



OPEN

Integration of machine learning and CFD for modeling mass transfer in water treatment using membrane separation process

Kamal Y. Thajudeen¹✉, Mohammed Muqtader Ahmed² & Saad Ali Alshehri¹

This study aims to assess the efficacy of machine learning models in predicting solute concentration (C) distribution in a membrane separation process, using the input parameters which are spatial coordinates. Computational fluid dynamics (CFD) was employed with machine learning for simulation of process. The models evaluated include Kernel Ridge Regression (KRR), Radius nearest neighbor regression (RNN), K-nearest neighbors (KNN), LASSO, and Multi-Layer Perceptron (MLP). Additionally, Harris Hawks Optimization (HHO) was utilized to fine-tune the hyperparameters of these models. Leading the way is the MLP model, which achieves a remarkable test R^2 value of 0.98637 together with very low RMSE and MAE values. Strongness and generalization capacity are shown by its consistent performance on test and training datasets. To conclude, the effectiveness of using machine learning regression methods more especially, KRR, KNN, RNN, LASSO, and MLP in estimating concentration from spatial coordinates was demonstrated in this work. For separation science via membranes where predictive modeling of spatial data is essential, the results offer important new perspectives by developing hybrid model.

Keywords Membranes, Mass transfer, CFD, Multi-layer perceptron, Harris Hawks optimization

Membrane process is one of the recently developed separation systems which has been developed over the last decades for separation of liquid and gas components. The primary usage of this process was for water treatment, but it has obtained wide applications in separation and reaction applications¹⁻³. For developing membrane process, advanced materials have been developed with tailored separation and operational properties. For nonporous membranes, the separation mechanism is mainly driven by solution-diffusion, while other mechanisms govern transport through porous membranes^{4,5}.

For membrane development, one needs to apply computational methods to simulate the separation process and understand the underlying parameters and their influence on the separation efficiency. Different computational models can be used and tested for membrane separation including computational fluid dynamics (CFD), molecular modeling, thermodynamics, machine learning, etc⁶⁻⁸. The major modeling strategy is CFD which is challenging and computationally expensive to implement for membrane separation, so other modeling techniques such as machine learning and integrated models are preferable to overcome this problem. For evaluating mass transfer and separation efficiency in the process, hybrid models can be also adopted and tuned for membrane separation by which the process is simulated to obtain concentration distribution of solute entire system⁹⁻¹¹. The hybrid strategy can take into account both major modeling approaches including CFD and machine learning so that the advantages of both methods can be benefited in the hybrid model. Feeding and training machine learning models via CFD data can be suggested as novel method for simulation of membrane separation. This approach can be performed by design and execution of advanced learning and optimization algorithms.

Accurate prediction of concentration in scientific and engineering applications is paramount for understanding complex systems and making informed decisions. This is also a major issue in membrane separation which can be addressed by using machine learning based models. Machine learning models have become highly regarded in process engineering due to their impressive capabilities for process prediction, classification, regression, and optimization. Machine Learning (ML) has sparked a revolution in various scientific disciplines by offering robust

¹Department of Pharmacognosy, College of Pharmacy, King Khalid University, Abha 62529, Saudi Arabia.

²Department of Pharmaceutics, College of Pharmacy, Prince Sattam Bin Abdulaziz University, P.O. Box 173, AlKharj 11942, Saudi Arabia. ✉email: kthajudeen@kku.edu.sa

methods for analyzing and modeling data^{12,13}. In this study, the effectiveness of various machine learning models in predicting concentration of solute in a membrane process based on spatial coordinates is evaluated. Models evaluated are Multi-Layer Perceptron (MLP), LASSO, K-Nearest Neighbors (KNN), Radius Nearest Neighbor Regression (RNN), and Kernel Ridge Regression (KRR). Hyper-parameter optimization is conducted using the innovative Harris Hawks Optimization (HHO) technique, known for its efficiency in fine-tuning model parameters.

KRR is a method that combines Ridge regression leveraging kernels structure. The algorithm acquires knowledge of a linear relationship within the space created by the specific kernel and the provided data. In essence, it achieves a balance between bias and variance by integrating regularization and nonlinearity using the kernel function¹⁴. KNN estimates the target variable using the values of its nearest neighbors in the training dataset. Nevertheless, it is susceptible to noisy data and necessitates meticulous adjustment of the number of neighbors (K). RNN, like KNN, makes predictions for the target variable by considering neighboring data points. However, rather than a predetermined quantity of neighbors, it takes into account all neighbors located within a specified radius. A particular kind of artificial neural network (ANN) with several layers is called a Multilayer Perceptron, or MLP. It is able to mimic complex connections by obtaining hidden representations. When a large enough number of hidden units are used, multi-layer perceptron (MLPs) can approximate any function with great degree of flexibility¹⁵. Mitigating overfitting requires regularization techniques. LASSO is a method that combines linear regression with L1 regularization. It promotes sparsity by imposing penalties on the absolute values of regression coefficients. The LASSO method is valuable for selecting features and is capable of handling data with a high number of dimensions¹⁶.

The selection of specific machine learning models such as Multi-Layer Perceptron (MLP), K-Nearest Neighbors (KNN), Radius Nearest Neighbor (RNN), Kernel Ridge Regression (KRR), and LASSO for this study is grounded in their unique strengths and applicability to the context of membrane separation. MLP, with its deep learning architecture, is capable of capturing complex nonlinear relationships, making it suitable for predicting intricate solute concentration variations. KNN and RNN offer simplicity and robustness in handling non-linear data by considering neighboring data points, while KRR effectively combines ridge regression with the kernel trick to manage nonlinearity and regularization. LASSO, known for its feature selection capabilities, aids in handling high-dimensional data by promoting sparsity. The Harris Hawks Optimization (HHO) algorithm was chosen for hyper-parameter tuning due to its superior exploration and exploitation capabilities compared to other optimization methods. Together, these methods provide a comprehensive approach to accurately model and predict concentration distributions in membrane separation processes, leveraging both their individual and collective advantages.

For professionals and researchers in fields such as engineering, chemistry, and environmental science, where accurate predictive modeling of spatial data is crucial, the findings of this research offer novel perspectives and practical implications. By demonstrating the efficacy of ML regression techniques in concentration estimation, this study contributes to advancing predictive modeling techniques in membrane science. The hybrid models in this work are built via utilization of both CFD and ML models in prediction of concentration distribution in a membrane process which is membrane contactor. The hybrid models developed in this work based on these machine learning models (MLP, KNN, RNN, KRR, and LASSO), and optimization method are developed for the first time in simulation of water treatment-based membrane separation. Potential predictive models can be derived from the successful instances of this study conducted on similar datasets by other researchers.

Process description

We generated a dataset for learning ML algorithms, and the dataset was obtained from CFD (computational fluid dynamics) simulation of a membrane contactor for liquid-phase separation. A geometry for hollow-fiber membrane contactor was considered, and the governing mass and momentum equations were solved via finite element technique as described in¹⁷⁻¹⁹. Steady-state and laminar flow conditions were considered for modeling the process. The meshed geometry of the membrane process analyzed in this work for liquid separation is illustrated in Fig. 1. From the left to the right side are feed side, membrane, and shell channel, respectively.

The dataset comprises over 25,000 data points, featuring r and z coordinates alongside the concentration (C) of a substance in moles per cubic meter (mol/m³). Figure 2 displays violin plots illustrating the distributions of these variables. The red-violin plot represents the r coordinates, the sky blue-violin plot represents the z coordinates, and the blue-violin plot represents the substance concentration. These visualizations offer insights into the variability and distribution of each variable. Figure 3 presents a correlation heatmap, revealing the relationships between r, z, and C. The heatmap aids in identifying any significant correlations between these variables, providing valuable insights into their interdependencies, facilitating further analysis and modeling endeavors. It should be pointed out that both Figs. 2 and 3 refer to the data before outlier removal.

In this study, outlier detection was conducted using the Isolation Forest method, a popular technique for identifying anomalies in datasets. The Isolation Forest algorithm operates by segregating observations within a dataset through the random selection of a characteristic and subsequently choosing a division point between the highest and lowest values of that characteristic. This process is iteratively repeated until all observations are completely separated. Outliers, due to their lower frequency and distinct values, are anticipated to necessitate fewer divisions in order to be separated from normal data points. By measuring the number of splits needed to isolate each observation, Isolation Forest assigns anomaly scores, with lower scores showing outliers. This approach provides various benefits, including scalability to large datasets and the ability to handle mixed data types²⁰. 2.1% of the data points were eliminated during the process of removing outliers. This is achieved by conducting an initial evaluation of the model with and without outliers, revealing that the removal of these 2.1% of data points results in a 4% increase in the mean R² score.

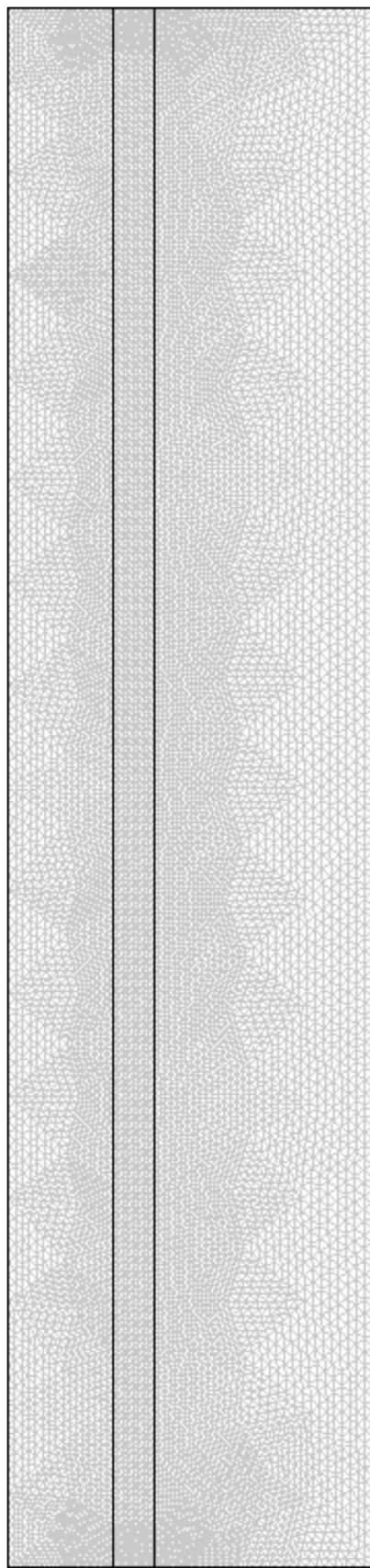


Fig. 1. Geometry of the contactor meshed for CFD computations.

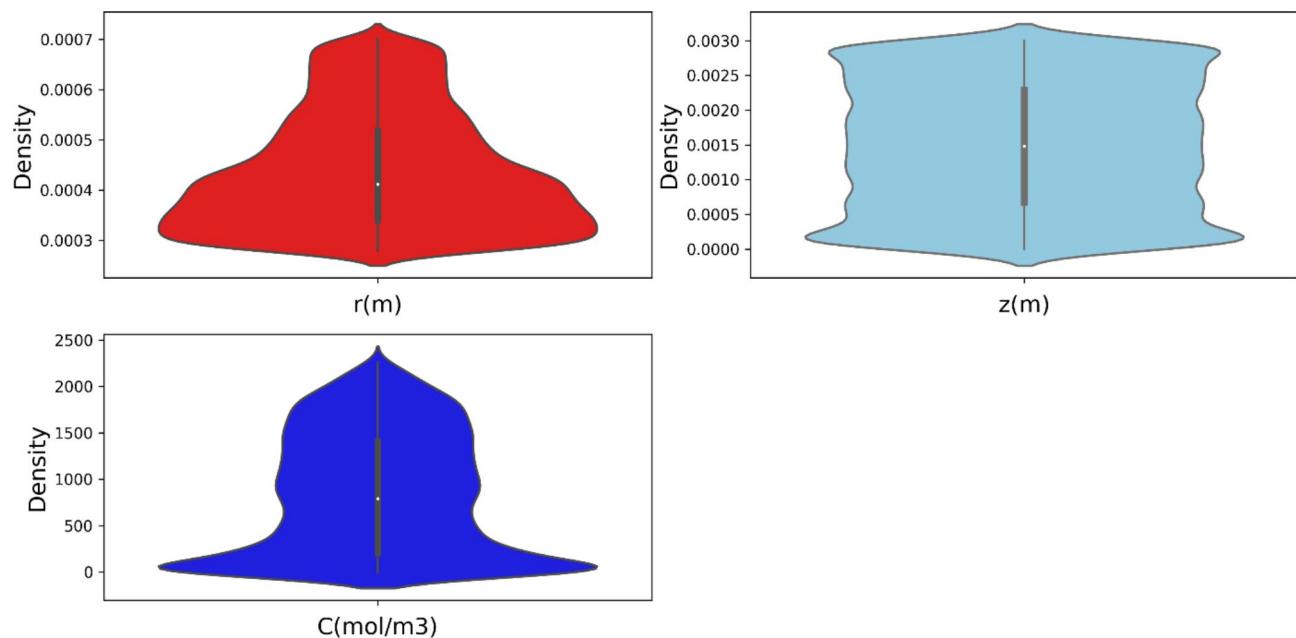


Fig. 2. Violin plot of membrane separation variables combining aspects of a box plot and a density plot to show both the central tendency and variability. The white point indicates the median for each variable.

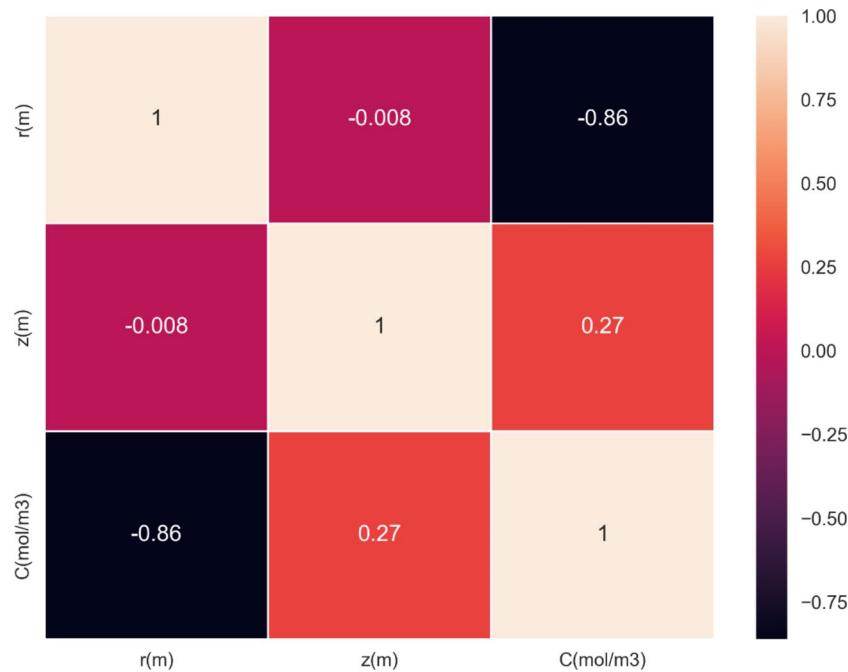


Fig. 3. Correlation Heatmap illustration.

Theoretical methods

Harris Hawks optimization (HHO)

The HHO algorithm draws inspiration from the cooperative hunting behavior exhibited by Harris hawks, a species commonly found in desert regions. These hawks are renowned for their coordinated hunting strategies, where they scout the desert landscape for small prey such as rabbits and then employ a collaborative hunting approach to capture their target. This method involves extracting the prey from its natural habitat, thereby facilitating its exhaustion and eventual capture. This hunting tactic is seen by the HHO algorithm as two-step: Exploration gave place to exploitation²¹. HHO tries to find the best solutions in the solution space during the exploration phase. Subsequently, in the exploitation phase, the algorithm refines its search, focusing on regions

identified during exploration to converge towards the optimal solution. Specific information regarding these phases is as follows:

- **Exploitation Phase:** During this phase, the hawk moves towards the prey by either shrinking the encircling mechanism or following an upward spiral-shaped path. The position of the hawk is updated according to the following Eqs^{21,22}:

$$X_{\text{next}} = \begin{cases} X_{\text{best}} - A \cdot |C \cdot X_{\text{best}} - X_c| & \text{if } r_2 < 0.5 \\ |X_{\text{best}} - X_c| \cdot e^{bl} \cdot \cos(2\pi l) + X_{\text{best}} & \text{if } r_2 \geq 0.5 \end{cases}$$

In recent equations, $A = 2 \cdot a \cdot r_1 - a$, $C = 2 \cdot r_1$, a decreases from 2 to 0, b denotes the spiral shape, l ranges from -1 to 1. Also, r_1 and r_2 are random values within the range of 0 to 1.

- **Exploration Phase:** In this phase, a random position X_{rand} is utilized to modify the position of the hawks. The vector A comprises random values ranging from -1 to 1, which compel a solution to steer away from the current optimal solution:

$$X_{\text{next}} = X_{\text{rand}} - A \cdot |C \cdot X_{\text{rand}} - X_c|$$

These equations dictate how the hawks adapt their positions during the optimization process to balance exploration and exploitation effectively, ensuring that they converge towards the optimal solution.

Notably, the original performance of HHO surpasses that of other well-known optimization methods such as MFO, PSO, GWO, DE, and WOA owing to its robust architecture. Unlike many swarm-based optimizers that need additional parameterization²³, HHO works effectively with only the swarm's initial population. Furthermore, HHO does not rely on derivative information, making it versatile and user-friendly. With each successive run, HHO's exploratory capabilities are enhanced, making it increasingly adept at tackling real-world problems. Consequently, HHO and its variants have garnered significant attention as effective solutions for a variety of practical applications²⁴.

Consider a scenario where one wants to find the highest peak in a mountainous region. Traditional methods might slowly climb the mountain, but HHO simulates hawks scouting from above, quickly diving towards promising peaks and collaborating to find the highest point more efficiently. This method helps in solving complex optimization problems by rapidly converging on the best solution.

To ensure robust and reliable optimization, we utilized the 3-fold mean R^2 score of each model as the objective function. This method entailed dividing the entire dataset into three folds, training the model on two folds, and validating it on the other fold. The procedure was iterated three times, each time with a different fold as the validation set, and the mean R^2 score across these folds was calculated. By using the 3-fold mean R^2 score as the objective function, we aimed to achieve a balanced evaluation that accounts for the model's performance variability and enhances its generalization capability. This methodology provided a comprehensive assessment of each model's predictive accuracy, guiding the HHO algorithm in identifying optimal hyper-parameter configurations. Also, HHO Parameters set to be as (The values chosen following thorough testing of several distinct combinations of the values):

- Population size: 30 hawks.
- Number of iterations: 100.

K-nearest neighbors (KNN)

KNN regression operates as a non-parametric algorithm, adopting a lazy learning approach to predict the output for a given test instance. Its principle lies in gauging similarity with neighboring data points^{25,26}. The prediction for an unknown test data point is computed using the equation²⁷:

$$\bar{y}_i = \frac{1}{\sum_{j=1}^k W_{ij}} \sum_{j=1}^k W_{ij} y_j^i$$

Here, the predicted value label (prediction) for the query point, denoted by \hat{y}_i , corresponds to the outcome anticipated for the particular data point. Typically, W_{ij} is computed as follows:

$$W_{ij} = \frac{1}{\sqrt{\sum_{i=1}^n (z_i - y_i)^2}}$$

where z_i and y_i denote the target and reference units, respectively.

Radius nearest neighbor regression (RNN)

Radius nearest neighbor regression (RNN) operates akin to KNN but diverges in determining neighboring elements by a fixed radius span R rather than by the k parameter. In this algorithm, R serves as an additional parameter to define²⁸. The response parameter is calculated by locally interpolating the nearest neighbor elements from the training portion within a specified radius R .

Working Principle of RNN can be summarized in following items:

1. Definition of Radius (R): The radius (R) is a critical hyperparameter in RNN. It defines the maximum distance within which data points are considered neighbors. The choice of R directly influences the model's performance, as it determines the number of neighbors used for prediction.
2. Neighbor Selection: For a given test point, the RNN algorithm identifies all training data points that lie within the radius R from the test point. This is achieved by calculating the Euclidean distance (or other distance metrics) between the test point and all training points.
3. Local Interpolation: Once the neighbors within the radius are identified, the prediction for the test point is made by averaging the target values of these neighbors. If no neighbors are found within the radius, it may be necessary to adjust the radius or handle such cases explicitly in the implementation.
4. Weighting Schemes: Similar to KNN, RNN can incorporate different weighting schemes to give more importance to certain neighbors. Commonly, inverse distance weighting is used, where closer neighbors have a higher influence on the prediction than those farther away.

Multi-layer Perceptron (MLP)

Feed-forward neural networks, also known as Multilayer Perceptron (MLPs), are composed of multiple hidden layers. Backpropagation is a commonly used method for training Multi-Layer Perceptron (MLPs) which utilizes the Error-Correction learning model. This rule entails the iterative adjustment of model parameters by moving in the opposite direction of the instantaneous slope, as determined by the error function^{11,29,30}.

The MLP architecture (Fig. 4) starts with an input layer and progresses through hidden layers until producing an output. These hidden layers optimize solver functions, activation functions, and the number of hidden layers. For a model with one hidden layer and a single output, the output can be formulated as^{11,30}:

$$\tilde{y} = \delta_2 \left(\sum_{i=1}^m \left(w_i^{(2)} \delta_1 (X) \right) + b^{(2)} \right) X = \sum_{j=1}^n \left(x_j w_{x_j}^{(1)} \right) + b^{(1)}$$

where, \tilde{y} indicates the model estimation vector for MLP, m denotes the count of neurons in the hidden layer, n signifies the count of input features, and x_j indicates the quantity of vectors in the dataset. $w^{(2)}$ represents the weights between the hidden and the output layers, whereas $w^{(1)}$ denotes the weights between the input features and the hidden layer. The output layer employs the activation function δ_2 , whereas the hidden layer utilizes the activation function δ_1 for its neurons. The hidden layers and the output layer each have their own bias vectors, denoted as $b^{(2)}$ and $b^{(1)}$ respectively¹¹.

Kernel Ridge regression (KRR)

A regularization approach called Kernel Ridge Regression (KRR) combines the advantages of kernel techniques with ridge regression. It shows to be especially helpful to address nonlinear connections between target outputs and the inputs³¹.

In a training dataset containing input features X and corresponding target values y , KRR aims to find a function $f(x)$ that minimizes the following objective function¹⁴:

$$J(f) = \frac{1}{2} (|y - f(X)|^2 + \alpha |f|^2)$$

where α is the regularization parameter and f represents the vector of function values at the training points.

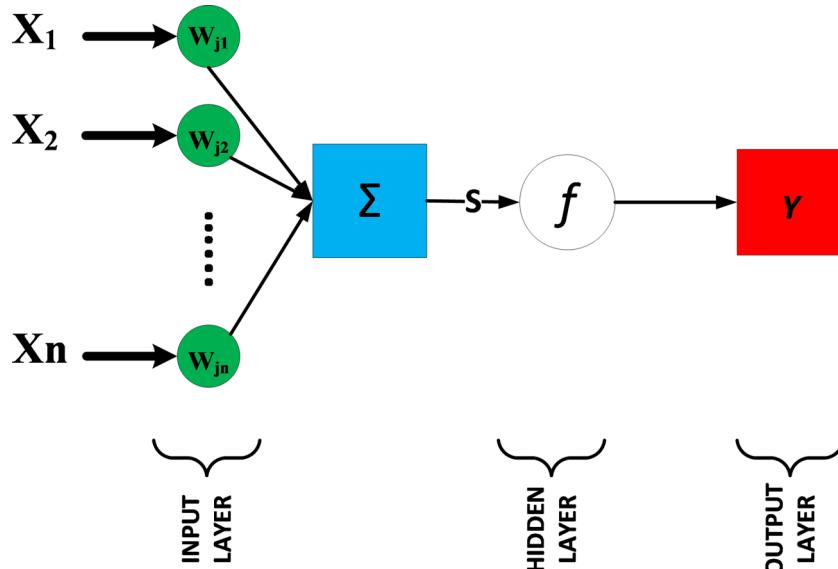


Fig. 4. The structure of MLP model.

Model	R ² Score	RMSE	MAE
KRR	0.95695	1.30482E+02	7.33020E+01
KNN	0.91238	1.59214E+02	1.31357E+02
RNN	0.90135	1.75017E+02	1.08226E+02
LASSO	0.76416	2.84738E+02	2.18471E+02
MLP	0.98664	7.55937E+01	3.93855E+01

Table 1. Train results.

Model	R ² Score	RMSE	MAE
KRR	0.95795	1.28837E+02	7.27184E+01
KNN	0.91176	1.59392E+02	1.32424E+02
RNN	0.90331	1.73007E+02	1.07858E+02
LASSO	0.76756	2.83712E+02	2.17855E+02
MLP	0.98637	7.61288E+01	3.98704E+01

Table 2. Test results.

Model	Mean R ² Score	Standard Deviation
KRR	0.95515	0.00072
KNN	0.90692	0.02163
RNN	0.89269	0.00085
LASSO	0.751585	0.00226
MLP	0.97832	0.00045

Table 3. 3-fold validation results.

The main concept of KRR is to use a kernel function to imply link the input features with a higher-dimensional space. Without specifically computing the transformed features, the kernel function estimates the inner product between the transformed feature vectors, enabling KRR to capture complex relationships. Among the often-used kernels are the linear, polynomial, and Gaussian (RBF) kernels.

LASSO

A kind of linear regression called LASSO regression seeks to use regularization to avoid overfitting and encourage model simplicity. Whereas OLS regression reduces the total squared residuals, LASSO adds a penalty term to the objective function according to the coefficients' absolute values. By promoting sparsity in the coefficient vector, this penalty essentially makes some coefficients zero¹⁶.

In situations where feature selection or dimensionality reduction are crucial, LASSO sparsity has major consequences. LASSO automatically selects a subset of original features during model fitting by shrinking coefficients to zero. Highlighting the most important predictors simplifies and improves model interpretation.

LASSO's sparse solutions make it ideal for high-dimensional data settings like genomics, text mining, and image processing, where the number of predictors exceeds the quantity of observations. In such cases, LASSO can identify relevant features and ignore irrelevant or redundant ones, improving model performance and generalization.

LASSO regularization has advantages, but it requires a tuning parameter (λ) to control the strength of regularization applied to coefficients. The selection of λ is crucial and often determined through cross-validation.

In this study, the polynomial features for this model were also tested in an initial evaluation. Because there was no discernible difference between the models, the raw version was selected for the final report in this research.

Results and discussion

The performance of five regression models—KRR, KNN, RNN, LASSO, and MLP—optimized with Harris Hawks optimization (HHO)—was assessed for predicting a substance's concentration (C) based on radial (r) and axial (z) coordinates. Over 25.000 data points made up the dataset. The results of statistical analyses for the models are listed in Tables 1 and 2 for training and testing steps, respectively. Also, k-fold validation results are listed in Table 3. As seen, three fitting criteria have been considered to compare the models for fitting the membrane dataset. The identified hyperparameters for each model are as follows:

- Kernel Ridge Regression (KRR):

Alpha: 9.9383526416

Kernel: polynomial
Gamma: 0.93028983658

- K-Nearest Neighbors (KNN):

Number of Neighbors: 2
Weights: distance
Algorithm: brute force

- Radius nearest neighbor regression (RNN):

Radius: 0.20315210
Weights: Uniform
Algorithm: brute force

- Lasso Regression:

Fit Intercept: True
Tolerance: 0.0003234
Alpha: 1.079

- Multilayer Perceptron (MLP) Regressor**:

Hidden Layer Sizes: 293
Activation Function: Tanh
Solver: sgd
Tolerance: 0.01386269999999999

These hyperparameters were determined to optimize the performance of each regression model, ensuring high predictive accuracy and generalization ability. The optimization process leveraged HHO's robust exploration and exploitation capabilities to effectively search the hyperparameter space and avoid local minima, leading to more reliable and accurate models. Numerical values from the tables derived from the training and test datasets are randomly selected in an 80% to 20% ratio.

Among the models evaluated, the MLP model exhibits outstanding performance in both test and train datasets, with the highest R^2 score of 0.98637 on the test data. This indicates that the MLP model captures a significant portion of the variance in the dataset, providing highly accurate predictions of the substance concentration. Furthermore, it demonstrates the lowest RMSE and MAE values, indicating superior precision and generalization ability compared to other models. Figure 5 displays a three-dimensional representation of the concentration modeled by MLP as a function of coordinates. It is useful to see how solute concentration changes in the shell side of membrane contactor. The main concentration variations are observed in the radial coordinate which is attributed to the molecular diffusion in the radial direction through the shell side of membrane contactor.

KRR also shows strong performance, with an R^2 of 0.95795 on the test data, closely following MLP. Despite slightly higher RMSE and MAE values compared to MLP, KRR still provides reliable predictions of substance concentration. Figure 6 depicts a three-dimensional visualization of the concentration predicted by KRR based on the coordinates.

KNN and RNN perform reasonably well, with R^2 scores around 0.91 on the test data. However, they exhibit higher RMSE and MAE values, indicating slightly lower precision in their predictions compared to MLP and KRR. Figures 7 and 8 depict a three-dimensional visualization of the concentration predicted by KNN and RNN models, based on the variables r and z .

The LASSO model, in contrast, exhibits the lowest performance among the models evaluated, as evidenced by its notably lower R^2 scores and higher RMSE and MAE values on both the training and test datasets. This suggests that the linear nature of LASSO might not adequately capture the complex relationships present in the dataset. Figure 9 depicts a three-dimensional visualization of the concentration predicted by LASSO based on the coordinates.

All things considered, the findings show how well MLP and KRR models (optimized with HHO) predict material concentration using $r(m)$ and $z(m)$ coordinates. Figures 10 and 11 are the impacts of inputs on concentration values using MLP model and Fig. 12 is the concentration distribution calculated by CFD model. It is seen that the mass transfer boundary layer has been formed in the shell side of membrane contactor and the model can predict it.

The MLP demonstrated strong performance in capturing complex nonlinear patterns (Parity Plots for this model is displayed in Fig. 13), but it required substantial computational resources and tuning efforts. KNN and RNN, while simpler and more intuitive, were less effective in handling high-dimensional data and exhibited sensitivity to the choice of the number of neighbors. KRR offered a good balance between performance and complexity, effectively managing nonlinearity and regularization but struggled with large datasets due to computational intensity. LASSO excelled in feature selection, promoting model interpretability by reducing dimensionality, but it sometimes led to oversimplification by excluding relevant features.

The HHO algorithm showed superior optimization capabilities, ensuring robust hyper-parameter tuning across all models, though its performance was contingent on proper parameter settings. These insights illustrate the necessity of choosing appropriate models and tuning methods based on the specific requirements and constraints of the membrane separation process, advocating for a tailored approach in practical applications.

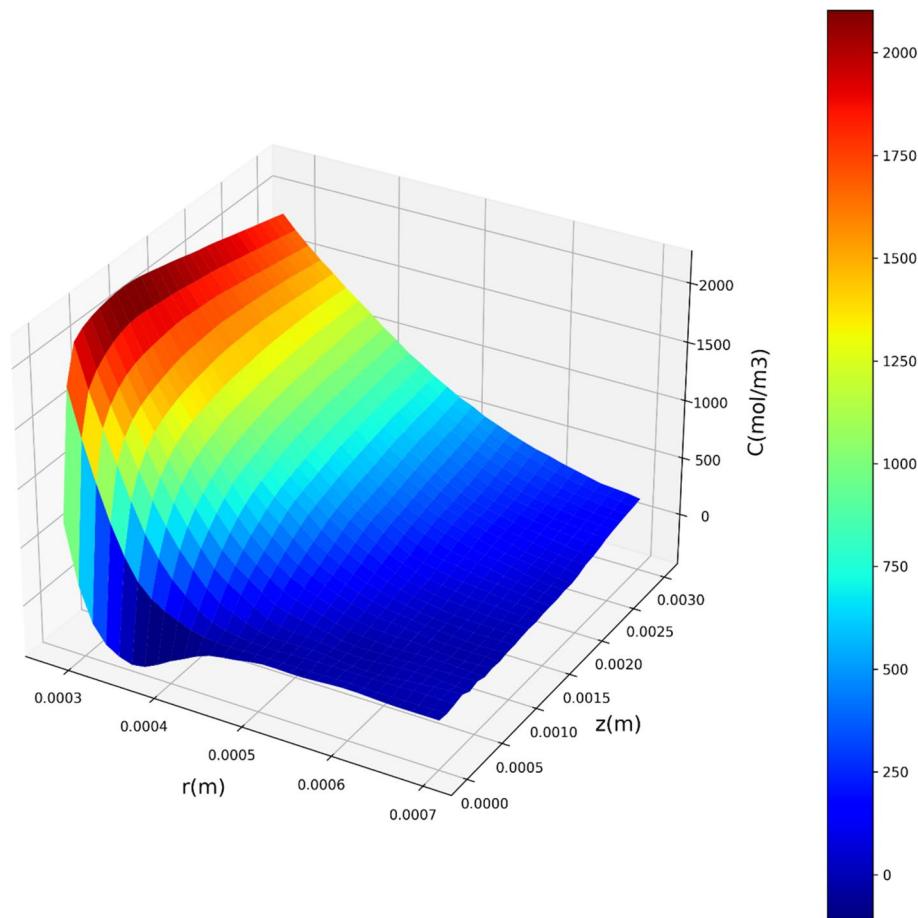


Fig. 5. 3D representation of MLP modeled concentration as a function of coordinates.

Furthermore, our work addresses the issue of overfitting in the models by employing the cross-validated objective function in HHO. The final cross-validations are also reported in this study.

Conclusion

Based on the provided results, the performance of various regression models for predicting the concentration of a substance (C) based on radial (r) and axial (z) coordinates in a membrane contactor process has been evaluated. The models considered are Kernel Ridge Regression (KRR), K-nearest neighbors (KNN), Radius nearest neighbor regression (RNN), LASSO, and Multi-Layer Perceptron (MLP). Hyper-parameter optimization has been conducted using Harris Hawks optimization (HHO).

Out of all the models evaluated, the MLP model stands out as the best performer in terms of its ability to accurately predict outcomes. The model obtained a remarkable R^2 score of 0.98637 when tested on the supplied data. This suggests that the MLP model effectively captures a substantial amount of variability in the data and produces dependable predictions of concentration using the provided inputs. Additionally, the MLP model demonstrates the lowest RMSE and MAE on both the training and test datasets, further reinforcing its superiority in terms of precision and generalization ability.

Following MLP, KRR also shows strong performance with high R^2 scores of around 0.95795 on the test data, indicating a robust ability to capture the underlying patterns in the dataset. However, it exhibits slightly higher RMSE and MAE compared to MLP, suggesting that MLP outperforms KRR in terms of prediction accuracy.

KNN and RNN perform reasonably well, with R^2 scores around 0.91 on the test data. However, they exhibit higher RMSE and MAE compared to MLP and KRR, indicating slightly lower precision in their predictions.

On the other hand, LASSO appears to be the weakest performer among the models considered, with significantly lower R^2 scores and higher RMSE and MAE values on both training and test datasets. This suggests that the linear nature of LASSO might not be well-suited for capturing the complex relationships present in the dataset.

In a nutshell the evaluation findings lead to the conclusion that, for the given dataset, the MLP model, enhanced with HHO, provides the best predictive performance, closely followed by KRR. These results can offer scholars and professionals working in the fields of concentration prediction and regression modeling in related situations important new perspectives. In future works, potential predictive models could be developed

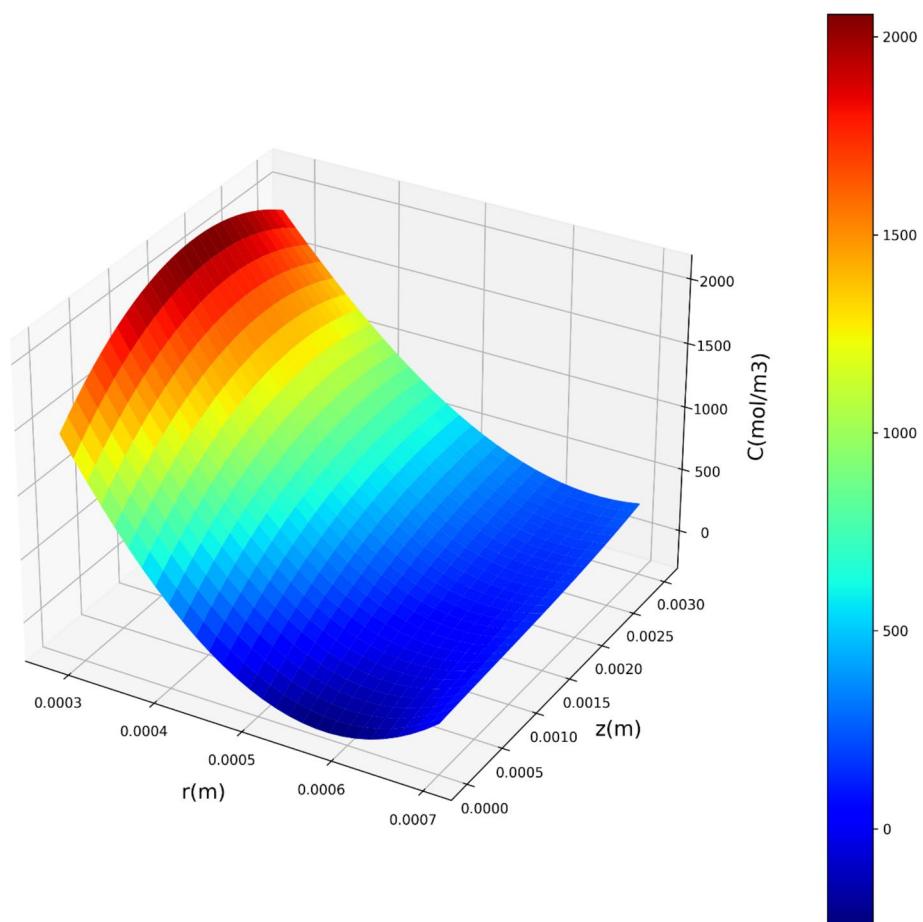


Fig. 6. 3D representation of KRR modeled concentration as a function of coordinates.

by applying the approach used in this study to similar datasets, allowing other researchers to validate and expand upon our findings.

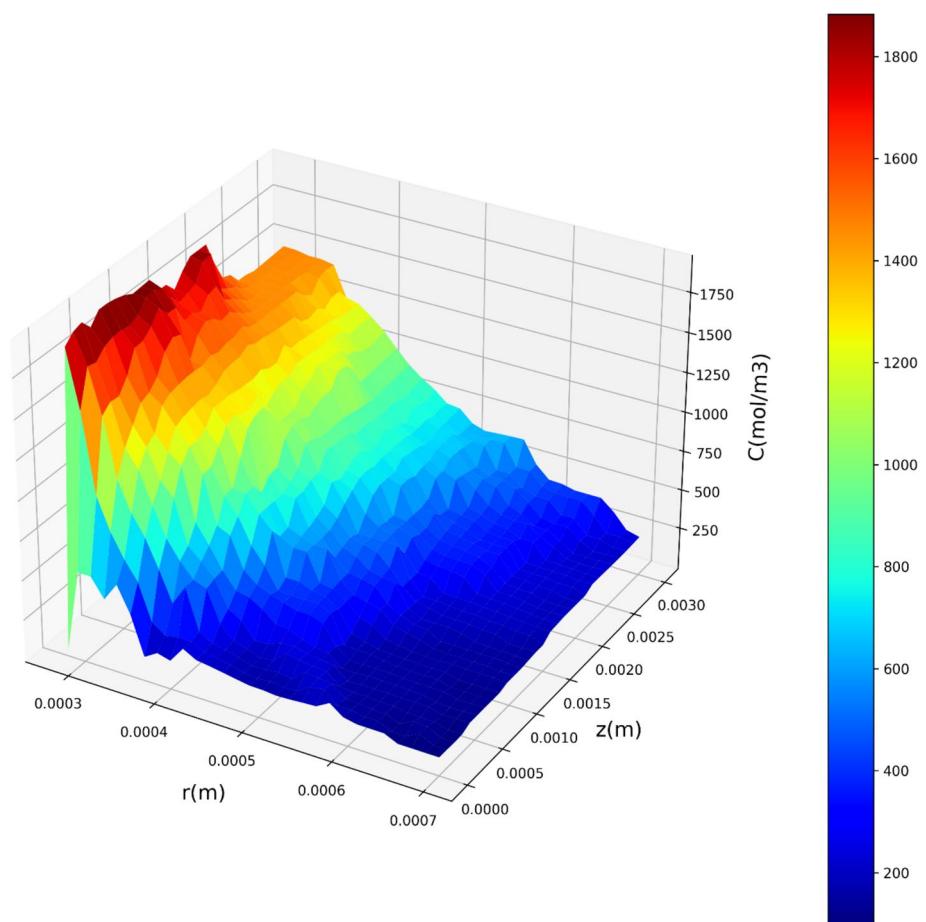


Fig. 7. 3D representation of KNN modeled concentration as a function of coordinates.

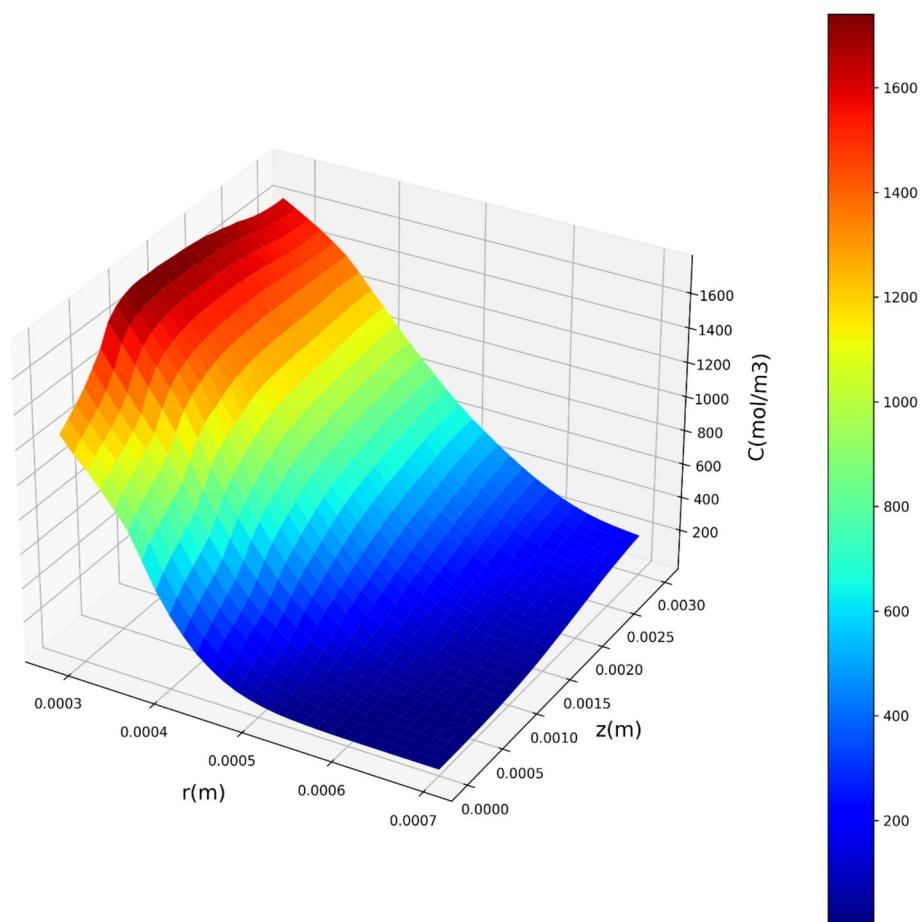


Fig. 8. 3D representation of RNN modeled concentration as a function of coordinates.

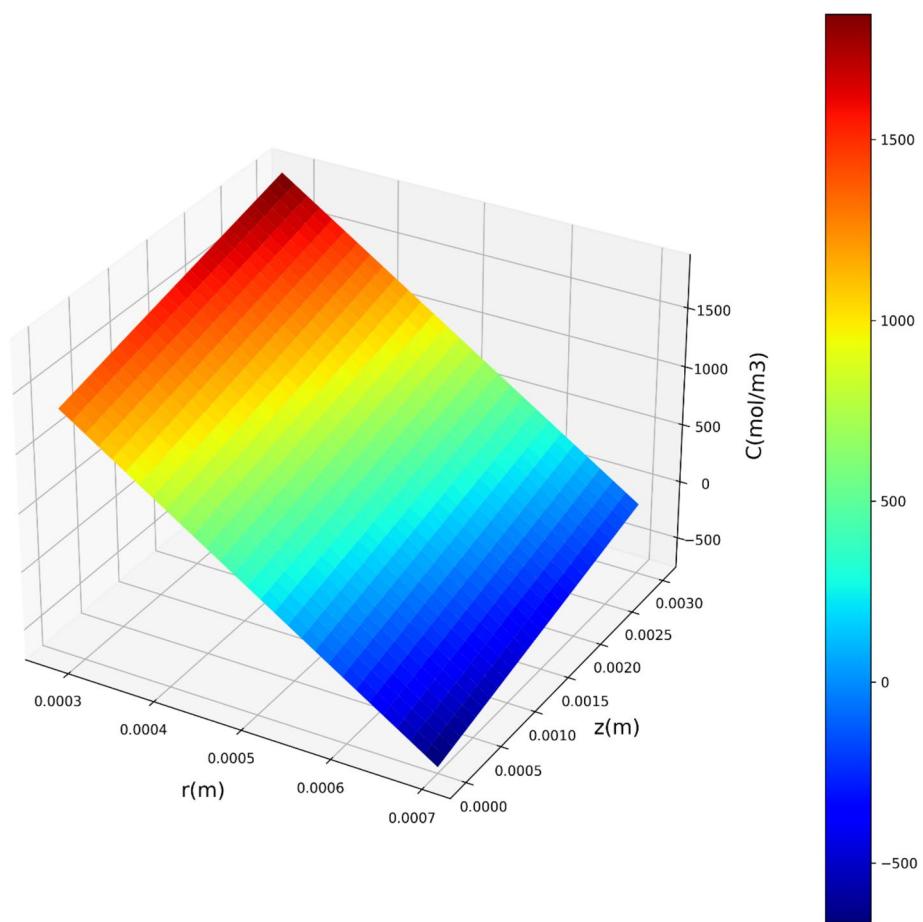


Fig. 9. 3D representation of LASSO modeled concentration as a function of coordinates.

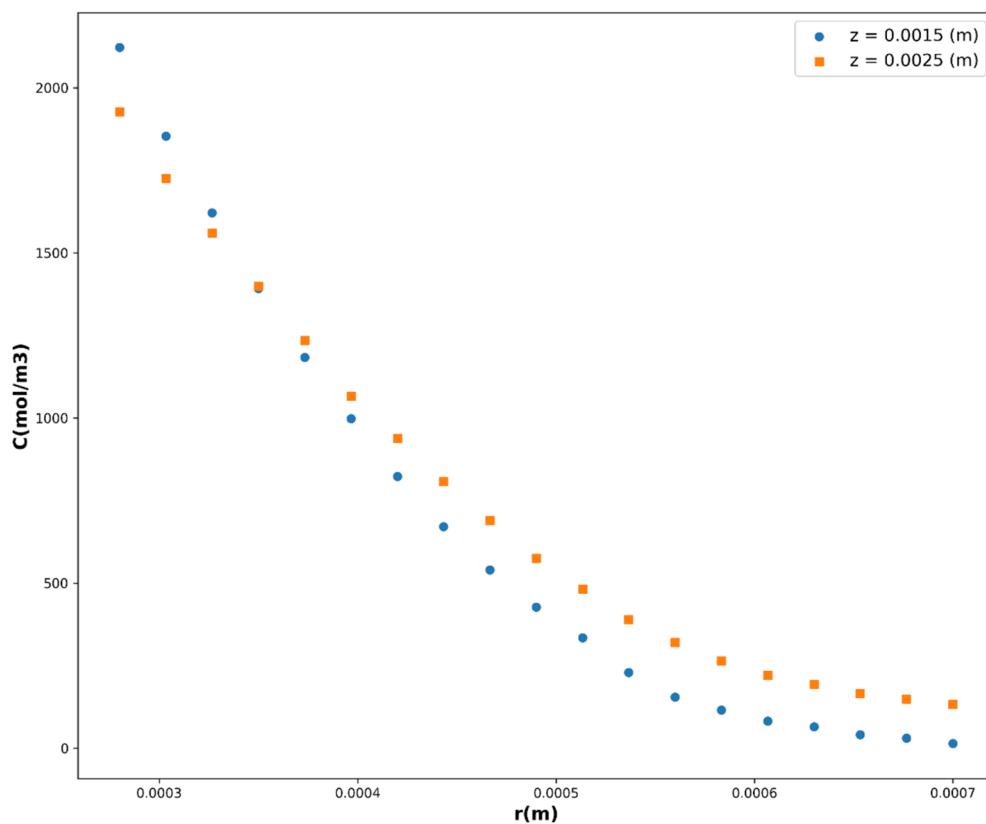


Fig. 10. Concentration profile of solute in radial direction.

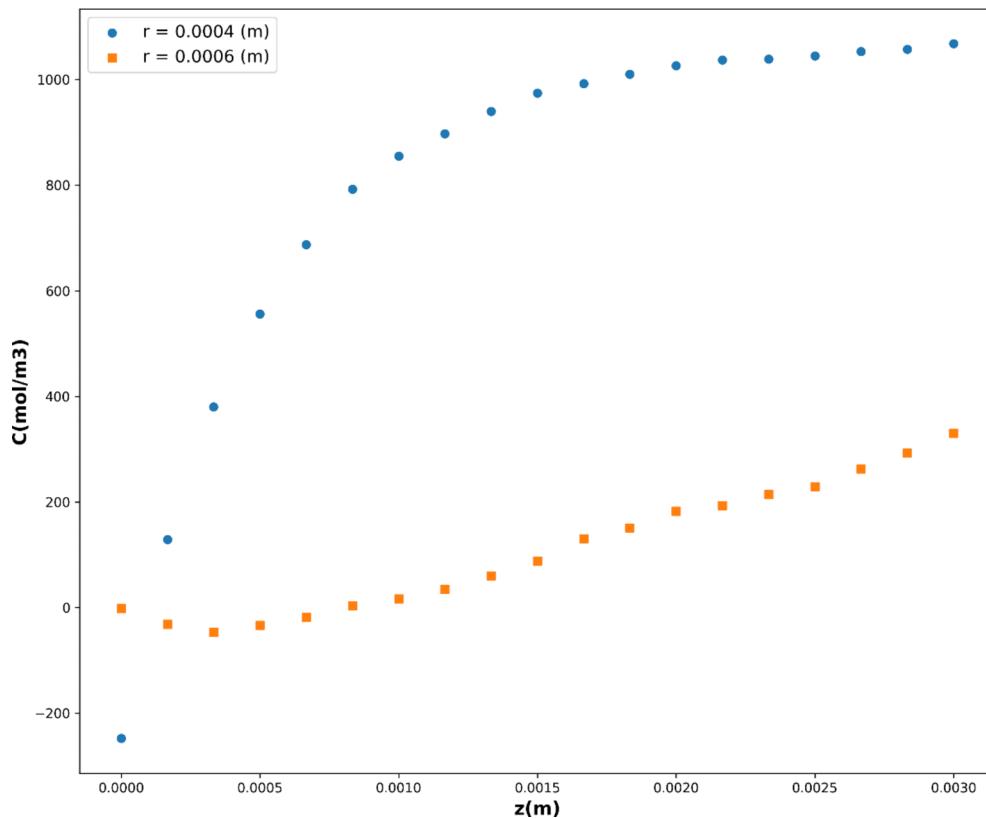


Fig. 11. Concentration profile of solute in axial direction.

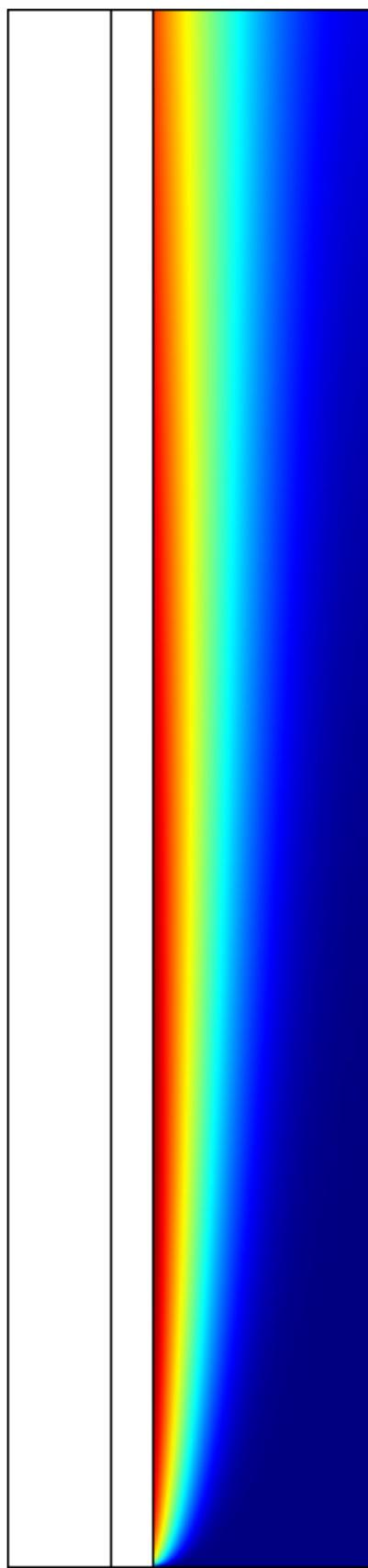


Fig. 12. Solute concentration distribution in the shell side of membrane contactor (CFD model).

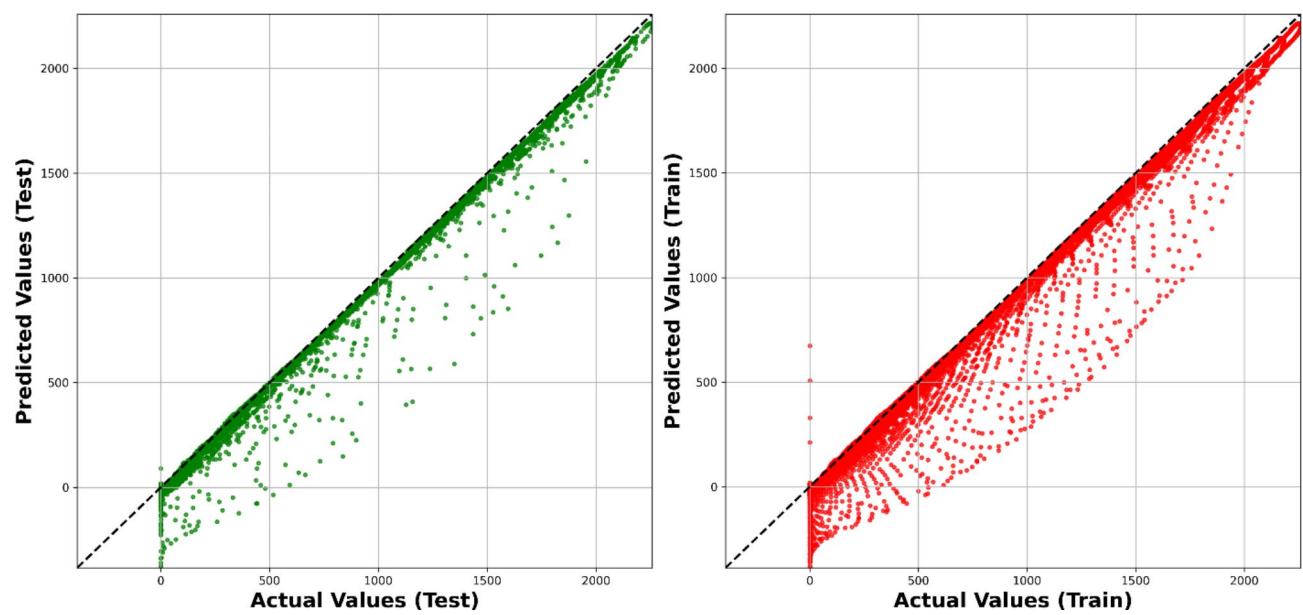


Fig. 13. Parity Plots for MLP model.

Data availability

The datasets used and analyzed during the current study are available from the corresponding author on reasonable request.

Received: 30 June 2024; Accepted: 26 September 2024

Published online: 14 October 2024

References

1. Albayati, N. et al. A comprehensive review on the use of Ti3C2Tx MXene in membrane-based water treatment. *Sep. Purif. Technol.* **345**, 127448 (2024).
2. Arthur T., Millar, G. J. & Love, J. Integration of waste heat recovered from water electrolysis to desalinate feedwater with membrane distillation. *J. Water Process. Eng.* **56**, 104426 (2023).
3. Beuscher, U., Kappert, E. & Wijmans, J. Membrane research beyond materials science. *J. Membr. Sci.* **643**, 119902 (2022).
4. Li, Q. et al. Effect of porous transport layer wettability on oxygen transportation in proton exchange membrane water electrolysis. *J. Power Sources.* **606**, 234554 (2024).
5. Liu, S. et al. Manipulating molecular orientation on porous membrane surface for fast transport. *J. Membr. Sci.* **692**, 122267 (2024).
6. Abdul Majid, O. et al. Impact of concentration polarization on membrane gas separation processes: from 1D modelling to detailed CFD simulations. *Chem. Eng. Sci.* **281**, 119128 (2023).
7. Cho, S. J. et al. Pore-filled composite membranes for water vapor separation: bench-scale advancements and semi-empirical modeling. *J. Environ. Chem. Eng.* **12**, 112986 (2024).
8. Ding, Y. & Jin, Y. Development of advanced hybrid computational model for description of molecular separation in liquid phase via polymeric membranes. *J. Mol. Liq.* **396**, 123999 (2024).
9. Almohana, A. I., Ali Bu sinnah, Z. & Al-Musawi, T. J. Combination of CFD and machine learning for improving simulation accuracy in water purification process via porous membranes. *J. Mol. Liq.* **386**, 122456 (2023).
10. Alsalhi, A. et al. *Theoretical Investigations on the liquid-phase Molecular Separation in Isolation and Purification of Pharmaceutical Molecules from Aqueous Solutions via Polymeric Membranes* 28p. 102925 (Environmental Technology & Innovation, 2022).
11. Liu, Y. et al. Computational simulation of mass transfer in membranes using hybrid machine learning models and computational fluid dynamics. *Case Stud. Therm. Eng.* **47**, 103086 (2023).
12. Kongadzem, E. M. L. *Machine Learning Application: Organs-on-a-chip in Parallel*. (2018).
13. Bishop, C. M. & Nasrabadi, N. M. *Pattern Recognition and Machine Learning* Vol. 4 (Springer, 2006).
14. Vovk, V. *Kernel Ridge Regression, in Empirical Inference* p. 105–116 (Springer, 2013).
15. Riedmiller, M. & Lernen, A. *Multi layer perceptron. Machine Learning Lab Special Lecture*, pp. 7–24 (University of Freiburg, 2014).
16. Ranstam, J. & Cook, J. LASSO regression. *J. Br. Surg.* **105** (10), 1348–1348 (2018).
17. Asadollahzadeh, M. et al. Simulation of nonporous polymeric membranes using CFD for Bioethanol Purification. *Macromol. Ther. Simul.* **27**(3), 1700084 (2018).
18. Cao, F. et al. Theoretical modeling of the mass transfer performance of CO₂ absorption into DEAB solution in hollow fiber membrane contactor. *J. Membr. Sci.* **593**, 117439 (2020).
19. Cao, Y. & Ghadiri, M. Numerical evaluation of the ozonation process in a hollow fibre membrane contactor. *Process Saf. Environ. Prot.* **170**, 817–823 (2023).
20. Najman, K. & Zieliński, K. *Outlier Detection with the use of Isolation Forests. In Data Analysis and Classification: Methods and Applications* 29 (Springer, 2021).
21. Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Future Generation Comput. Syst.* **97**, 849–872 (2019).
22. Kottath, R., Singh, P. & Bhowmick, A. Swarm-based hybrid optimization algorithms: an exhaustive analysis and its applications to electricity load and price forecasting. *Soft. Comput.* **27** (19), 14095–14126 (2023).
23. Akl, D. T. et al. IHHO: an improved Harris Hawks optimization algorithm for solving engineering problems. *Neural Comput. Appl.* **36**, 12185–12298 (2024).
24. Al-Betar, M. A. et al. Survival exploration strategies for Harris hawks optimizer. *Expert Syst. Appl.* **168**, 114243 (2021).
25. Acito, F. *k Nearest Neighbors*, in *Predictive Analytics with KNIME: Analytics for Citizen Data Scientists*, pp. 209–227 (Springer, 2023).
26. Kramer, O. & Kramer, O. *K-nearest neighbors*. Dimensionality reduction with unsupervised nearest neighbors, pp. 13–23 (2013).
27. Shataee, S. et al. Forest attribute imputation using machine-learning methods and ASTER data: comparison of k-NN, SVR and random forest regression algorithms. *Int. J. Remote Sens.* **33** (19), 6254–6280 (2012).
28. Wang, Z. et al. Entropy and gravitation based dynamic radius nearest neighbor classification for imbalanced problem. *Knowl. Based Syst.* **193**, 105474 (2020).
29. Du, K. L. et al. Perceptron: Learning, generalization, model selection, fault tolerance, and role in the deep learning era. *Mathematics*. **10** (24), 4730 (2022).
30. Bisong, E. & Bisong, E. *The multilayer perceptron (MLP). Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 401–405 (2019).
31. Zhang, S. et al. Kernel ridge regression for general noise model with its application. *Neurocomputing*. **149**, 836–846 (2015).

Acknowledgements

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through Large Research Project under grant number RGP2/580/45 .

Author contributions

Kamal Y. Thajudeen: Investigation, Validation, Methodology, Writing. Mohammed Muqtader Ahmed: Writing, Validation, Formal analysis. Saad Ali Alshehri: Investigation, Supervision, Formal analysis, Resources.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to K.Y.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024