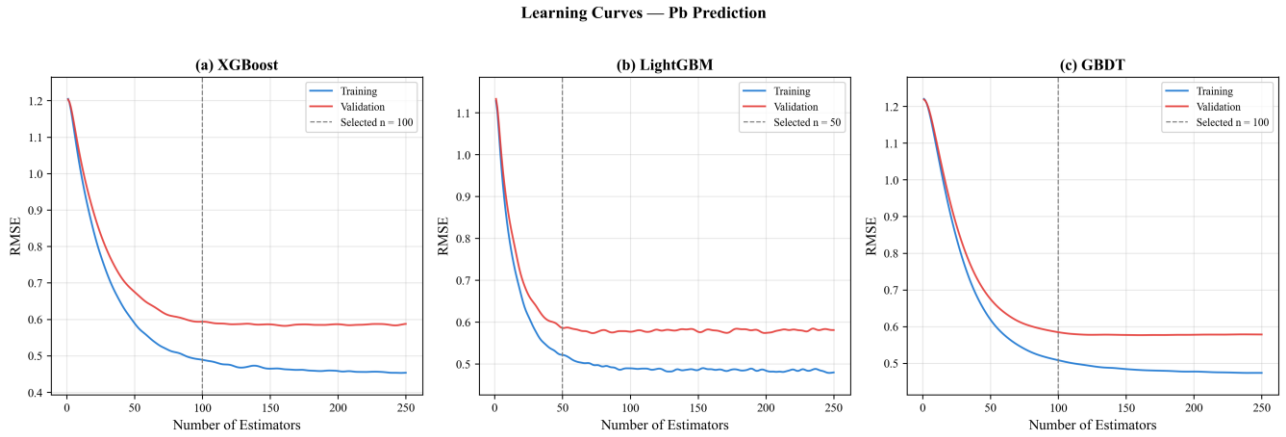

Figure S4. Learning curves of boosting tree models showing training and validation RMSE as a function of the number of estimators: (a) XGBoost, (b) LightGBM, and (c) GBDT in Pb prediction