

Optimizing neural network algorithms for submerged membrane bioreactor: A comparative study of OVAT and RSM hyperparameter optimization techniques

Syahira Ibrahim  and Norhaliza Abdul Wahab *

Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor Bahru 81310, Malaysia

*Corresponding author. E-mail: aliza@fke.utm.my

 SI, 0000-0001-6933-5536; NA, 0000-0001-8522-2507

ABSTRACT

Hyperparameter tuning is an important process to maximize the performance of any neural network model. This present study proposed the factorial design of experiment for screening and response surface methodology to optimize the hyperparameter of two artificial neural network algorithms. Feed-forward neural network (FFNN) and radial basis function neural network (RBFNN) are applied to predict the permeate flux of palm oil mill effluent. Permeate pump and transmembrane pressure of the submerge membrane bioreactor system are the input variables. Six hyperparameters of the FFNN model including four numerical factors (neuron numbers, learning rate, momentum, and epoch numbers) and two categorical factors (training and activation function) are used in hyperparameter optimization. RBFNN includes two numerical factors such as a number of neurons and spreads. The conventional method (one-variable-at-a-time) is compared in terms of optimization processing time and the accuracy of the model. The result indicates that the optimal hyperparameters obtained by the proposed approach produce good accuracy with a smaller generalization error. The simulation results show an improvement of more than 65% of training performance, with less repetition and processing time. This proposed methodology can be utilized for any type of neural network application to find the optimum levels of different parameters.

Key words: computational time, DoE implementation, neural network, optimization techniques, submerged membrane bioreactor data

HIGHLIGHTS

- Membrane fouling in filtration is a complex process, and understanding the behaviour of the dynamic is crucial.
- This is the study of artificial neural network (ANN)-based dynamic model and optimization for a submerged membrane bioreactor (SMBR) filtration process.
- The ANN structure was able to model the dynamic behaviour of the filtration process under normal conditions.
- The optimization method improves the ANN structure for SMBR filtration model development.

1. INTRODUCTION

Malaysia is the second-largest producer and exporter of palm oil (PO) after Indonesia (Kushairi *et al.* 2018). Currently, PO and palm kernel oil constitute 34% of the world's oil and fat production, crucial for sustaining the global population. Consequently, the PO industry plays a pivotal role in Malaysia's agricultural and economic growth, making a substantial contribution to gross domestic product, gross national income, foreign exchange reserves, and employment (Kushairi *et al.* 2018). Simultaneously, the high production of PO results in a significant volume of industrial wastewater. It is estimated that the extraction of one tonne of crude PO necessitates between five and seven tonnes (5000–7000 kg) of water, with over 50% of this water becoming palm oil mill effluent (POME) (Ahmad *et al.* 2015).

POME comprises wastewater streams originating from three primary processing steps in the mill, namely, sterilizer condensate, clarification wastewater, and hydrocyclone wastewater (Liew *et al.* 2015). POME is naturally non-toxic, and direct discharge of untreated POME into rivers initiates natural decomposition and rapidly depleting dissolved oxygen levels in the river water. This leads to the degradation of aquatic life and the natural ecosystem. Recognizing the environmental impact of the PO industry, the adoption of advanced wastewater treatment technologies for POME discharge is imperative

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

to meet the standards set forth by the Malaysian government through the enactment of the Environmental Quality Acts (EQA) in 1978.

In recent years, membrane technology has gained significant importance in the field of wastewater treatment, such as municipal, domestic, and industrial wastewater including POME. This preference arises from its simplicity of operation, lower weight, reduced space requirements, and high efficiency. The membrane bioreactor (MBR) technology has also demonstrated its reliability as a filtration system for treating POME discharge (Teow *et al.* 2023). However, fouling remains the principal drawback of MBR filtration systems, leading to increased energy consumption and maintenance costs (Ghani *et al.* 2018). Fouling occurs when solid materials clog the membrane pores, resulting in decreased permeate flux and elevated transmembrane pressure.

According to Li *et al.* (2022), fouling can be mitigated by controlling various fouling variables. Stuckey (2012) demonstrated that fouling is influenced by several parameters, including membrane form, hydrodynamic conditions, the composition of the biological system, and the operating conditions of the reactor and the chemical system. Operationally, fouling can be managed and reduced through techniques such as air bubble (aeration) control, backwashing, relaxation, and chemical cleaning (Yusuf *et al.* 2019). Given that the MBR filtration system involves numerous variables and is highly nonlinear due to the intricate nature of fouling variables, the construction of mathematical models for predictive purposes proves to be a valuable tool (Xiong *et al.* 2019). A reliable prediction model is critical for supporting decision-making that impacts the system performance.

Machine learning (ML) algorithm primarily aims to develop models that can predict new values based on historical data. Linear and nonlinear regression models are the two general categories of ML algorithms (Garza-Ulloa 2022). For nonlinear models, frequently employed approaches include k-nearest neighbour, decision trees, support vector machines, and artificial neural networks (ANNs) (Basha & Rajput 2019). According to Bunmahotama *et al.* (2017), prediction models play an indispensable role in optimizing parameters by conducting independent data processing, forecasting outputs based on input datasets, adjusting the operation of a series of equipment, and furnishing timely and automated decisions for production. Furthermore, modelling aids in streamlining operations, conserving manpower, and reducing energy consumption within the system. Therefore, an accurate prediction model is pivotal in supporting the decision-making process, which directly impacts system performance.

The ANN stands as a well-established ML algorithm, noted for its efficacy in modelling complex and nonlinear processes. The concept of ANN is inspired by the functioning of the biological human brain in problem-solving processes. ANN has several merits, including strong learning capacity, parallelism, and robustness in handling noise (Hemeida *et al.* 2020). The most common ANN structures employed in wastewater treatment are the multilayer perceptron or feed-forward neural network (FFNN) and the radial basis function neural network (RBFNN). However, the primary challenge in constructing effective prediction models using ANN lies in the tuning process of the hyperparameters (Feurer & Hutter 2019; Probst 2019b; Pannakkong *et al.* 2022). Most ML techniques, including ANN, encompass hyperparameters, each requiring specific configurations. Tuning these hyperparameters to identify the optimal configuration that minimizes the loss value is known as hyperparameter optimization (HPO) (Feurer & Hutter 2019).

HPO plays a critical role in enhancing the performance and reproducibility of ML models (Vincent & Jidesh 2023). Consequently, HPO has gained substantial commercial interest, aligning with the increased utilization of ML across industries. However, it has usually found application in internal tools and ML cloud services. HPO serves as a crucial resource for data analysts and researchers in comprehending ML hyperparameters (Ali *et al.* 2023). Nevertheless, resolving the HPO problem is challenging, and achieving the ideal configuration in a few trials is crucial, necessitating a suitable approach. Several ML studies propose methods to tune hyperparameters for decision tree (Alawad *et al.* 2018), support vector machine (Duarte & Wainer 2017), deep neural network (Zhou *et al.* 2019), random forest (RF) (Probst 2019b), FFNN (Ibrahim & Wahab 2022), etc. These studies show that well tuning the hyperparameters of each algorithm increases the performance of the ML training process (Nematzadeh *et al.* 2022). There exists no one-size-fits-all rule for tuning hyperparameters, and this optimization process is contingent on the complexity of the modelled system.

In the literature, various methods for HPO or tuning hyperparameters are outlined. The simplest approach with minimal effort entails using recommended values from prior studies or default values suggested by ML libraries. Nonetheless, these hyperparameter values may not universally perform optimally across all ML models, and they may require modification based on different or updated input data (Lujan-Moreno *et al.* 2018; Probst 2019a). Conventional HPO is often conducted through a trial-and-error method, where studies may focus solely on selecting the number of neurons and neglecting other

hyperparameters (Badrnezhad & Mirza 2014; Onukwuli *et al.* 2021). This approach may prove impractical, especially for less-experienced modellers when an extensive array of configurations is involved. Frequently, this technique may not guarantee finding the optimal hyperparameter values.

Other studies (Pakravan *et al.* 2015; Said *et al.* 2018) have applied a one-variable-at-a-time (OVAT) method to determine the optimal number of neurons. Among conventional methods, OVAT is more efficient, as it is user-friendly and requires minimal expertise from the modeller. For instance, Jawad *et al.* (2021) individually trained the log-sigmoid and tan-sigmoid as activation functions with neurons ranging from 1 to 20. However, these tuning procedures are time-consuming and laborious, particularly when numerous parameters are examined simultaneously (Elfghi 2016). According to Ibrahim *et al.* (2020), for three different levels of each ANN variable, approximately 245 ($=3^5$) different configurations of hyperparameters would be required.

The design of experiment (DoE) technique has gained considerable attention and has been successfully applied as a modelling and optimization tool to address complex and nonlinear problems in various industrial fields (Pashaei *et al.* 2023). The technique is favoured for its reliability, simplicity, systematic strategies, low trial requirements, and reduced processing time. While considering all hyperparameter combinations in the application of HPO may be time-consuming, especially when involving mixed parameters (e.g., numerical and categorical), the DoE technique aids in overcoming these drawbacks. In addition, it provides richer information by exploring a larger design space in an engaging manner (e.g., two dimensional (2D) and three dimensional (3D)), offering an easy means to discern the interaction effects of the hyperparameters.

In hyperparameter tuning for optimization problems, DoE techniques have been employed, although with limited room for study. For instance, the study by Kechagias *et al.* (2018) utilized the full factorial design method for HPO, optimizing the number of neurons in the hidden layer, learning rate, and momentum constant. While this method accurately identifies the main and interaction effects of the hyperparameters, it is restricted to linear models and experiences an exponential increase in the number of experiments with a higher number of factors. Typically, this factorial design is applicable to three primary objectives: screening factors, optimization, and stability testing (Chen *et al.* 2013). Consequently, response surface methodology (RSM) proves indispensable if curvature is present due to higher-order relationships. Moreover, studies by Lujan-Moreno *et al.* (2018) and Probst (2019b) have employed DoE methodology to tune RF hyperparameters such as ntree, mtry, replace, nodesize, classwt, cutoff, and maxnodes value. This DoE approach combines linear (factorial design) and non-linear (RSM) design methods for screening and tuning hyperparameters, respectively.

Among RSM designs, the most popular are central composite design (CCD) and Box–Behnken designs (BBD). A study by Pashaei *et al.* (2023) has compared CCD and BBD, with CCD standing out for its high applicability and performance with only a few test points. In another work presented by Nourbakhsh *et al.* (2014), only 52 different settings were created using CCD for five different types of hyperparameters, including the number of neurons, training epochs, step size, training percentage, and momentum coefficient. A quadratic model was developed from training results to mean-square error (MSE) as the response. The results indicated that step size, momentum coefficient, and training epoch significantly influenced the model development. However, the study did not account for any categorical factors. Furthermore, it did not compare the conventional method with the proposed RSM technique in terms of optimization processing time and model accuracy. As such, the fairness and effectiveness of the RSM method could not be conclusively established. This underscores the need to expand knowledge on mixed hyperparameters concerning prediction model performance across various ML types using the RSM approach.

In light of this, the work here aims to propose a combined DoE and RSM optimization approach with ANN models to predict the permeate flux of submerged membrane bioreactors (SMBRs). In this case, two common network structures – FFNN and RBFNN algorithms – are employed. A combination of numerical and categorical hyperparameters is utilized to achieve high model accuracy. The outcomes of ANN using the proposed RSM optimization and OVAT optimization are compared to gain a comprehensive understanding of the respective models' capacities for accurate prediction and to enhance ANN training performance.

The evaluation of model performance is based on real input–output data from the SMBR treatment plant for POME wastewater. Subsequently, model accuracy is analyzed through performance statistical errors, including the correlation coefficient (R), coefficient of determination (R^2), MSE, and root-mean-square error (RMSE). The total number of repetitions and computational time are also measured and compared. To demonstrate the practical implications of the proposed HPO, the outcomes of applying two types of HPO techniques to the neural network learning algorithms (FFNN and RBFNN) are presented. The simulation and experimental findings from the application of HPO on a POME dataset are discussed thoroughly.

2. MATERIALS AND METHODS

2.1. Experimental setup

MBR filtration treatment is conducted at a pilot scale with a working volume of 30 L. Samples are collected from the final pond of the treatment plant, possessing a biochemical oxygen demand of less than $10,000 \text{ mg L}^{-1}$, to induce fouling in the SMBR filtration process. The plant is equipped with a single bioreactor tank containing submerged hollow fibre membranes. These hollow fibre membranes are fabricated using polyethersulfone, with a pore size of approximately 80–100 kDa and an effective surface area of about 0.35 m^2 . Figure 1 depicts the schematic of the pilot plant setup for the experiment, and Table S1 (see Supplementary Material) specifies the instruments utilized in the pilot plant. Data acquisition was controlled and monitored through the National Instruments Labview 2009 software, interfaced with NI USB 6009 hardware.

2.1.1. Data collection

A total of 4,000 data were collected for each parameter during the experiment, involving airflow (standard litres per minute (SLPM)), permeate pump voltage, transmembrane pressure (mbar), and permeate flux ($\text{Lm}^{-2}\text{h}^{-1}$). Random magnitude steps vary from 1.4 to 3 volts for the suction pump and from 0 to 270 mbar for the transmembrane pressure (TMP), to produce the dynamic behaviour of the filtration process. Both permeate pump and TMP were OFF during the relaxation period. The settings for the permeate-to-relaxation period were maintained at 120 s for permeate and 30 s for relaxation, with continuous aeration airflow. Airflow was regulated between 6 and 8 SLPM during filtration to sustain a high intensity of bubble flow for membrane surface cleansing. Figure 2 illustrates the SMBR filtration dataset, including permeate pump voltage and TMP as inputs and permeate flux as output. Data analysis was conducted using MATLAB R2015a and Design-Expert version 12.0.

2.1.2. Experimental analyses

Filtration process performance is assessed based on permeate flux, which is calculated as follows:

$$J = \frac{v}{A \times t}, \quad (1)$$

where J is permeate flux in ($\text{L m}^{-2} \text{ h}^{-1}$), v is the volume flowrate in litres (L), A is the membrane surface area (m^2), and t is the time (h).

2.2. Preprocessing data

In this study, data normalization and division were conducted before developing the neural network model. Given that the input data for the SMBR system encompass different magnitudes and scales, the dataset was scaled to fall within the range of 0–1. This was implemented to prevent the dominance of large original input data in the solution, as well as to mitigate

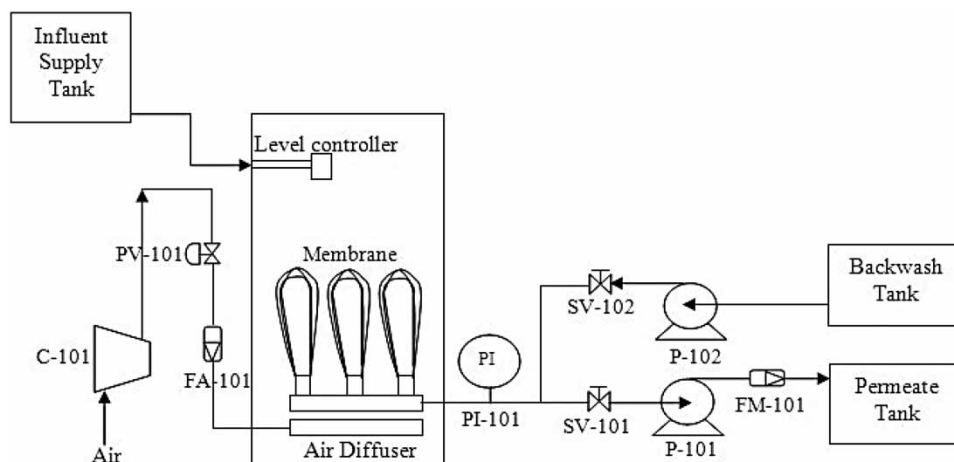


Figure 1 | Schematic diagram of the SMBR system.

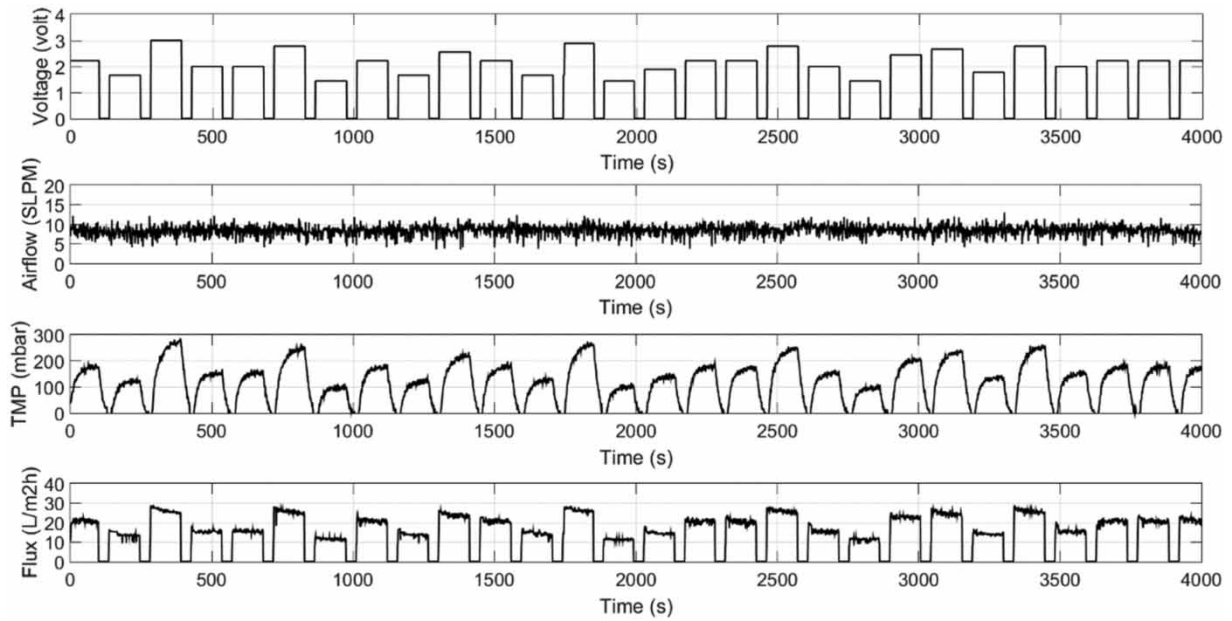


Figure 2 | Dataset from the SMBR filtration experiment.

numerical challenges during calculations (Kumar 2011). Equation (2) outlines the normalization formula:

$$X' = \frac{y_i - y_{\min}}{y_{\max} - y_{\min}} \quad (2)$$

where X' represents the scaled value, y_i is the i th actual value of the data, y_{\max} is the maximum value of the data, and y_{\min} is the minimum value of the data.

To assess the feasibility of the predictive model, all data were randomly divided into training dataset (T_{training}) and testing dataset (T_{testing}). The training dataset was employed for network training and evaluation, while the testing dataset was used to validate network performance. The dataset was partitioned into 60% for training and 40% for testing. Hyperparameters were tuned, and network performance was evaluated using the training dataset (T_{training}). This process was repeated three times, with the average MSE calculated. The hyperparameters yielding the minimum MSE were selected. Finally, the network was validated on the testing dataset (T_{testing}) to provide an unbiased estimate of its performance on unseen data.

2.3. Simulation modelling

The present study employs the schematic structures of FFNN and RBFNN to predict the permeate flux of POME during SMBR filtration processes. The input variables comprise transmembrane pressure and permeate pump voltage, while the output variable is the permeate flux of POME.

2.3.1. Feed-forward neural network

The FFNN consists of two or more layers of processing elements linked by weight connections. It comprises three layers: an input layer, a hidden layer, and an output layer (Figure S1 in Supplementary Material) (Ghasemi *et al.* 2023). An earlier work (Said *et al.* 2018) reported that a network with one hidden layer is commonly used in practice. The error is fed forward through the network during training to adjust the weights and minimize the error. This process is repeated until the minimum error is achieved or a specific number of epochs is reached. The FFNN used in this study is based on the following equation:

$$\hat{y}_1(t) = E_i \left[\sum_{j=1}^{nh} W_{ij} f_j \left(\sum_{l=1}^{n\varphi} w_{lj} + w_{j0} \right) + W_{i0} \right], \quad (3)$$

where $\hat{y}_1(t)$ represents the prediction output, f_j is the function of the network, and φ is the input vector. W_{ij} and w_{ij} represent the network connection layer weights and biases, respectively.

2.3.2. Radial basis function neural network

The RBFNN is structurally similar to the FFNN, comprising three layers. In RBFNN, neurons in the hidden and output layers are interconnected by weights. Typically, the neurons in RBFNN are based on the Gaussian function (Kasiviswanathan & Agarwal 2012). Its structure is depicted in Figure S2 and defined in Equation (4):

$$y(t) = \sum_{i=1}^k w_i \theta_i(t); \dots \dots \dots i = 1, 2, 3 \dots k. \quad (4)$$

The output of the hidden layer can be represented by Equation (5):

$$\theta_i = e^{-\left(\frac{\|x(t) - c_i(t)\|^2}{\psi_i^2}\right)}, \quad (5)$$

where x is the input vector, c is the centre of the hidden node, k is the number of hidden nodes, and ψ is the width of the hidden node. w represents the weight number of the output layer, and y indicates the output of the network.

To achieve optimal network performance in RBFNN, it is essential to regulate certain hyperparameters, namely, the number of spreads and the number of neurons in the hidden layer. An incorrect choice of the spread constant can lead to underfitting or overfitting. High spread values result in data points being scattered over a considerable distance from the centre, consequently reducing the maximum function response (Kasiviswanathan 2012). It is worth noting that the default spread value in the Matlab toolbox is set to one.

2.3.3. Hyperparameter configurations

Optimizing hyperparameters is crucial for achieving good performance in an ANN model. Careful selection of appropriate ranges for hyperparameters is essential. For the FFNN model structure, the training process involves various numerical hyperparameters, including the number of neurons, learning rate, number of epochs, and momentum coefficient. In addition, categorical factors such as training function and activation function are considered to enhance model prediction performance.

Activation functions, sometimes referred to as ‘transfer functions,’ determine how the weighted sum of inputs is transformed into an output from nodes in a layer of the network. The choice of activation function in the hidden and output layers significantly influences the network’s ability to learn from the training dataset. Typically, the hidden layer employs non-linear functions, while the output layer uses a linear transfer function (*purelin*). Commonly used activation functions in hidden layers include sigmoid (or logistic) and hyperbolic tangent (*Tanh*) (Rasamoelina *et al.* 2020). In recent years, the rectified linear unit (*ReLU*) has gained popularity among practitioners and deep learning researchers for its simplicity and superior training performance over common activation functions (Rasamoelina *et al.* 2020).

Training functions used in this case are Levenberg–Marquardt (lm) and gradient descent with momentum (gdm). The lm training function offers capabilities such as fast training, nonlinear regression with low MSE, and memory reduction features (Keong *et al.* 2016; Johnson Santhosh *et al.* 2021). On the other hand, gdm has advantages such as avoiding local minima, accelerating learning, and stabilizing convergence (Winiczenko *et al.* 2016). The gdm training function primarily depends on two parameters: the learning rate and momentum parameters. Thus, the impact of different types of activation and transfer functions on the network performance of SBR filtration for POME models needs to be defined.

The number of epochs or training cycles is crucial in determining network models. Too few epochs limit the network’s ability, while too many can lead to overfitting and increased error (Winiczenko *et al.* 2016). In training the FFNN model with different training functions (br, lm, and gd), the maximum setting for the number of epochs is 1,000 (Keong *et al.* 2016). Next, the parameter aims to find the minimum weight space. A learning rate that is too high may lead to increased oscillations in MSE, while too low of a learning rate causes smaller steps in the weight space. A low learning rate can reduce the network’s ability to escape local minima in the error surface.

The following parameter defines the amount of momentum, where low momentum leads to less sensitivity of the network to local gradients, while high momentum may cause divergence of adaptation, resulting in unusable weights. Moreover, studies by Winiarczyk *et al.* (2016) and Yousif & Kazem (2021) have found that optimal values of the learning rate and momentum provide smooth behaviour and speed up convergence. They also noted that excessively low values of the learning rate and momentum slow down the convergence process, while overly high values may lead to network instability and training divergence.

The number of neurons in the hidden layer plays a pivotal role in network performance. Too few hidden neurons result in limited adaptation in simulation modelling. Conversely, too many may lead to system memory errors or over training (Winiarczyk *et al.* 2016) and increase the calculation time (Nazghelichi *et al.* 2011). There is no systematic method for determining the structure (number of hidden neurons) of the network. To achieve an accurate neural network approximation, there exists an upper bound for the number of hidden neurons, as proposed by Hecht-Nielsen (1992), given in Equation (6):

$$N^\alpha \leq 2N^\beta + 1, \quad (6)$$

where N^α is the number of hidden neurons and N^β is the number of inputs.

Rogers & Dowla (1994) proposed the relationship between the number of training data samples and the number of hidden neurons, which is given in Equation (7), to prevent an overfitting issue in the training data:

$$N^\alpha \leq \frac{N^\theta}{N^\beta + 1}, \quad (7)$$

where N^θ is the number of samples in the training data.

For a conventional-based ANN to search for optimal conditions, networks were trained under a wide range of hyperparameter settings based on the OVAT method. The conventional HPO was carried out on the training dataset (T_{training}). The selection of corresponding FFNN and RBFNN hyperparameters is based on the lowest MSE for each parameter, which is then used for the testing dataset (T_{testing}) to verify the reliability of the network models. In this present study, the number of neurons in the hidden layer, the learning rate, the momentum, the number of epochs, the number of spreads, training function, and activation function are considered in building an optimum network structure. The range of hyperparameters for conventional-based FFNN and RBFNN models is presented in Table 1.

Table 1 | The range of ANN hyperparameters for conventional models

Conventional-based FFNN method						
Symbol	Model Hyperparameters	FFNN-1	FFNN-2 Range settings	FFNN-3	FFNN-4	
A	Number of neurons	1–30	1–30	1–30	1–30	
B	Learning rate	0.1–1	0.1–1	0.1–1	0.1–1	
C	Momentum	0.1–1	0.1–1	0.1–1	0.1–1	
D	Number of epochs	100–1,000	100–1,000	100–1,000	100–1,000	
E	Training function	<i>trainlm</i>	<i>trainlm</i>	<i>traingdm</i>	<i>traingdm</i>	
F	Activation function	<i>tansig</i>	<i>ReLU</i>	<i>tansig</i>	<i>ReLU</i>	
	Number of repetitions	60	60	60	60	
Conventional-based RBFNN method						
Symbol	Model Hyperparameters	RBFNN Range setting				
A	Number of neurons	1–30				
B	Number of spreads	0.5–3				
	Number of repetitions	36				

2.3.4. Proposed ANN-RSM hyperparameter optimization method

2.3.4.1. Screening method. This section outlines the proposed RSM-based ANN (ANN-RSM) HPO method. Figure S3 illustrates the screening steps using the two-level full factorial method. Initially, a screening process using a two-level full factorial design model was conducted with a few sets of hyperparameter configurations. This approach greatly aids in identifying appropriate ranges or scales for RSM optimization. A first-order model using a polynomial function is generally sufficient to approximate the process. It is assumed to behave similarly to the network within a small range or region of the response surface. The first-order model is represented as follows:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_i x_i + \varepsilon, \quad (8)$$

where Y represents the predicted response or dependent variable, x_i are the independent variables, β_i are the constants, and ε is the error term.

Under this screening method, the region of the response surface was defined using several scales. Three different scales of hyperparameters for FFNN and two different scales for RBFNN were assessed (refer to Table S2 in the Supplementary Material). The scale of hyperparameters exhibiting the best performance was selected for the subsequent RSM optimization procedure. The screening results for FFNN and RBFNN hyperparameters (2LF-FFNN and 2LF-RBFNN, respectively) are provided in Table S2.

Analysis of variance (ANOVA) at a 5% level of significance was used for determining the experimental design, interpretations, and analyses of the training data. The ANOVA included evaluation terms such as the coefficient of determination (R^2), adjusted R^2 , and predicted R^2 , which were used to assess the significance of the model. The predicted response was transformed to bring the distribution of the response variable closer to normal distribution. The Box-Cox plot was applied to improve model fitting, employing various transformations $\lambda = -1$, $\lambda = -1$, $\lambda = 0$, $\lambda = 0.5$, representing inverse, inverse square root, natural log, square root, and no transformation functions, respectively (Nazghelichi *et al.* 2011).

Pareto chart is applied to illustrate the sequencing of the statistically significant for main and interaction effect factors and to compare their relative value. Factors exceeding the reference line are considered significant at a confidence level (t -value) of 95%. Figure S4(a) shows that the main factor of training function (E) hyperparameters contributes the highest effect to the MSE value of 2LF-FFNN-1 algorithm, followed by momentum (C), neuron numbers (A), and learning rate (B). In addition, there are several interaction factors significantly affecting the MSE performance, such as AD, AE, AF, BC, BE, CD, CE, CF, DE, DF, EF, ACE, ADE, ADF, BCE, and ACDE. In Figure S4(b), the number of neurons has the most significant impact on the MSE performance of 2LF-RBFNN-2 algorithm.

The ANOVA reveals that the range setting for 2LF-FFNN-1 yielded the best performance with $R^2 = 0.9534$, $R_{\text{adj}}^2 = 0.9247$, and $R_{\text{pred}}^2 = 0.8744$ using inverse square root transformation to develop the highest factorial model. For 2LF-RBFNN screening hyperparameters, natural log transformation was used to develop the highest two-level factorial model. Based on the ANOVA results, 2LF-RBFNN-2 produced an excellent model performance ($R^2 = 0.9898$, $R_{\text{adj}}^2 = 0.9847$, and $R_{\text{pred}}^2 = 0.9593$). Therefore, the scale and configuration hyperparameters of 2LF-FFNN-1 and 2LF-RBFNN-2 were selected for FFNN and RBFNN, respectively, to execute the higher-order optimization method using the RSM technique.

2.3.4.2. RSM hyperparameters optimization method

i. Data Insertion:

RSM is a sequential procedure used to fit a second-order model, which works well in modelling curvature around promising regions. The overall steps of data insertion were the same as Figure S2. In the case of FFNN, six hyperparameters are taken into account: four numerical variables (number of neurons, learning rate, momentum, and number of epochs) and two categorical components (training function and activation function). Meanwhile, in the RBFNN case, two numerical variables (number of neurons and spreads) are considered. The lowest and highest levels of variables were coded as -1 and $+1$, respectively, including the axial star points of $(-\alpha$ and $+\alpha)$, where α is the distance of the axial points from the centre and makes the design rotatable. The α value was set to 1 (face centred). The total number of experimental combinations was calculated based on the concept of CCD using Equation (9):

$$2^k + 2k + n_0, \quad (9)$$

where k is the number of factor (numerical factor only) and n_0 is the number of experiments (tuning) repeated at the centre point. Then, 2^k and $2k$ are stated as the factorial point and an axial point, respectively.

In the FFNN case, with four numerical factors (number of neurons, learning rate, momentum, and number of epochs), $k = 4$ and $n_0 = 6$ were set. Since this study involves two categorical factors with two levels – training function (*trainlm* and *traingdm*) and activation function (*tansig* and *ReLU*) – the number of experiments repeated at the factorial point, axial point, and centre point are doubled for each categorical factor ($2_{\text{trainlm}}^4 = 16$, $2_{\text{traingdm}}^4 = 16$; $2_{\text{tansig}}^4 = 16$, $2_{\text{ReLU}}^4 = 16$; $2k_{\text{trainlm}} = 8$, $2k_{\text{traingdm}} = 8$; $2k_{\text{tansig}} = 8$, $2k_{\text{ReLU}} = 8$; $n_{0 \text{ trainlm}} = 6$, $n_{0 \text{ traingdm}} = 6$; $n_{0 \text{ tansig}} = 6$, $n_{0 \text{ ReLu}} = 6$). Meanwhile, in the case of RBFNN, with two numerical factors (neuron numbers and spread numbers), $k = 2$ and $n_0 = 5$ were set. Then, the number of experiments repeated at the factorial point, axial point, and centre point is four, four, and five, respectively. Therefore, the total number of runs needed for FFNN and RBFNN were 120 and 13, respectively.

A matrix of 120 experiments for FFNN and 13 experiments for RBFNN was generated using the software package Design-Expert version 12.0. Tables S3 and S4 show the complete design matrix of the experiments performed and the obtained results of the MSE for FFNN and RBFNN, respectively. The centre points were used to determine the experimental error and reproducibility of the data.

ii. Data Transformation:

The predicted response was transformed to bring the distribution of the response variable closer to normal distribution. The Box-Cox plot was applied to improve model fitting, employing various transformations ($\lambda = -1$, $\lambda = -1$, $\lambda = 0$, $\lambda = 0.5$, and $\lambda = 1$) representing inverse, inverse square root, natural log, square root, and no transformation functions, respectively (Nazghelichi *et al.* 2011).

iii. Model Selection:

Quadratic models are established using the least square method to describe the dynamic behaviour of the network process. For k factors, the second-order model is utilized as shown in Equation (10) (Ogunjiofor & Ayodele 2023):

$$Y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k \beta_{ij} x_i x_j + \varepsilon, \quad (10)$$

where Y is the predicted response, x_i and x_j are the factors, β_i and β_j are the constants, and ε is the error term. The term β_0 is the intercept term, β_i is the linear term, β_{ii} and β_{jj} are the squared terms, and β_{ij} is the interaction term between the variables.

iv. ANOVA, Diagnostic Plot, and Model Graph:

The accuracy of the RSM model was determined using ANOVA and diagnostic plot. The ANOVA included evaluation terms such as the coefficient of determination (R^2), adjusted R^2 , predicted R^2 , adequate precision, F -value, and p -value, which were used to assess the significance of the model. The statistical test factor, F -value, was used to evaluate the significance of the model at the 95% confidence level (Nourbakhsh *et al.* 2014). The p -value served as a tool to ensure the importance of each coefficient at a specified level of significance. Generally, a p -value less than 0.050 indicated the most significance and contributed largely toward the response. The smaller the p -value, the more significant the corresponding coefficient. Values greater than 0.050 are less significant. Then, the residuals analysis is performed to diagnose the model adequacy and reliability. The plot of response and the interaction factors were obtained from the model graph of the Design-Expert version 12.0 software.

v. Set the Goal and Optimum Value:

In the numerical optimization phase, there are five possibilities of goal to construct the desirability indices including ‘maximize’, ‘minimize’, ‘target’, ‘in range’, and ‘equal to’. Desirability range is from zero to one for any given response. A value of one represents the ideal case, while zero indicates that one or more responses fall outside the desirable limit. Then, RSM will suggest several sets of the solution for hyperparameter configuration from the most desirable results to the least.

The detailed methodology of the development of the proposed ANN-RSM is depicted in Figure 3.

2.3.5. Performance evaluation

The performance evaluation of the ANN model development is measured using the MSE, RMSE, correlation coefficient (R), and determination coefficient (R^2), as given in Equations (11)–(14) (Nourbakhsh *et al.* 2014; Jawad *et al.* 2021):

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^n (\hat{y}_i - y_i)^2, \quad (11)$$

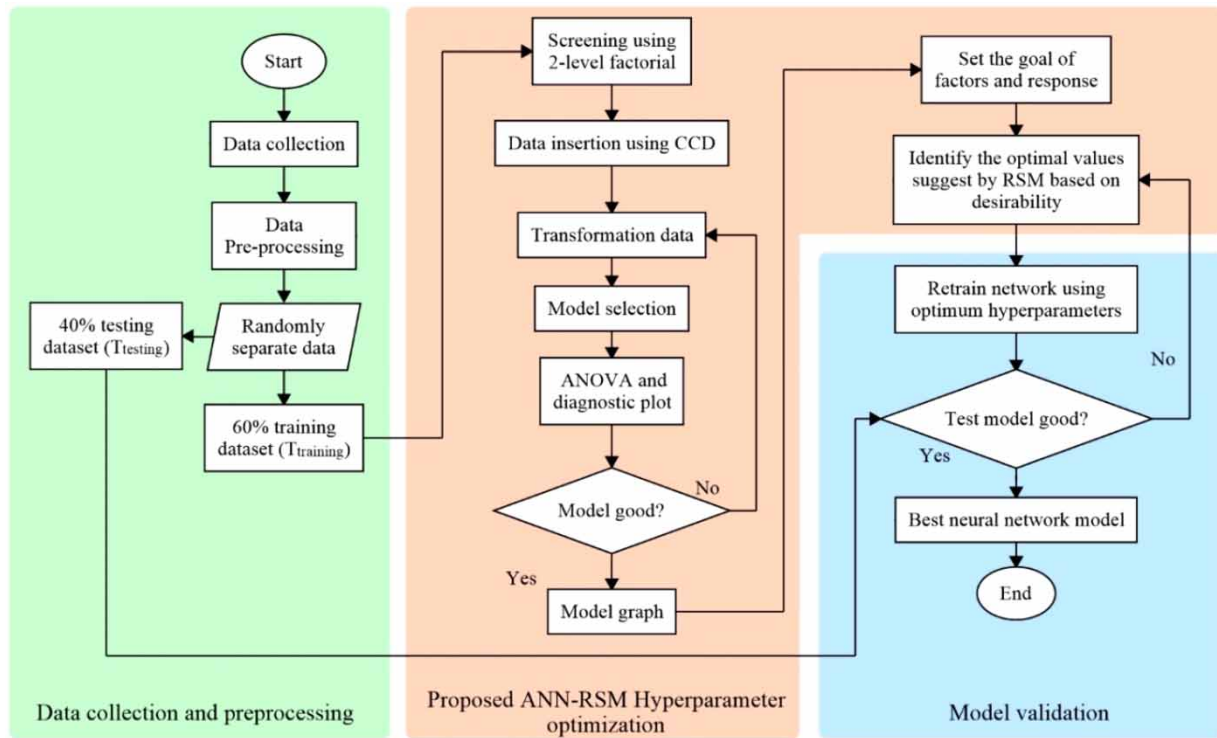


Figure 3 | Overall process of proposed ANN-RSM hyperparameters tuning method.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \tag{12}$$

$$R = \frac{\sum_{i=1}^n (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\left\{ \sum_{i=1}^n (y_i - \bar{y})^2 \right\} \left\{ \sum_{i=1}^n (\hat{y}_i - \bar{\hat{y}})^2 \right\}}}, \tag{13}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \tag{14}$$

where y_i , \hat{y}_i , \bar{y} , and $\bar{\hat{y}}$ denote the i th independent variable, i th dependent variable, the mean of independent variables, and the mean of dependent variables, respectively. The independent and dependent variables are measured by the permeate flux and predicted permeate flux of POME, respectively. Thus, the correlation coefficient was used to assess the strength of the relationship between the inputs (permeate pump and TMP) and the permeate flux output. The MSE (or RMSE) value near zero, and the R (or R^2) value near one indicate the high accuracy of the prediction model (Ibrahim & Wahab 2022; Idris *et al.* 2022).

3. RESULTS AND DISCUSSION

This section is divided into three parts. The first part describes the results of the conventional-based ANN training for selecting hyperparameters. The second part introduces the proposed RSM-based ANN training (ANN-RSM) along with the initial screening results. Finally, the model validations for permeate flux using optimal parameters from both techniques

(conventional and ANN-RSM) are presented and discussed. The accuracy of the ANN models is measured and compared using training and testing regressions.

3.1. Conventional-based HPO method

For the FFNN model, optimal performance is achieved by optimizing four numerical hyperparameters and two categorical hyperparameters: number of neurons, learning rate, momentum, number of epochs, training function, and activation function (as outlined in Section 3.1.1). Then, the RBFNN model's optimal configuration involves optimizing two numerical hyperparameters: number of neurons and number of spreads (as outlined in Section 3.1.2). This optimization is conducted using the OVAT method based on the lowest MSE value.

3.1.1. Conventional-based FFNN training

Figure 4(a) depicts the results for optimizing the number of neurons in the hidden layer. The number of neurons is varied from 1 to 30, requiring 30 runs. The legend bars represent MSE for FFNN-1 and FFNN-2 using the *trainlm* function, while the solid lines represent MSE for FFNN-3 and FFNN-4 using the *traingdm* function, respectively. In general, the bar chart shows a decreasing trend with respect to the number of neurons, while the solid line shows an upward tendency with respect to the number of neurons.

At this neuron plot, it is evident that the lowest MSE for FFNN-1, FFNN-2, and FFNN-3 were obtained at 28, 29, and 1, with MSE values of 0.0221, 0.0225, and 0.0394, respectively. Then, for FFNN-4, it was found that 3, 7, 8, 11, 13, and 19 with MSE values of 0.0352, 0.0357, 0.0382, 0.0366, 0.0343, and 0.0317, respectively, may produce a good prediction outcome.

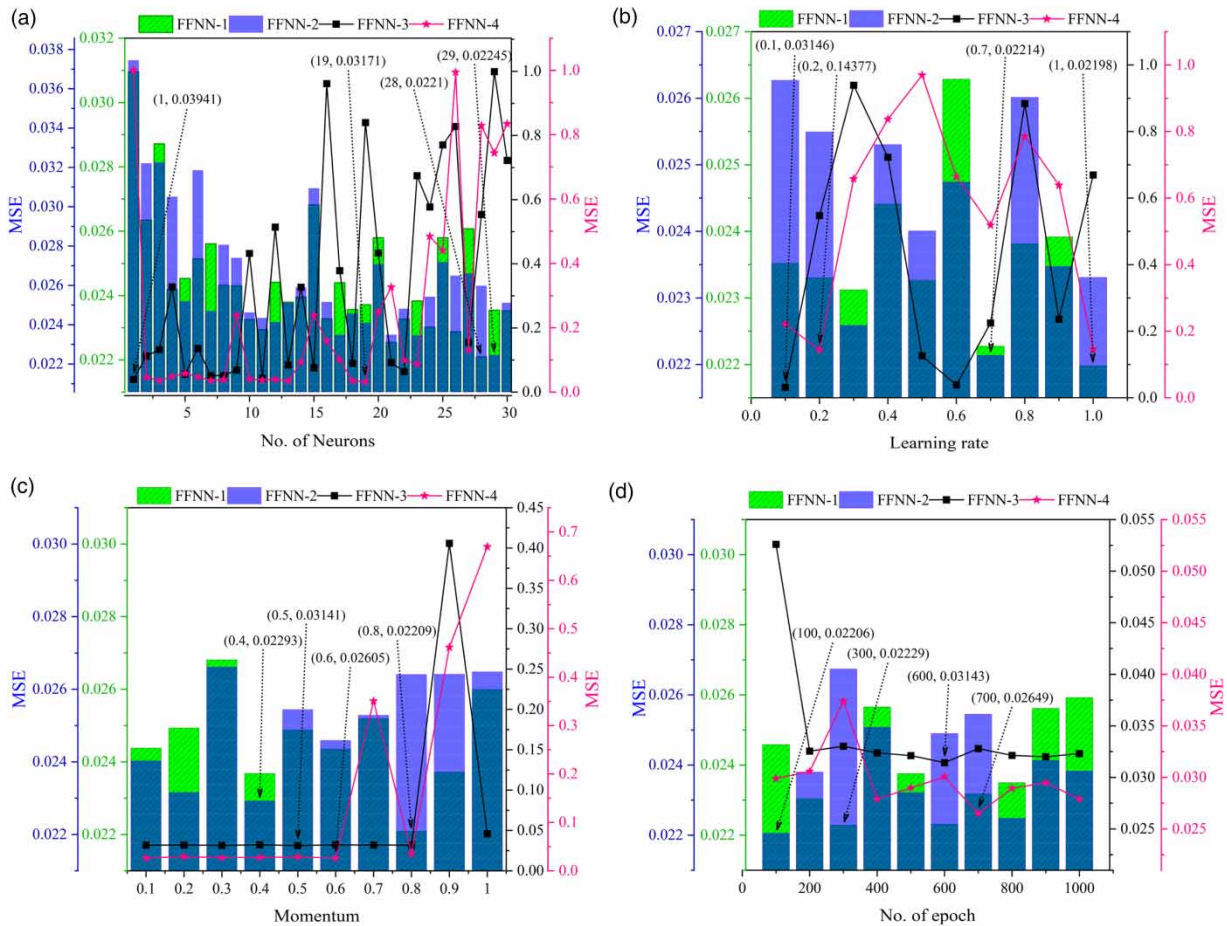


Figure 4 | MSE with varying number of (a) neurons, (b) learning rate, (c) momentum, and (d) epochs.

However, 11 neurons have been selected since it produced the lowest MSE value of 0.0317. It seems that the interaction of neuron number and activation function significantly influences the MSE of FFNN model development.

Figure 4(b) and 4(c) present the MSE with varying values of learning rate and momentum, respectively. In this case, both learning rate and momentum were trained with values set from 0.1 to 1, requiring 10 runs for each hyperparameter. As shown in Figure 4(b), the MSE values generated by the *traingdm* function tend to fluctuate between zero and one as the learning rate varies, compared to the *trainlm* function, which fluctuates between 0.022 and 0.027. Notably, learning rates of 0.8, 0.4, 0.1, and 0.2 produce favourable prediction outcomes for FFNN-1, FFNN-2, FFNN-3, and FFNN-4, respectively.

The optimal values of the momentum coefficient for FFNN-1, FFNN-2, FFNN-3, and FFNN-4 are depicted in Figure 4(c). At this momentum value, it is evident that 0.8 represents the optimum momentum value for FFNN-1, with an MSE value of 0.0221. For FFNN-2, the bar chart shows an increasing trend with respect to the number of momentums, and the lowest MSE was obtained at 0.4 (0.0229). Meanwhile, the MSE values of FFNN-3 and FFNN-4 exhibit a similar trend: they remain slightly consistent at the beginning, but experience an increase towards the end of the momentum range. For instance, in the case of FFNN-3, the MSE values exhibit slight consistency at the beginning, but then show a sudden increment at momentums of 0.7, 0.9, and 1. Therefore, 0.5 and 0.6 momentum values were selected for FFNN-3 and FFNN-4, resulting in MSE values of 0.0314 and 0.0261, respectively.

The MSE results for FFNN-1, FFNN-2, FFNN-3, and FFNN-4 to determine the optimal number of epochs are depicted in Figure 4(d). The MSE values for FFNN-1 and FFNN-2 exhibit random fluctuations within the range of 0.022–0.0227. Based on these results, it was determined that epoch values of 300 and 600 could potentially yield good results for FFNN-1 with a lower MSE. However, an epoch value of 300 was chosen since it resulted in the lowest MSE of 0.0223. FFNN-2 also produced the lowest MSE value (0.0221) with 100 epochs.

Regarding the momentum for FFNN-3, it is shown to have an optimal value at 700 with an MSE value of 0.0265. Furthermore, the MSE values for FFNN-4, from the first 100 to 200 epochs, experienced a sudden decrease from 0.0525 to 0.0325. Subsequently, they stabilized around 0.0325 until the end. Ultimately, FFNN-4 achieved optimal results with 600 epochs, resulting in the smallest MSE value of 0.0314.

In summary, the lowest MSE values were achieved by FFNN when utilizing the *trainlm* function for training, as opposed to *traingdm*. This is attributed to the advantages of the Levenberg–Marquardt algorithm, which include its rapid training process and its effectiveness in function fitting (nonlinear regression), leading to a lower MSE (Ibrahim & Wahab 2022).

3.1.2. Conventional-based RBFNN training

Figure 5(a) and 5(b) show the MSE with varying numbers of neurons and spreads for RBFNN training. Similar to FFNN, the number of neurons is varied from 1 to 30, requiring 30 runs. Meanwhile, the number of spreads varied from 0.5 to 3

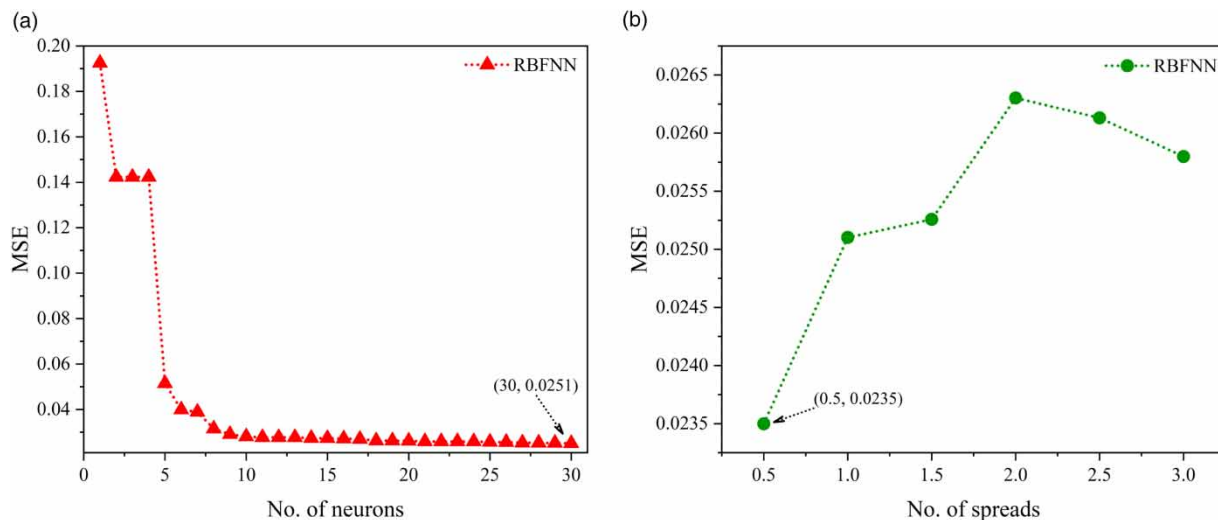


Figure 5 | MSE with varying (a) number of neurons and (b) number of spreads.

in increments of 0.5, with six runs required. It can be observed in Figure 5(a) that the MSE values for the number of neurons suddenly decrease and then slightly stabilize at minimum values starting at eight until 30, with MSEs ranging from 0.0315 to 0.0251, respectively. In the meantime, as depicted in Figure 5(b), the MSE shows an increasing trend with respect to the number of spreads, then it slightly decreases after the number of spreads is set to two. Therefore, the lowest MSE for the number of neurons and the number of spreads were obtained at 30 (0.0251) and 0.5 (0.0235), respectively.

3.2. RSM-based HPO method

This section outlines the outcomes of RSM-based FFNN and RSM-based RBFNN training, utilizing the selected scale obtained in Section 2.3.4.1.

3.2.1. RSM-based FFNN training

For this study, a quadratic model was selected to establish the relationship between FFNN hyperparameters (input) and the corresponding MSE response (output). Employing the Box-Cox method, the MSE response was transformed into the inverse of MSE (1/MSE), with α set at 1. This transformation aligns the response distribution closer to normality and enhances the model's fit to the data (Nazghelichi *et al.* 2011). In addition, inverting the MSE values is beneficial for a small value data because it will transform a metric into a larger value. This allows for a more effective representation of improved performance.

Comparisons of actual and predicted 1/MSE responses, based on 120 runs with various hyperparameter configurations using CCD, are provided in Table S3 in the Supplementary Material. The equation, in terms of coded factors, allows for predictions of the response at given levels of each factor. Typically, high levels of factors are coded as +1, while low levels are coded as -1. This coded equation is valuable for assessing the relative impact of the factors by comparing their coefficients. The quadratic model, in terms of coded factors for 1/MSE of the FFNN-RSM model, is expressed in Equation (15):

$$\begin{aligned} \frac{1}{\text{MSE}} = & 27.8796 + 4.7072 A + 1.9849 B - 2.5610 C + 1.0684 D - 11.5799E - 2.3678F - 1.0215 AB - 0.0688 AC \\ & - 0.3811AD - 3.5479 AE + 3.1920 AF - 2.7498 BC - 0.5089 BD + 0.5671 BE - 0.1462 BF + 0.4303 CD \\ & - 2.7918CE + 0.2225 CF + 1.0511 DE - 0.0169DF + 0.0231 EF - 1.9888 A^2 - 1.8045 B^2 - 1.9809 C^2 + 0.3722 D^2 \end{aligned} \quad (15)$$

where parameters A, B, C, D, E, and F represent the coded values of the number of neurons, learning rate, momentum, number of epochs, training function, and activation function, respectively. Positive values lead to an increase in the response, while negative values indicate a decrease. Due to the inherently small MSE response values of the network, the coefficients' values are relatively low (Podstawczyk *et al.* 2015).

The accuracy of the RSM model is assessed through ANOVA, as presented in Table 2. *P*-values less than 0.050 (A, B, C, E, F, AE, AF, BC, and CE) indicate significant effects on the predicted process. The analysis reveals that the linear term of training functions (E) and neuron number (A) are the most significant factors in the 1/MSE response. Following these, in order of significance, are the activation function, momentum, and learning rate. In contrast, the number of epochs shows a less substantial effect on the response, with a small *F*-value (1.16) and a *p*-value (0.2839). AB, AC, AD, BD, BE, BF, CD, CF, DE, DF, EF, A², B², C², and D² have *p*-values greater than 0.050, indicating their lesser importance in the FFNN training process.

The lack-of-fit test for the model yielded an insignificance with an *F*-value of 0.8309 and a *p*-value of 0.7237, indicating that the model suitably fits the experimental data. Model fitness was assessed using the determination coefficient (*R*²) for various models (linear, two-factorial, and quadratic). As a practical guideline, an *R*² equal to or higher than 0.75 is recommended, representing the total deviation of observed activity values from their mean (Elfghi 2016). The closer the *R*² is to unity, the better the model performs in predicting response values. Here, *R*² = 0.7741, *R*_{adj}² = 0.7140, and *R*_{pred}² = 0.6309 signify a strong agreement between observed and predicted inverse MSE values from the fitted model.

An adequate precision of 13.2814, exceeding four, affirms a reliable signal, indicating the model's accuracy. Interestingly, it appears that the number of epochs has a lesser impact on the overall performance accuracy of the FFNN training model. While the number of epochs showed significance in two-factor interactions with other factors in the two-level factorial method (Figure S3(a)), it was estimated insignificant in the RSM approach. In RSM, parameter significance is evaluated

Table 2 | Analysis of variance for the FFNN hyperparameters quadratic model

Source	Sum of squares	df	Mean square	F-value	p-Value	Remark
Model	22791.82	25	911.67	12.88	<0.0001	Significant
A: No. of neurons	1595.33	1	1595.33	22.54	<0.0001	Significant
B: Learning rate	283.68	1	283.68	4.01	0.0482	Significant
C: Momentum	472.24	1	472.24	6.67	0.0113	Significant
D: No. of epoch	82.19	1	82.19	1.16	0.2839	
E: Train. Func.	16091.19	1	16091.19	227.38	<0.0001	Significant
F: Activ. Func.	672.78	1	672.78	9.51	0.0027	Significant
AB	66.78	1	66.78	0.9437	0.3338	
AC	0.3032	1	0.3032	0.0043	0.9480	
AD	9.29	1	9.29	0.1313	0.7179	
AE	906.33	1	906.33	12.81	0.0005	Significant
AF	733.12	1	733.12	10.36	0.0018	Significant
BC	483.92	1	483.92	6.84	0.0104	Significant
BD	16.57	1	16.57	0.2342	0.6295	
BE	23.15	1	23.15	0.3272	0.5687	
BF	1.54	1	1.54	0.0217	0.8831	
CD	11.85	1	11.85	0.1674	0.6833	
CE	561.16	1	561.16	7.93	0.0059	Significant
CF	3.57	1	3.57	0.0504	0.8229	
DE	79.55	1	79.55	1.12	0.2918	
DF	0.0205	1	0.0205	0.0003	0.9865	
EF	0.0640	1	0.0640	0.0009	0.9761	
A ²	40.99	1	40.99	0.5792	0.4485	
B ²	33.75	1	33.75	0.4769	0.4915	
C ²	40.67	1	40.67	0.5746	0.4503	
D ²	1.44	1	1.44	0.0203	0.8870	
Residual	6652.29	94	70.77			
Lack of fit	5019.59	74	67.83	0.8309	0.7237	Not significant
Pure error	1632.70	20	81.64			
Cor total	29444.12	119				
R²	0.7741					
Adjusted R²	0.7140					
Predicted R²	0.6309					
Adeq. precision	13.2814					

using statistical test based on the coefficients of these terms in the regression model. If a parameter's coefficient does not significantly differ from zero, it is considered insignificant.

The residuals analysis is a powerful tool to validate the model adequacy and reliability. Figure S5(a) displays the normal probability graph versus the studentized residuals, with points lying on a straight line. It confirms that the model is appropriate because errors are typically distributed with a constant (Pashaei *et al.* 2023). Assessing the hypothesis of constant variance at specific levels is demonstrated in Figure S5(b), showing a random distribution of points above and below the x -axis between -3.6612 and $+3.6612$ for externally studentized residuals. Both ANOVA and residual analysis support the model's validity.

The perturbation graph is crucial for understanding how the response ($1/\text{MSE}$) changes in relation to alterations in its factors from the reference point, while keeping other factors at constant reference values. Figure 6(a) illustrates perturbation

plots comparing the effect of all numerical factors including the number of neurons (A), learning rate (B), momentum (C), and number of epochs (D) at the reference point, whereas categorical factors including training function and activation function were set at *trainlm* and *ReLU*, respectively.

In this case, the graph representing $1/\text{MSE}$ would behave in the opposite way compared to MSE, where the value of $1/\text{MSE}$ should be as high as possible. Observations from the perturbation plot indicate that the number of neurons exerts a relatively higher effect on the change of $1/\text{MSE}$, while learning rate and momentum exhibit a significant interaction effect, and epoch number exerts a very small effect. This is further confirmed by the single plot as shown in Figure 6(b). Lower epoch numbers indicate faster training, requiring fewer iterations for convergence. This occurs because training stops once the minimum

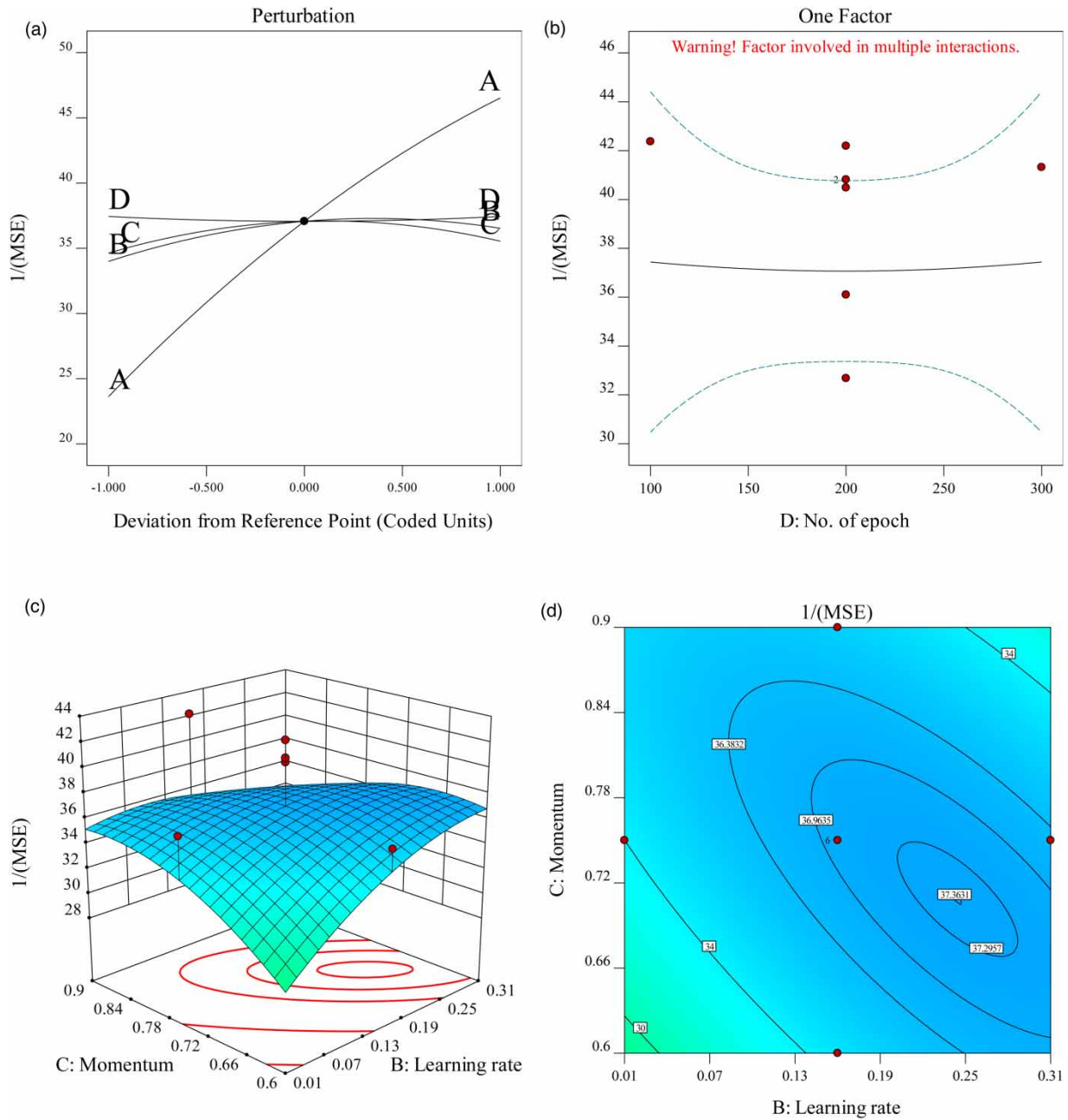


Figure 6 | (a) Perturbation plot, (b) single epoch plot versus inverse MSE, (c) interaction plot using three dimension, and (d) contour plot of learning rate and momentum versus inverse MSE.

error is achieved, often before the specified number of epochs is reached (Salam *et al.* 2021). These results align with the simulation results obtained by the conventional method in the previous section (Section 3.1.1) in Figure 4(d), where lower MSE values were produced for all FFNN models with minimal variation in their trends.

Furthermore, the relationship between 1/MSE response and the combined factors of learning rate and momentum is illustrated in a three-dimensional (3D) surface plot and a two-dimensional (2D) plot. The blue colour region indicates the highest 1/MSE values. Figure 6(c) and 6(d) demonstrates a significant increase in 1/MSE values with respect to increasing learning rate and momentum. It displays an ideal curve for learning rate and momentum. Both hyperparameters interact, leading to accelerated learning. The curve remains linear if the learning rate is too low and resembles an inverse exponential curve if the learning rate is high. However, excessively high learning rates can decay the loss rapidly and lead to getting stuck at local minima. The observation suggests that the range of maximum 1/MSE value appears to be between 0.13 and 0.31 for learning rate and momentum around 0.6–0.84.

In conclusion, these three hyperparameters (neuron number, learning rate, and momentum) play an essential role in influencing the 1/MSE value, which is consistent with the results obtained from the regression model ANOVA. Through numerical optimization in Design-Expert software, the goal for all hyperparameter factors was set at 'in range,' while the MSE response was specified at 'minimum.' Therefore, the optimum values suggested by RSM with a desirability value of one are as follows: number of neurons = 15, learning rate = 0.23, momentum = 0.65, number of epochs = 287, and *trainlm* and *ReLU* have been selected as the training and activation functions, respectively. These optimum values were applied for predicting the permeate flux of POME.

3.2.2. RSM-based RBFNN training

Regression analysis was applied to the design matrix data of RBFNN hyperparameters and fitting the response (MSE) to a quadratic model. The MSE response was transformed to natural log MSE ($\ln(\text{MSE})$). The quadratic model in terms of coded values for $\ln(\text{MSE})$ is expressed in Equation (16):

$$\ln(\text{MSE}) = -3.6265 - 1.1606 A - 0.0109 B + 0.0649 AB + 1.0952 A^2 - 0.0086 B^2 \quad (16)$$

where A and B are the coded values for the number of neurons and the number of spreads, respectively. Positive and negative signs represent synergistic and antagonistic effects between mutual interaction and individual parameters.

Table 3 presents the ANOVA of the quadratic model for the RBFNN-RSM training response. The highest F -value (21809.26) with a p -value <0.0001 confirms the model's statistical significance. An R^2 of 0.9999 indicates a strong correlation

Table 3 | Analysis of variance for the RBFNN hyperparameters quadratic model

Source	Sum of squares	df	Mean square	F-value	p-Value	Remark
Model	11.95	5	2.39	21809.26	<0.0001	Significant
A: No. of neurons	8.08	1	8.08	73738.65	<0.0001	Significant
B: No. of spreads	0.0007	1	0.0007	6.47	0.0384	Significant
AB	0.0168	1	0.0168	153.69	<0.0001	Significant
A^2	3.31	1	3.31	30225.83	<0.0001	Significant
B^2	0.0002	1	0.0002	1.87	0.2141	
Residual	0.0008	7	0.0001			
Lack of fit	0.0008	3	0.0003			
Pure error	0.0000	4	0.0000			
Cor total	11.95	12				
R^2	0.9999					
Adjusted R^2	0.9999					
Predicted R^2	0.9993					
Adeq. precision	344.6088					

between predicted and actual response values. Predicted R^2 (0.9993) closely aligns with adjusted R^2 (0.9999), underscoring model significance. The adequate precision (344.6068) implies a satisfactory signal-to-noise ratio. A ratio greater than 4 is desirable. Thus, the model can be used to navigate the design space (Nourbakhsh *et al.* 2014).

All main factors (A and B) and two-factor interaction (AB) exhibit p -values < 0.050 , signifying their significant impact on the prediction process. The analysis reveals the first-order and second-order effects of neurons (A and A^2) as the most significant terms in the $\ln(\text{MSE})$ response for the RBFNN model. Meanwhile, the quadratic term for spreads (B^2) exhibits a p -value of 0.2141, indicating less influence on the response.

Comparisons of actual and predicted $\ln(\text{MSE})$ responses, based on 13 runs with various hyperparameter configurations of RBFNN using CCD, are provided in Table S4. The actual and predicted $\ln(\text{MSE})$ values align, confirming the applicability of the quadratic model (Equation (16)) in establishing the relationship between $\ln(\text{MSE})$ and RBFNN hyperparameters.

Residual analysis further supports these findings. Figure S6(a) and S6(b) display the normal probability diagram of studentized residuals and residuals plot versus predicted response values, respectively. The points cluster near the straight line in Figure S6(a), indicating model adequacy. Figure S6(b) shows random scatter across the graph between the red line (-4.5612 and $+4.5612$), affirming the model's adequacy and unobvious error.

The perturbation scheme, comparing the effects of two hyperparameters (neurons and spreads) at the reference point, is illustrated in Figure 7(a). Neurons (A) exhibit a curvature with a significant decrease at a specific point, then increase with respect to $\ln(\text{MSE})$. Spreads (B), conversely, maintain a constant trend along their midpoint. Figure 7(b) and 7(c) plots the response ($\ln(\text{MSE})$) with respect to neuron numbers and spread numbers, respectively. Factor A exerts a relatively higher effect on changes in the reference point, while factor B has a minimal effect.

The curves in Figure 7(c) reveal a significant increase of spread numbers at beginning, and then it decreases with respect to $\ln(\text{MSE})$. These findings confirm the statistical results from Table 3, affirming the significance of all hyperparameters of RBFNN to $\ln(\text{MSE})$. Furthermore, the shift of spread numbers being insignificant in a two-level factorial design (Figure S4(b)) to becoming significant in RSM can be attributed to the fact that the two-level factorial design is not well suited for capturing the more complex relationships or curvature in the response surface of spread numbers at multiple levels (more than two).

Figure 7(d) and 7(e) presents the three-dimensional (3D) surface plot and two-dimensional (2D) contour plot of $\ln(\text{MSE})$ against combinations of neuron and spread numbers. The red regions denote the highest $\ln(\text{MSE})$, while the blue regions signify the lowest $\ln(\text{MSE})$. The minimum $\ln(\text{MSE})$ occurs within the range of 13–31 neurons. The numerical goal setting of neurons number and spreads number were set to 'in range', while 'minimum' for MSE response. Then, RSM suggests optimal values of 29 neurons and 1.5 spreads for minimizing $\ln(\text{MSE})$ with a desirability value of one. These parameters were applied to predict permeate flux in POME.

3.3. Model validation

The validation of FFNN and RBFNN models is certified based on the validity of training and testing dataset. To evaluate the validity of training and testing models, the permeate flux output plot and regression plot of developed models are discussed in this section. The comparison of the results is based on two important criteria: the lowest MSE and the highest correlation coefficient (R) values.

Figure S7(a)–S7(f) shows training results for the permeate flux outputs for the FFNN-1, FFNN-2, FFNN-3, FFNN-4, and FFNN-RSM models. These models are plotted based on the best hyperparameters obtained in Sections 3.1.1, 3.2.1, and 3.2.2. Figure S7(a) shows that the predicted datasets for FFNN-1 through FFNN-4 and FFNN-RSM training models have similar trends to the actual or measured datasets. Moreover, FFNN-2 model showed the highest accuracy with an R -value and MSE of 0.9881 and 0.0236, respectively, followed by FFNN-RSM (0.9878 and 0.0243), FFNN-1 (0.9877 and 0.0245), FFNN-4 (0.9862 and 0.0274), and FFNN-3 (0.9837 and 0.0323).

As shown in Figure 8(a)–8(f), the training model is then validated using the testing dataset, achieving good agreement with the actual dataset. The permeate flux models for FFNN-3 have a slight offset due to a slightly lower performance of MSE. All models demonstrate good prediction with comparable R and MSE results for the testing dataset. The testing performance of FFNN-2 model was higher, followed by FFNN-RSM, FFNN-1, FFNN-4, and FFNN-3 with R and MSE values of 0.9871 and 0.0256, 0.9867 and 0.0264, 0.9865 and 0.0268, 0.9854 and 0.0290, and 0.9827 and 0.0343, respectively.

Figure S8(a) shows the comparison plot between RBFNN and RBFNN-RSM permeate flux models for the SMBR filtration system during the training dataset. It can be seen that both models demonstrated good prediction with slightly higher

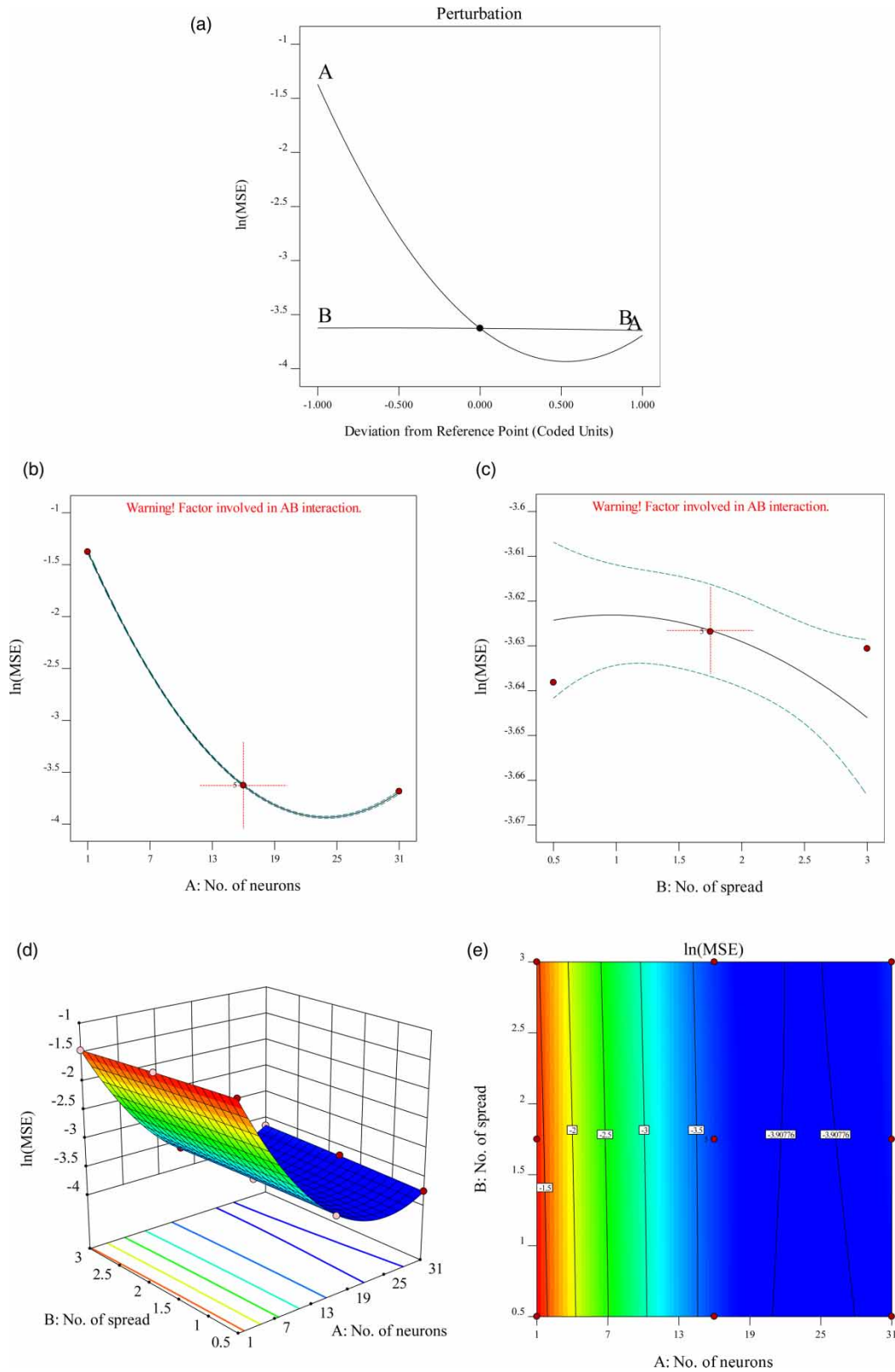


Figure 7 | (a) Perturbation plot, (b) single factor plot for neurons number, (c) single factor plot for number of spread, (d) interaction plot using 3-dimensional and (e) contour plot of neurons and spreads number versus $\ln(\text{MSE})$.

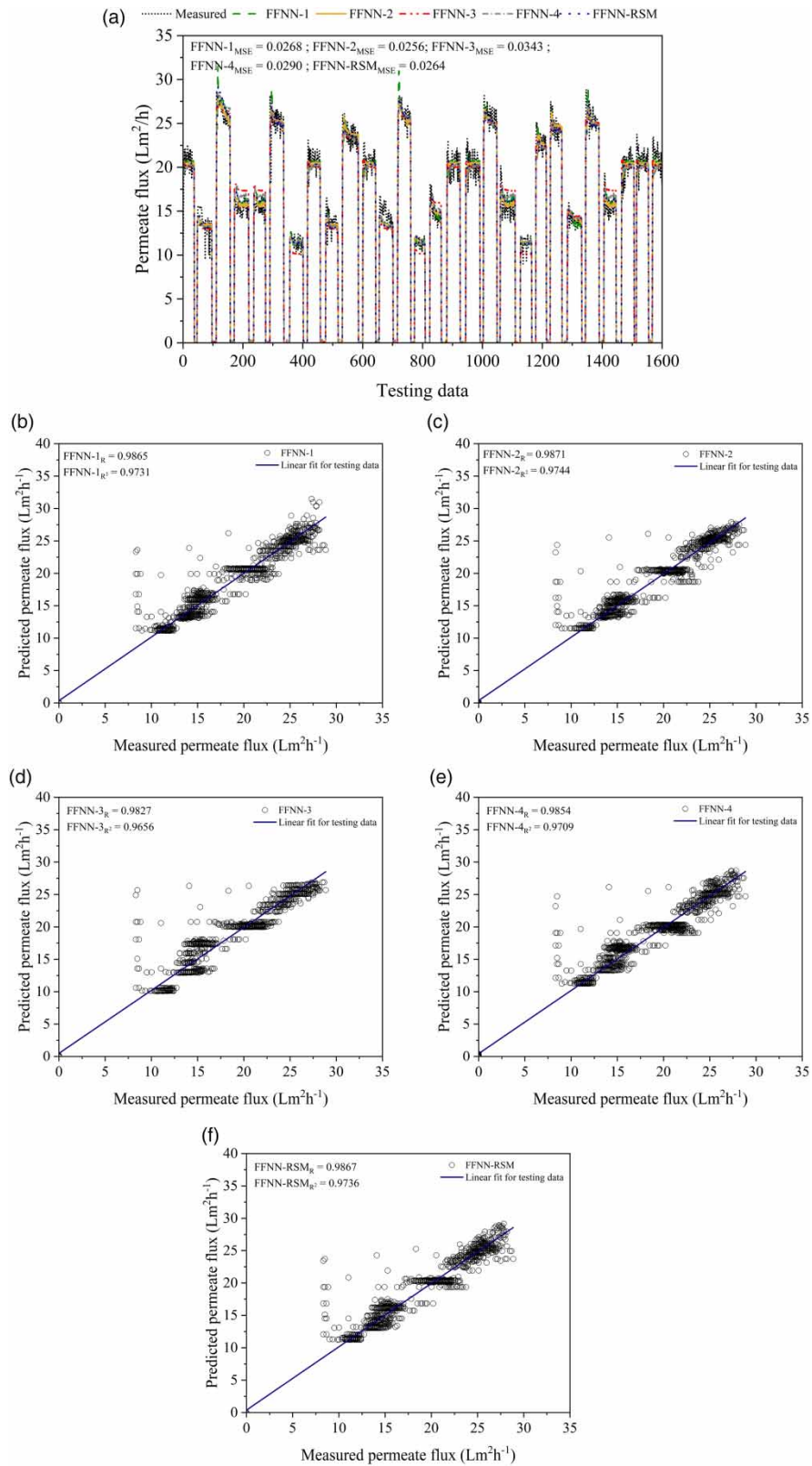


Figure 8 | (a) Permeate flux for FFNN – measured, FFNN – 1, FFNN – 2, FFNN – 3, FFNN – 4, and FFNN – RSM models. Comparison of the measured and predicted permeate flux of POME between (b) FFNN – 1, (c) FFNN – 2, (d) FFNN – 3, (e) FFNN – 4, and (f) FFNN – RSM for the testing dataset.

performance of accuracy for RBFNN than RBFNN-RSM. Overall training performance in terms of R and MSE value for both models is depicted in Figure S8(a) through Figure S8(c). The RBFNN model resulted in 0.9882 and 0.0235 for R and MSE, respectively. The R and MSE for RBFNN-RSM are 0.9873 and 0.0253, respectively.

The testing model of conventional RBFNN was then compared with the testing model of RBFNN-RSM. It can be seen from Figure 9(a) that both models demonstrated good prediction with comparable MSE results for testing dataset. The RBFNN model resulted in 0.9867 and 0.0265 for R and MSE, respectively, while the R and MSE performance of the RBFNN-RSM model are 0.9857 and 0.0284, respectively (Figure 9(a)–9(c)).

Table 4 shows the summary of overall performance for all models. The FFNN models (FFNN-1, FFNN-2, FFNN-3, FFNN-4, and FFNN-RSM) demonstrated good and comparable prediction models, with a slightly higher performance of accuracy for FFNN-2 (0.9871), followed by FFNN-RSM (0.9867), FFNN-1 (0.0865), FFNN-4 (0.9854), and FFNN-3 (0.9827). However, FFNN-RSM only required 120 runs in 150.38 s to determine the optimal value of hyperparameters. Meanwhile, the conventional method required 60 runs for each model (a total of 240 runs), with a total computational time of 253.24 s for all models, which is 59.58 s for FFNN-1, 57.68 s for FFNN-2, 72.95 s for FFNN-3, and 63.03 s for FFNN-4. Moreover, RBFNN-RSM only needed 98.82 s for 13 runs compared to the conventional RBFNN, which required 269.02 s for 36 runs to determine the optimal hyperparameters.

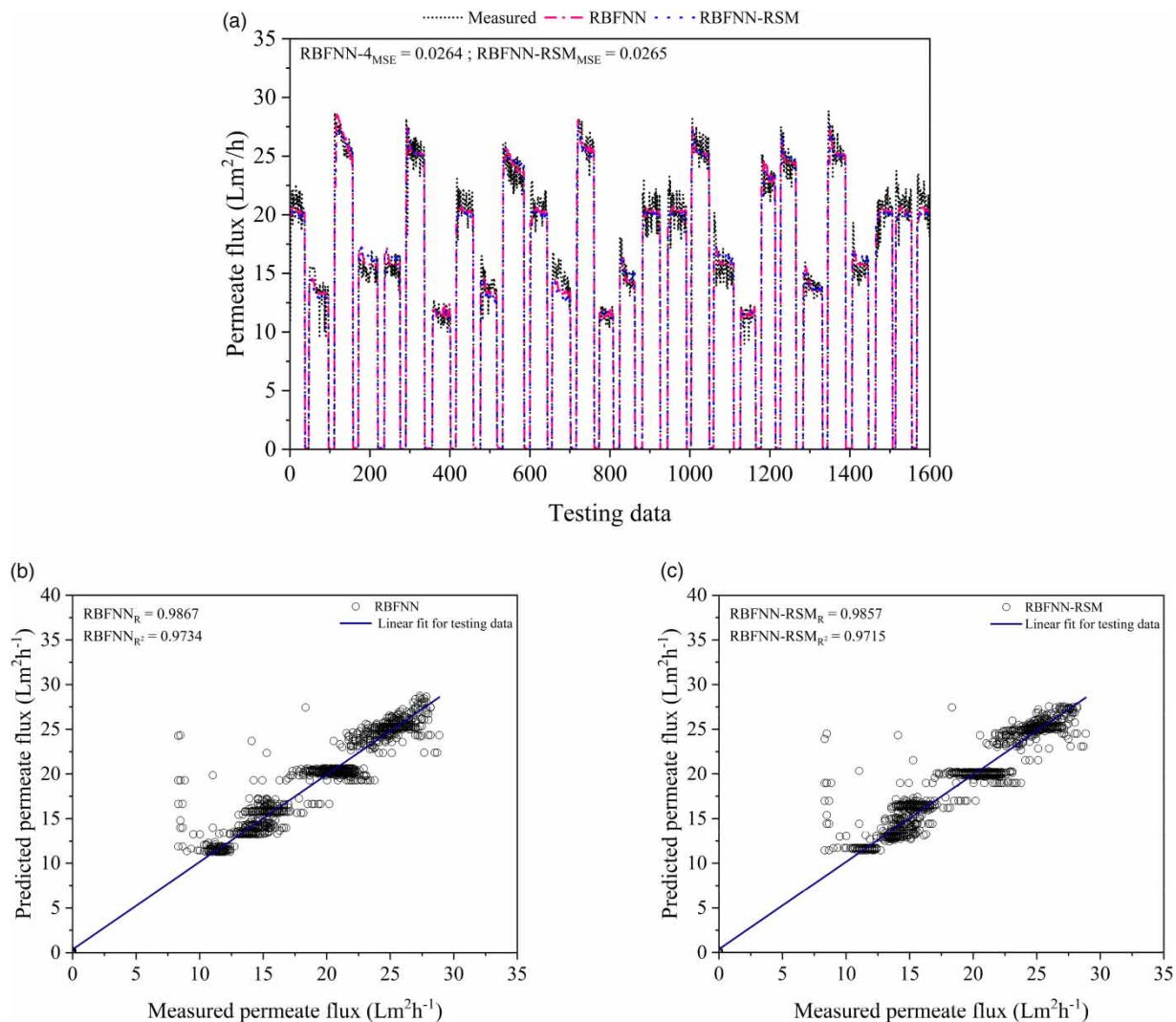


Figure 9 | (a) Permeate flux for RBFNN – measured, RBFNN – conventional, and RBFNN – RSM models. Comparison of the measured and predicted permeate flux of POME between (b) RBFNN – conventional and (c) RBFNN – RSM for the testing dataset.

Table 4 | Overall performance accuracy of the FFNN – conventional, FFNN – RSM, RBFNN – conventional, and RBFNN – RSM models for the SMBR filtration system

Model	Training				Testing				No. of run	Computational time (sec)	Total no. of run	Total computational time (sec)
	MSE	RMSE	R	R ²	MSE	RMSE	R	R ²				
FFNN-1	0.0245	0.1567	0.9877	0.9754	0.0268	0.1636	0.9865	0.9731	60	59.58	240	253.24
FFNN-2	0.0236	0.1535	0.9881	0.9764	0.0256	0.1600	0.9871	0.9744	60	57.68		
FFNN-3	0.0323	0.1797	0.9837	0.9677	0.0343	0.1851	0.9827	0.9656	60	72.95		
FFNN-4	0.0274	0.1654	0.9862	0.9726	0.0290	0.1703	0.9854	0.9709	60	63.03		
FFNN-RSM	0.0243	0.1559	0.9878	0.9757	0.0264	0.1624	0.9867	0.9736	120	150.38	120	150.38
RBFNN	0.0235	0.1533	0.9882	0.9765	0.0265	0.1628	0.9867	0.9734	36	269.02	36	269.02
RBFNN-RSM	0.0253	0.1589	0.9873	0.9747	0.0284	0.1686	0.9857	0.9715	13	93.82	13	98.82

Overall, the RBFNN required more computational time compared to FFNN, consistent with prior findings (Yang 2023). This can be attributed to the larger number of neurons necessary for RBFNN, providing more freedom for weight adjustment and subsequently increasing calculation time (Nazghelichi *et al.* 2011). Moreover, slightly better performance accuracy was observed for FFNN compared to RBFNN. However, this finding contrasts with results reported by Yang (2023) for predicted membrane distillation performance. It appears that the combination of the training algorithm (*trainlm*) and activation function (*ReLU*) used in FFNN was optimal for permeate flux prediction, exhibiting reduced computational time and high-performance accuracy. This is attributed to the advantages of the Levenberg–Marquardt algorithm, which facilitates a rapid training process with lower MSE (Ibrahim & Wahab 2022), and the superior training performance of the *ReLU* function over common activation functions (Rasamoelina *et al.* 2020).

It is noteworthy that the proposed ANN-RSM approach reduces the number of repetitions required to find optimum hyperparameters. Consequently, computational operation time can be significantly reduced compared to the time required when the OVAT method is employed in this study. Indeed, the proposed ANN-RSM emerges as a systematic, superior, and faster optimization technique for determining appropriate ANN hyperparameters when contrasted with the traditional OVAT method. Therefore, the integrated ANN and RSM approach stands as a viable alternative to the OVAT method, effectively reducing computational time and expediting ANN model development.

4. CONCLUSIONS

The FFNN and RBFNN model have been successfully developed for the permeate flux of POME during the submerged MBR filtration process. The proposed combined FFNN-RSM and RBFNN-RSM model has been developed to determine the ANN hyperparameters, and it has been compared with conventional one-variables-at-time models (FFNN-1, FFNN-2, FFNN-3, FFNN-4, and RBFNN). The model validation results showed that good validity of the training and testing models was obtained for all models. The simulation results showed the comparable performance accuracy of the proposed ANN-RSM models (FFNN-RSM and RBFNN-RSM) in relation to the conventional ANN models (FFNN-1, FFNN-2, FFNN-3, FFNN-4, and RBFNN). The optimization of the ANN hyperparameters for the FFNN-RSM models shows an improved computational time and number of repetitions – by about 41 and 50%, respectively – compared to the conventional FFNN model (FFNN-1, FFNN-2, FFNN-3, and FFNN-4). Meanwhile, an RBFNN-RSM model indicates an enhanced computational time and repetition number by 65 and 64%, respectively, compared to the conventional RBFNN model. The benefit of RSM is due to the application of the DoE, which requires less repetition of the training process but can provide a huge sum of information. In this work, the RSM successfully determines the best hyperparameters for the FFNN and RBFNN models. Moreover, it learned the significance and the relationship between the ANN hyperparameters and the MSE values even though the model involves with mix parameters (e.g., numerical and categorical). The proposed ANN-RSM technique proved an improvement over the conventional ANN models in terms of the number of repetitions, the computational time, and the estimation capabilities. While this study demonstrates the applicability of the proposed ANN-RSM for optimizing ANN hyperparameters in the permeate flux process by the SMBR system, further investigations into this methodology are essential. Exploring alternative training functions, such as Bayesian regularization (BR) and scaled conjugate gradient, along

with varying activation functions like logsig, scaled exponential linear unit, and leak rectified linear unit, is warranted. Subsequently, the methodology outlined in this study should be replicated not only in other ML models but also in diverse wastewater treatment areas. This would contribute to a comprehensive understanding of the proposed approach's generalizability and effectiveness across different contexts. Furthermore, this significant improvement can be later used in control system development to improve membrane operation.

ACKNOWLEDGEMENTS

This work was supported in part by the Universiti Teknologi Malaysia High Impact University Grant (UTMHI) vote QJ130000.2451.08G74 and the Ministry of Higher Education under Prototype Research Grant Scheme (PRGS/1/2019/TK04/UTM/02/3).

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Ahmad, M. A., Abdullah, L. C., Yaw, T. C. S. & Mohammad, A. W. 2015 Overview on application of response surface methodology (RSM) in treatment of palm oil mill effluent (POME). *Journal of Environmental Science and Engineering B* 4 (3), 111–118. doi:10.17265/2162-5263/2015.03.001.
- Alawad, W., Zohdy, M. & Debnath, D. 2018 Tuning hyperparameters of decision tree classifiers using computationally efficient schemes. In: *Proceedings of 2018 1st IEEE International Conference on Artificial Intelligence and Knowledge Engineering, AIKE 2018*, pp. 168–169. doi:10.1109/AIKE.2018.00038.
- Ali, Y. A., Awwad, E. M., Al-Razgan, M. & Maarouf, A. 2023 Hyperparameter search for machine learning algorithms for optimizing the computational complexity. *Processes* 11 (349), 1–21. doi:10.3390/pr11020349.
- Badrnezhad, R. & Mirza, B. 2014 Modeling and optimization of cross-flow ultrafiltration using hybrid neural network-genetic algorithm approach. *Journal of Industrial and Engineering Chemistry. The Korean Society of Industrial and Engineering Chemistry* 20 (2), 528–543. doi:10.1016/j.jiec.2013.05.012.
- Basha, S. M. & Rajput, D. S. 2019 Survey on Evaluating the Performance of Machine Learning Algorithms: Past Contributions and Future Roadmap. In: *Deep Learning and Parallel Computing Environment for Bioengineering Systems*. Academic Press, pp. 153–164. doi:10.1016/B978-0-12-816718-2.00016-6.
- Bunmahotama, W., Hung, W. N. & Lin, T. F. 2017 Prediction of the adsorption capacities for four typical organic pollutants on activated carbons in natural waters. *Water Research* 111, 28–40. Elsevier Ltd. doi:10.1016/j.watres.2016.12.033.
- Chen, M.-Y., Fan, M.-H., Chen, Y.-L. & Wei, H.-M. 2013 Design of experiments on neural network's parameters optimization for time series forecasting. *Neural Network World* 4 (13), 369–393.
- Duarte, E. & Wainer, J. 2017 Empirical comparison of cross-validation and internal metrics for tuning SVM hyperparameters. *Pattern Recognition Letters* 88, 6–11. Elsevier B.V. doi:10.1016/j.patrec.2017.01.007.
- Elfghi, F. M. 2016 A hybrid statistical approach for modeling and optimization of RON: A comparative study and combined application of response surface methodology (RSM) and artificial neural network (ANN) based on design of experiment (DOE). *Chemical Engineering Research and Design* 113, 264–272. doi:10.1016/j.cherd.2016.05.023.
- Feurer, M. & Hutter, F. 2019 Hyperparameter Optimization. In: *Automated Machine Learning*. The Spring. Springer, pp. 3–33. doi:10.1007/978-3-030-05318-5_1.
- Garza-Ulloa, J. 2022 Machine Learning Models Applied to Biomedical Engineering. In: *Applied Biomedical Engineering Using Artificial Intelligence and Cognitive Models*. Academic Press, pp. 175–334. doi:10.1016/B978-0-12-820718-5.00002-7.
- Ghasemi, M., Samadi, M., Soleimani, E. & Chau, K. W. 2023 A comparative study of black-box and white-box data-driven methods to predict landfill leachate permeability. *Environmental Monitoring and Assessment* 195 (7), 1–17. doi:10.1007/s10661-023-11462-9.
- Ghani, M. S. H., Haan, T. Y., Lun, A. W., Mohammad, A. W., Ngteni, R. & Yusof, K. M. M. 2018 Fouling assessment of tertiary palm oil mill effluent (POME) membrane treatment for water reclamation. *Journal of Water Reuse and Desalination* 8 (3), 412–423. doi:10.2166/wrd.2017.198.
- Hecht-Nielsen, R., 1992 Theory of the backpropagation neural network. In: *International 1989 Joint Conference on Neural Networks* (Wechsler, H., ed.). Washington, DC, USA, pp. 593–605. doi: 10.1109/IJCNN.1989.118638.

- Hemeida, A. M., Hassan, S. A., Mohamed, A. A. A., Alkhalaf, S., Mahmoud, M. M., Senjyu, T., El-Din, A. B. & Alsayyari, A. 2020 Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. *Ain Shams Engineering Journal* **11** (3), 659–675. Faculty of Engineering, Ain Shams University. doi:10.1016/j.asej.2020.01.007.
- Ibrahim, S. & Wahab, N. A. 2022 Improved artificial neural network training based on response surface methodology for membrane flux prediction. *Membranes* **12** (726), 1–25. <https://doi.org/10.3390/mem-branes12080726> Academic.
- Ibrahim, S., Wahab, N. A., Ismail, F. S. & Sam, Y. 2020 Optimization of artificial neural network topology for membrane bioreactor filtration using response surface methodology. *IAES International Journal of Artificial Intelligence (IJ-AI)* **9** (1), 117–125. doi:10.11591/ijai.v9.i1.pp117-125.
- Idris, I., Ahmad, Z., Roslee Othman, M., Sholahudin Rohman, F., Ilyas Rushdan, A. & Azmi, A. 2022 Application of artificial neural network to predict water flux from pre-treated palm oil mill effluent using direct contact membrane distillation. *Materials Today: Proceedings* **63**, S411–S417. doi:10.1016/j.matpr.2022.04.084.
- Jawad, J., Hawari, A. H. & Zaidi, S. J. 2021 Modeling and sensitivity analysis of the forward osmosis process to predict membrane flux using a novel combination of neural network and response surface methodology techniques. *Membranes* **11** (1), 1–19. doi:10.3390/membranes11010070.
- Johnson Santhosh, A., Tura, A. D., Jiregna, I. T., Gemechu, W. F., Ashok, N. & Murugan, P. 2021 Optimization of CNC turning parameters using face centred CCD approach in RSM and ANN-genetic algorithm for AISI 4340 alloy steel. *Results in Engineering* **11**, 1–9. <https://doi.org/10.1016/j.rineng.2021.100251>.
- Kasiviswanathan, K. S. & Agarwal, A. 2012 Radial basis function artificial neural network: Spread selection. *International Journal of Advanced Computer Science* **2** (11), 394–398. Available from: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.428.6010&rep=rep1&type=pdf>.
- Kechagias, J., Tsiolikas, A., Asteris, P. & Vaxevanidis, N. 2018 Optimizing ANN performance using DOE: Application on turning of a titanium alloy. In *MATEC Web of Conferences*, Vol. 178, pp. 1–5. (01017) doi:10.1051/mateconf/201817801017.
- Keong, K. C., Mustafa, M., Mohammad, A. J., Sulaiman, M. H. & Abdullah, N. R. H. 2016 Artificial neural network flood prediction for Sungai Isap residence. In: *2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*. IEEE, Shah Alam, Malaysia, pp. 236–241. doi:10.1109/I2CACIS.2016.7885321.
- Kumar, S. 2011 *Neural Networks, A Classroom Approach*, 2nd edn. Mc Graw Hill Education, Chennai.
- Kushairi, A., Loh, S. K., Azman, I., Hishamuddin, E., Ong-Abdullah, M., Izuddin, Z. B. M. N., Razmah, G., Sundram, S. & Parveez, G. K. A. 2018 Oil palm economic performance in Malaysia and R&D progress in 2017. *Journal of Oil Palm Research* **30** (2), 163–195. doi:10.21894/jopr.2018.0030.
- Li, Y.-L., Wu, J.-J., Ma, J., Li, S.-S., Xue, X., Wei, D., Shan, C.-L., Hua, X.-Y., Zheng, M.-X. & Xu, J.-G. 2022 Alteration of the individual metabolic network of the brain based on Jensen-Shannon divergence similarity estimation in elderly patients with type 2 diabetes mellitus. *Diabetes* **71** (5), 894–905. doi:10.2337/DB21-0600.
- Liew, W. L., Mohd Azraai, K., Khalida, M., Soh Kheang, L. & Affam, A. C. 2015 Conventional methods and emerging wastewater polishing technologies for palm oil mill effluent treatment: A review. *Journal of Environmental Management* **149**, 222–235. Elsevier Ltd doi:10.1016/j.jenvman.2014.10.016.
- Lujan-Moreno, G. A., Howard, P. R., Rojas, O. G. & Montgomery, D. C. 2018 Design of experiments and response surface methodology to tune machine learning hyperparameters, with a random forest case-study. *Expert Systems with Applications* **109**, 195–205. doi:10.1016/j.eswa.2018.05.024.
- Nazghelichi, T., Aghbashlo, M. & Kianmehr, M. H. 2011 Optimization of an artificial neural network topology using coupled response surface methodology and genetic algorithm for fluidized bed drying. *Computers and Electronics in Agriculture* **75** (1), 84–91. doi:10.1016/j.compag.2010.09.014.
- Nematzadeh, S., Kiani, F., Torkamanian-Afshar, M. & Aydin, N. 2022 Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases. *Computational Biology and Chemistry* **97** (107619), 1–29.
- Nourbakhsh, H., Emam-Djomeh, Z., Omid, M., Mirsaedghazi, H. & Moini, S. 2014 Prediction of red plum juice permeate flux during membrane processing with ANN optimized using RSM. *Computers and Electronics in Agriculture* **102**, 1–9. doi:10.1016/j.compag.2013.12.017.
- Ogunjiofor, E. I. & Ayodele, F. O. 2023 Utilization of response surface methodology in optimization of locally sourced aggregates. *Journal of Asian Scientific Research* **13** (1), 54–67. doi:10.55493/5003.v13i1.4771.
- Onukwuli, D. O., Esonye, C., Ofoefule, A. U. & Eyisi, R. 2021 Comparative analysis of the application of artificial neural network-genetic algorithm and response surface methods-desirability function for predicting the optimal conditions for biodiesel synthesis from *Chrysophyllum albidum* seed oil. *Journal of the Taiwan Institute of Chemical Engineers* **125**, 153–167. doi:10.1016/j.jtice.2021.06.012.
- Pakravan, P., Akhbari, A., Moradi, H., Azandaryani, A. H., Mansouri, A. M. & Safari, M. 2015 Process modeling and evaluation of petroleum refinery wastewater treatment through response surface methodology and artificial neural network in a photocatalytic reactor using poly ethyleneimine (PEI)/titania (TiO₂) multilayer film on quartz tube. *Applied Petrochemical Research* **5** (1), 47–59. doi:10.1007/s13203-014-0077-7.
- Pannakkong, W., Thiwa-Anont, K., Singthong, K., Parthanadee, P. & Buddhakulsomsiri, J. 2022 Hyperparameter tuning of machine learning algorithms using response surface methodology: A case study of ANN, SVM, and DBN. *Mathematical Problems in Engineering* **2022**, 1–17. doi:10.1155/2022/8513719.

- Pashaei, H., Mashhadimoslem, H. & Ghaemi, A. (2023) Modeling and optimization of CO₂ mass transfer flux into Pz-KOH-CO₂ system using RSM and ANN. *Scientific Reports* **13**(1), 1–25. doi:10.1038/s41598-023-30856-w.
- Podstawczyk, D., Witek-Krowiak, A., Dawiec, A. & Bhatnagar, A. 2015 Biosorption of copper(II) ions by flax meal: Empirical modeling and process optimization by response surface methodology (RSM) and artificial neural network (ANN) simulation. *Ecological Engineering* **83**, 364–379. doi:10.1016/j.ecoleng.2015.07.004.
- Probst, P., Boulesteix, A. L. & Bischl, B. 2019a Tunability: Importance of hyperparameters of machine learning algorithms. *Journal of Machine Learning Research* **20**, 1–32.
- Probst, P., Wright, M. N. & Boulesteix, A. L. 2019b Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **9** (3), 1–15. doi:10.1002/widm.1301.
- Rasamoelina, A. D., Adjailia, F. & Sincak, P. 2020 A review of activation function for artificial neural network. In: *IEEE 18th World Symposium on Applied Machine Intelligence and Informatics*. IEEE, Herl'any, Slovakia, pp. 281–286.
- Rogers, L. L. & Dowla, F. U. 1994 Optimization of groundwater remediation using artificial neural networks with parallel solute transport modeling has been successfully applied to a variety of optimization. *Water Resources Research* **30** (2), 457–481.
- Said, M., Ba-abbad, M., Rozaimah, S. & Abdullah, S. 2018 Artificial neural network (ANN) for optimization of palm oil mill effluent (POME) treatment using reverse osmosis. *Journal of Physics: Conference Series* **1095** (012021), 1–10. doi:10.1088/1742-6596/1095/1/012021.
- Salam, A., Hibaoui, A. E. & Saif, A. 2021 A comparison of activation functions in multilayer neural network for predicting the production and consumption of electricity power. *International Journal of Electrical and Computer Engineering* **11** (1), 163–170. doi:10.11591/ijece.v11i1.pp163-170.
- Stuckey, D. C. 2012 Recent developments in anaerobic membrane reactors. *Bioresource Technology* **122**, 137–148. doi:10.1016/j.biortech.2012.05.138.
- Teow, Y. H., Zulkifli, E. & Wikramasinghe, S. R. 2023 Performance and resilience of the PolyCera[®] Titan membrane for industrial wastewater treatment. *Water Science and Technology* **87** (5), 1056–1071. doi:10.2166/wst.2023.034.
- Vincent, A. M. & Jidesh, P. 2023 An improved hyperparameter optimization framework for AutoML systems using evolutionary algorithms. *Scientific Reports* **13** (1), 4737. doi:10.1038/s41598-023-32027-3.
- Winiczenko, R., Górnicki, K., Kaleta, A. & Janaszek-Mańkowska, M. 2016 Optimisation of ANN topology for predicting the rehydrated apple cubes colour change using RSM and GA. *Neural Computing and Applications* **30** (6), 1795–1809. doi:10.1007/s00521-016-2801-y.
- Xiong, J., Zuo, X., Zhang, S., Liao, W. & Chen, Z. 2019 Model-based evaluation of fouling mechanisms in powdered activated carbon/membrane bioreactor system. *Water Science and Technology* **79** (10), 1844–1852. doi:10.2166/wst.2019.167.
- Yang, C. 2023 Neural networks for predicting air gap membrane distillation performance. *Journal of the Indian Chemical Society* **100** (2), 100921. doi:https://doi.org/10.1016/j.jics.2023.100921.
- Yousif, J. H. & Kazem, H. A. 2021 Case studies in thermal engineering prediction and evaluation of photovoltaic-thermal energy systems production using artificial neural network and experimental dataset. *Case Studies in Thermal Engineering* **27** (101297), 1–13. doi:10.1016/j.csite.2021.101297.
- Yusuf, Z., Wahab, N. A. & Sudin, S. 2019 Soft computing techniques in modelling of membrane filtration system: A review. *Desalination and Water Treatment* **161**, 144–155. doi:10.5004/dwt.2019.24294.
- Zhou, Y., Cahya, S., Combs, S. A., Nicolaou, C. A., Wang, J., Desai, P. V. & Shen, J. 2019 Exploring tunable hyperparameters for deep neural networks with industrial ADME data sets. *Journal of Chemical Information and Modeling* **59** (3), 1005–1016. doi:10.1021/acs.jcim.8b00671.

First received 31 October 2023; accepted in revised form 10 March 2024. Available online 26 March 2024