A Prediction Model for Heat Exchanger Fouling Factor based on Stacking Model

Zhiping Chen, Yongle Meng, Haoshan Yu, Ruiqi Wang, and Wenwu Zhou

***Abstract*—Given the pressing demand for energy conservation, the petrochemical sector faces increasingly stringent energy-saving mandates. Heat exchangers, essential to this sector, suffer efficiency losses and increased energy consumption due to fouling. To ensure optimal operation of heat exchange systems, regular assessment of solid deposits and the implementation of cleaning schedules are imperative. However, the multitude of influencing factors renders traditional estimation methods unreliable. Consequently, we developed a stacking model to predict the fouling factor of heat exchangers. Specifically, we first constructed fouling factor prediction models using various machine learning techniques, then selected the best-performing models—random forest, extreme gradient boosting , and light gradient boosting machine—for integration. Finally, the predictions from these three models were fed into a linear regression layer to form the final stacking model. The results indicate that the constructed stacking model significantly enhances the accuracy of fouling factor prediction. This model not only surpasses traditional multilayer perceptron neural network methods but also outperforms the well-performing gaussian process regression. This achievement not only validates the effectiveness of our model but also provides robust support for future research and applications in related fields.**

Link to graphical and video abstracts, and to code: https://latamt.ieeer9.org/index.php/transactions/article/view/9050

***Index Terms*—Fouling Factor Prediction, Heat Exchanger Fouling, Stacking Model.**

I. Introduction

Heat exchangers, crucial equipment in the petrochemical industry, facilitate thermal energy transfer between fluids, playing a pivotal role in enhancing energy efficiency [1]. However, fouling—a common issue in heat exchangers—occurs during operation when solid deposits form on heat transfer surfaces due to the presence of particulate matter or chemical reaction products in the fluid [2], [3]. This phenomenon leads to reduced heat transfer performance, increased pressure drop, intensified corrosion, and ultimately affects process efficiency, equipment lifespan, energy consumption, and safety risks. Consequently, proactive fouling prevention and timely removal are vital for ensuring efficient heat exchanger operation [4]–[6]. Unfortunately, measuring fouling severity is both challenging and time-consuming, often yielding imprecise results. Therefore, there is an urgent need to explore novel methods for predicting fouling factor (R_f) from easily measurable variables. The R_f serves as a fundamental parameter for planning cleaning schedules. Its formation and severity depend on various factors, including heat transfer surface shape, material, roughness, temperature, fluid properties, flow velocity, flow rate, concentration, pH, etc. To accurately predict the R_f, nonlinear relationships and interactions among multiple variables must be considered. Consequently, predicting heat exchanger R_f becomes a complex and challenging task [7]–[10].

Artificial neural networks (ANNs), computational models inspired by biological neural systems, excel at learning complex patterns from data. By utilizing different input variables and configurations, ANNs can predict and analyze fouling in heat exchangers [11]–[13]. For instance, using input variables such as crude oil flow rate, temperature, and pipe diameter, ANNs can predict output variables like fouling layer thickness, fouling coefficient, and thermal resistance within heat exchangers [14]–[16]. These variables reflect fouling phenomena and their impact on heat exchanger performance. Aminian et al [17]. employed a four-layer feedforward neural network model with crude oil flow rate, pipe surface temperature, and diameter as input variables to predict fouling rates in crude oil preheaters, achieving an average relative error (MRE) of 26.23%. Similarly, Aminian et al [18]. predicted fouling thresholds in crude oil preheaters using surface temperature, Reynolds number, and Prandtl number, achieving an absolute mean relative error (AMRE) of 15.83%. Other studies have explored dynamic and static ANN modeling techniques for fault detection [19], isolation, and adaptation in heat exchanger closed-loop temperature control. Additionally, optimized ANN models based on moving window technology have been used for online monitoring and prediction of crude oil fouling behavior in industrial shell-and-tube heat exchangers [20], achieving early fouling estimation with an MRE of approximately 8%. Local linear wavelet neural network models have also been applied to predict temperature differences and efficiency in heat exchangers [21], with predictions closely aligned with experimental results. While ANNs offer powerful fouling coefficient prediction capabilities, they require substantial training data. Insufficient or low-quality datasets may limit their generalization ability across diverse fouling scenarios. Consequently, researchers are increasingly exploring ensemble learning algorithms to achieve more accurate predictions [22], [23]. Ensemble learning algorithm, as meta-algorithms, enhance model accuracy and robustness while reducing bias and variance by constructing and combining multiple machine

Z. Chen is with Xi'an University of Science and Technology and State Key Laboratory of Green and Low-carbon Development of Tar-rich Coal in Western China, China (e-mail: cupczp@163.com).

Y. Meng, H. Yu, R. Wang, and W Zhou are with Xi'an University of Science and Technology, China (e-mails: xust-myl@163.com, a892430275@gmail.com, ruiqi.wang@xust.edu.cn, and Zhww1015@163.com).

learning (ML) models. Hosseini et al [24] successfully predicted R_f in heat exchangers using various ML methods, with their proposed gaussian process regression (GPR) model outperforming ANNs and other ML approaches. Leveraging data characteristics and distributions, ensemble learning algorithms automatically select suitable ML methods and parameters, minimizing manual intervention and enhancing model reliability and usability. Additionally, these algorithms construct multiple ensemble models by employing different algorithms in the first layer of the ensemble architecture, allowing diverse algorithms to capture trends in training data and produce accurate results [25].

In our study, we utilized 11,626 actual samples analyzed by Davoudi et al [26]. After comparing the predictive performance of several classical ML models, we employ ensemble learning to construct a stacking model that integrates high-performing ML methods. Compared to traditional ML approaches, our model exhibits superior robustness and more accurate fouling coefficient calculations within heat exchangers.

II. Research Methodology

This section elaborates on experimental data, ML models, the model development process, and performance evaluation criteria. Relevant analyses and ML algorithms provide theoretical and practical foundations for the model. Fig.1 illustrates the regression learning workflow employed in our ML study.

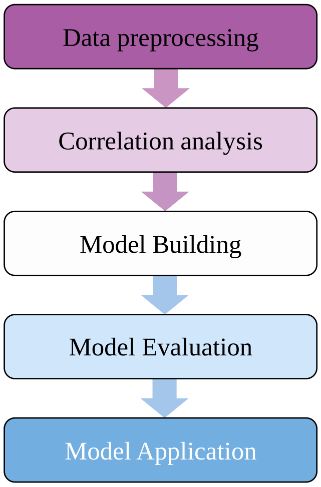


Fig. 1. Flowchart of the ML regression process.

A. Data Collection

The experimental dataset used in this study is sourced from [26]. As previously mentioned, we leveraged 11,626 laboratory-measured R_f to develop and test the proposed methods. Complete details about this database are provided in the supplementary materials.

B. Modeling of ML Methods

In our study, we selected k-nearest neighbors (knn) [27], random forest [28], bootstrap aggregating (bagging) [29], extreme gradient boosting (xgboost) [30], light gradient boosting machine (lightgbm) [31], and GPR [32] for comparison based on their diverse methodologies and widespread usage in the field of machine learning. Knn was chosen for its simplicity and effectiveness in handling non-linear data patterns. Random forest and bagging were included for their robustness as ensemble methods, improving model stability and accuracy by reducing variance. Xgboost and lightgbm were selected for their advanced boosting techniques, known for high performance and efficiency in processing large datasets. Finally, GPR was included for its probabilistic approach, providing valuable uncertainty estimates. Comparing these varied techniques allows for a comprehensive evaluation of their performance in fouling factor prediction, providing deeper insights into their respective strengths and applications.

In all of the following model training process, we divided the dataset into a training set and a test set, which account for 80% and 20% of the total dataset respectively. This division method effectively evaluates the generalization ability of the model and avoids overfitting.

1) Knn: Initially, we preprocessed the collected data by addressing missing values, handling outliers, and standardizing the data to ensure consistency and accuracy. The features relevant to R_f were utilized as input variables for our model. To determine the optimal K value, we employed 5-fold cross-validation. Through this rigorous validation process, we identified K=5 as optimal for implementing the knn algorithm, striking a balance between bias and variance and ensuring robust classification performance on our dataset [33]. Subsequently, having selected the optimal K, we trained the knn model using the entire training set. Finally, we assessed the model's predictive performance on the testing set. Based on these results, we further refined the model parameters to enhance accuracy and reliability [34].

2) Random Forest: Random forest offers advantages such as reduced overfitting risk, improved generalization, adaptability to high-dimensional and large-scale data, and feature importance assessment. Input variables included operational and design parameters of the heat exchanger. By using random sampling and feature subset selection, we increased diversity among decision trees, thereby enhancing prediction accuracy and stability [35]. The objective function of random forest can be expressed as:

$$f(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x), \tag{1}$$

Where $f(x)$ represents the random forest prediction, B denotes the number of trees in the forest, and $T_b(x)$ represents the prediction of the b-th tree for input x.

3) Bagging: Bagging is an ensemble learning technique used for both regression and classification tasks. It involves training multiple base models independently and in parallel on different subsets of the training data [36]. These subsets are generated using bootstrap sampling, where the data points are randomly selected with replacement. In the case of the bagging classifier, the final prediction is made by aggregating the predictions of all base models using majority voting. For regression models, the final prediction is obtained by averaging the predictions of all base models. Bagging improves

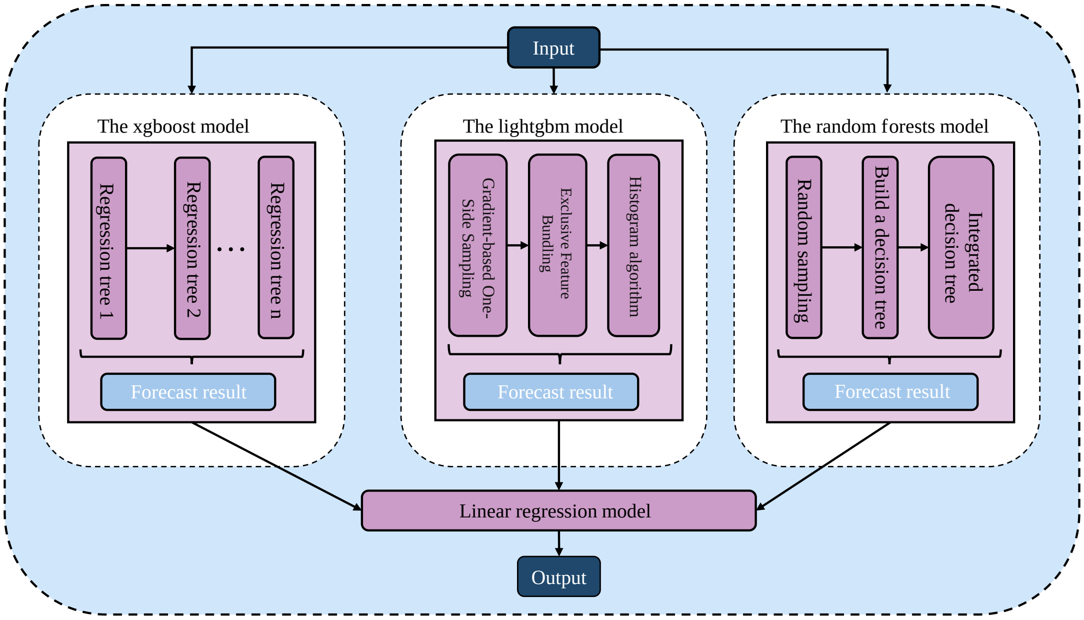


Fig. 2. Network structure diagram of stacking Model.

accuracy and reduces overfitting, especially in models with high variance.

4) Xgboost: Xgboost is an ensemble model of decision trees used for predicting R_f in heat exchangers. After data preprocessing and feature selection, the dataset is split into training and testing sets. The xgboost regression model is initialized and tuned, and its performance is evaluated on the testing set [37]. Xgboost excels in data processing speed, prediction performance, and feature importance assessment. Its objective function can be expressed as:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{2}$$

Where n is the number of samples, K represents the number of trees, y_i is the true value of the i-th sample, $\hat{y}_i$ is the predicted value of the i-th sample, f_k is the function of the K-th tree, and $\Omega(f_k)$ is the prediction of the K-th tree.

5) GPR: Firstly, after data cleaning and preprocessing, we used the prepared training data to train the GPR model. GPR estimates relationships between data points based on the selected kernel function and the covariance matrix of the training data, establishing a mapping between R_f and operating conditions. For new data samples, the model uses Gaussian processes for interpolation or regression to predict R_f. Additionally, GPR provides uncertainty estimates related to the predictions, crucial for assessing prediction reliability. Finally, we evaluated the model's performance using the test set. [38].

6) Lightgbm: After cleaning and preprocessing the data, we train the lightgbm model using the prepared training data. We employ a gradient boosting approach, constructing multiple decision tree models iteratively to enhance the model. We obtain the final prediction through weighted averaging or voting. During training, lightgbm adjusts tree parameters based on the gradient of the loss function to minimize the loss. For new data samples, the model combines predictions from an ensemble of decision trees. Finally, we evaluate the model's performance using a testing set and optimize it based on the evaluation results [39].

C. Stacking Model

The stacking model combines multiple individual models into a multi-output prediction system. Compared to standalone ML models, stacking models leverage the strengths of multiple individual models, significantly enhancing prediction accuracy and stability by learning the relationships between different models [40], [41]. Our constructed stacking model consists of two layers (Fig.2):

Base Model Layer (First Layer): This layer comprises several ML models, including xgboost, lightgbm, and random forest. Each model independently predicts the training data, and their predictions are then passed to the next layer. The purpose is to exploit the unique advantages of each model in handling specific data types or tasks, capturing different aspects of the data. Notably, xgboost and lightgbm are gradient-boosted decision tree models, highly effective in handling nonlinear relationships and feature interactions. Random forest, on the other hand, is an ensemble learning method based on bagging, adept at handling high-dimensional data and mitigating overfitting.

Meta Model or Final Output Layer (Second Layer): In this layer, the first-layer predictions serve as input features for the final prediction. In our approach, the second layer consists of a linear regression model. It aims to learn the optimal combination of predictions from different base models, thereby improving prediction accuracy and stability. The key advantage of stacking model lies in its ability to overcome limitations inherent in individual models.

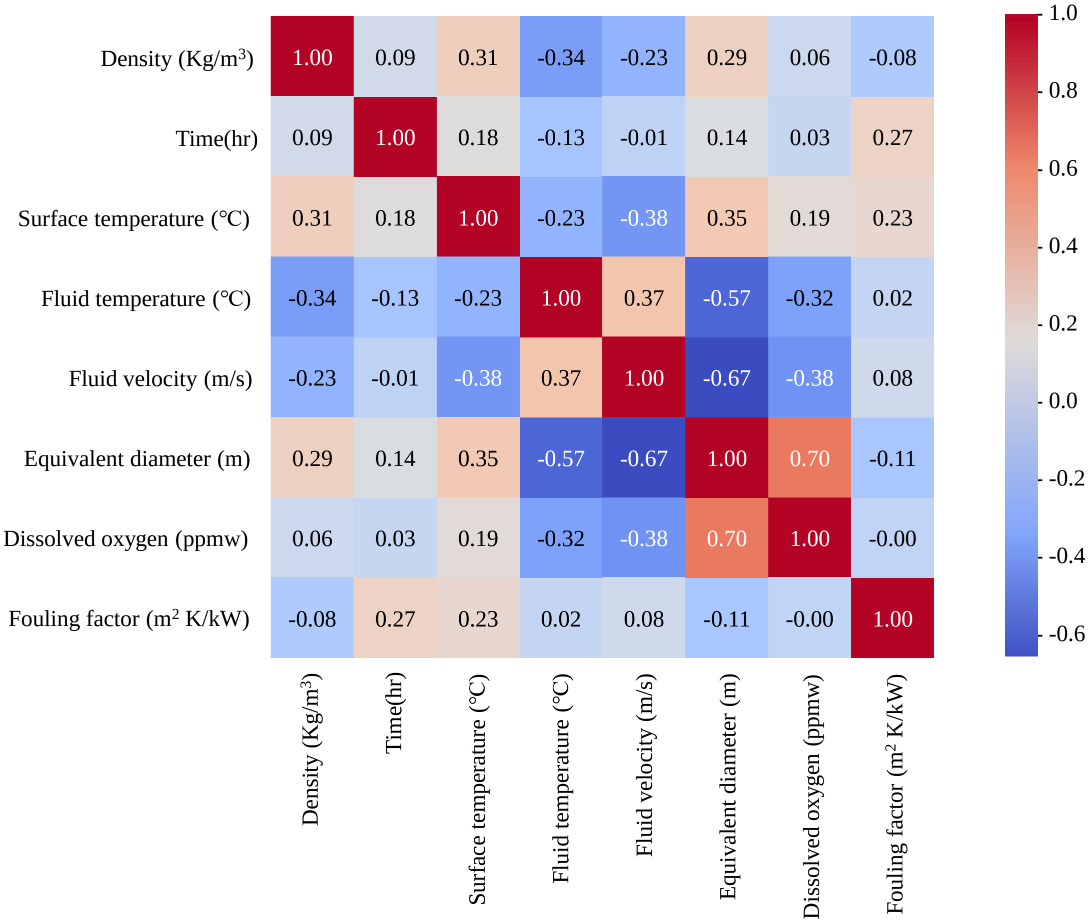


Fig. 3. Correlation heat map of factors affecting heat exchanger fouling.

Model parameter settings significantly impact the performance of stacking model. We split the dataset into 80% training data and 20% testing data, adjusting base model parameters for optimal performance. For xgboost, we set the learning rate to 0.1, the number of trees to 800, and the maximum tree depth to 8. In lightgbm, the typical learning rate is 0.05, with 800 trees and a maximum of 16 leaves. For random forest, we use 300 trees and a maximum depth of 6.

D. Data Preprocessing

1) Data Cleaning: Data cleaning is a fundamental step in data analysis and mining, enhancing data quality and model accuracy [7]. In this study, we initially removed rows containing missing values to reduce the number of gaps. Next, we employed mean imputation, replacing missing values with the dataset's average to maintain data integrity [8]. Finally, we replaced negative values with zeros to avoid unreasonable values. These steps completed the data cleaning process.

2) Feature Scaling: Feature scaling is a common preprocessing step in ML, ensuring that features with different scales have similar ranges. This step is crucial for algorithms that are sensitive to the scale of input data. In our study, we applied mean normalization [42] to scale the feature values to a range of [-1, 1] and centered the data around a mean of 0.

The formula for mean normalization is:

$$X' = \frac{X - \mu}{X_{\text{max}} - X_{\text{min}}}, \tag{3}$$

where X is the original feature value, μ is the mean of the feature, X_{max} is the maximum value of the feature, X_{min} is the minimum value of the feature.

E. Correlation Verification

To assess the correlation between R_f and relevant features, we employed Kendall's correlation coefficient. This non-parametric statistic measures the degree of consistency between two ordered variables. It does not rely on parameter estimation or hypothesis testing and is robust against outliers [43]. Kendall's coefficient considers only the ranking relationship between data points, making it suitable for different data scales [44]. The formula for Kendall's correlation coefficient (τ) is:

$$\tau = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \text{sgn}(x_i - x_j) \cdot \text{sgn}(y_i - y_j) \tag{4}$$

Kendall's τ correlation coefficient measures the ranking consistency between two variables by comparing all pairs of data in the sample. n in the formula is the total number of samples,

$\text{sgn}(x_i - x_j)$ and $\text{sgn}(y_i - y_j)$ are symbolic functions used to determine the relative size between variables.

The Kendall correlation heatmap (Fig. 3) reveals the following relationships between different variables: density shows a moderate positive correlation with surface temperature (correlation coefficient of 0.31) and a moderate negative correlation with fluid temperature (correlation coefficient of -0.34). Time and R_f exhibit a moderate positive correlation (correlation coefficient of 0.27). Surface temperature and equivalent diameter have a relatively strong positive correlation (correlation coefficient of 0.35), but surface temperature is moderately negatively correlated with flow rate (correlation coefficient of -0.38). Fluid temperature and flow rate exhibit a moderate positive correlation (correlation coefficient of 0.37), while fluid temperature and equivalent diameter have a relatively strong negative correlation (correlation coefficient of -0.57). Additionally, flow rate and equivalent diameter exhibit a relatively strong negative correlation (correlation coefficient of -0.67), while equivalent diameter and dissolved oxygen show a strong positive correlation (correlation coefficient of 0.70). The R_f is moderately positively correlated with time and weakly positively correlated with surface temperature, with no significant correlations with other variables.

III. Results and Discussion

In this section, we evaluate the performance of the constructed ML models. Our evaluation is based on training and testing datasets, and we compare our model with existing ones.

A. Evaluation Metrics

In ML, model construction involves efficient algorithms, consideration of prediction accuracy, determination of optimal model structures and parameters, and comparison with other scenarios. To assess model performance, we employ five statistical uncertainty metrics [45], [46]: Mean Squared Error (MSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), Relative Average Absolute Error (RAE%), and Mean Absolute Percentage Error (MAPE%). These metrics quantitatively measure the differences between actual and predicted data, objectively assessing model fit quality and generalization ability. Each metric serves a specific role in evaluation: MSE and MAE assess overall fit, R^2 measures explanatory variability, while RAE% and MAPE focus on relative and percentage errors. Lower values of MAE, RAE, MSE, and MAPE indicate higher regression model accuracy. The R^2 value ranges from 0 to 1, with values closer to 1 indicating better model fit and values closer to 0 indicating poorer fit. Establishing ML models requires comprehensive consideration of various factors, and these uncertainty metrics provide a comprehensive, objective approach to guide model selection and optimization, leading to improved performance across diverse application scenarios. The formulas for different evaluation metrics are as follows:

$$\text{MAPE\%} = \left(\frac{100}{n}\right) \times \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right|, \tag{5}$$

$$\text{MSE} = \left(\frac{1}{n}\right) \times \sum_{i=1}^{n} (\hat{y}_i - y_i)^2, \tag{6}$$

$$\text{MAE} = \left(\frac{1}{n}\right) \times \sum_{i=1}^{n} |\hat{y}_i - y_i|, \tag{7}$$

$$\text{RAE\%} = 100 \times \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |y_i - \overline{y}|}, \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (y_i - \overline{y})^2}, \tag{9}$$

Where y_i represents the actual value, $\hat{y}_i$ represents the predicted value, and n is the number of samples,$\overline{y}$ represents the mean of the actual values.

B. Comparison and Validation of ML Models

We evaluate the predictive performance of knn, random forest, bagging, xgboost, GPR, and lightgbm ML algorithms using five metrics. Table 1 reveals that random forest outperforms other methods in terms of MSE, MAE, and MAPE%, while achieving the highest R^2 and RAE%. This indicates that random forest is the most suitable method among the six for predicting R_f. Conversely, knn performs poorly across all metrics, indicating its unsuitability for this task. GPR exhibits subpar performance in MSE but moderate performance in other metrics, suggesting some adaptability for fouling factor prediction but lacking stability. Bagging, xgboost, and lightgbm perform well across all metrics, demonstrating high adaptability and stability for this problem. These methods exhibit significantly smaller MSE, MAE, and MAPE% than knn and GPR, with superior R^2 and RAE%. Notably, xgboost excels in R^2, while lightgbm performs best in MAE(Table I).

TABLE I
Performance Comparison of Classical ML Prediction Models

Methods	MSE	MAE	MAPE%	R^2	RAE%
Knn	6.94×10^{-4}	7.08×10^{-3}	19.19	0.98029	4.63
Random forest	1.30×10^{-4}	4.40×10^{-3}	16.68	0.99631	2.87
Bagging	1.54×10^{-4}	4.80×10^{-3}	16.06	0.99561	3.13
Xgboost	1.65×10^{-4}	4.82×10^{-3}	16.23	0.99671	3.09
Lightgbm	1.50×10^{-4}	4.34×10^{-3}	16.59	0.99622	3.28
GPR	4.50×10^{-4}	5.18×10^{-3}	16.21	0.99221	5.81

In order to more directly reflect the experimental results, we compared the experimental predicted R_f values of knn, random forest, bagging, xgboost, GPR and lightgbm with the real R_f values by regression graph (Fig. 4).

Fig. 4 presents regression graphs for various models, commonly used to display the relationship between predicted and actual values, thereby assessing model performance. Typically, the x-axis represents actual values, while the y-axis represents predicted values. Each point corresponds to an observation, with its x-coordinate representing the actual value and its y-coordinate representing the model's prediction. The regression line (best-fit line) indicates the degree of fit between actual and predicted values. Specifically, if the regression line coincides with the diagonal (y = x), it signifies perfect alignment between model predictions and actual values. Deviation from

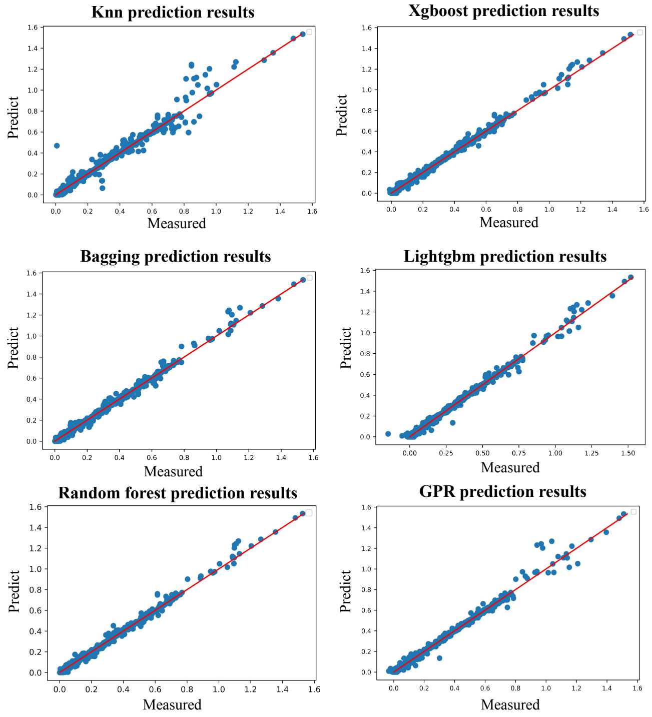


Fig. 4. Regression plots of R_f predictions from different ML methods against experimentally measured data.

the diagonal indicates discrepancies between predictions and actual values.

From Fig. 4, we observe that the predicted results align with the evaluation metrics. Notably, knn and GPR exhibit poor predictive performance. Conversely, xgboost, lightgbm, and random forest yield predictions close to actual values, demonstrating robust predictive capabilities. We employ ensemble learning to construct a stacking model, which will be discussed separately in Section III-C.

C. Comparison and Verification of Stacking Models

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS

Methods	MSE	MAE	MAPE%	R^2	RAE%
MLPNN	1.88×10^{-3}	2.01×10^{-2}	33.13	0.9463	15.7
GPR	4.5×10^{-4}	5.18×10^{-3}	16.21	0.99221	5.81
Stacking	1.26×10^{-4}	4.38×10^{-3}	15.94	0.99683	2.39

In previous studies, Davoudi and Vaferi [26] used the same dataset to build the multilayer perceptron neural network (MLPNN) model for predicting the fourth root $\sqrt[4]{R_f}$ of R_f, while Saleh Hosseini [24] used the GPR method to predict the quadratic root $\sqrt{R_f}$ of R_f. Excellent results have been obtained in the prediction effect (Table II). By comparing the forecasting performance indicators, it is not difficult to see that the ensemble model we constructed has better performance in predicting R_fs. Table 1 outperforms a single ML model on all evaluation metrics, specifically: The Mean Squared Error MSE of the stacking model is the lowest, only 1.26×10^{-4}. The MSE of the MLPNN model and the GPR method are 1.88×10^{-3} and 4.51×10^{-4}, respectively, which means that the difference between the predicted value and the real value is minimal. The Mean Absolute Error (MAE) of the stacking model is also the lowest, only 4.38×10^{-3}. The MAE of the MLPNN model and the GPR method are 2.01×10^{-2} and 5.18×10^{-3}, respectively, which means that the absolute error between the predicted value and the real value is the smallest. The Mean Absolute Percentage Error MAPE% of the stacking model is also the lowest, only 15.94%. The MAPE% of the MLPNN model and the GPR method are 33.13% and 16.21%, respectively, which means that the relative error between the predicted value and the real value is minimal. The Coefficient of Determination R^2 of the stacking model is the highest, reaching 0.99683. The R^2 of the MLPNN model and the GPR method are 0.9463 and 0.99221, respectively, which means that it has the highest linear correlation between the predicted value and the real value. The Relative Absolute Error RAE% of the stacking model is also the lowest, at 2.39%. The RAE% of the MLPNN model and GPR method are 15.70% and 5.81% respectively, which means that the relative absolute error between the predicted value and the true value is minimal compared with the simple average method.

Our research results robustly demonstrate the significant advantages of our designed stacking model in predicting pollution factors. Its predictive performance not only surpasses traditional MLPNN methods but even outperforms well-performing GPR methods. This achievement not only confirms the effectiveness of our model but also provides strong support for future research and applications in related fields.

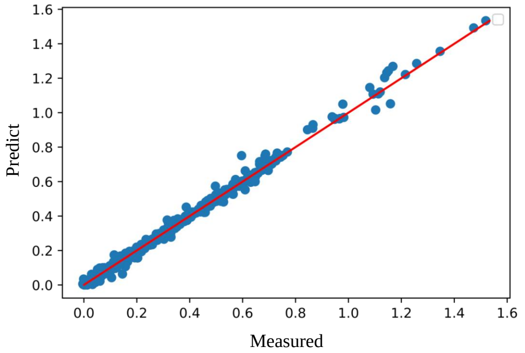
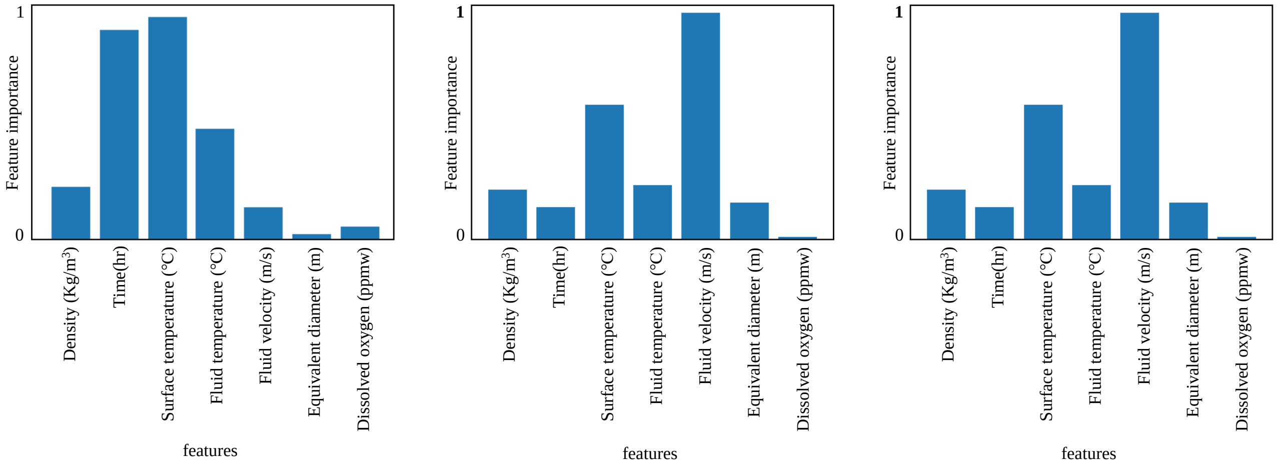


Fig. 5. Regression plots of R_f predictions from stacking Model methods against experimentally measured data.

For a more intuitive representation of experimental results, we have created a regression graph (Fig. 5) comparing the stacking model's experimental predicted R_f values with the actual R_f values. This regression graph further validates the consistency and accuracy of the model's R_f predictions. The underlying idea of the regression graph is that for a perfect model, the regression line (fitted to the scatter plot) would align with the diagonal reference line (y = x).

It is clear that each prediction is closely related to the ground truth. From the figure, we notice that most of the predictions are very close to the ideal line and the regression line itself almost coincides with the y = x diagonal line. This



(a) Feature importance-xgboost

(b) Feature importance-random forest

(c) Feature importance-lightgbm

Fig. 6. Feature contribution diagram of different methods in stacking model.

further emphasizes the minimal error and variance between the predicted and actual values. Therefore, we can assert that the model has achieved near-perfect results.

D. Feature Contributions of the Stacking Methods

To explore the factors influencing R_f, we analyzed the feature contributions of the three ensemble learning models—xgboost, lightgbm, and random forest—within the stacking model (Fig. 6).

From the data, we deduce that the three ensemble learning models within the stacking model exhibit varying influence weights and feature contributions. For xgboost and lightgbm, the most influential features are Time, Surface temperature, and Fluid temperature. The least influential features are Equivalent diameter and Dissolved oxygen. In the case of random forest, the most influential features are Fluid velocity and Surface temperature, while the least influential features are Time and Dissolved oxygen. By integrating random forest, xgboost, and lightgbm, our stacking model optimally considers different feature contributions, overcoming the challenges of arbitrary model combinations and selection order, resulting in superior predictive performance.

IV. Conclusions

In this study, we focus on enhancing the prediction accuracy of R_f using ensemble learning. Under the premise of ensuring data quality and suitability through preprocessing, we employ Kendall's correlation coefficient to validate the data's interrelationships. This analysis provides valuable insights into data trends, informing subsequent model development. We train knn, random forest, bagging, xgboost, GPR, and lightgbm methods, comprehensively comparing their performance and integrating the best-performing methods into a stacking model. Rigorous model construction and performance comparison robustly demonstrate the significant advantages of the stacking method in predicting R_f. Specifically, the stacking model excels across all evaluation metrics, surpassing not only traditional MLPNN methods but also outperforming well-performing GPR methods. Notably, the stacking model achieves an outstanding R^2 value of 0.99683, signifying high accuracy and robustness. Our research introduces novel insights and methods for predicting R_f in heat exchangers, providing valuable saupport for future applications.

Acknowledgments

The authors gratefully acknowledge the financial supports from the State Key Laboratory of Heavy Oil Processing (Program No. SKLOP202202004).

References

[1] J. Berce, M. Zupančič, M. Može, and I. Golobič, "A review of crystallization fouling in heat exchangers," *Processes*, vol. 9, no. 8, p. 1356, 2021. doi:10.3390/pr9081356.

[2] W. F. Alfwzan, G. A. Alomani, L. A. Alessa, and M. M. Selim, "Sensitivity analysis and design optimization of nanofluid heat transfer in a shell-and-tube heat exchanger for solar thermal energy systems: A statistical approach," *Arabian Journal for Science and Engineering*, pp. 1–17, 2023. doi:10.1007/s13369-023-08568-0.

[3] Y. Cao, A. Taghvaie Nakhjiri, S. M. Sarkar, and M. Ghadiri, "Integration of ann and nsga-ii for optimization of nusselt number and pressure drop in a coiled heat exchanger via water-based nanofluid containing alumina and ag nanoparticles," *Arabian Journal for Science and Engineering*, vol. 48, no. 7, pp. 8861–8869, 2023. doi:10.1007/s13369-022-07480-3.

[4] E. Sandrin, A. Hissanaga, J. Barbosa Jr, and A. da Silva, "On the relation between maximal thermal performance and minimal fouling deposition rate in heat exchanger-like devices," *Applied Thermal Engineering*, vol. 243, p. 122518, 2024. doi:10.1016/j.applthermaleng.2024.122518.

[5] R. Ranjan and S. Kumar, "An efficient cascaded effect based parallel flow heat exchanger using nonlinear model predictive controller based fuzzy optimization technique," *Arabian Journal for Science and Engineering*, vol. 48, no. 3, pp. 3227–3239, 2023. doi:10.1007/s13369-022-07120-w.

[6] M. Hiba, A. F. Ibrahim, S. Elkatatny, and A. Ali, "Application of machine learning to predict the failure parameters from conventional well logs," *Arabian Journal for Science and Engineering*, vol. 47, no. 9, pp. 11709–11719, 2022. doi:10.1007/s13369-021-06461-2.

[7] S. Sundar, M. C. Rajagopal, H. Zhao, G. Kuntumalla, Y. Meng, H. C. Chang, C. Shao, P. Ferreira, N. Miljkovic, S. Sinha, *et al.*, "Fouling modeling and prediction approach for heat exchangers using deep learning," *International Journal of Heat and Mass Transfer*, vol. 159, p. 120112, 2020. doi:10.1016/j.ijheatmasstransfer.2020.120112.

[8] S.-Z. Tang, M.-J. Li, F.-L. Wang, Y.-L. He, and W.-Q. Tao, "Fouling potential prediction and multi-objective optimization of a flue gas heat exchanger using neural networks and genetic algorithms," *International Journal of Heat and Mass Transfer*, vol. 152, p. 119488, 2020. doi:10.1016/j.ijheatmasstransfer.2020.119488.

[9] Y. K. Dossumbekov, N. Zhakiyev, M. A. Nazari, M. Salem, and B. Abdikadyr, "Sensitivity analysis and performance prediction of a micro plate heat exchanger by use of intelligent approaches," *International Journal of Thermofluids*, p. 100601, 2024. doi:10.1016/j.ijft.2024.100601.

[10] E. M. El-Said, M. Abd Elaziz, and A. H. Elsheikh, "Machine learning algorithms for improving the prediction of air injection effect on the thermohydraulic performance of shell and tube heat exchanger," *Applied Thermal Engineering*, vol. 185, p. 116471, 2021. doi:10.1016/j.applthermaleng.2020.116471.

[11] Z. Karimi Shoar, H. Pourpasha, S. Zeinali Heris, S. B. Mousavi, and M. Mohammadpourfard, "The effect of heat transfer characteristics of macromolecule fouling on heat exchanger surface: A dynamic simulation study," *The Canadian Journal of Chemical Engineering*, vol. 101, no. 10, pp. 5802–5817, 2023. doi:10.1002/cjce.24832.

[12] E. Reynoso-Jardón, A. Tlatelpa-Becerro, R. Rico-Martínez, M. Calderón-Ramírez, and G. Urquiza, "Artificial neural networks (ann) to predict overall heat transfer coefficient and pressure drop on a simulated heat exchanger," *International Journal of Applied Engineering Research*, vol. 14, no. 13, pp. 3097–3103, 2019. doi:.

[13] A. K. Gupta, P. Kumar, R. K. Sahoo, A. K. Sahu, and S. K. Sarangi, "Performance measurement of plate fin heat exchanger by exploration: Ann, anfis, ga, and sa," *Journal of Computational Design and Engineering*, vol. 4, no. 1, pp. 60–68, 2017. doi:10.1016/j.jcde.2016.07.002.

[14] T. N. Verma, P. Nashine, D. V. Singh, T. S. Singh, and D. Panwar, "Ann: Prediction of an experimental heat transfer analysis of concentric tube heat exchanger with corrugated inner tubes," *Applied Thermal Engineering*, vol. 120, pp. 219–227, 2017. doi:10.1016/j.applthermaleng.2017.03.126.

[15] L. M. Romeo and R. Gareta, "Fouling control in biomass boilers," *Biomass and bioenergy*, vol. 33, no. 5, pp. 854–861, 2009. doi:10.1016/j.biombioe.2009.01.008.

[16] A. Boloorchi and M. Jafari Nasr, "A model for fouling of plate-and-frame heat exchangers in food industry," *Asia-Pacific Journal of Chemical Engineering*, vol. 7, no. 3, pp. 427–433, 2012. doi:10.1002/apj.585.

[17] J. Aminian and S. Shahhosseini, "Evaluation of ann modeling for prediction of crude oil fouling behavior," *Applied thermal engineering*, vol. 28, no. 7, pp. 668–674, 2008. doi:10.1016/j.applthermaleng.2007.06.022.

[18] J. Aminian and S. Shahhosseini, "Neuro-based formulation to predict fouling threshold in crude preheaters," *International Communications in Heat and Mass Transfer*, vol. 36, no. 5, pp. 525–531, 2009. doi:10.1016/j.icheatmasstransfer.2009.01.020.

[19] R. F. Garcia, "Improving heat exchanger supervision using neural networks and rule based techniques," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3012–3021, 2012. doi:10.1016/j.eswa.2011.08.163.

[20] M. N. Kashani, J. Aminian, S. Shahhosseini, and M. Farrokhi, "Dynamic crude oil fouling prediction in industrial preheaters using optimized ann based moving window technique," *Chemical Engineering Research and Design*, vol. 90, no. 7, pp. 938–949, 2012. doi:10.1016/j.cherd.2011.10.013.

[21] D. K. Mohanty and P. M. Singru, "Fouling analysis of a shell and tube heat exchanger using local linear wavelet neural network," *International journal of heat and mass transfer*, vol. 77, pp. 946–955, 2014. doi:10.1016/j.ijheatmasstransfer.2014.06.007 .

[22] L. Goliatt, C. Saporetti, L. Oliveira, and E. Pereira, "Performance of evolutionary optimized machine learning for modeling total organic carbon in core samples of shale gas fields," *Petroleum*, vol. 10, no. 1, pp. 150–164, 2024. doi:10.1016/j.petlm.2023.05.005.

[23] R. M. Adnan, R. R. Mostafa, H.-L. Dai, S. Heddam, A. Kuriqi, and O. Kisi, "Pan evaporation estimation by relevance vector machine tuned with new metaheuristic algorithms using limited climatic data," *Engineering Applications of Computational Fluid Mechanics*, vol. 17, no. 1, p. 2192258, 2023. doi:10.1080/19942060.2023.2192258.

[24] S. Hosseini, A. Khandakar, M. E. Chowdhury, M. A. Ayari, T. Rahman, M. H. Chowdhury, and B. Vaferi, "Novel and robust machine learning approach for estimating the fouling factor in heat exchangers," *Energy Reports*, vol. 8, pp. 8767–8776, 2022. doi:10.1016/j.egyr.2022.06.123.

[25] K. M. Ting and I. H. Witten, "Issues in stacked generalization," *Journal of artificial intelligence research*, vol. 10, pp. 271–289, 1999. doi:10.1613/jair.594.

[26] E. Davoudi and B. Vaferi, "Applying artificial neural networks for systematic estimation of degree of fouling in heat exchangers," *Chemical Engineering Research and Design*, vol. 130, pp. 138–153, 2018. doi:10.1016/j.cherd.2017.12.017.

[27] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007. doi:10.1016/j.patcog.2006.12.019.

[28] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017. doi:.

[29] N. Altman and M. Krzywinski, "Ensemble methods: bagging and random forests," *Nature Methods*, vol. 14, no. 10, pp. 933–935, 2017. doi:.

[30] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016. doi:10.1145/2939672.2939785.

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

[32] E. Schulz, M. Speekenbrink, and A. Krause, "A tutorial on gaussian process regression: Modelling, exploring, and exploiting functions," *Journal of Mathematical Psychology*, vol. 85, pp. 1–16, 2018. doi:10.1016/j.jmp.2018.03.001.

[33] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, *et al.*, "Top 10 algorithms in data mining," *Knowledge and information systems*, vol. 14, pp. 1–37, 2008. doi:10.1007/s10115-007-0114-2.

[34] D. Cheng, S. Zhang, Z. Deng, Y. Zhu, and M. Zong, "knn algorithm with data-driven k value," in *Advanced Data Mining and Applications: 10th International Conference, ADMA 2014, Guilin, China, December 19-21, 2014. Proceedings 10*, pp. 499–512, Springer, 2014. doi:10.1007/978-3-319-14717-8_39.

[35] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009. doi:10.1007/978-0-387-21606-5.

[36] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, pp. 123–140, 1996. doi:10.1007/BF00058655.

[37] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting xgboost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications*, vol. 9, no. 40, pp. 651–662, 2016. doi:.

[38] C. Rasmussen and Z. Ghahramani, "Infinite mixtures of gaussian process experts," *Advances in neural information processing systems*, vol. 14, 2001. doi:.

[39] Y. Ju, G. Sun, Q. Chen, M. Zhang, H. Zhu, and M. U. Rehman, "A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting," *Ieee Access*, vol. 7, pp. 28309–28318, 2019. doi:10.1109/ACCESS.2019.2901920.

[40] X. Guo, Y. Gao, D. Zheng, Y. Ning, and Q. Zhao, "Study on short-term photovoltaic power prediction model based on the stacking ensemble learning," *Energy Reports*, vol. 6, pp. 1424–1431, 2020. doi:10.1016/j.egyr.2020.11.006.

[41] N. Ke, G. Shi, and Y. Zhou, "Stacking model for optimizing subjective well-being predictions based on the cgss database," *Sustainability*, vol. 13, no. 21, p. 11833, 2021. doi:10.3390/su132111833.

[42] J. Han, M. Kamber, and J. Pei, "Data mining: Concepts and," *Techniques, Waltham: Morgan Kaufmann Publishers*, 2012.

[43] D. Valencia, R. E. Lillo, and J. Romo, "A kendall correlation coefficient between functional data," *Advances in Data Analysis and Classification*, vol. 13, pp. 1083–1103, 2019.

[44] C. Croux and C. Dehon, "Influence functions of the spearman and kendall correlation measures," *Statistical methods & applications*, vol. 19, pp. 497–515, 2010.

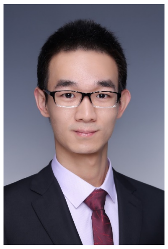
[45] Y. Cao, E. Kamrani, S. Mirzaei, A. Khandakar, and B. Vaferi, "Electrical efficiency of the photovoltaic/thermal collectors cooled by nanofluids: Machine learning simulation and optimization by evolutionary algorithm," *Energy Reports*, vol. 8, pp. 24–36, 2022. doi:10.1016/j.egyr.2021.11.252.

[46] W. Qiao, Y. Wang, J. Zhang, W. Tian, Y. Tian, and Q. Yang, "An innovative coupled model in view of wavelet transform for predicting short-term pm10 concentration," *Journal of Environmental Management*, vol. 289, p. 112438, 2021. doi:10.1016/j.jenvman.2021.112438 .

Zhiping Chen graduated from China University of Petroleum in 2014 with a Ph.D. degree. He is currently employed as an associate professor at Xi'an University of Science and Technology. He has presided over and participated in more than 20 scientific research and teaching reform projects at various levels, including the National Natural Science Foundation, and published more than 40 related papers.

Yongle Meng received the bachelor's degree from Guangxi University for Nationalities, Nanning, China, in 2020. She is currently pursuing the master's degree with the Xi'an University of Science and Technology, Xi'an, China.Her research focuses on deep learning and data mining.

Haoshan Yu was born on July 21, 2005 and is currently studying at Xi'an University of Science and Technology. His research focuses on ML.



Ruiqi Wang graduated from China University of Petroleum in 2021 with a Ph.D. degree. He is currently employed at Xi'an University of Science and Technology, mainly engaged in research on chemical system engineering and chemical process optimization. He has participated in many National Natural Science Foundation of China general projects, excellent young projects and horizontal projects.

Wenwu Zhou selected for the National Postdoctoral Innovative Talents Support Program and Shaanxi Province's High-Level Talent Introduction Program, among other talent development projects. He has published over 20 SCI papers, with a single paper achieving a maximum impact factor of 19.503 and a cumulative impact factor of approximately 186.