

Interpretation of Drop Size Predictions from a Random Forest Model Using Local Interpretable Model-Agnostic Explanations (LIME) in a Rotating Disc Contactor

Hardik Prabhu, Aamod Sane, Renu Dhadwal, Naren Rajan Parlikkad,* and Jayaraman Krishnamoorthy Valadi*

Cite This: <https://doi.org/10.1021/acs.iecr.3c00808>

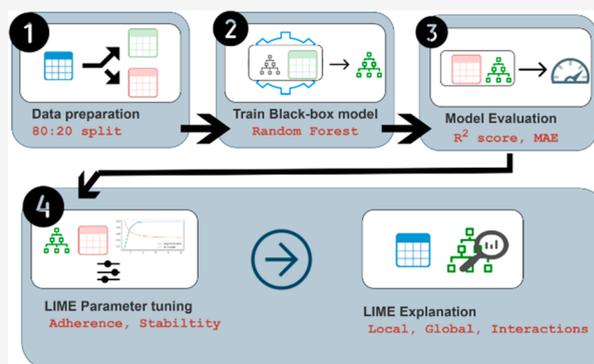
Read Online

ACCESS |

Metrics & More

Article Recommendations

ABSTRACT: Drop size is a crucial parameter for the efficient design and operation of the rotating disc contactor (RDC) in liquid–liquid extraction. The current work focuses on providing local and global explanations for the prediction of the drop size in a rotating disc contactor (RDC). The Random Forest (RF) regression model is a robust machine learning algorithm that can accurately capture complex relationships in the data. However, the interpretability of the model is limited. In order to address the issue of interpretability of the developed RF model, in the current work, we employed Local Interpretable Model-Agnostic Explanations (LIME) of the predictions of the RF model. This provides both local and global views of the model and thereby helps one to gain insights into the factors influencing predictions. We have provided local explanations depicting the impact of different attributes on the prediction of the output for any given input example. We have also obtained global feature importance, providing the top subset of informative attributes. We have also developed local surrogate models incorporating second order attribute interactions. This has provided important information about the effect of interactions on the drop size prediction. By augmenting the random forest model with LIME, it is possible to develop a more accurate and interpretable model for estimating the drop size in RDCs, ultimately leading to improved performance and efficiency.



1. INTRODUCTION

Estimating the drop size in a rotating disc contactor (RDC) is central to its design and efficient operation. An RDC is commonly employed in the liquid–liquid extraction process. A typical construction of an RDC consists of a rotating disc and a stator disc placed at a regular distance along a cylindrical column. Liquids that are essentially immiscible in nature are pumped into the RDC.¹ A swirl flow is created in the RDC by the combined action of the rotating discs and stator discs. This results in the shearing of the dispersed phase fluid into droplets and facilitates mass and heat transport between the dispersed and the continuous phase.

In our previous work,² empirical correlations for the estimation of drop size were evaluated. Further, a machine learning model based on the random forest was developed. The results of our previous work² demonstrated that the model achieved considerable accuracy in estimating drop size in the RDC system. Building upon this foundation, our current work aims to enhance the random forest model by introducing an explainability technique, allowing us to gain deeper insights into the factors that influence drop size prediction. By

combining the accuracy of the random forest model with improved interpretability, this work shall pave the way for further optimization of RDC design and its performance.

The remaining sections of the paper are structured as follows: In **Section 2**, the need for explainable machine learning models and the concept of explainability in machine learning are introduced by illustrating examples of prior work in the field. Further, the lacuna in terms of dearth of research work in process engineering involving explainable AI models is highlighted. This is followed by a brief overview of existing works on drop size prediction in an RDC, and the scope of the current work is presented in **section 2**. The subsequent section (**Section 3**) presents a comprehensive overview of various

Special Issue: Vivek Ranade Festschrift

Received: March 13, 2023

Revised: July 10, 2023

Accepted: July 12, 2023

aspects related to model interpretability using a local surrogate model, description of LIME, and chosen explanation methodology in this work. Section 4 presents the data set used in this work and the experimental simulation setup employed in our current study. The results of our experiment are presented in Section 5, followed by the conclusion of our work in Section 6.

2. EXPLAINABLE AI MODEL FOR DROP SIZE ESTIMATION IN AN RDC

2.1. Need for Explainable Machine Learning Models.

The need for explainable machine learning models arises from the growing importance of transparency and trust in the decision-making process of artificial intelligence systems. As AI applications become more prevalent in various industries, including chemical engineering, it becomes crucial to understand and interpret the rationale behind the decisions made by these models. There is often a desire to have simple models that perform well on a given task. Simple models are advantageous because they are easier to understand, interpret, and implement. They often have fewer parameters, making them computationally efficient. However, as the complexity of the task increases, simple models may struggle to capture the intricate patterns and relationships present in the data. Complex problems often require models with a higher capacity and flexibility to learn and represent the underlying complexities accurately. In such cases, simple models may fail to achieve the desired level of performance and may be outperformed by more complex models. There exists a trade-off between model complexity and interpretability. Highly complex models, such as deep neural networks, random forest, and support vector machines, often exhibit superior performance but lack transparency. On the other hand, interpretable models, like decision trees or linear regression, provide clear explanations but may sacrifice accuracy. Explainable AI aims to provide insights into how any given model, whether transparent or black-box, arrives at its conclusions, making it easier for humans to comprehend, validate, and correct the outcomes.

2.2. Previous Work on Explainable Machine Learning Models.

2.2.1. Explainability. Explainability refers to the ability to understand how and why a particular outcome was arrived at by a machine learning model. It can provide a visual and quantitative/qualitative explanation of factors involved in predictions for individual instances. It involves understanding the biases and errors in the model's logic. It facilitates the users (who may or may not be well versed with modeling concepts and procedures) a clear and concise understanding of the outcomes. The issue of explainability in AI and ML gained momentum with the development of complex models. These black-box models, though accurate, provided very little insights into the working of these models. The review by Burkhart and Huber³ explores the principles and methodologies of explainable Supervised Machine Learning (SML) and provides key definitions. It reviews recent approaches to explainable SML and classifies them based on the introduced definitions. The review by Vilone and Longo⁴ categorizes theories related to the concept of explainability and the evaluation approaches for eXplainable Artificial Intelligence (XAI) methods. The review also critically examined the gaps and limitations and proposes future research directions. Explainability methods can be broadly categorized into intrinsic and posthoc approaches. Intrinsic methods, also known as transparent or interpretable models, are models that are inherently interpretable and have

understandable decision-making mechanisms. Decision trees, linear regression, and logistic regression are examples of ante-hoc models. On the other hand, posthoc methods involve interpreting the predictions of a black-box model without modifying the model itself. These methods can be further classified into model-agnostic and model-specific methods. Model-agnostic methods, such as LIME⁵ (Local Interpretable Model-Agnostic Explanations) and SHAP⁶ (SHapley Additive exPlanations), can be applied to any black-box model and are not dependent on the architecture of the model. Model-specific methods, such as activation maximization and saliency maps, are designed specifically for certain types of models such as convolutional neural networks.

2.2.2. Model-Agnostic Methods. SHAP⁶ (SHapley Additive exPlanations) is a model-agnostic method used for explainability. It assigns each feature in the input data a Shapley value that measures its contribution to the prediction. SHAP has been used in various applications, such as in predicting mortality risk in patients with sepsis and predicting energy consumption in buildings. Liu and Aldrich⁷ employed SHAP for explaining anomalies in coal processing data. Younis et al.⁸ provided Shap explanations for anomaly detection. Togo et al.⁹ provided an Explainable framework for toxicity prediction. Jang et al.¹⁰ augmented Fault Diagnosis of Industrial Processes modeling with SHAP explanations. Fatahi et al.¹¹ provided details of effects of cement rotary kiln variables with Shapley Value explanations.

LIME⁵ (Local Interpretable Model-Agnostic Explanations) is another popular model-agnostic method used to explain black-box models. It works by generating local interpretable models around a specific instance to approximate the behavior of the black-box model in that locality. It was developed by Ribeiro et al.⁵ and has become very popular in prediction of complex machine learning models. Nguyen et al.¹² employed LIME for Prediction of Parkinson's Disease Depression. Mardaoui et al.¹³ provided an analysis based on LIME for interpreting text data. Çiçek et al.¹⁴ provided a lucid account of application of LIME for detection of risk factors of PCOS patients. Jain et al.¹⁵ explained sentiment analysis results on social media texts through visualization. Thus, LIME has been used to provide in-depth insight into black-box model predictions.

Sensitivity analysis is a valuable technique used in various domains to understand the behavior of a model and the impact of input variables on its output. It helps quantify the sensitivity of the model to changes in the input variables and provides insights into the relative importance of those variables. Sensitivity analysis can be performed at both the global and local levels, each serving different purposes. Typically, local and global sensitivity analyses are treated as separate endeavors. In recent years, global sensitivity analysis has gained significant popularity and recognition in various fields and disciplines. For example, global sensitivity analysis employing Monte Carlo simulations has been performed previously.¹⁶ These methods accurately provide global feature importance. This is done by gauging the extent of variance of variables for local changes in the input features. Recently Inapkurti et al.¹⁷ successfully employed Monte Carlo based global sensitivity analysis in their deep learning-based prediction of particulate matter in air. Techniques such as LIME on the other hand are designed to explain individual predictions at a local level, providing insights into the factors influencing specific instances. Furthermore, LIME extends its interpretability beyond the local level by

aggregating the feature importance measures to derive a global perspective. LIME stands out by offering the flexibility to provide both local and global perspectives, thereby enhancing its applicability to interpreting model predictions.

2.3. Lack of Explainable Models in Process Engineering. Rigorous machine learning and deep learning approaches have been employed in chemical and process engineering and multiphase flow system applications for building accurate prediction models.^{18–22} In most of these approaches, explainability and interpretability aspects have not been addressed. Owing to the increasing importance of eXplainable Artificial Intelligence (XAI), it may be useful to employ explainability for interpretation of rigorously derived black-box approaches. This work is a step in that direction. As we had explained in the [Introduction](#), we have attempted to provide explanations to our earlier work, which deals with the prediction of drop size in an RDC.

2.4. Brief Overview of Existing Works on Drop Size Prediction in an RDC. Drop size and phase holdup are critical hydrodynamic parameters that influence the transport parameters such as mass and heat transport coefficients and thus influence the performance of the RDC. Drop size governs the interfacial surface area that is available for mass and heat transport across the phases in a multiphase system. The smaller the drop size, the greater the interfacial surface area per unit volume and, hence, the greater the rate of mass and heat transport. Hence, a reliable estimation of the drop size is significant in the design and operation of the RDC.

In our previous work,² we consolidated the available data points on dispersed phase holdup and drop size in the RDC from available literature sources. A partial list of sources for experimental data on the RDC includes Ghalehchian,²³ Ismail Al-Rahawi,²⁴ Korchinsky,²⁵ and Olney.²⁶ In the reported experimental work,^{23–26} the drop size denotes the average drop size expressed as volume surface mean diameter or called the Sauter mean diameter. Further, ten empirical correlations available in the literature for the estimation of drop size in the RDC were consolidated. For instance, Ismail Al-Rahawi²⁴ has performed experiments using a Toluene-Water system with acetone as solute and reported the average drop size or Sauter mean diameter under varied conditions. Further, Ismail Al-Rahawi²⁴ has proposed empirical correlation for the estimation of the mean drop size. Since, we have not generated any new data in our work, we have adopted the definition of the drop size as per the available literature. More comprehensive information on the experimental data and the empirical correlations is made available as Supporting Information in our previous work² and is not reproduced here for the sake of brevity. The Supporting Information can be accessed at <https://pubs.acs.org/doi/10.1021/acs.iecr.0c04149>

We evaluated the empirical correlations for the estimation of the drop size. The prediction efficacy was evaluated by computing the average absolute relative error (AARE) values. The AARE was higher than 50% for most data points. Hence, none of the compiled literature correlations served as reliable correlations for the estimation of drop size. The empirical correlations and the AARE values can be found in the Supporting Information of our previous work.²

The random forest model with a 5-fold cross-validation was developed as part of our previous work.² Machine learning models have been widely employed in chemical engineering for estimation of bubble size and holdup in a bubble column, flow regime identification, estimation of mass transport coefficient,

etc.^{20–22} Notwithstanding, prior to our previous work, we could not find any reports of machine learning based models for drop size estimation in an RDC. In our earlier work, prior to development of the random forest model, the linear regression model was also developed with stepwise regression for feature selection. However, it was observed that the linear regression model gave a poor prediction performance as R^2 and AARE for the test set were found to be 0.6230 and 27.21%, respectively.² The random forest model based on random forest models with top features was developed. The prediction performance of the random forest for a test drop size data set was found to be $R^2 = 0.8725$ and AARE = 15.7946%. For a further description of the model and the results, one can refer to our earlier work² which is not reproduced here for the sake of brevity.

2.5. Scope of Current Work. Random forest²⁷ (RF) and other robust and rigorous machine learning algorithms can capture complex relationships in the data and provide accurate predictions. However, the interpretability of the model is often limited, making it difficult to understand the factors that influence the predictions. There is a need to unbox the black-box model by employing explainable model paradigms. Hence, in our current work, we employed Local Interpretable Model-Agnostic Explanations⁵ (LIME) along with the RF model to address the issue of interpretability of the RF model previously developed by us.² The LIME with RF provides both local and global views of the model and thereby helps to gain insights into the factors influencing predictions.

3. EXPLANATION METHODOLOGY EMPLOYED

In this section, we provide a comprehensive overview of various aspects related to model interpretability using a local surrogate model. We start with an introduction to the random forest algorithm and discuss its relevance as a black-box model. Then we introduce LIME (local interpretable model-agnostic explanations) and explain its significance in generating explanations for complex models. [Section 3.3](#) explores the scope of explanation, focusing on both global and local interpretability. [Section 3.4](#) delves into the details of fitting a local linear surrogate model using LIME, while [section 3.4](#) presents local explanations derived from this model. [Section 3.5](#) investigates the relationship between adherence and stability of the LIME explanations in relation to locality. In [Section 3.6](#), we discuss global explanations, and in [section 3.7](#), we introduce second-order feature interactions within the local model in order to capture feature interaction effects on the random forest model predictions.

3.1. Random Forest: Choice of the Black-Box Model.

Random forest²⁷ regression is a powerful and flexible black-box model. Its ability to handle nonlinear relationships, interactions between features, and noisy data makes it a popular choice among machine learning practitioners.^{28,29} Random forest is an ensemble learning method that evolved from decision trees. Decision tree regression works by recursively partitioning the input space into regions, followed by assigning a constant value to each region based on the average or median value of the target variable in that region. One of the main limitations of decision tree regression is its tendency to overfit the data. Decision trees have a high variance, which means that they are sensitive to small changes in the data and can easily overfit. Bagging (Bootstrap Aggregating) is an ensemble learning technique that reduces variance in machine learning models.^{30,31} It builds multiple models on multiple samples

randomly drawn with a replacement from the training set. By averaging the predictions of all models, the random variations in each individual model are smoothed out. The random forest model further decreases the variance by combining multiple decision trees using bagging and by introducing randomness into the tree-building process. This is done by randomly selecting a subset of all of the available attributes to partition the data on a given node. Random forest black-box models have been interpreted by using explainability techniques in different fields. Liu and Aldrich⁷ proposed a general anomaly detection and explanation method based on random forest and isolation forest with treeSHAP. The case studies are related to composition of coal and contain five, six, and twenty-two input attributes. Ribeiro et al.⁵ have explained random forest and neural networks using LIME for text and image classification, respectively.

3.2. LIME. Local Interpretable Model-Agnostic Explanations⁵ (LIME) provides an interpretable understanding of how a model is making predictions by approximating the complex black-box model with a simple interpretable surrogate model in the vicinity of the point of interest. Linear Regression is a popular choice for the local surrogate model. This is because the coefficients in a linear regression model directly represent the effect of each feature on the predicted outcome. The local model may not be fitted in the same feature space. For example, in the case of image data, the image is first partitioned into a collection of d superpixels. The image x is transformed into $x' \in \{0,1\}^d$. A local model is then fitted in $\{0,1\}^d$, a binary space of interpretable components. In the current work, tabular data consisting of numerical features is considered. The implementation is less complex when compared to unstructured data such as images. In the case of tabular data, no such transformation is required.

3.2.1. Scope of the Explanation. Local explanations refer to an explanation of a single prediction made by the model. In this approach, LIME generates a linear surrogate model that approximates the behavior of the original model in the vicinity of the instance to be explained. This approximated model is then used to generate feature importance values for local explanation. The usefulness of the generated LIME values is illustrated in detail in section 3.4.

Global explanations, on the other hand, refer to an explanation of the overall behavior of the model. In this approach, LIME generates a set of local explanations for a set of instances, and then an analysis of the feature importance values across all the instances in the set is done to understand the global behavior of the model. The usefulness of the generated LIME values is illustrated in detail in section 3.6.

3.3. Fitting a Local Model. According to the methodology provided in the LIME,⁵ the following steps are undertaken to fit a local surrogate model (f_1) for any given black-box model (f_b) around a data instance x .

1. Sample a set (S) of M points by performing perturbations on x , $S = \{s_1, s_2, s_3, \dots, s_M\}$.
2. Based on the proximity to the instance x , give weights W to all the instances in the generated sample. $W = \{\pi_x(s_1), \pi_x(s_2), \dots, \pi_x(s_M)\}$, where $W_i = \pi_x(s_i)$ is the weight assigned to the instance s_i by using function π_x . The closer a point is to the instance x , the higher the weight it gets.
3. Fit a surrogate model f_1 on the sample S by first using the trained black-box model predictions on S as the target

variable, $f_b(S) = \{f_b(s_1), f_b(s_2), \dots, f_b(s_M)\}$. Then using a weighted loss function train the surrogate model; the weights for all the sample point are given by W . The weighted loss function assigns higher weights to the sampled points that are closer to the instance x . This is because these points are likely to be more representative of the local behavior of the black-box model in the vicinity of x , and therefore, the local model should be better at approximating the black-box model in this region. For additional details, refer to section 3.3.1. The weighted mean square error (eq 2) has been used as the loss function to train the local surrogate model in the LIME python package. We are using the python package for our implementations. The code can be found at <https://github.com/marcotcr/lime>.

3.3.1. Implementation Details. Let $x \in \mathbb{R}^n$ be a data instance for which the black-box model prediction is $f_b(x)$. A Gaussian distribution is used for generating a neighborhood S around x (eq 1). The distribution mean is x itself, and the covariance matrix Σ is a diagonal matrix consisting of the variance along each feature. The training data could be used to estimate the variance.

$$S = \{x + e_j \mid \forall j \in \{1, 2, 3, \dots, N\}, \quad e_j \sim N(0, \Sigma)\} \quad (1)$$

Not all the points in the generated neighborhood are equally important. As LIME operates under the assumption that for a small enough locality the black-box model is approximately linear, it is important to understand that as we move closer in the generated neighborhood to x , the tighter the fit of the local surrogate model to the black-box model. This is captured by a weighted loss function to train the linear model. The fit of the local model to the black-box model is measured by taking the weighted mean square error between the local model (f_1) predictions and the black-box model (f_b) predictions over the generated neighborhood S (eq 2).

$$\sum_{i=1}^N W_i (f_1(s_i) - f_b(s_i))^2 \quad (2)$$

An exponential kernel with the parameter kernel width (k) is applied over the Euclidean distance (D) between a sampled point (s_i) and the instance x to generate the weight given to the sampled point (eq 3). The closer the point is to x , the higher its weight.

$$W_i = \pi_x(s_i) = \exp(-D(x, s_i)^2 / k^2) \quad (3)$$

Weighted Ridge regression is used as the local linear model in the LIME package. Ridge regression adds a penalty term to the linear regression cost function, which helps to prevent overfitting by shrinking the coefficients of the regression model toward zero. The cost function used for training the model is given below (eq 4)

$$\sum_{i=1}^N W_i (f_1(s_i) - f_b(s_i))^2 + \lambda \sum_1^n \theta_i^2 \quad (4)$$

where $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$ represents the feature coefficients of the local model, and λ is the regularization parameter. The local linear model is given below (eq 5).

$$f_1(x) = \theta^T x + \theta_0 \quad (5)$$

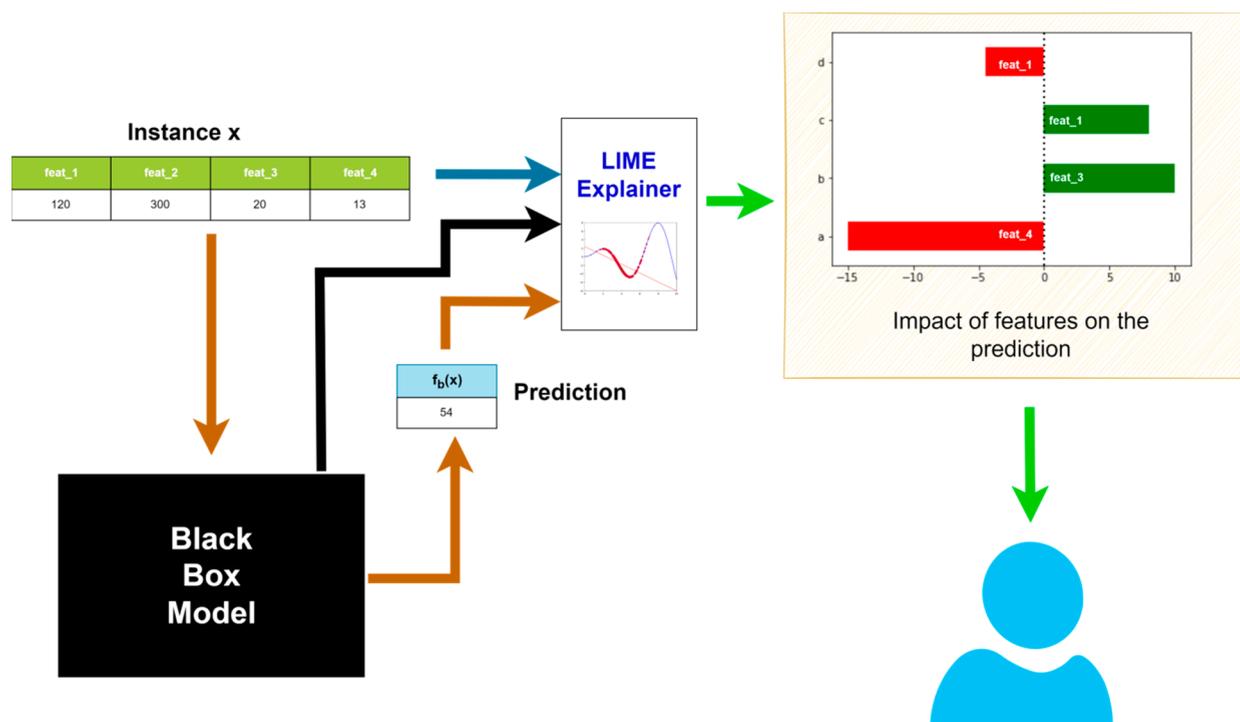


Figure 1. LIME explanation for the black-box model prediction made on instance x .

In sections 3.4 and 3.6, we will see how the locally fitted linear model helps to explain the black-box model predictions.

3.4. Local Explanations. LIME can be used to get both the local and global view of a black-box model's behavior, making it a powerful tool for understanding complex machine learning models. Once the black-box model is trained, its predictions made on the test data set could be explained by LIME. At the local level, LIME explains the prediction of a specific test instance by generating an interpretable local model around that instance. In section 3.3, ridge regression was selected as the local interpretable model. The outcome of the linear model is a linear combination of the feature values (eq 5). The importance of each feature can be quantified by examining the coefficients of the linear model. They indicate the direction and magnitude of the effects that each feature has on the prediction. As the local model is a good approximation of the black-box model around the instance, the larger the magnitude of a coefficient, the more important the corresponding feature is to the black-box model's prediction. Figure 1 illustrates the process of getting an explanation for the black-box model prediction on instance x . Instance x and the black-box model prediction on it are taken as inputs along with other inputs which are described in Algorithm 1. A bar plot created using the coefficients (LIME values) of the local linear model is returned.

As mentioned earlier, random forest regression is the complex black-box model used for making predictions. In order to generate an explanation for its prediction on a given instance, the following choices are made. The Gaussian distribution with the instance to be explained as the mean, and the training data standard deviation is used to generate a sample around the instance. Note that the exponential kernel only considers the Euclidean distance; therefore, the training data mean and the standard deviation is used to scale (transform) the sampled data, and then the weighted local model is fitted in the transformed space while the original

sampled data is used to get the black-box model predictions which are used as the target values to fit the local linear model. It also makes sense to have all of the features on an equal scale considering that the local model coefficients are taken as the LIME values. LIME also allows an additional feature selection step. It is done before fitting the local model to the sample generated around the instance to be explained. This is done to make the explanations compact and human-readable. The number of features to be selected (m) is decided by the user of the LIME package. A Lasso regression model is first fitted, and only the top m features are selected. The local model is then fitted by considering only the selected features. We have decided to keep all of the features as the number of total features is already small. Algorithm 1 summarizes the steps undertaken to explain the single instance.

Algorithm 1 Explain instance($x, f_b, M, k, \mu, \Sigma$)
Require: Instance x , black-box model f_b , size of the neighbourhood generated M , kernel width k and summary statistics (mean and variance) from the training dataset of the black-box model; μ and Σ .
 $S \leftarrow \text{GenerateSample}(x, M, \Sigma)$
 $Y \leftarrow f_b(S)$ \triangleright Query the black-box model to generate target values.
 $Z \leftarrow \text{StandardScaler}(S, \mu, \Sigma)$
 $x' \leftarrow \text{StandardScaler}(x, \mu, \Sigma)$ \triangleright Scale the sample as well as the instance x
 $W \leftarrow \text{GenerateWeights}(x', Z, k)$
 $\beta \leftarrow \text{WeightedRidge}(\text{data} = Z, \text{target} = Y, \text{weights} = W)$ \triangleright Get the coefficients corresponding to each feature from the trained local model
return β

3.5. Adherence and Stability Relationship with the Locality. The concept of locality is crucial to LIME. The kernel width controls the size of the locality within the generated neighborhood, which consists of only the significantly weighted points, which may influence the local surrogate model (eq 3). If the locality of significant points around the point of interest is too large, it may cause the LIME model to not adhere to the black-box model in the considered locality. The kernel width is the hyper-parameter which governs the size of the locality. As we move away from the instance of interest, the weights assigned to the generated points decay at a higher rate if the kernel width is small. This can be seen in the

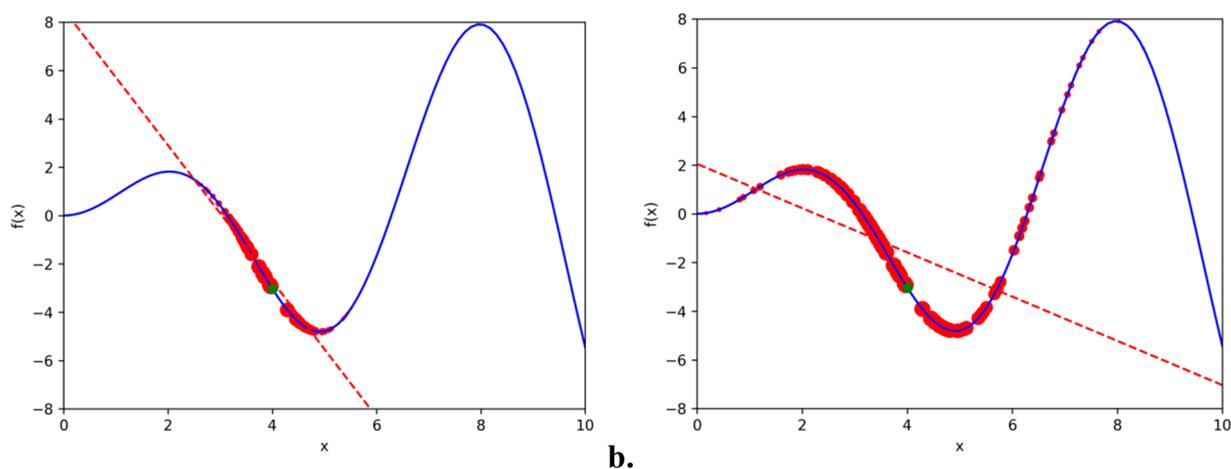


Figure 2. Effect of the kernel size on the local predictor.

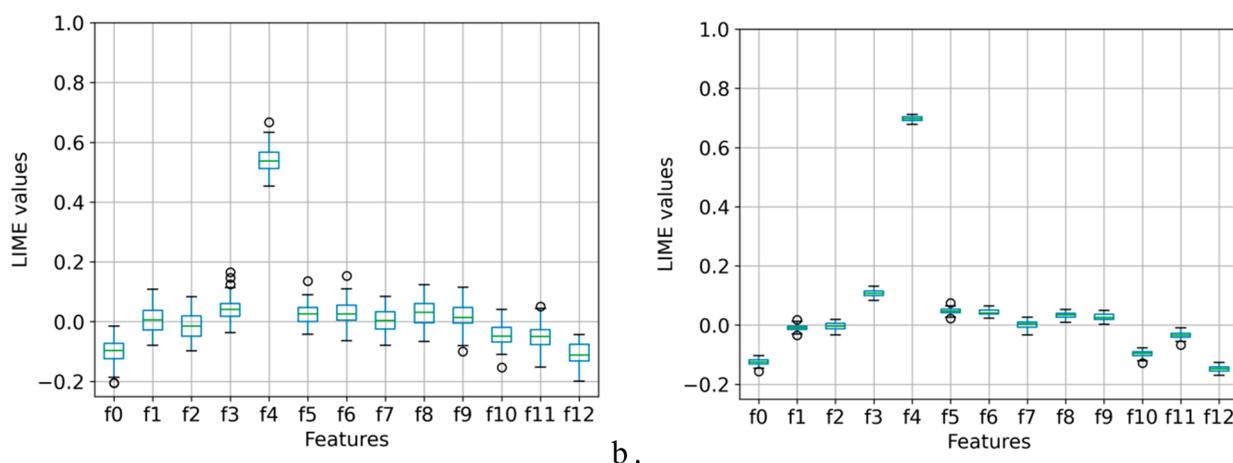


Figure 3. Box plots for coefficients of the local surrogate model for repeated LIME calls made to explain a single instance with different kernel widths.

Figure 2. The black-box model is shown with a blue line. The green point is the instance of interest. A neighborhood of points is generated around the instance of interest using a Gaussian distribution.

The generated points are colored red and have sizes proportional to the weights assigned to them. The weights are assigned to the neighborhood points by using two different kernel widths. The left subfigure (Figure 2a) uses a smaller kernel width of $k = 0.5$ compared to the right subfigure (Figure 2b) which uses a kernel width of $k = 1.5$. As a result, the locality of relevant points in the left subfigure is smaller than that in the right subfigure. A weighted linear regression is then fitted on the neighborhood data. As seen from the figure, the smaller the locality, the tighter the fit, and the lower the error between the black-box model and the local surrogate model at x .

Stability of the LIME explanation also depends on the locality. In order to fit a local model around an instance to be explained, first a random sample of perturbed data points is generated. If LIME is implemented to explain the same instance multiple times, every implementation of LIME will result in the fitting of a local model with a different generated sample. This may lead to slightly different values of the local model coefficients. This slight variation is tolerable if the explanations do not drastically change each time. Kernel width

and stability are expected to be directly proportional.³² One quick way to see this is to select an instance and a kernel width and perform repeated LIME calls in order to explain the same instance and observe the spread of coefficient values assigned to each feature by the local model with the help of a box plot. In Figure 3, this effect is demonstrated on a randomly chosen test instance from our data set. Two different kernel widths are used to plot the subfigures in Figure 3. The value of k used for the left subfigure (Figure 3a) is 0.5, while the right subfigure (Figure 3b) uses a kernel width of 1.5. The x -axis consists of the 13 features (f_0 - f_{12}). The y -axis represents the spread of value assigned to the coefficients across the repeated LIME calls. It can be observed that for the higher value of kernel width (Figure 3b), the variability decreases.

This issue of stability has been studied extensively in the work of Visani et al.³³ The authors have come up with two complementary indices to measure the stability for the repeated LIME calls, provided the local model considered is ridge regression. The Variables Stability Index (VSI) is used to measure whether the same set of features is selected in the feature selection step prior to fitting the local model. The coefficients stability index (CSI) is used to measure whether the coefficients attributed to the same feature can be considered equal. As the feature selection step has been skipped, only the CSI is used for quantifying stability. A high

CSI score indicates that the coefficients attributed to the same feature are consistent and stable across all of the samples generated for repeated LIME calls. The code used for calculation of the CSI value for any given instance is available at https://github.com/giorgiovisani/LIME_stability. The use of adherence and stability measurements for tweaking the kernel width is discussed in section 4.

3.6. Global Explanations. The explanation provided at the global level describes which features are generally important to the black-box model. We are primarily interested in understanding the behavior of the model on the data coming from the data distribution. To estimate the global behavior of the model, a sufficiently large data set which is representative of the data distribution should be considered. In order to perform an analysis on the global level, first, a LIME coefficient matrix is created. It is done by performing LIME calls on all the instances in the data set and storing the coefficients corresponding to all the features in a matrix with the same number of columns as features and rows as instances. For obtaining the coefficient matrix, refer to Algorithm 2 described below.

Algorithm 2 Generate Coeff Matrix ($X, f_b, M, k, \mu, \Sigma$)

Require: Dataset X , black-box model f_b , size of the neighbourhood generated M , kernel width k and summary statistics (mean and variance) from the training dataset of the black-box model; μ and Σ .
 $i = 0$
 $M \leftarrow$ no. of rows in X
 while $i \leq M$ do
 $\beta_i \leftarrow$ Explain instance($x_i, f_b, M, k, \mu, \Sigma$)
 $i = i + 1$
 end while
 $C = [\beta_1^T, \beta_2^T, \dots, \beta_M^T]$ \triangleright The matrix of coefficients generated by the local models
 return C

The coefficient matrix could be used to directly calculate the relevance of a feature on a global scale. It is done for each of the features (columns) by calculating the mean of the absolute values of the column corresponding to the feature. For a

feature, the trend in the LIME coefficient values could also be observed by plotting the values that a feature takes across the entire data set against the LIME coefficient values assigned to it. Figure 4 gives an illustration of the entire process.

3.7. Capturing Feature Interactions. Black-box models may capture feature interactions while making predictions, which may not be immediately apparent while exploring the data. A drawback of using a linear local surrogate model is that it assumes a linear relationship between the dependent variable and the independent variables, and therefore, they cannot directly express the captured feature interactions of the black-box model. However, it is possible to include interaction terms in the local linear model to express these interaction effects. An interaction term is a new variable that is created by multiplying two or more independent variables together. We propose to include second-order interaction variables in our local surrogate model (f_i) so that we could express the captured feature interaction of the black-box model locally (eq 6).

$$f_i(x) = \sum_{j=1}^n \theta_j x_j + \sum_{i < j} \theta_{ij} x_i x_j + \theta_0 \quad (6)$$

In our implementation, the weighted ridge regression penalty still applies, and the weights are calculated the same way as before. An additional feature engineering step is included before fitting the linear model. It modifies the input data set by including all the second order interaction features in the input data set. It is done by elementwise multiplying two columns together at a time. The total number of features including interactions increases by $O(n^2)$ where n is the number of features in the original data set. In order to keep the explanation compact, a feature selection step may also be employed on the modified data set before fitting the final model; this optional step is already present in LIME. The implementation is given in Algorithm 3.

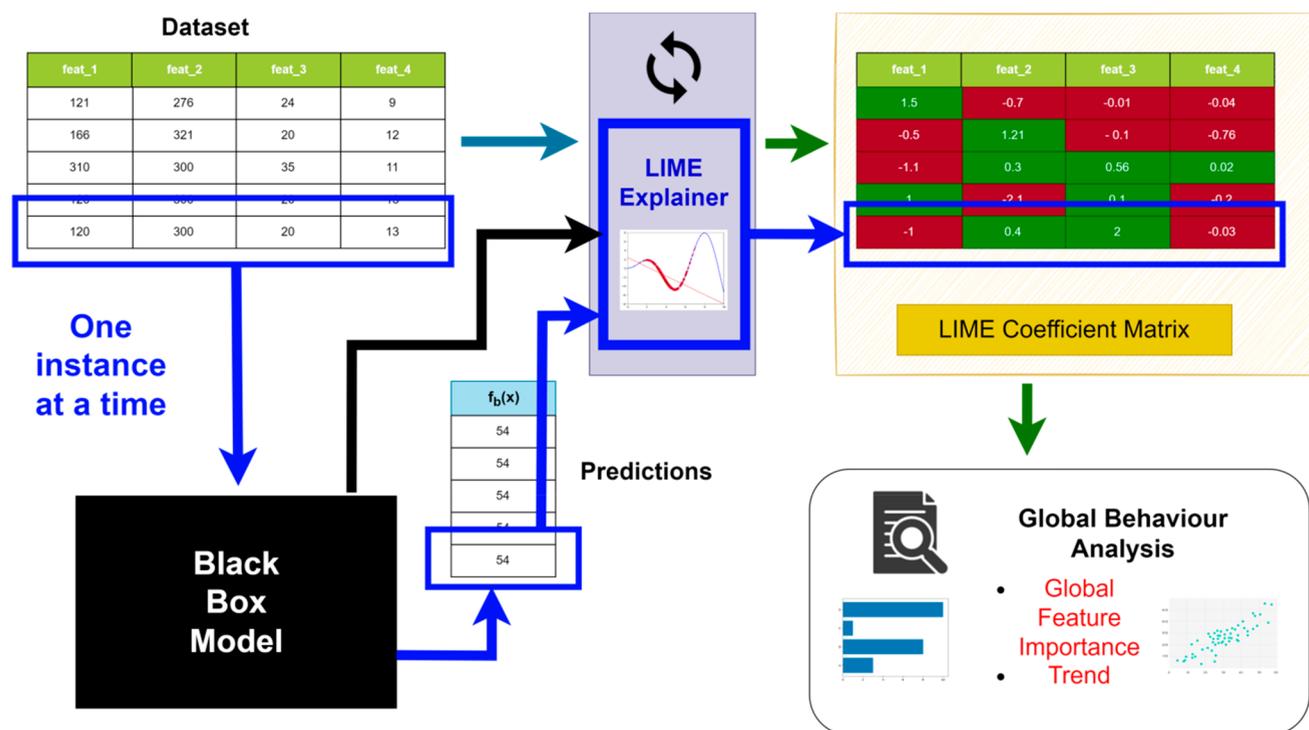


Figure 4. Global explanation of the black-box model predictions on a data set.

Algorithm 3 Explain instance with interactions ($x, f_b, M, k, \mu, \Sigma$)

Require: Instance x , black-box model f_b , size of the neighbourhood generated M , kernel width k and summary statistics (mean and variance) from the training dataset of the black-box model; μ and Σ .

```

S ← GenerateSample(x, N, Σ)
Y ← f_b(S)           ▷ Query the black-box model to generate target values.
Z ← StandardScaler(S, μ, Σ)
x* ← StandardScaler(x, μ, Σ) ▷ Scale the sample as well as the instance x
Z ← FeatureEngineering(Z) ▷ Include the 2nd order features
W ← GenerateWeights(x*, Z, k)
β ← WeightedRidge(data = Z, target = Y, weights = W) ▷ Get the
coefficients corresponding to each feature from the trained local model
return β

```

Another benefit of adding second-order interactions to the local surrogate model is that it allows for a more accurate representation of the relationships between the input features and the output variable. Second-order interactions can capture nonlinear relationships between input features that are not captured by a linear model that includes only first-order features.

4. DATA SET AND EXPERIMENTAL SETUP FOR SIMULATIONS

4.1. Data Set. In the current work, we have used the same data set compiled in our earlier work² and made available at 10.7910/DVN/HMNWIC.² Water was used as the continuous phase liquid in most of the studies. However, the dispersed phase fluid employed in an RDC varied in nature. We employed the DBSCAN algorithm for removing outliers from the data set. DBSCAN identifies three types of points including the outliers in the data set. Core points are those that have a sufficient number of neighboring points. Border points are those that are not core points but have at least one core point in their neighborhood. Outliers are points that are not core points and do not have any core points in their neighborhood. Outliers are located in regions of low point density. After removal of outliers, 572 drop size data points were used as the input data set for the work. Each data point is uniquely

described by 13 features: Dispersed Phase Density, Continuous Phase Density, Dispersed Phase Viscosity, Continuous Phase Viscosity, Interfacial tension, Diameter of the column, Height of the column, Compartment height, Diameter of the rotor disc, Diameter of the stator, Rotor speed, Dispersed Phase Velocity, and Continuous Phase Velocity. Dispersed phase droplet size is the output or target variable. The dispersed phase drop size denotes the mean drop size reported as the volume average diameter or the Sauter mean diameter, in line with the experimental studies.

4.2. Experimental Setup for Simulations. In section 3, we saw how LIME could be used to explain black-box model predictions, making it more interpretable. Our objective is to show the following: 1. The random forest model can accurately estimate the drop size in our data set. 2. Utilize LIME to explain the random forest model both locally and globally, as well as discover how feature interactions affect model predictions. The entire end to end experiment could be summarized in 5 steps (refer to Figure 5 for an illustration).

Step 1 (Data Processing). The data set containing 572 rows is first randomly split into a train and test data set using an 80:20 split. The train data set has 457 rows, and the test data set has 115 rows.

Step 2 (Train Model). The random forest model with a default set of hyper-parameters provided in the scikit-learn python package is selected as the black-box model. It is trained on the train data set.

Step 3 (Model Evaluation). The trained model is used to predict the drop size for the test data set. The predictions are compared with the actual target values using metrics R^2 and MAE.

Step 4 (Model Interpretation Using LIME). 4.1. LIME Parameter Tuning: The parameter kernel width (k) is tuned by taking a grid of several values; the best value is found by

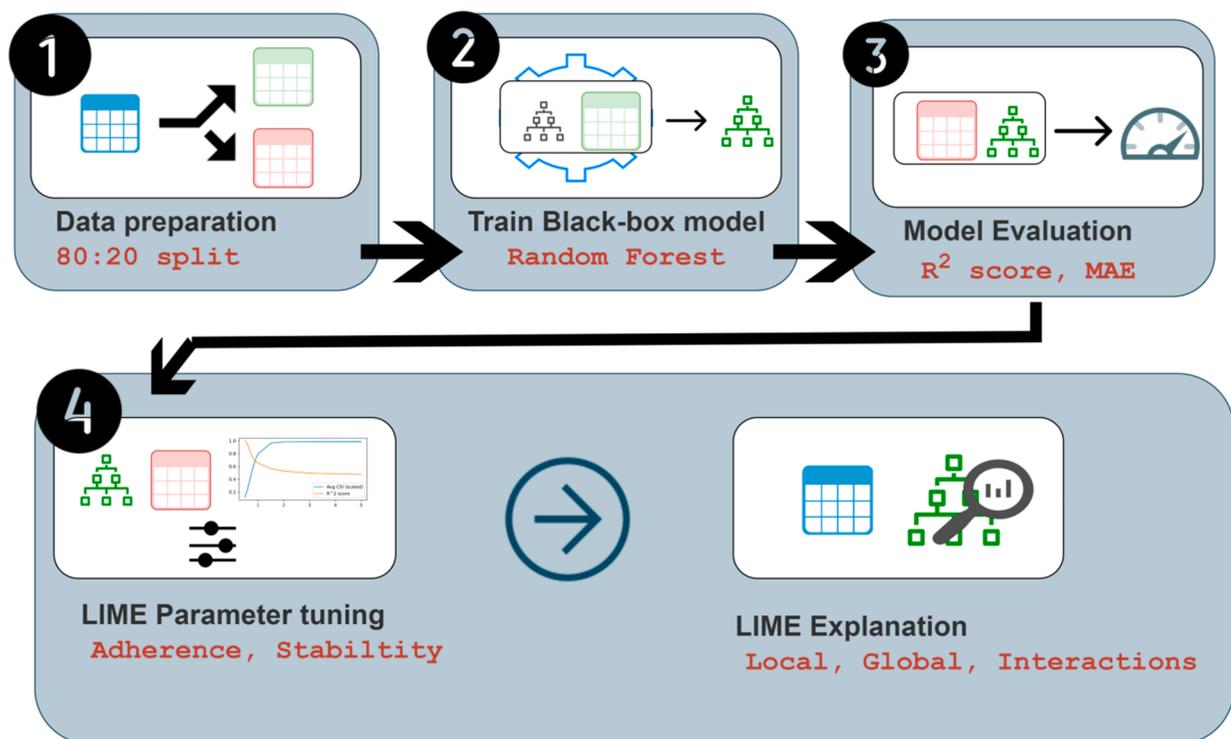


Figure 5. Flowchart for the steps undertaken in the experiment.

analyzing the adherence and stability of the local models on a data set of points. For a fixed kernel width, the stability of the local models over the data set is calculated by first measuring the coefficients stability index (CSI) of all the LIME calls on each of the instances in the data set. Section 3 contains details regarding the stability index and how it could be calculated. Then the average CSI value is calculated. And for adherence, the black-box model predictions are calculated over the entire data set. Then a set of predictions is made by using the local model of each of the instances to predict its drop size. Then the R^2 score is calculated between the black-box model predictions and the set of predictions. This process is repeated for all the different values of kernel width (k), and then an optimum kernel width is selected after analyzing the R^2 and avg CSI scores.

4.2. LIME Explanations: LIME could be used to locally explain one particular test instance, or it could be used to understand the overall behavior of the black-box model with respect to all the features over a data set. Algorithm 1 is used for explaining a single instance. In order to understand the model globally, a matrix of feature coefficients of each of the local models (one model per test instance) is created, as shown in Algorithm 2. In order to understand the feature interaction in the model predictions, Algorithm 3 is used. The LIME python package is used for the implementation. For Algorithm 3, we made some modifications to the existing LIME package.

5. RESULTS AND DISCUSSION

The random forest regression model is fitted in order to predict the target variable for our data set. We did not perform any hyper-parameter tuning and used the default hyper-parameters for the model provided in the scikit-learn python package. Several excellent papers in process engineer literature deal with hyper-parameter optimization. These include the following: (i) robust optimization of computationally expensive networks employ rigorous cross-validation methodology for hyperparameter tuning³⁴ and (ii) multiobjective optimization of cascaded mixed-suspension mixed-product removal crystallizers.³⁵ In the current work, the data set is randomly split into a training set and a testing set using an 80–20 split. We use R -squared (R^2) and Mean Absolute Error (MAE) as evaluation metrics to assess the performance of our model on the test set (Table 1). Predictions made by the model are explained by using LIME.

Table 1. Evaluation of the Random Forest Regression on the Test Data Set

Model	R^2	MAE
Random Forest	0.8831	0.3571

5.1. LIME Hyper-Parameter Tuning. Kernel width is the most important hyper-parameter to tune. It determines the size of the locality around an instance, which influences the weighted local linear model to explain the behavior of a black-box model. The default value of this hyperparameter is $0.75 \times \sqrt{\text{(number of features)}}$ in the LIME python package. The test data set is used for tuning the hyper-parameter.

For a selected value of kernel width, the CSI values for all the instances in the test data set are calculated, and then the average CSI value over all the instances is returned. A grid of kernel width (k) values is taken starting at 0.5, and the corresponding avg CSI and R^2 values are plotted in Figure 6.

The CSI index is defined to be between 0 and 100 but has been scaled between 0 and 1. The details of computation of the CSI and R^2 values are provided in section 4.

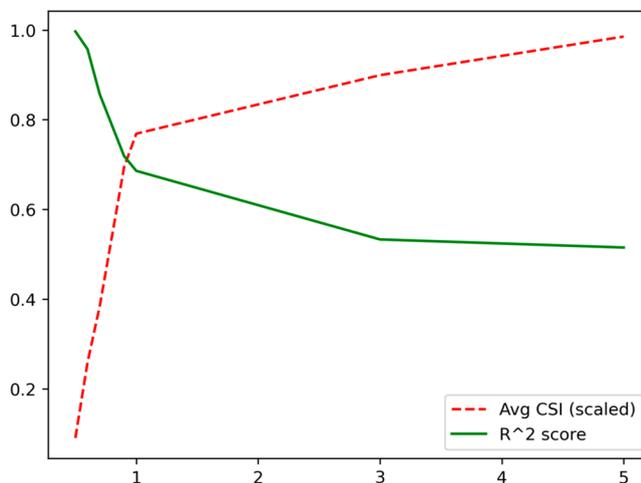
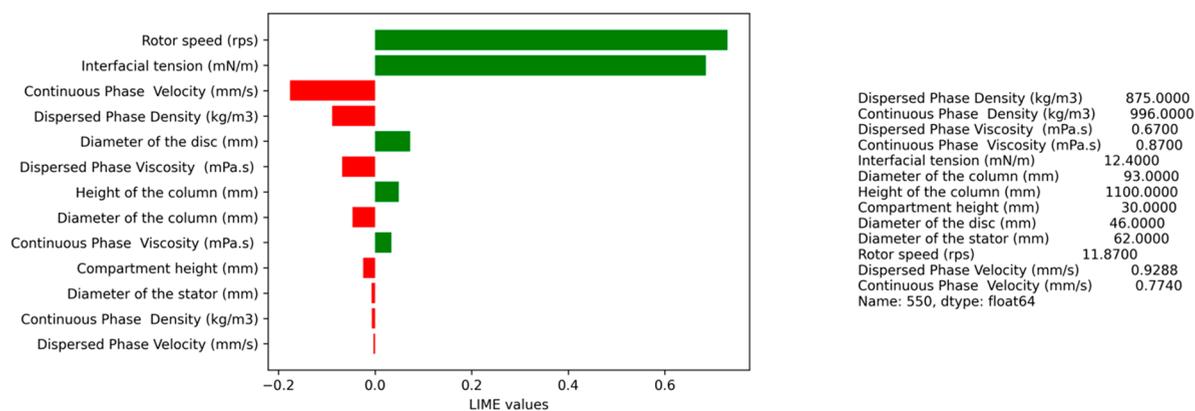


Figure 6. Effect of the kernel width on adherence and stability.

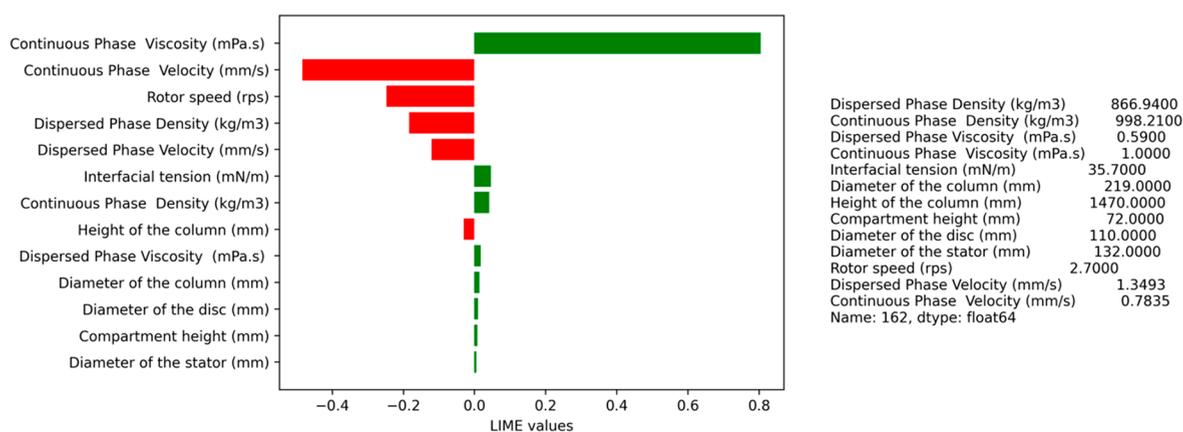
Figure 6 illustrates the effects of taking different k values. It is seen that there exists a trade-off between adherence and stability for a k value ≥ 0.5 . The selected k value is 0.9. This is appropriate considering we are trying to maximize both stability and adherence. The selected value is used for the computations.

5.2. LIME Explanations. By providing both local and global explanations, LIME can help users understand how a machine-learning model works in both general and specific instances. The local explanation for a given instance is given by the coefficient values (also known as the LIME values) of the local linear model. Explanations returned by LIME for 3 randomly selected instances from the test data set are shown in Figure 7. A positive coefficient means the corresponding feature has a positive impact, and a negative coefficient means the corresponding feature has a negative impact. For each of the instances, the values that all the features take are given on the right, and the LIME coefficient (importance) values are given on the left. It can be observed that the feature Rotor speed has a positive impact on instance “a”, while it has a negative impact on instances “b” and “c”. The value the feature Rotor speed takes (11.8) is significantly higher in instance “a” than the rest of the instances, instances “b” (4) and “c” (2.7). The impact of continuous phase viscosity is positive throughout the three instances, but the magnitude is lower in instance “a”, where the feature takes a lower value (0.87) compared to instance “b” and instance “c” (1). It could also be observed that for all the instances, not all the features contribute toward the black-box model prediction, the contributions are sparse, and the features with significantly large LIME values are few.

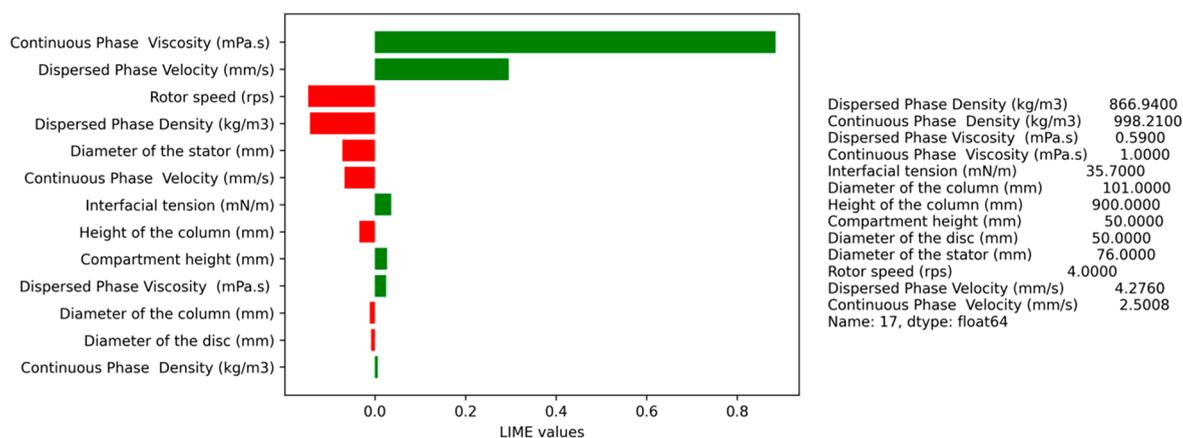
In the context of the LIME (Local Interpretable Model-Agnostic Explanations) framework, the identification of a positive or negative impact of a feature suggests that introducing slight perturbations (keeping other features fixed) to its value tends to lead to higher or lower predicted outcomes by the model, provided that the new data point remains in close proximity to the original instance. This observation is a consequence of LIME’s ability to provide a local linear approximation of the underlying complex model.



Instance a.



Instance b.



Instance c.

Figure 7. Local explanations for 3 test instances.

This observation is different from a feature's correlation with the model output. A feature's correlation with the model output reflects the general statistical relationship between that feature and the predicted outcomes across the entire data set. Correlation measures the strength and direction of the linear relationship between the feature and the model outcome independent of other features; it does not capture the local

behavior of the model around a specific instance. It is important to consider that the impact of slight perturbations on a feature value of any given instance, whether positive or negative, depends on the local context, which considers the values that the other features take for that instance. It is worth noting that even if two data points have the same value for a particular feature, they can still be significantly distant from

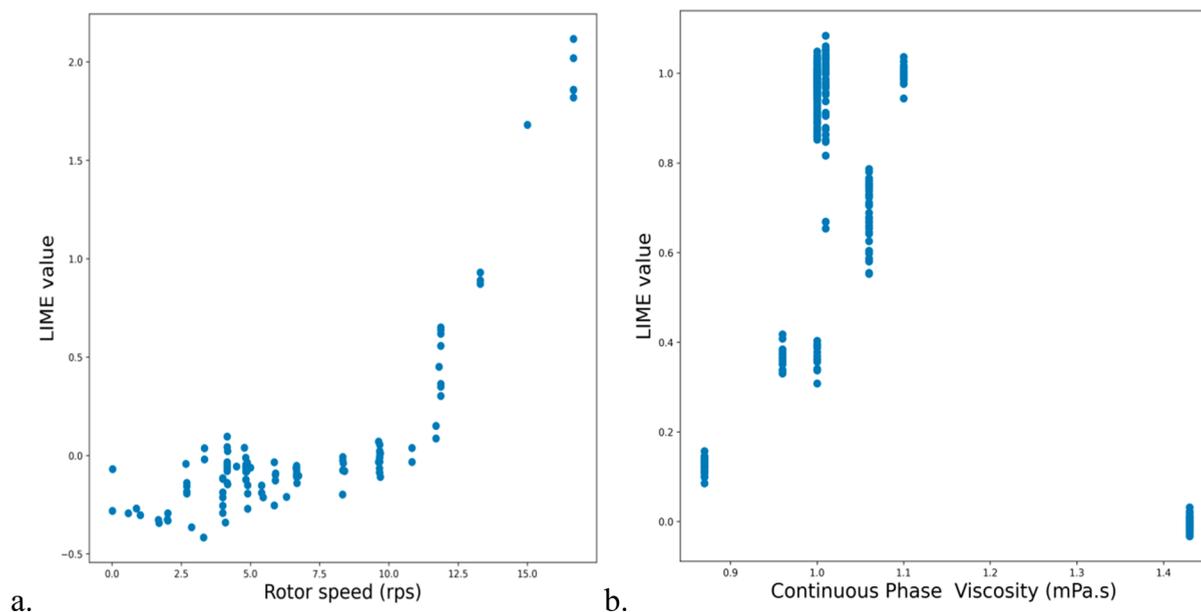


Figure 8. LIME scatterplot for features.

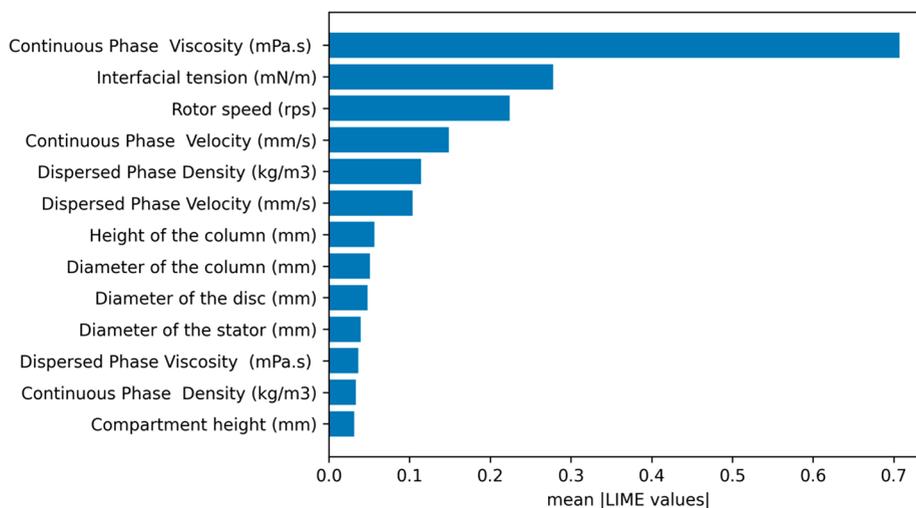


Figure 9. Global feature importance calculated for the random forest model predictions over the entire data set.

each other in the feature space. This may be due to the difference in values taken from the other features. In such scenarios, a single local surrogate model may not adequately capture the combined localities of these points. Instead, each data point will have its own local linear surrogate model, and the local model coefficients (LIME values) for that particular feature of these models may differ from each other and can even have impact in opposite directions.

In order to explain the model behavior globally, a large data set should be considered. We select the entire available data set; as the data set is larger from the data distribution, the better the global understanding of the model. We employ Algorithm 2 to get the LIME coefficient matrix for the data set. It contains the LIME values associated with each feature across all the instances in the data set. While comparing the local explanation of 3 instances, we observed that there might exist some interesting patterns of LIME values as the feature takes different values. The patterns can be well studied by plotting a LIME value scatterplot for the feature where each individual point is an instance with the feature value on the x -axis and the

LIME value on the y -axis. Figure 8a shows the LIME value scatterplot for rotor speed, and Figure 8b shows the scatterplot for continuous phase viscosity. Rotor speed shows an increasing trend in the LIME value with respect to the value of rotor speed in the entire data set. Notably, it is observed that for examples with lower rotor speed values, the feature's influence tends to have a negative impact. Our findings suggest that as we move toward the region of higher rotor speed values within the data distribution, the positive influence of rotor speed on the random forest model prediction strengthens.

On the other hand, when considering continuous phase viscosity, we note that the positive impact initially increases as we move toward higher values. However, beyond a certain threshold, a decline in a positive impact is observed, and in fact, the feature even exhibits a negative impact within the region characterized by higher continuous phase viscosity values.

These observations shed light on the intricate relationships between the features and model predictions. They suggest that the influence of the rotor speed becomes increasingly positive

as it approaches higher values within the data distribution. Meanwhile, the positive impact of continuous phase viscosity initially strengthens until a critical point, after which it undergoes a substantial decline, eventually even resulting in a negative impact within the region marked by higher values.

For calculating global feature importance, we compute the mean of the absolute LIME values of each feature across the entire data set. It is done by taking the column-wise mean of the absolute values from the LIME coefficient matrix. Figure 9 contains the global feature importance for explaining the random forest model. It shows the overall influence of a particular feature on the model in order to make predictions. The feature that is more important will have a greater impact on the model output. For the drop size data set, the most influential features for the model are continuous phase viscosity, rotor speed, and Interfacial tension. These features could have both positive and negative impacts on model prediction. The global feature importance plot indicates just the overall impact, while it may be possible to have examples where a particular feature impacts the model prediction positively in some cases and it may impact negatively in others.

The global feature importance values can be used for feature selection, as well. The relevant features according to LIME could be identified by referring to the global feature importance values obtained in Figure 9. Table 2 examines the random forest model performance after selecting subsets of the highly influential features according to LIME (selected from the top of Figure 9).

Table 2. Evaluation of the Random Forest Regression on the Test Data Set with Feature Selection

Features selected	R^2	MAE
3	0.785	0.479
5	0.770	0.480
6	0.811	0.424
7	0.881	0.365
All	0.882	0.361

As seen from Table 2, selecting the top 3 features and training the random forest model yield a decent R^2 of 0.78 over the test data set. As more features are added, the performance of the model increases. Selecting the top 7 features according to the global importance gives the same performance as the model trained with the entire set of features. The features with lower feature importance values do not influence the black-box model prediction by much, and hence, some of them could be dropped as shown in Figure 10. We can see that the model trained after removing up to 6 features gives a similar performance as the model created using all the features.

5.3. Capturing Black-Box Model Feature Interactions.

In section 3.7, we outlined an algorithm that incorporates second order feature interactions in the local surrogate linear model. We implemented the algorithm by slightly modifying the existing LIME Python package. The local model contains 91 features (including all of the second order interactions). The kernel width was optimized. In Figure 11, the LIME values for the local model for an instance are shown in the form of a heatmap. The coefficient values for each of the original features are represented by the diagonal entries, whereas the coefficient value for the interaction between features i and j is represented by the nondiagonal entry (i th row, j th column). For the given instance, the explanation is

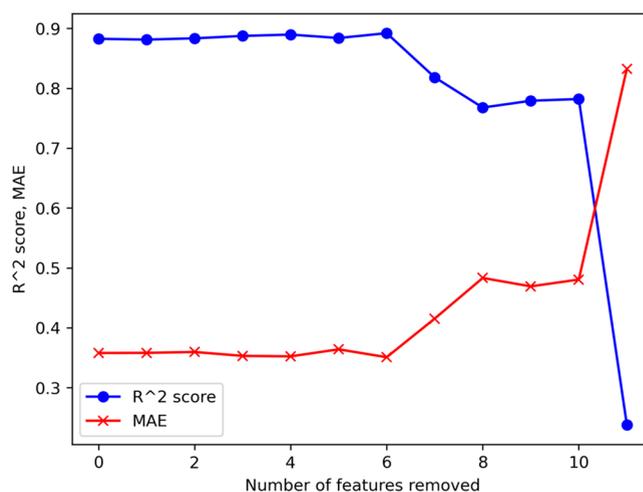


Figure 10. Impact on the model performance by removing the features with the lowest global importance one by one.

sparse, and the same has been observed in the local explanations in Figure 7. The features continuous phase viscosity (CPVisc) and interfacial tension (IT) are two features with a significant positive impact, and moreover, their interaction also has a positive impact.

As we have noted in section 3.7, a feature selection step in the local model is also employed to keep the explanation compact. In Figure 12, we create a local model by selecting 10 of the 91 features in order to explain an instance. From the figure it is apparent that the black-box is influenced by the feature interaction between continuous phase viscosity and interfacial tension, which are also the two features with the most impact for the black-box model prediction on the given instance. Their interaction has a positive contribution toward the prediction.

The global feature importance (including interactions) could be calculated just as the way it is done for Figure 9. We take the mean of absolute LIME values across the entire data set for each of the features of the local models; it includes the second order features. This is also shown as a bar plot in Figure 13. Only the top 9 features (including interactions) out of 91 are shown in the bar plot. Continuous phase viscosity and interfacial tension are the two features with the highest overall importance, which also have a slight interaction impact on the black-box model (7th best global feature importance). It is clear from the figure that the interactions between the top features (ranked 1,2,3) are more significant than the interactions between lower ranked features.

The top 8 features (including 1 interaction) ranked based on the global importance are 1. Continuous Phase Viscosity, 2. Interfacial Tension, 3. Rotor Speed, 4. Continuous Phase Velocity, 5. Dispersed Phase Velocity, 6. Dispersed Phase Density, 7. Interfacial Tension - Continuous Phase Viscosity, and 8. Height of the column. This ranked list clearly indicates that the black-box model is able to capture some feature interactions (seventh). We could also use the global importance for further feature engineering. The random forest model is influenced by the feature interaction between interfacial tension and Continuous phase viscosity. It may be useful to add a second order interaction feature between the two to the data set, as it may help to uncover more complex relationships involving these features.³⁶ We trained the random forest model on the train data set with the 1. Top 6 individual



Figure 11. Local explanation with feature interaction.

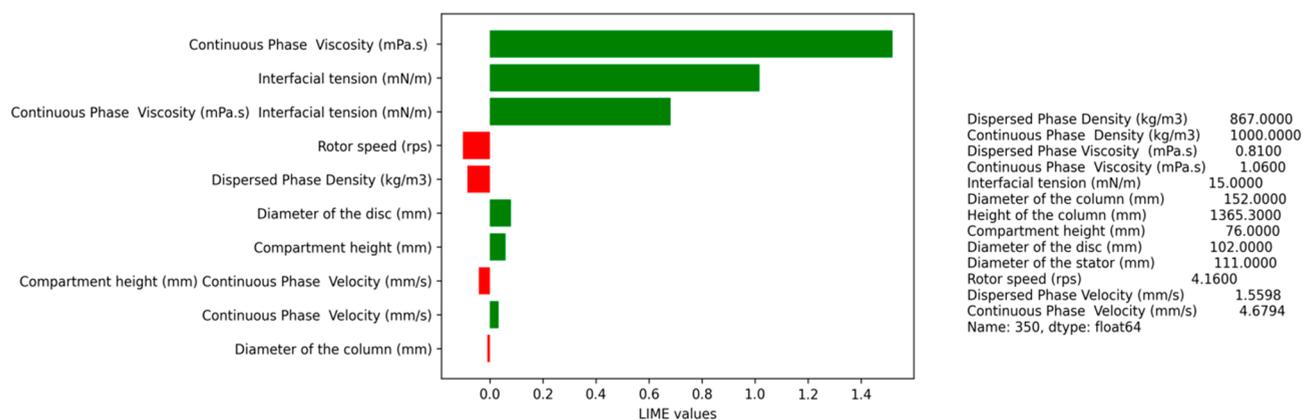


Figure 12. Local explanation with feature interaction with 10 features (including interaction terms) selected.

features (from the above list), 2. Top 7 features (including a second order interaction term), and 3. Top 8 features including second order interaction terms. Then we evaluated the models on the test data set.

From Table 3, it is clear that featuring second order interaction terms in the model did not improve the predictions on the test data set. However, it could be useful in general, and if inclusion of second order features is desired, an initial understanding of the feature interaction effect on the black-box model may be useful.

6. CONCLUSIONS

The estimation of the drop size in a rotating disc contactor is crucial for the effective design and operation of the contactor.

In this study, we have shown that the random forest model achieves high accuracy in predicting the drop size on our data set, with an R^2 score of 0.881 and a mean absolute error (MAE) of 0.3571. To ensure that the model is explainable and interpretable, we implemented LIME, a tool that provides explanations for the random forest model predictions on the local and global levels. At the local level, the utilization of bar plots with LIME values offers a valuable visual explanation to clarify the black-box model's predictions for specific instances. This visualization provides essential information by quantifying the relative contributions of different features toward the black-box model's prediction for each instance. On the global level, the analysis of global feature importance plots has yielded a comprehensive list of informative attribute subsets. Notably,

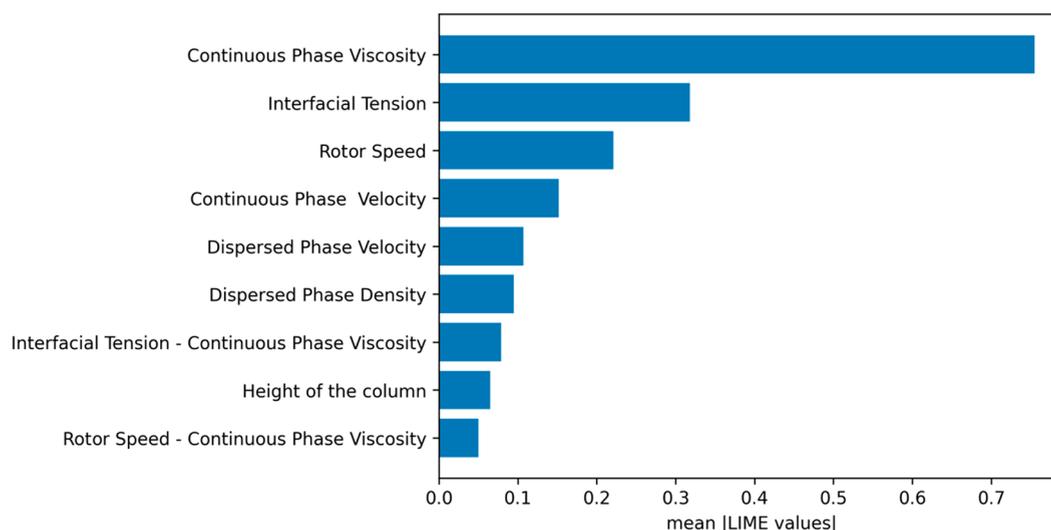


Figure 13. Global feature importance (including interactions) calculated for the random forest model predictions over the entire data set (Only the top 9 features are shown in the bar plot.).

Table 3. Evaluation of the Random Forest Regression on the Test Data Set with/without 2nd Order Feature

Features selected	R ²	MAE
6	0.81	0.41
7 (1 interaction)	0.81	0.41
8 (1 interaction)	0.88	0.35

when considering only the top 7 features identified by LIME global analysis in Figure 9, we observe a comparable performance to that of the original model (0.8831 R² score and 0.365 MAE). To delve deeper into the contributions of pairwise feature interactions on the random forest model's predictions for individual instances, we have incorporated second order feature interaction terms into the local surrogate model. This inclusion allows us to better comprehend the influence of these pairwise interactions. The revised list of top 8 informative attributes, as identified in Figure 13, also encompasses pairwise feature interactions, such as "Interfacial Tension - Continuous Phase Viscosity" and "Rotor Speed - Continuous Phase Viscosity". This discovery further reinforces the significance of these pairwise interactions in influencing the random forest model's predictions. Furthermore, we explicitly engineered second order interaction feature terms based on the top 8 features identified earlier. Interestingly, this approach achieved a level of performance similar to that of the original model, with an R² score of 0.88 and an MAE of 0.35. These results suggest that the random forest model is sufficiently capable of capturing such feature interactions, rendering the explicit addition of second order interaction features unnecessary for achieving further improvement. By demonstrating that the random forest model maintains its accuracy when trained on a smaller subset of features, we enhance our understanding of the relationship between the drop size and the relevant features. This analysis allows us to identify and discard irrelevant features that do not significantly contribute to the model's predictive performance. In conclusion, the use of a random forest model and the LIME can be useful in the design and optimization of rotating disc contactors.

AUTHOR INFORMATION

Corresponding Authors

Naren Rajan Parlikkad – School of Chemical & Biotechnology, SASTRA Deemed to be University, Tamil Nadu, India 613401; orcid.org/0000-0002-7589-0728; Email: prnaren@scbt.sastra.edu

Jayaraman Krishnamoorthy Valadi – Computing and Data Sciences, FLAME University, Pune, India 412115; Centre for Mathematical Modelling, FLAME University, Pune, India 412115; orcid.org/0000-0003-0185-9039; Email: valadi@gmail.com

Authors

Hardik Prabhu – Computing and Data Sciences, FLAME University, Pune, India 412115; Centre for Mathematical Modelling, FLAME University, Pune, India 412115

Aamod Sane – Computing and Data Sciences, FLAME University, Pune, India 412115

Renu Dhadwal – Computing and Data Sciences, FLAME University, Pune, India 412115; Centre for Mathematical Modelling, FLAME University, Pune, India 412115

Complete contact information is available at: <https://pubs.acs.org/10.1021/acs.iecr.3c00808>

Author Contributions

The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript. All the authors have contributed equally

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

N.R.P. was mentored and guided for his Ph.D. by Prof. Vivek V. Ranade during his stay at the National Chemical Laboratory, Pune, India during 2005–2009. Prior to it, N.R.P. was also taught by Prof. Vivek V. Ranade at the Institute of Chemical Technology, Mumbai. N.R.P. is ever

grateful to the research supervision and mentoring of Prof. Vivek V. Ranade. J.K.V. was a collaborator and a peer colleague of Prof. Vivek V. Ranade during his stay at the National Chemical Laboratory, Pune, India. J.K.V. wishes Prof. Vivek V. Ranade a long-healthy life on his 60th birthday and success at his endeavors at the University of Limerick.

■ ABBREVIATIONS

AARE	Average Absolute Relative Error
CSI	Coefficients Stability Index
LIME	Local Interpretable Model-Agnostic Explanations
MAE	Mean Absolute Error
RDC	Rotating Disc Contactor
RF	Random Forest
SHAP	SHapley Additive exPlanations
VSI	Variables Stability Index
XAI	eXplainable Artificial Intelligence

NOMENCLATURE

D	Euclidean distance
D_C	Column diameter, m
d_r	Rotor diameter, m
d_s	Stator diameter, m
f_b	Black-box model
f_l	Local surrogate model
g	Acceleration due to gravity, m/s^2
H	Column height, m
h_c	Compartment height, m
k	Kernel width
N	Rotor speed, rps
s_i	i^{th} instance in the neighborhood S around x
S	Generated neighborhood around instance x
V_c	Velocity of the continuous phase, m/s
V_d	Velocity of the dispersed phase, m/s
x	Instance to be explained

Greek symbols

μ_c	Viscosity of the continuous phase, Pa·s
μ_d	Viscosity of the dispersed phase, Pa·s
ρ_c	Density of the continuous phase, kg/m^3
ρ_d	Density of the dispersed phase, kg/m^3
σ	Interfacial tension, N/m

■ REFERENCES

- (1) Murugesan, T.; Regupathi, I. Prediction of Continuous Phase Axial Mixing in Rotating Disc Contactors. *J. Chem. Eng. Jpn.* **2004**, *37* (10), 1293–1302.
- (2) Saraswathi K, S.; Bhosale, H.; Oihal, P.; Parlikkad Rajan, N.; Valadi, J. K. Random forest and autoencoder data-driven models for prediction of dispersed-phase holdup and drop size in rotating disc contactors. *Ind. Eng. Chem. Res.* **2021**, *60*, 425–435.
- (3) Burkart, N.; Huber, M. F. A survey on the explainability of supervised machine learning. *J. Artif Intell Res.* **2021**, *70*, 245–317.
- (4) Vilone, G.; Longo, L. Notions of explainability and evaluation approaches for explainable artificial intelligence. *Information Fusion* **2021**, *76*, 89–106.
- (5) Ribeiro, M. T.; Singh, S.; Guestrin, C. Why should I trust you? Explaining the predictions of any classifier. *KDD '16: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*; 2016; pp 1135–1144.
- (6) Lundberg, S. M.; Lee, S. A unified approach to interpreting model predictions. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA*; , 2017; p 4768.

(7) Liu, X.; Aldrich, C. Explaining anomalies in coal proximity and coal processing data with Shapley and tree-based models. *Fuel* **2023**, *335*, 126891.

(8) Younis, R.; Ahmad, A.; Abu Al-Haija, Q. Explaining Intrusion Detection-Based Convolutional Neural Networks Using Shapley Additive Explanations (SHAP). *Big Data Cogn* **2022**, *6*, 126.

(9) Togo, M. V.; Mastroiorito, F.; Ciriaco, F.; Trisciuzzi, D.; Tondo, A. R.; Gambacorta, N.; Bellantuono, L.; Monaco, A.; Leonetti, F.; Bellotti, R.; Altomare, C. D.; Amoroso, N.; Nicolotti, O. TIREZIA: An eXplainable Artificial Intelligence Platform for Predicting Developmental Toxicity. *J. Chem. Inf Model.* **2023**, *63*, 56–66.

(10) Jang, K.; Pilario, K. E. S.; Lee, N.; Moon, I.; Na, J. Explainable Artificial Intelligence for Fault Diagnosis of Industrial Processes. *IEEE Trans. Industr. Inform.* **2023**, *1*.

(11) Fatahi, R.; Nasiri, H.; Homafar, A.; Khosravi, R.; Siavoshi, H.; Chelgani, S. C. Modeling operational cement rotary kiln variables with explainable artificial intelligence methods—a “conscious lab” development. *Part. Sci. Technol.* **2023**, *41*, 715.

(12) Nguyen, H. V.; Byeon, H. Prediction of Parkinson’s Disease Depression Using LIME- Based Stacking Ensemble Model. *Mathematics* **2023**, *11*, 708.

(13) Mardaoui, D.; Grreau, D. An analysis of LIME for text data. *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics, PMLR*; 2021; Vol. 130, pp 3493–3501.

(14) Balıkcı Çiçek, I.; Küçükakçali, Z.; Yağın, F. H. Detection of risk factors of PCOS patients with Local Interpretable Model-agnostic Explanations (LIME) Method that an explainable artificial intelligence model. *J. Cogn. Syst.* **2021**, *6*, 59–63.

(15) Jain, R.; Kumar, A.; Nayyar, A.; Dewan, K.; Garg, R.; Raman, S.; Ganguly, S. Explaining sentiment analysis results on social media texts through visualization. *Multimed Tools Appl.* **2023**, *82*, 22613.

(16) Sulieman, H.; Alzaatreh, A. A Supervised feature selection approach based on global sensitivity. *Archives of Data Science, Series A* **2018**, *5*, 1.

(17) Inapakurthi, R. K.; Miriyala, S. S.; Mitra, K. Deep learning based dynamic behavior modelling and prediction of particulate matter in air. *Chem. Eng. J.* **2021**, *426*, 131221.

(18) Dobbelaere, M. R.; Plehiers, P. P.; Van de Vijver, R.; Stevens, C. V.; Van Geem, K. M. Machine Learning in Chemical Engineering: Strengths, Weaknesses, Opportunities, and Threats. *Engineering* **2021**, *7*, 1201–1211.

(19) Schweidtmann, A. M.; Esche, E.; Fischer, A.; Kloft, M.; Repke, J. U.; Sager, S.; Mitsos, A. Machine learning in chemical engineering: A perspective. *Chemie Ingenieur Technik* **2021**, *93*, 2029–2039.

(20) Gandhi, A. B.; Joshi, J. B.; Jayaraman, V. K.; Kulkarni, B. D. Development of support vector regression (SVR)-based correlation for prediction of overall gas hold-up in bubble column reactors for various gas-liquid systems. *Chem. Eng. Sci.* **2007**, *62*, 7078–7089.

(21) Gandhi, A. B.; Gupta, P. P.; Joshi, J. B.; Jayaraman, V. K.; Kulkarni, B. D. Development of unified correlations for volumetric mass-transfer coefficient and effective interfacial area in bubble column reactors for various gas-liquid systems using support vector regression. *Ind. Eng. Chem. Res.* **2009**, *48*, 4216–4236.

(22) Zhu, L. T.; Chen, X. Z.; Ouyang, B.; Yan, W. C.; Lei, H.; Chen, Z.; Luo, Z. H. Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. *Ind. Eng. Chem. Res.* **2022**, *61*, 9901–9949.

(23) Ghalehchian, J. S. Prediction of Hydrodynamics of Rotating Disc Contactors Based on a New Monte-Carlo Simulation for Drop Breakage. *J. Chem. Eng. Jpn.* **2002**, *35*, 604–612.

(24) Ismail Al-Rahawi, A. M. New Predictive Correlations for the Drop Size in a Rotating Disc Contactor Liquid-Liquid Extraction Column. *Chem. Eng. Technol.* **2007**, *30*, 184–192.

(25) Korchinsky, W. J.; Loke, C. T.; Cruz-Pinto, J. J. C. Optimization of Drop Size in Countercurrent Flow Liquid - Liquid Extraction Columns. *Chem. Eng. Sci.* **1982**, *37*, 781–786.

(26) Olney, R. B. Droplet Characteristics in a Countercurrent Contactor. *AIChE J.* **1964**, *10*, 827–835.

(27) Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.

- (28) Simon, S. M.; Glaum, P.; Valdovinos, F. S. Interpreting random forest analysis of ecological models to move from prediction to explanation. *Sci. Rep.* **2023**, *13*, 3881.
- (29) Zhang, Y.; Zhang, X.; Razbek, J.; Li, D.; Xia, W.; Bao, L.; Mao, H.; Daken, M.; Cao, M. Opening the black box: interpretable machine learning for predictor finding of metabolic syndrome. *BMC Endocr. Disord.* **2022**, *22*, 214.
- (30) Bauer, E.; Kohavi, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Mach Learn* **1999**, *36*, 105–139.
- (31) Breiman, L. Bagging predictors. *Machine learning* **1996**, *24*, 123–140.
- (32) Visani, G.; Bagli, E.; Chesani, F. OptiLIME: Optimized LIME explanations for diagnostic computer algorithms. *Proceedings of the CIKM 2020 Workshops, 2020, October 19–20, Galway, Ireland*.
- (33) Visani, G.; Bagli, E.; Chesani, F.; Poluzzi, A.; Capuzzo, D. Statistical stability indices for LIME: Obtaining reliable explanations for machine learning models. *J. Oper. Res. Soc.* **2022**, *73*, 91–101.
- (34) Miriyala, S. S.; Mittal, P.; Majumdar, S.; Mitra, K. Comparative study of surrogate approaches while optimizing computationally expensive reaction networks. *Chem. Eng. Sci.* **2016**, *140*, 44–61.
- (35) Inapakurthi, R. K.; Naik, S. S.; Mitra, K. Toward Faster Operational Optimization of Cascaded MSMPR Crystallizers Using Multiobjective Support Vector Regression. *Ind. Eng. Chem. Res.* **2022**, *61*, 11518–11533.
- (36) Ibidoja, O. J.; Shan, F. P.; Sulaiman, J.; Ali, M. K. M. Robust M-estimators and Machine Learning Algorithms for Improving the Predictive Accuracy of Seaweed Contaminated Big Data. *J. Niger. Soc. Phys. Sci.* **2023**, 1137–1137.