

Machine Learning-Driven Prediction of Organic Compound Adsorption onto Microplastics in Freshwater

Ya Wang *, Peng Zhao, Honghong Yi and Xiaolong Tang *

School of Energy and Environmental Engineering, University of Science and Technology Beijing, Beijing 10083, China; m202310194@xs.ustb.edu.cn (P.Z.); yihonghong@ustb.edu.cn (H.Y.)

* Correspondence: yaanne@126.com or wyanne@ustb.edu.cn (Y.W.); txiaolong@126.com (X.T.)

Contents

Table S1. Categories of organic compounds used in PE predictive model	4
Table S2. Hyperparameter optimization for RF, GBDT, XGBoost, CatBoost, LightGBM and SVM	5
Table S3. Outliers and their standardized residuals (δ^*) and leverage values (h)	7
Figure S1. Application domains with Euclidean distance-based approach for linear models of (a) CPE, (b) PBS, (c) PCL and (d) LDPE MPs.....	8
Figure S2. Application domains with Euclidean distance-based approach for six nonlinear models with different algorithms for PE MPs: (a) RF, (b) GBDT, (c) XGBoost, (d) CatBoost, (e) LightGBM and (f) SVM.....	9

Q^2_v and Q^2 can be calculated according to the following equations.

$$Q^2_v = 1 - \frac{\sum_{i=1}^{n_v} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n_v} (y_i - \bar{y}_v)^2} \quad (1)$$

where y_i and $\hat{y}_i$ are the experimental value and predicted value for the i th compound, respectively, and $\bar{y}_v$ is the average value.

$$Q^2 = \frac{1}{K} \sum_{k=1}^K R^2_{(k)}, K = 5 \quad (2)$$

where $K = 5$; $R^2_{(K)}$ is the coefficient of determination.

Table S1. Categories of organic compounds used in PE predictive model.

Number of compounds	Category
1	Alcohols
4	Alkanes
16	Aromatic compounds
5	Aromatic halogen compounds
33	Biphenyls and chlorinated biphenyls
1	Carboxylic acids
4	Cycloalkanes
1	Esters
4	Ethers
2	Ketones
9	Nitrogen heterocycles
1	Oxygen heterocycles
2	Phenols
3	Phthalates
19	Polycyclic aromatic hydrocarbons

Table S2. Hyperparameter optimization for RF, GBDT, XGBoost, CatBoost, LightGBM and SVM.

Algorithm	Optimization Strategy	Random Seeds	Stopping Criteria	Search Ranges / Fixed Values	Software/Library Versions
RandomForestRegressor	Fixed hyperparameters + 5-fold CV evaluation	random_state=42	Training stops after fixed iterations	n_estimators=300; max_depth=6; min_samples_split=10; min_samples_leaf=5; max_features='sqrt'	scikit learn v1.3
GBDT	Fixed hyperparameters + 5-fold CV evaluation	random_state=42	Training stops after fixed iterations	n_estimators=200; learning_rate=0.05; max_depth=3; min_samples_split=15; min_samples_leaf=10; max_features=0.5; subsample=0.8; ccp_alpha=0.001	scikit learn v1.3
XGBoost	Grid Search with 5-fold CV	n_estimators=[140,150,160]; learning_rate=[0.10,0.12,0.14]; max_depth=[4,5]; gamma=[0.20,0.25,0.30]; reg_alpha=[1.2,1.5,1.8]; reg_lambda=[2.5,3.0,3.5]; colsample_bytree=[0.45,0.5,0.55]; subsample=[0.75,0.8,0.85]; min_child_weight=[2,3,4]	Best R ² from CV; refit best model	n_estimators=[140,150,160]; learning_rate=[0.10,0.12,0.14]; max_depth=[4,5]; gamma=[0.20,0.25,0.30]; reg_alpha=[1.2,1.5,1.8]; reg_lambda=[2.5,3.0,3.5]; colsample_bytree=[0.45,0.5,0.55]; subsample=[0.75,0.8,0.85]; min_child_weight=[2,3,4]	XGBoost v2.0, scikit-learn v1.3
CatBoost	Grid Search with 5-fold CV	random_state=42	Early stopping not explicitly set; grid search selects best R ²	iterations: [500, 1000]; learning_rate: [0.01, 0.1]; depth: [4, 6, 8]; l2_leaf_reg: [1, 3, 5]	CatBoost v1.2 (Python API)
LightGBM	Fixed hyperparameters + 5-fold CV evaluation	random_state=42	Training stops after fixed iterations	n_estimators=150; learning_rate=0.03; max_depth=4; min_child_samples=20;	LightGBM v4.0, scikit learn v1.3

				min_data_in_leaf=15; feature_fraction=0.6; bagging_fraction=0.7; bagging_freq=1; reg_alpha=0.01; reg_lambda=0.01	
SVM	Grid Search with 5-fold CV	random_state=42	Best R^2 from CV; refit best model	Linear kernel: C=[0.01,0.05,0.1,0.5], epsilon=[0.2,0.3,0.4,0.5]; RBF kernel: C=[0.05,0.1,0.5,1], gamma=[0.001,0.005,0.01,'scale'], epsilon=[0.3,0.4,0.5,0.6]	scikit-learn v1.3

Table S3. Outliers and their standardized residuals (δ^*) and leverage values (h).

Compound	δ^*	h	Model
2,2',3,3',4,4',6-heptachlorobiphenyl	-3.18	0.08	GBDT
2,2',3,3',4,4',6-heptachlorobiphenyl	-3.55	0.07	XGBoost
3,3',4,4',5,5'-hexachlorobiphenyl	3.25	0.05	XGBoost
2,2',3,3',4,4',6-heptachlorobiphenyl	-3.21	0.07	LightGBM
2,3,4,5,6,2',3',4',5'-nonachlorobiphenyl	-3.02	0.18	LightGBM

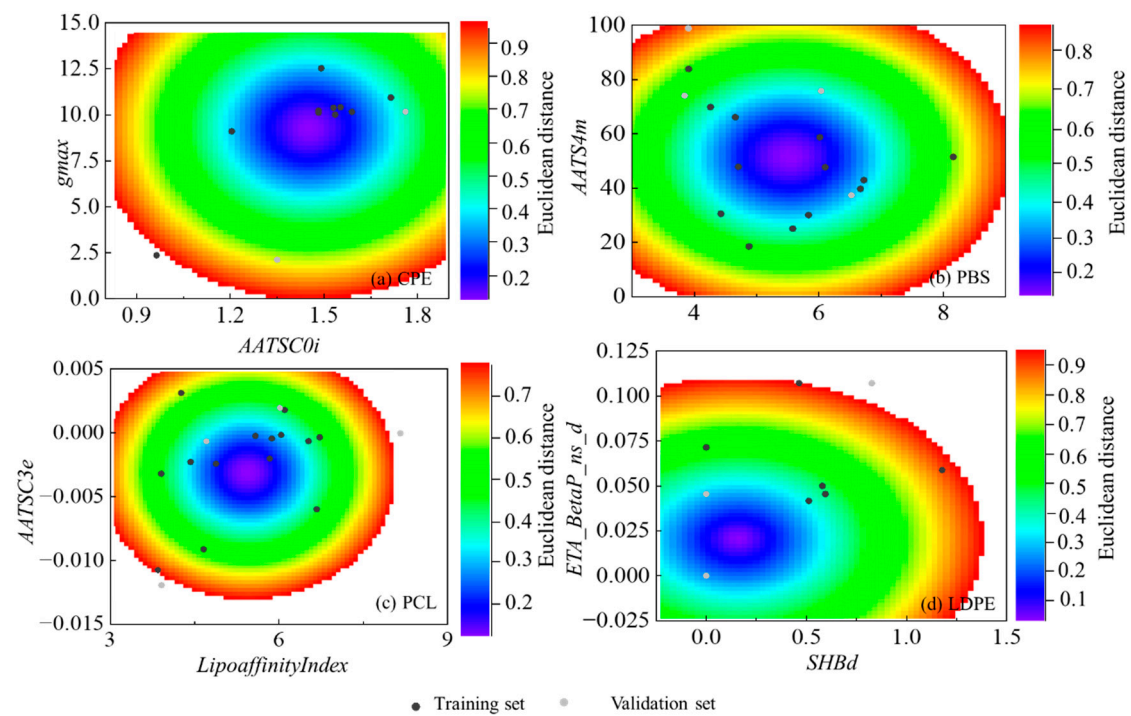


Figure S1. Application domains with Euclidean distance-based approach for linear models of (a) CPE, (b) PBS, (c) PCL and (d) LDPE MPs.

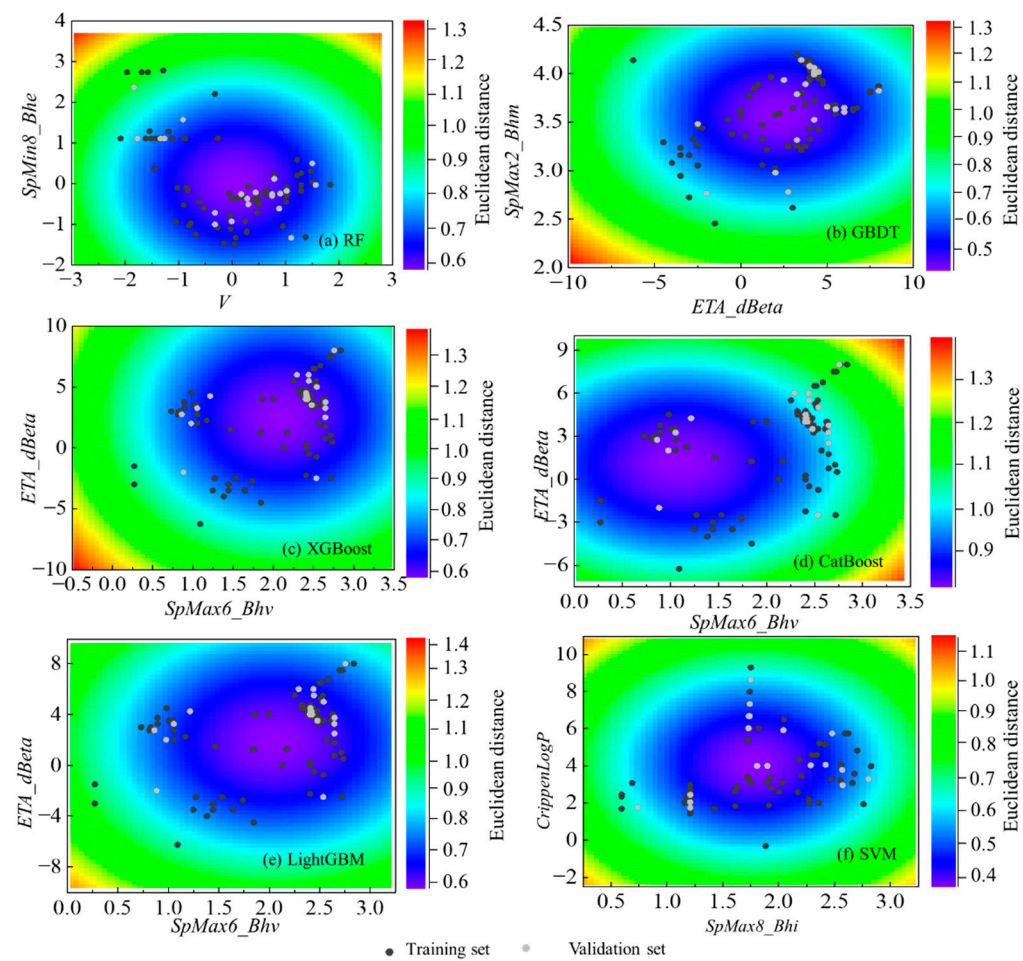


Figure S2. Application domains with Euclidean distance-based approach for six nonlinear models with different algorithms for PE MPs: (a) RF, (b) GBDT, (c) XGBoost, (d) CatBoost, (e) LightGBM and (f) SVM.

Pseudocode

Step 1: Load dataset

- Read data from CSV file.
- Extract chemical feature columns (S, B, A, V, SpMax6_Bhv, ..., CrippenLogP).
- Extract target variable logkd.

Step 2: Feature Selection (RFECV)

- Use XGBoost (n_estimators=100, max_depth=3, learning_rate=0.1) as the base model.
- Perform Recursive Feature Elimination (RFE) with 5-fold cross-validation.
- Retain the optimal subset of features (at least five features).

Step 3: Data Split and Scaling

- Split dataset into training set (80%) and test set (20%).
- Standardize selected features using StandardScaler.

Step 4: Hyperparameter Optimization

- Define parameter grid:
 - n_estimators $\in \{140, 150, 160\}$;
 - learning_rate $\in \{0.10, 0.12, 0.14\}$;
 - max_depth $\in \{4, 5\}$;
 - gamma $\in \{0.20, 0.25, 0.30\}$;
 - reg_alpha $\in \{1.2, 1.5, 1.8\}$;
 - reg_lambda $\in \{2.5, 3.0, 3.5\}$;
 - colsample_bytree $\in \{0.45, 0.5, 0.55\}$;
 - subsample $\in \{0.75, 0.8, 0.85\}$;
 - min_child_weight $\in \{2, 3, 4\}$.
- Apply GridSearchCV with 5-fold cross-validation.

- Select the best parameter combination (based on R^2 score).

Step 5: Train Final Model

- Fit XGBoost model on the training set using the best parameters.
- Save trained model to file (trained_xgboost_model.pkl).

Step 6: Model Evaluation

- Predict logkd on training and test sets.
- Compute performance metrics:
 R^2 , RMSE, MAE.
- Save performance report.

Step 7: Feature Importance and SHAP Analysis

- Extract feature importance from XGBoost.
- Use SHAP TreeExplainer to compute SHAP values.
- Generate bar plot and violin plot of feature importance.
- Save plots and CSV file.

Step 8: Outlier Detection

- Apply Isolation Forest (n_estimators=200, contamination=0.105).
- Detect outliers (samples outside the applicability domain).
- Save outlier report (outliers_report.csv).

End Algorithm