Communication

Catalytic Activity Prediction of α-Diimino Nickel Precatalysts toward Ethylene Polymerization by Machine Learning

Zaheer Abbas [1,2], Md Mostakim Meraz [1,2], Wenhong Yang [3,*], Weisheng Yang [3] and Wen-Hua Sun [1,2,*]

[1] Key Laboratory of Engineering Plastics and Beijing National Laboratory for Molecular Science, Institute of Chemistry, Chinese Academy of Sciences, Beijing 100190, China
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] PetroChina Petrochemical Research Institute, Beijing 102206, China
* Correspondence: whyang@iccas.ac.cn (W.Y.); whsun@iccas.ac.cn (W.-H.S.)

Computational Details

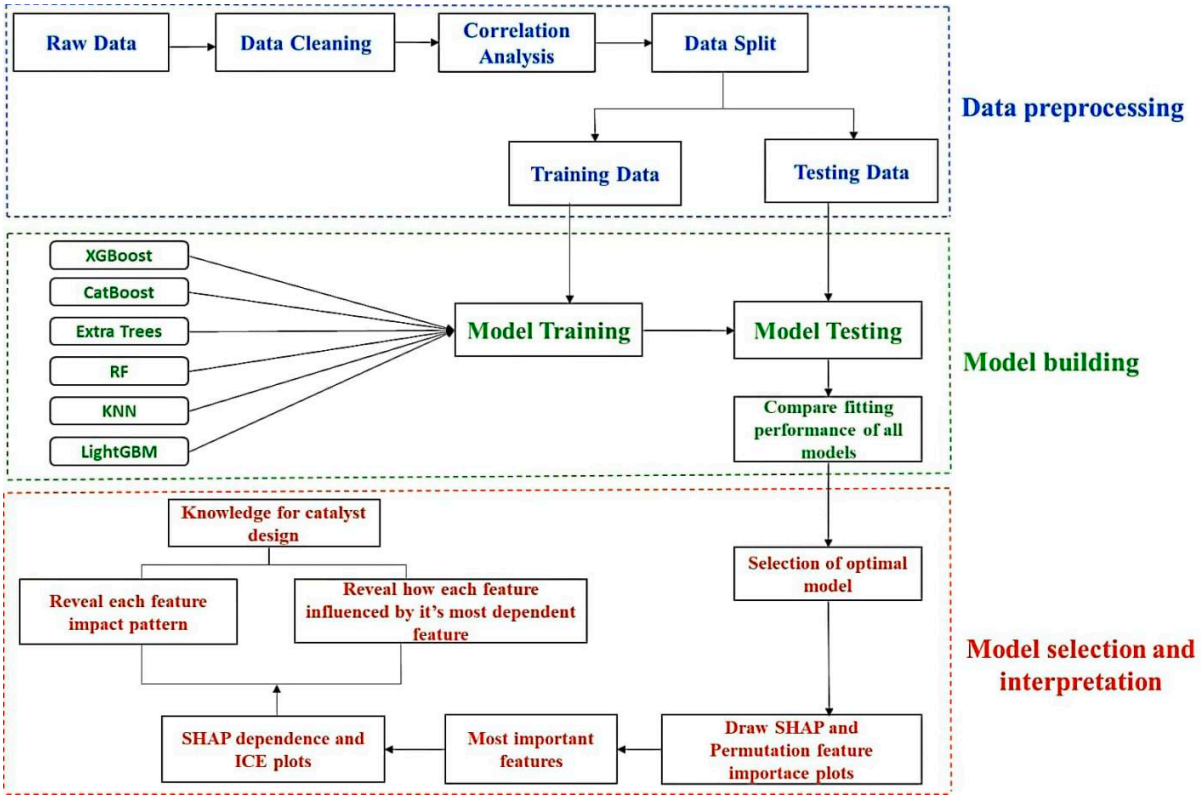

Figure S1. The methodology implemented for predicting and interpreting catalytic activities.

Descriptor calculation and selection

Table S1. The selection of descriptors and the corresponding values of R^2, Q^2, *RMSE*, and *MAE*.

No.	Train R^2	Test R^2	Train *MAE*	Test *MAE*	Train *RMSE*	Test *RMSE*	Q^2 (n_split=4)
124	0.999	0.846	0.055	0.874	0.068	1.198	0.558
108	0.999	0.818	0.057	0.933	0.068	1.303	0.520
35	0.999	0.817	0.086	0.968	0.108	1.305	0.531
25	0.999	0.921	0.094	0.697	0.120	0.859	0.561
20	0.999	0.730	0.101	1.225	0.128	1.589	0.484

Overview of ML Models

XGBoost

XGBoost is an open-source algorithm that provides a fast and scalable way to train gradient-boosted tree models. XGBoost uses a gradient descent algorithm to minimize prediction errors and generates a model composed of a series of weak predictive models, typically in the form of decision trees. It employs an iterative approach, sequentially fitting decision trees to the residuals (errors) of the preceding model. This progressive refinement, where each tree learns from the shortcomings of its predecessor, leads to a cumulative improvement in prediction accuracy. The iterative generation of a robust learner can be expressed by Equation S1:

$$f(x) = \sum_{j=1}^{N} f_j(x) = \hat{y} \quad \text{(S1)}$$

where *f(x)* represents the final predictive model, f_j represents the weak learner after j^{th} iterations, *N* is the total number of weak learners, and $\hat{y}$ is the predicted outcome.

XGBoost provides the user with the flexibility to tailor the loss function and manage tree complexity through the incorporation of a regularization term into the objective function. XGBoost also uses two more techniques—shrinkage and column subsampling—to improve its performance. Shrinkage helps to reduce overfitting, while column subsampling helps to speed up the training process.

CatBoost

CatBoost a gradient-boosting library that employs a consistent set of functions for building left and right splits in decision trees at each tree level. Similar to XGBoost, it builds multiple binary decision trees in each iteration to reduce the error. Moreover, CatBoost is particularly known for its efficiency in handling categorical data; it is also quite effective at handling regression problems. It employs a method known as ordered boosting to automatically identify and leverage the hierarchical arrangement of categorical variables, which enables the algorithm to grasp intricate data relationships and generate precise predictions.

LightGBM

Another gradient-boosting framework, LightGBM, employs two novel techniques to improve performance: gradient-based one-side sampling (GOSS) and exclusive feature bundling (EFB). GOSS downsamples the training data by focusing on instances with larger gradients, while EFB bundles mutually exclusive features together. These techniques can significantly speed up training and improve accuracy. In addition, LightGBM uses leaf-wise growth, while XGBoost uses level-wise growth. Leaf-wise growth is more efficient and can reduce overfitting.

Extra Trees (ETRs)

ETRs regression is an ML algorithm similar to RF. However, it diverges by generating a larger number of decision trees and making random selections of features and split points at each, which helps to improve the accuracy of the model by reducing overfitting. ETRs and RF systems exhibit two notable differences. First, ETRs utilize random points for partition nodes by selecting cutting points. Moreover, they mitigate bias by nurturing trees using the complete learning sample. The Extra Trees algorithm is less affected by noisy or irrelevant features than other machine learning algorithms, which can improve its performance.

Random Forest (RF)

RF is a machine-learning algorithm that uses multiple decision trees to make predictions. Using bootstrap sampling, a random subset of the training data is created, and each base tree model is trained on this subset. This randomization of the subset aids in mitigating the potential for overfitting and enhances the algorithm's resilience. Then, all linear nodes in the tree are pruned.

This process is repeated for each base tree model, and the final prediction is made by averaging the predictions of all the trees. This algorithm can handle noisy and incomplete data, offers ease of tuning, and supports parallel computation.

k-Nearest Neighbor (KNN)

KNN is a simple and effective ML algorithm that can be used for both classification and regression tasks. It works by finding the k most similar data points to a target data point and then using the average or weighted average of the target values of those k neighbors to predict the target value of the new data point. KNN is a good basic algorithm to try first because it is easy to understand and implement, and it can often perform well with a small amount of data. However, it is important to note that KNN can be sensitive to the choice of the k value, and it can be computationally expensive to train and predict on large datasets. KNN can perform poorly on high-dimensional data because it becomes more difficult to calculate the distance between data points as the number of dimensions increases.

Results and Discussion

Table S2. Comparison of the bond lengths and bond angles between the calculated geometry and experimental data for complex **1,** along with the standard deviation (**δ**).

Complex 01	Experiment	Calculated
Bond lengths (Å)		
Ni(1)-Br(1)	2.3363	2.336
Ni(1)-Br(2)	2.341	2.341
Ni(1)-N(1)	2.063	2.06
Ni(1)-N(2)	2.038	2.035
N(1)-C(1)	1.289	1.291
N(1)-C(13)	1.439	1.439
N(2)-C(11)	1.28	1.282
N(2)-C(58)	1.451	1.451
δ	-	**0.114**
Bond angle (°)		
N(1)-Ni(1)-N(2)	83.4	76.943
Br1(1)-Ni(1)-Br(2)	123.26	124.279
N(1)-Ni(1)-Br(1)	109.8	110.476
N(1)-Ni(1)-Br(2)	111.7	112.352

N(2)-Ni(1)-Br(1)	111.8	112.455
N(2)-Ni(1)-Br(2)	109.9	110.574
δ	-	**2.898**

Table S3. Comparison of the bond lengths and bond angles between the calculated geometry and experimental data for complex **34,** along with the standard deviation (**δ**).

Complex 34	Experiment	Calculated
Bond lengths (Å)		
Ni(1)-Br(1)	2.3205	2.32
Ni(1)-Br(2)	2.3406	2.34
Ni(1)-N(1)	2.014	2.011
Ni(1)-N(2)	2.051	2.047
N(1)-C(1)	1.279	1.281
N(1)-C(45)	1.449	1.449
N(2)-C(12)	1.296	1.298
N(2)-C(13)	1.428	1.428
δ	-	**0.124**
Bond angle (°)		
N(1)-Ni(1)-N(2)	82.85	77.292
Br1(1)-Ni(1)-Br(2)	124.52	125.402
N(1)-Ni(1)-Br(1)	115.21	115.707
N(1)-Ni(1)-Br(2)	104.15	104.707
N(2)-Ni(1)-Br(1)	108.24	108.836
N(2)-Ni(1)-Br(2)	114.25	114.849
δ	-	**2.964**

Table S4. Comparison of the bond lengths and bond angles between the calculated geometry and experimental data for complex **61,** along with the standard deviation (**δ**).

Complex 61	Experiment	Calculated
Bond lengths (Å)		
Ni(1)-Br(1)	2.3213	2.322
Ni(1)-Br(2)	2.3267	2.326
Ni(1)-N(1)	2.045	2.042
Ni(1)-N(2)	2.015	2.012
N(1)-C(12)	1.293	1.295
N(1)-C(13)	1.432	1.431
N(2)-C(1)	1.28	1.282
N(2)-C(45)	1.446	1.446

δ	-	**0.118**
Bond angle (º)		
N(1)-Ni(1)-N(2)	82.72	77.436
Br1(1)-Ni(1)-Br(2)	123.6	124.416
N(1)-Ni(1)-Br(1)	110.3	110.87
N(1)-Ni(1)-Br(2)	112.59	113.165
N(2)-Ni(1)-Br(1)	113.44	113.984
N(2)-Ni(1)-Br(2)	106.76	107.311
δ	-	**2.827**

Table S5. Comparison of the bond lengths and bond angles between the calculated geometry and experimental data for complex **62,** along with the standard deviation (**δ**).

Complex 62	**Experiment**	**Calculated**
Bond lengths (Å)		
Ni(1)-Br(1)	2.3374	2.337
Ni(1)-Br(2)	2.3192	2.318
Ni(1)-N(1)	2.07	2.067
Ni(1)-N(2)	2.039	2.036
N(1)-C(12)	1.28	1.282
N(1)-C(13)	1.44	1.44
N(2)-C(1)	1.282	1.284
N(2)-C(45)	1.449	1.449
δ	-	**0.115**
Bond angle (º)		
N(1)-Ni(1)-N(2)	81.96	76.676
Br1(1)-Ni(1)-Br(2)	122.2	123.051
N(1)-Ni(1)-Br(1)	110.4	110.869
N(1)-Ni(1)-Br(2)	110.07	110.646
N(2)-Ni(1)-Br(1)	105.99	106.555
N(2)-Ni(1)-Br(2)	119.01	119.638
δ	-	**2.854**

Table S6. Detailed information of the 25 selected descriptors.

No.	Molecular descriptor	Category
1	MOMI-YZ	Moment of inertia
2	GATS4e	Autocorrelation
3	AATS8p	Autocorrelation
4	Avg electroph. react. index for a N atom	Molecular orbital related
5	Moment of inertia A	Geometrical
6	Balaban index	Topological
7	FNSA-3 fractional PNSA (PNSA-3/TMSA)	CPSA
8	GATS8e	Autocorrelation
9	AATS7p	Autocorrelation
10	RDF65m	Radial distribution function
11	MOMI-XZ	Moment of inertia
12	Relative number of single bonds	Constitutional
13	GATS3e	Autocorrelation
14	RDF120m	Radial distribution function
15	SIC1	Information content
16	Min (>0.1) bond order of a Ni atom	Molecular orbital related
17	Polarity parameter/square distance	Electrostatic
18	MATS4m	Autocorrelation
19	ATSC4i	Autocorrelation
20	AATS4i	Autocorrelation
21	MATS5i	Autocorrelation
22	ATSC8m	Autocorrelation
23	Min nucleoph. react. index for a C atom	Molecular orbital related
24	SIC4	Information content
25	LOBMIN	Length over breadth

	No.1																								
No.1	1.00	No.2																							
No.2	0.24	1.00	No.3																						
No.3	0.01	0.07	1.00	No.4																					
No.4	0.06	0.00	0.03	1.00	No.5																				
No.5	0.44	0.44	0.05	0.00	1.00	No.6																			
No.6	0.45	0.56	0.04	0.11	0.29	1.00	No.7																		
No.7	0.12	0.64	0.00	0.10	0.24	0.48	1.00	No.8																	
No.8	0.02	0.55	0.14	0.05	0.21	0.20	0.31	1.00	No.9																
No.9	0.17	0.00	0.50	0.01	0.07	0.02	0.03	0.06	1.00	No.10															
No.10	0.12	0.04	0.26	0.00	0.48	0.02	0.03	0.03	0.10	1.00	No.11														
No.11	0.98	0.20	0.05	0.05	0.45	0.37	0.11	0.00	0.22	0.18	1.00	No.12													
No.12	0.01	0.00	0.23	0.09	0.00	0.01	0.13	0.08	0.30	0.00	0.03	1.00	No.13												
No.13	0.06	0.23	0.04	0.38	0.03	0.15	0.25	0.01	0.04	0.00	0.06	0.05	1.00	No.14											
No.14	0.15	0.12	0.01	0.10	0.10	0.00	0.08	0.32	0.06	0.04	0.17	0.07	0.01	1.00	No.15										
No.15	0.12	0.56	0.17	0.01	0.10	0.44	0.42	0.40	0.01	0.00	0.07	0.05	0.09	0.03	1.00	No.16									
No.16	0.01	0.02	0.01	0.04	0.00	0.10	0.01	0.00	0.06	0.00	0.01	0.00	0.14	0.01	0.00	1.00	No.17								
No.17	0.00	0.00	0.14	0.00	0.01	0.01	0.00	0.00	0.07	0.18	0.00	0.05	0.00	0.00	0.00	0.03	1.00	No.18							
No.18	0.02	0.17	0.00	0.00	0.21	0.05	0.16	0.37	0.01	0.18	0.02	0.00	0.00	0.11	0.19	0.06	0.18	1.00	No.19						
No.19	0.04	0.24	0.05	0.00	0.15	0.25	0.41	0.24	0.15	0.08	0.05	0.10	0.00	0.08	0.07	0.00	0.04	0.10	1.00	No.20					
No.20	0.12	0.44	0.06	0.00	0.14	0.44	0.36	0.34	0.00	0.00	0.10	0.00	0.11	0.04	0.21	0.10	0.01	0.01	0.50	1.00	No.21				
No.21	0.12	0.15	0.01	0.00	0.28	0.09	0.02	0.19	0.14	0.19	0.14	0.10	0.08	0.01	0.05	0.03	0.03	0.06	0.01	0.11	1.00	No.22			
No.22	0.22	0.03	0.36	0.00	0.19	0.10	0.04	0.00	0.46	0.23	0.28	0.03	0.03	0.02	0.04	0.06	0.19	0.16	0.26	0.03	0.00	1.00	No.23		
No.23	0.00	0.02	0.05	0.08	0.00	0.02	0.00	0.00	0.02	0.01	0.01	0.00	0.08	0.04	0.00	0.10	0.00	0.00	0.02	0.00	0.00	0.09	1.00	No.24	
No.24	0.01	0.06	0.01	0.05	0.28	0.02	0.01	0.11	0.00	0.17	0.01	0.00	0.00	0.21	0.06	0.01	0.00	0.00	0.21	0.16	0.12	0.00	0.02	1.00	No.25
No.25	0.76	0.18	0.00	0.08	0.37	0.27	0.06	0.02	0.03	0.04	0.67	0.00	0.07	0.10	0.09	0.00	0.03	0.02	0.00	0.05	0.15	0.07	0.00	0.00	1
Act.	0.04	0.04	0.08	0.11	0.07	0.02	0.05	0.08	0.07	0.08	0.04	0.01	0.04	0.01	0.02	0.01	0.02	0.09	0.14	0.06	0.01	0.16	0.09	0.01	0.01

Figure S2. Triangular matrix of correlations among the selected 25 descriptors and activity.

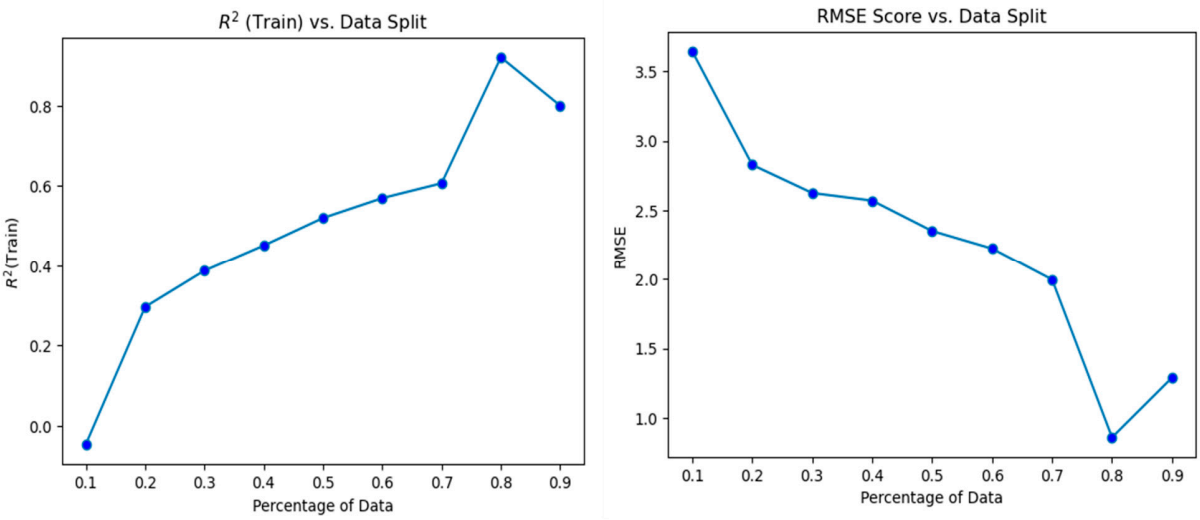

Figure S3. R^2 and *RMSE* scores as functions of different percentages of training data.

Table S7. Hyperparameters of the ML models.

Model	Parameter	Search Range	Value
XGBoost	n_estimators	[200, 300, 400, 500, 600, 700]	200
	max_depth	[1, 2, 3, 4, 5, 7, 9, 11, 12]	3
	learning_rate	[0.01, 0.1, 0.2, 0.3, 0.5]	0.1
	alpha	[0.03, 0.05, 0.09, 0.1, 0.2]	0.1
	lambda	[0.01, 0.1, 0.5, 0.8, 1.0]	0.1
CatBoost	Iterations	[50, 100, 200, 350, 550, 800]	100
	learning_rate	[0.05, 0.1, 0.2, 0.5, 0.6]	0.2
	max_depth	[1, 2, 3, 5, 6, 7, 8, 9]	5
Extra Trees	n_estimators	[100, 200, 500, 700, 800, 1000]	100
	max_features	[2, 5, 10, 15, 20, 25, 30, 35]	25
	min_samples_split	[2, 3, 6, 9, 12, 14, 15]	2
	max_depth	[5, 10, 20, 25, 30, None]	None
	n_estimators	[100, 150, 300, 500, 600, 900]	100
Random Forest	max_features	[5, 6, 8, 9, 12, 15, 18, 25]	9
	min_samples_leaf	[1, 2, 3, 5, 6, 8, 9]	1
	bootstrap	[True, False]	True
	min_samples_split	[2, 3, 4, 6, 8, 9, 11]	2
k-Nearest Neighbors	n_neighbors	[1, 2, 3, 4, 5, 7, 8, 10]	4
	weights	[uniform, distance]	distance
LightGBM	num_leaves	[10, 20, 30, 40, 50, 60, 70]	10
	boosting_type	[gbdt, rf]	gbdt

The performance of ML models, in particular the XGBoost model, can be significantly impacted by hyperparameters, such as the maximum depth, learning rate, and the number of trees. Maximum depth controls the complexity of the ensemble model. Increasing the value of the max depth enhances the model's ability to capture complex interactions between the input and output variables but may also lead to overfitting. The learning rate is also known as a shrinkage factor, which reduces feature weights to make the boosting process more conservative. Smaller values of the learning rate effectively minimize the loss function and reduce the risk of overfitting, while a higher number of trees helps prevent overfitting and contributes to regularization. Furthermore, examining the relationship between deviance and the number of iterations for the learning rate revealed consistent decreases in both training and testing errors,

indicating effective model training and strong generalization capabilities. Additionally, Figure **S4** shows that early stopping halted training in less than 250 iterations instead of 500, indicating that the model's performance did not improve after this iteration, thus preventing overfitting.

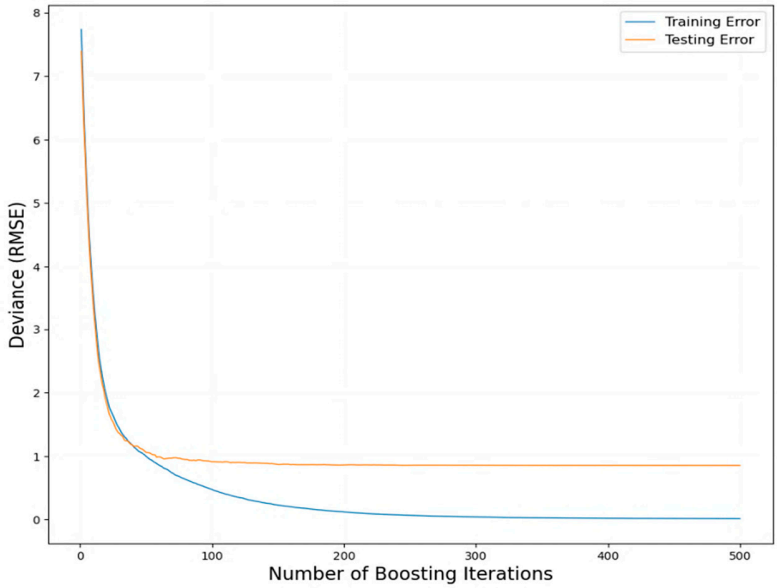

Figure S4. Training and test sets deviance against boosting iterations.

Dependency plots

In SHAP dependency plots, each dot within the plot represents an observation and accumulates along each feature row to visualize density. The x-axis of the dot is determined by the SHAP value of individual data points and the wide-area expresses the concentration of data points. Furthermore, the dots are color-coded based on their original value for that particular feature, ranging from red (higher) to blue (lower), indicating the feature's impact on the model. A higher feature value corresponds to a redder dot color.

Permutation Feature Importance

The global impact of features on catalytic activity prediction is also determined with the aid of PFI. Figure **S5** illustrates the contribution of each feature to the prediction output based on permutation importance. Following the permutation importance analysis, the five most important

features, ranked in descending order of significance, are MOMI-YZ, MOMI-XZ, AATS8p, AATS7p, and SIC1. The feature's value in the figure indicates the change in performance after reshuffling. The rankings of most of the feature importance in both SHAP and PFI exhibit a degree of similarity, as four key features (AATS8p, MOMI-XZ, MOMI-YZ, and AATS7p) appear in the top five features by both methodologies. However, the differences in order can be attributed to the fact that SHAP takes into account interaction effects among features, while PFI only considers the impact of each feature on the model score individually without capturing any interactions. It is noteworthy that SHAP offers a much more detailed analysis compared to PFI.

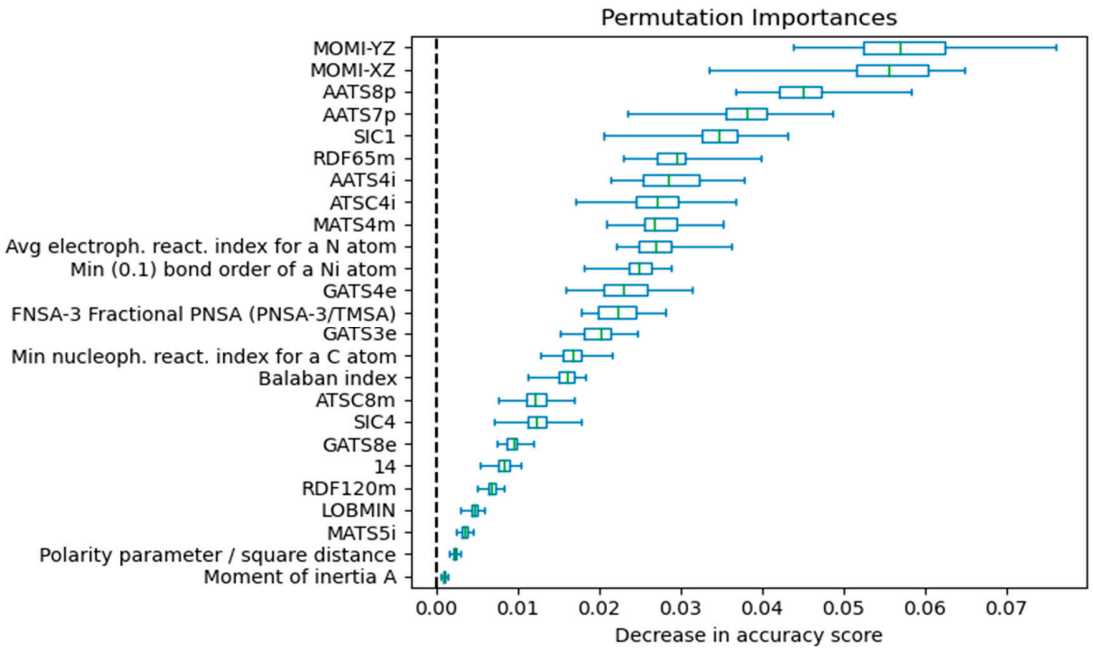

Figure S5. Feature importance ranking based on permutation feature importance (PFI).

Interpretation of RDF65m and RDF120m descriptors

This work utilized two radial distribution function (RDF) descriptors (RDF65m, RDF120m) to characterize the local atomic environment around the central atom. These descriptors provide information about the packing efficiency of the material. The RDF65m descriptor was calculated

at a distance of 6.5 Å, while RDF120m was calculated at a larger distance of 12.0 Å. These descriptors take into account the atomic masses of the atoms in the calculation. This weighting scheme emphasizes the contribution of heavier atoms to the overall mass distribution around the central atom. A higher descriptor value indicates a denser packing of atoms around the central atom, which can be associated with catalytic activity.