PREPRINT

PERTANIKA JOURNALS

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

Comparative Study of Nonlinear Regression Models for Cleanliness Factor Prediction in Coal-Fired Utilities

Nur Aina Mohammad Abdul Aziz[1], Azura Che Soh[1]*, Asnor Juraiza Ishak[1], Nor Mohd Haziq Norsahperi[1] and Amir Redzuan Mohd Ibrahim[2]

[1]Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

[2]Sultan Azlan Shah Power Station, Tenaga Nasional Berhad, Teluk Rubiah, 32040 Seri Manjung, Perak, Malaysia

ABSTRACT

This study presents a comparative analysis of nonlinear regression models integrated with feature selection for predicting the cleanliness factor (CF) in coal-fired utilities. The models evaluated are regression trees (RT), support vector regression (SVR), ensembles of trees, and artificial neural networks (ANN). Different boiler designs introduce various operational parameters that influence cleanliness, making it more challenging to predict real-time data accurately. To enhance the model's predictive accuracy, the minimum redundancy maximum relevance (MRMR) feature selection technique was integrated, facilitating improved model performance by selecting the best subsets of variables. Model performance was assessed accordingly, where the number of selected features varies between 138 and 10. The results indicate that a combination of bagged trees and MRMR with 10 features achieved R^2 values of 0.973 for the training set and 0.976 for the test set, with a mean squared error (MSE) of 0.001 for both datasets. Compared to SVR and ANN, bagged trees consistently demonstrated superior predictive accuracy with reduced computational complexity. These findings confirm that ensemble-based models, particularly bagged trees with MRMR, provide the most accurate and computationally efficient approach for CF prediction. An accurate CF prediction creates more reliable information for a data-driven approach that solves the soot-blowing operational system. The system has the risk of either underblowing or overblowing steam during boiler cleaning. This risk, if not properly handled, may lead to more severe ash fouling and slagging issues, such as emergency shutdowns, metal corrosion, and declining heat transfer efficiency in coal-fired utilities. Overall, improving real-time boiler monitoring minimizes steam waste during soot-blowing operations.

Keywords: Cleanliness factor, coal-fired boiler, feature selection, machine learning, nonlinear regression, soot blowing optimization

ARTICLE INFO

Article history:
Received: 11 March 2025
Accepted: 04 July 2025
Published: 14 October 2025

DOI: https://doi.org/10.47836/pjst.33.6.14

E-mail addresses:
gs69737@student.upm.edu.my (Nur Aina Mohammad Abdul Aziz)
azuracs@upm.edu.my (Azura Che Soh)
asnorji@upm.edu.my (Asnor Juraiza Ishak)
nmhaziq@upm.edu.my (Nor Mohd Haziq Norsahperi)
amir.redzuan@tnb.com.my (Amir Redzuan Mohd Ibrahim)
* Corresponding author

INTRODUCTION

Coal-fired power plants generate substantial soot, leading to issues such as ash fouling and slagging in boiler sections (Wei et al., 2020). These issues not only reduce heat transfer efficiency but can also cause damage to boiler components, raising operational costs and creating additional maintenance demands. Soot-blowing mechanisms are employed to mitigate these effects, using high-temperature, high-pressure steam to remove soot from boiler walls and pipes (Kumari & Srinivasan, 2019). However, soot-blowing traditionally follows a fixed schedule, relying on operator experience, which can result in inconsistent cleaning due to human error (P. Li et al., 2023; Q. Li et al., 2020; Shi et al., 2021; Wen et al., 2022). To evaluate cleaning effectiveness, the CF is used as a metric, comparing the boiler's current condition to an ideally clean state (Shi et al., 2022).

A coal-fired power plant is known to operate with a high-dimensional dataset (Menn & Chudnovsky, 2021). Despite improvements in boiler design and technology, research on identifying key parameters that affect boiler cleanliness has been limited, particularly in utilizing data mining techniques for predictive modeling. Thota and Syed (2024) mentioned that while coal-fired utilities have much operational data, the usage of unwanted characteristics and past data is not being addressed properly, which later causes the prediction performance to degrade. Different boiler designs and configurations will introduce varying parameters that can impact the cleanliness of the boiler. Thus, relying on the expert advice based on previous boilers may overlook hidden correlations within the boiler.

This study addresses the need for feature selection methods to identify the most relevant parameters for accurate CF predictions. Feature selection algorithms help identify the best subsets of variables that enhance model performance by focusing on critical factors and reducing complexity (Bezerra et al., 2024; Jemai & Zarrad, 2023). Given that the CF ranges continuously from 0 to 1, regression learners are well-suited for this task, as they can capture continuous outputs based on the model's input features. Furthermore, considering the high-dimensional dataset, nonlinear relationships will be utilized to identify patterns among key parameters that influence boiler cleanliness.

Therefore, this research explores various nonlinear regression models that incorporate feature selection techniques, comparing their performance to determine the most effective approach for analyzing data from coal-fired boilers. The objective is to develop robust regression models that use real production data to predict the CF within the convective sections of coal-fired power plants, ultimately enhancing cleaning efficiency and operational stability.

Overview of Coal-Fired Boiler

The overview of an ultra-supercritical boiler power plant operated by one of the major utility providers in Malaysia is presented in Figure 1. The plant includes several key components: the boiler system, furnace system, boiler clean-up and start-up system, and

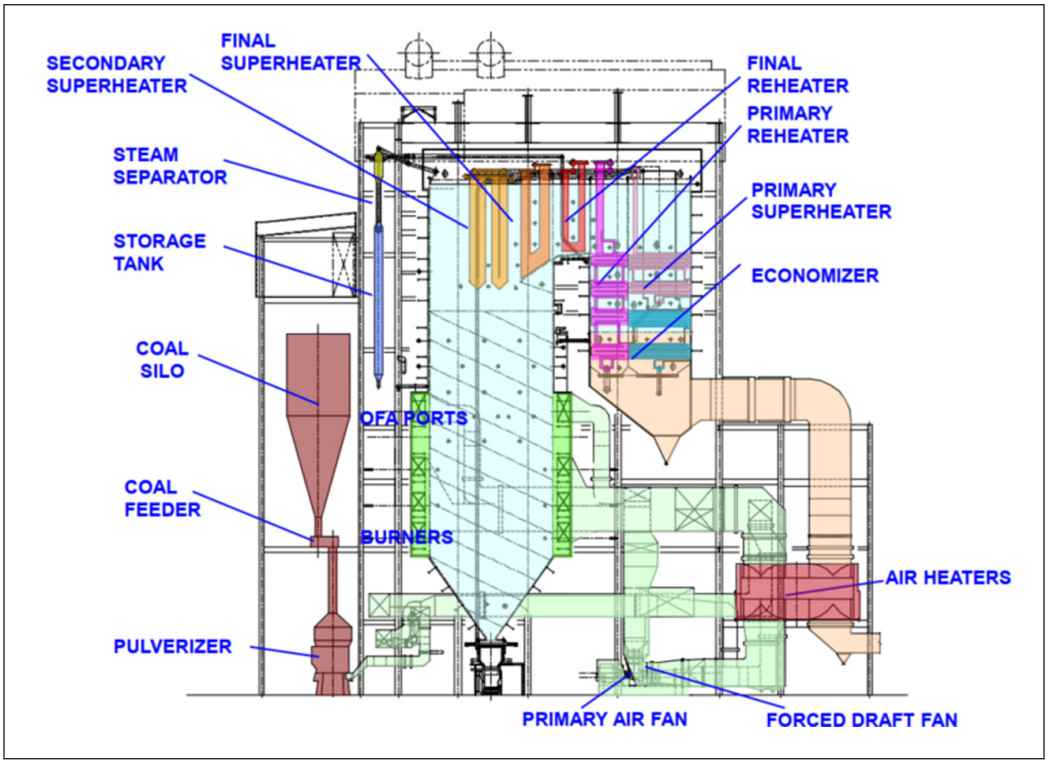

Figure 1. Overview of ultra-supercritical boiler power plant (Source: Tenaga Nasional Berhad [TNB], 2024)

air and gas system. Additionally, the metal temperature throughout the power plant is considered. The components are primarily categorized based on their respective heat exchangers.

The boiler system's pulverizer consists of six coal flow storage units, as listed in Table 1. Of these, five will be operated simultaneously, while one will serve as a backup for the coal flow. Pulverizers grind coal into a fine powder, which is subsequently used as fuel for combustion in the boiler to optimize combustion efficiency.

Table 1
Parameters of pulverizer

No.	Parameter	Unit
1.	Pulv A coal flow	t/h
2.	Pulv B coal flow	t/h
3.	Pulv C coal flow	t/h
4.	Pulv D coal flow	t/h
5.	Pulv E coal flow	t/h
6.	Pulv F coal flow	t/h

Note. Pulv = Pulverizer

The following parameters are essential for the main soot-blowing operation system, as shown in Table 2. The main steam pressure, flow, and temperature sensors measure the pressure, flow, and temperature generated within the boiler. An optical pyrometer sensor is used to monitor the flue gas components. Two types of fans are utilized: the primary air fan (PAF), which transports pulverized coal from the pulverizers to the furnace area, and the

forced draft fan (FDF), which supplies fresh air to the furnace for the combustion process. Additionally, feedwater pressure and flow sensors monitor the feedwater as it enters the boiler. The target load measures the power output required from the steam turbine, while the steam blow-off ring (SBR) steam flow sensor measures steam flow as part of the power plant's safety and operational systems.

The economizer is a component of the convective section of the boiler used to preheat feedwater before it enters the boiler. This process reduces the amount of coal required to produce high heat for steam generation. Additionally, the economizer cools down the flue gas before it is released into the atmosphere. Sensors monitor parameters, including the oxygen percentage, gas inlet and outlet temperatures, and feedwater inlet and outlet temperatures within the economizer. The detailed parameters for the economizer are listed in Table 3.

The primary superheater (PSH), located within the convective section of the boiler, serves as the initial stage in the superheating process. Its primary function is to elevate the temperature of the steam without increasing its pressure, optimizing thermal efficiency while maintaining safe operational limits. Key parameters monitored by sensors include steam inlet and outlet temperatures, as well as spray water flow, ensuring effective and controlled temperature increases. The detailed parameters for the PSH are listed in Table 4.

Table 2
Parameters of the main operation

No.	Parameter	Unit
1.	Main steam pressure	MPag
2.	Main steam flow	t/hr
3.	Main steam temperature 1	degC
4.	Optical pyrometer sensor 1	degC
5.	Optical pyrometer sensor 2	degC
6.	PAF A air volume flow	t/h
7.	PAF B air volume flow	t/h
8.	FDF A air volume flow	t/h
9.	FDF B air volume flow	t/h
10.	Feed water pressure	MPa
11.	FW flow	t/h
12.	Main steam (A) pressure	Mpag
13.	Main steam (B) pressure	MPag
14.	Target load (net)	MW
15.	SBR steam flow	t/h

Note. PAF = Primary air fan; FDF = Force draft fan; FW = Feedwater; SBR = Steam blow-off ring

Table 3
Parameters of economizer

No.	Parameter	Unit
1.	ECON A out gas O_2 anal	%
2.	ECON B out gas O_2 anal	%
3.	ECON out gas TEMP 1	degC
4.	ECON out gas TEMP 2	degC
5.	ECON inlet FW TEMP	degC
6.	ECON outlet FW TEMP	degC
7.	ECON B outlet FW TEMP	degC
8.	ECON inlet FW flow 3	t/h

Note. ECON = Economizer; O_2 = Oxygen; TEMP = Temperature; FW = Feedwater

The secondary superheater (SSH), also situated in the convective section, is the second stage of superheating and further elevates the steam's temperature to ultra-high levels. This additional temperature increase enhances the thermal energy available for power generation. Similar to the PSH, sensors monitor critical parameters such as the steam inlet and outlet

temperatures and spray water flow, ensuring precise temperature control at this advanced heating stage. The detailed parameters for the SSH are listed in Table 5.

Following the SSH, the final superheater (FSH), the third stage of superheating, maximizes the steam's temperature just before it enters the high-pressure turbine. This maximization step is crucial for achieving the highest possible efficiency in power generation. Sensors in the FSH monitor vital parameters, including steam inlets and outlet temperatures, to ensure optimal performance and safety. The detailed parameters for the FSH are listed in Table 6.

The reheater (RH) component, part of the boiler's convective section, reintroduces heat to the steam after it exits the high-pressure turbine, raising its temperature before it enters the low and intermediate-pressure turbines. This reheating process is essential for sustaining efficiency and preventing moisture formation, which could damage turbine blades. Key parameters, such as steam inlet and outlet temperatures, are continuously monitored by sensors to regulate this reheat phase effectively. The detailed parameters for the RH are listed in Table 7.

Table 4
Parameters of primary superheaters (PSH)

No.	Parameter	Unit
1.	PSH A inlet steam TEMP	degC
2.	PSH B inlet steam TEMP	degC
3.	PSH A out steam TEMP	degC
4.	PSH B out steam TEMP	degC
5.	PSH spray water flow (1st stage)	t/h

Note. TEMP = Temperature

Table 5
Parameters of secondary superheaters (SSH)

No.	Parameter	Unit
1.	SSH A inlet steam TEMP	degC
2.	SSH B inlet steam TEMP	degC
3.	SSH A out steam TEMP	degC
4.	SSH B out steam TEMP	degC
5.	SSH spray water CV A POS	%
6.	SSH spray water CV B POS	%
7.	SSH spray water flow	t/h

Note. TEMP = Temperature; CV = Control valve; POS = Position

Table 6
Parameters of final superheaters (FSH)

No.	Parameter	Unit
1.	FSH A inlet steam TEMP	degC
2.	FSH B inlet steam TEMP	degC
3.	FSH A out steam TEMP	degC
4.	FSH B out steam TEMP	degC

Note. TEMP = Temperature

Table 7
Parameters of reheaters (RH)

No.	Parameter	Unit
1.	PRH steam TEMP A	degC
2.	PRH steam TEMP B	degC
3.	FRH A out steam TEMP	degC
4.	FRH B out steam TEMP	degC
5.	HRH A steam pressure (A)	MPa
6.	HRH B steam pressure (B)	MPa
7.	CRH steam TEMP A	degC
8.	CRH steam TEMP B	degC
9.	CRH steam A pressure	MPa
11.	RH spray water CV A POS	%
12.	RH spray water CV B POS	%
13.	RH spray water flow	t/h
14.	HORIZ RH out gas TEMP 1	degC

Note. PRH = Primary reheater; FRH = Final reheater; HRH = Hot reheat; CRH = Cold reheat; HORIZ = Horizontal; TEMP = Temperature; CV = Control valve; POS = Position

The air heater (AH), located in the convective section of the boiler, preheats the combustion air by utilizing residual heat from the flue gases, thereby enhancing overall combustion efficiency. By raising the temperature of the incoming air, the AH reduces the energy required for combustion, thereby improving the boiler's thermal efficiency and reducing fuel consumption. Sensors within the AH monitor key parameters, such as the inlet gas temperature, to ensure optimal heat transfer and maintain efficient preheating conditions. This monitoring enables precise adjustments, maximizing both energy conservation and combustion efficiency. The detailed parameters for the AH are listed in Table 8.

Metal temperature sensors make up the largest number of sensors in the power plant, serving to monitor the temperature of various metal components. Due to the extensive area required for heat transfer, numerous sensors are installed within the boiler. Accurate temperature control is imperative for optimizing coal power plant efficiency. The metal temperature sensors are listed in Table 9.

Table 8
Parameters of air heaters (AH)

No.	Parameter	Unit
1.	AH A inlet gas TEMP	degC
2.	AH B inlet gas TEMP	degC

Note. TEMP = Temperature

Table 9
Metal temperature sensors

No.	Furnace	Superheaters	Reheaters
1.	MNJ:05:_BSC1_AI_0186	MNJ:05:_BSC2_AI_0281	MNJ:05:_BSC2_AI_0317
2.	MNJ:05:_BSC1_AI_0187	MNJ:05:_BSC2_AI_0282	MNJ:05:_BSC2_AI_0319
3.	MNJ:05:_BSC1_AI_0188	MNJ:05:_BSC2_AI_0283	MNJ:05:_BSC2_AI_0322
4.	MNJ:05:_BSC1_AI_0189	MNJ:05:_BSC2_AI_0284	MNJ:05:_BSC2_AI_0324
5.	MNJ:05:_BSC1_AI_0190	MNJ:05:_BSC2_AI_0285	MNJ:05:_BSC2_AI_0318
6.	MNJ:05:_BSC1_AI_0191	MNJ:05:_BSC2_AI_0286	MNJ:05:_BSC2_AI_0321
7.	MNJ:05:_BSC1_AI_0193	MNJ:05:_BSC2_AI_0287	MNJ:05:_BSC2_AI_0323
8.	MNJ:05:_BSC1_AI_0194	MNJ:05:_BSC2_AI_0289	MNJ:05:_BSC2_AI_0325
9.	MNJ:05:_BSC1_AI_0195	MNJ:05:_BSC1_AI_0165	MNJ:05:_BSC2_AI_0326
10.	MNJ:05:_BSC1_AI_0234	MNJ:05:_BSC1_AI_0166	MNJ:05:_BSC2_AI_0327
11.	MNJ:05:_BSC1_AI_0235	MNJ:05:_BSC1_AI_0167	MNJ:05:_BSC2_AI_0329
12.	MNJ:05:_BSC1_AI_0236	MNJ:05:_BSC1_AI_0244	MNJ:05:_BSC2_AI_0330
13.	MNJ:05:_BSC1_AI_0237	MNJ:05:_BSC1_AI_0245	MNJ:05:_BSC2_AI_0331
14.	MNJ:05:_BSC1_AI_0238	MNJ:05:_BSC1_AI_0246	MNJ:05:_BSC2_AI_0332
15.	MNJ:05:_BSC1_AI_0239	MNJ:05:_BSC1_AI_0259	MNJ:05:_BSC2_AI_0333
16.	MNJ:05:_BSC1_AI_0241	MNJ:05:_BSC1_AI_0260	MNJ:05:_BSC2_AI_0334
17.	MNJ:05:_BSC1_AI_0242	MNJ:05:_BSC1_AI_0261	
18.	MNJ:05:_BSC1_AI_0243	MNJ:05:_BSC1_AI_0262	
19.	MNJ:05:_BSC1_AI_0169	MNJ:05:_BSC1_AI_0263	
20.	MNJ:05:_BSC1_AI_0170	MNJ:05:_BSC1_AI_0265	

Table 9 *(continue)*

No.	Furnace	Superheaters	Reheaters
21.	MNJ:05:_BSC1_AI_0171	MNJ:05:_BSC1_AI_0266	
22.	MNJ:05:_BSC1_AI_0172	MNJ:05:_BSC1_AI_0310	
23.	MNJ:05:_BSC1_AI_0173	MNJ:05:_BSC2_AI_0308	
24.	MNJ:05:_BSC1_AI_0217	MNJ:05:_BSC2_AI_0309	
25.	MNJ:05:_BSC1_AI_0218	MNJ:05:_BSC2_AI_0310	
26.	MNJ:05:_BSC1_AI_0219	MNJ:05:_BSC2_AI_0311	
27.	MNJ:05:_BSC1_AI_0220	MNJ:05:_BSC2_AI_0313	
28.	MNJ:05:_BSC1_AI_0221	MNJ:05:_BSC2_AI_0314	
29.		MNJ:05:_BSC2_AI_0315	
30.		MNJ:05:_BSC2_AI_0316	

Note. MNJ:05: BSC1_AI_xxxx = Manjung unit 5 boiler superheater coil 1 analog input no. xxxx; MNJ:05: BSC2_AI_xxxx = Manjung unit 5 boiler superheater coil 2 analog input no. xxxx

METHODOLOGY

This research focuses on developing nonlinear regression models, enhanced with feature selection techniques, to analyze and identify the most robust model for gaining insights into new parameters within the convective sections of the boiler. The model development process begins with data collection from the power plant, followed by data preprocessing to ensure the quality and accuracy of the data. The dataset is then partitioned into training and testing sets.

A crucial part of the analysis involves computing CF, which serves as the target variable to measure boiler efficiency. Feature selection techniques are applied to identify key predictive variables, after which nonlinear regression models are developed, trained, and tested. Finally, prediction results are evaluated and compared to determine the model's accuracy and effectiveness, guiding improvements in operational efficiency for the boiler's convective components.

Dataset Preparation: Preprocessing and Partitioning Data

This dataset was extracted from a 1,000 MW ultra-supercritical coal-fired power plant in Malaysia, covering the years 2018 and 2023. Data from 2018 is considered the reference state of cleanliness for the boiler, while data from 2023 represent the current state of the boiler under investigation. According to plant engineers and technicians, the data from 2018 is considered to represent the cleanest operating conditions, as it marks the beginning of full-scale operations at the power plant. Before that, the plant was still in its trial phase. A total of 138 parameters were collected throughout the power plant. The raw data is primarily sourced from heat exchanger components to identify the parameters most

influencing boiler cleanliness. This process includes data preprocessing, CF computation, and data partitioning.

Preprocessing is essential to ensure model quality, as it helps eliminate outliers and manage missing values. Null data, where values were not recorded properly, is removed because most algorithms cannot process such entries. Outliers, defined as data with a target load below 750 MW, are also excluded. Data within the target load range is retained, as values below this threshold often correspond to the boiler's start-up phase before combustion stabilizes and are unsuitable for model development. After preprocessing, the dataset is randomly partitioned into training and testing sets, with an 80:20 ratio, where 80% of the data is used for training and 20% is reserved for testing.

Computing the Cleanliness Factor

In this study, the CF is selected as the dependent variable to assess fouling conditions in the convective section of the boiler. Equation 1 defines the calculation method for CF, which has been validated as an effective means of determining the cleanliness status of heat exchangers. CF is defined as the ratio of the real-time heat transfer rate, Q_r, to the heat transfer rate under clean conditions, Q_c (Breeding et al., 2010).

$$CF = \frac{Q_r}{Q_c} \quad [1]$$

The equations for heat transfer rates under real-time and clean conditions ($Q_{r/c}$) are fundamentally similar, differing only by the timeframe of data collection. Reference data for the clean condition were recorded in 2018, marking the start of full-scale operations at the power plant. Therefore, this timeframe is considered to represent the boiler's cleanest state. Equation 2 defines the computation of the heat transfer rate, providing an accurate average based on the inlet and outlet temperatures of the gas and steam (Madejski et al., 2018).

$$Q_{r/c} = m\,(H_{outlet} - H_{inlet}) \quad [2]$$

Here, m is the mass flow rate, and H_{outlet} and H_{inlet} represent the enthalpy values from the superheated steam table, based on the steam temperature and pressure within the boiler. This calculation ensures data consistency. Ultimately, this equation is adapted to align with the dataset, with CF designated as the target variable for training the nonlinear regression model.

Model Development

Model training in this study incorporates nonlinear regression models alongside feature selection techniques. Models with all 138 features represent those without feature selection,

as illustrated in the research model framework in Figure 2. Feature selection is performed using the MRMR algorithm, resulting in varying numbers of selected features. RT, SVR, ensembles of trees, and ANN are then developed based on the selected features.

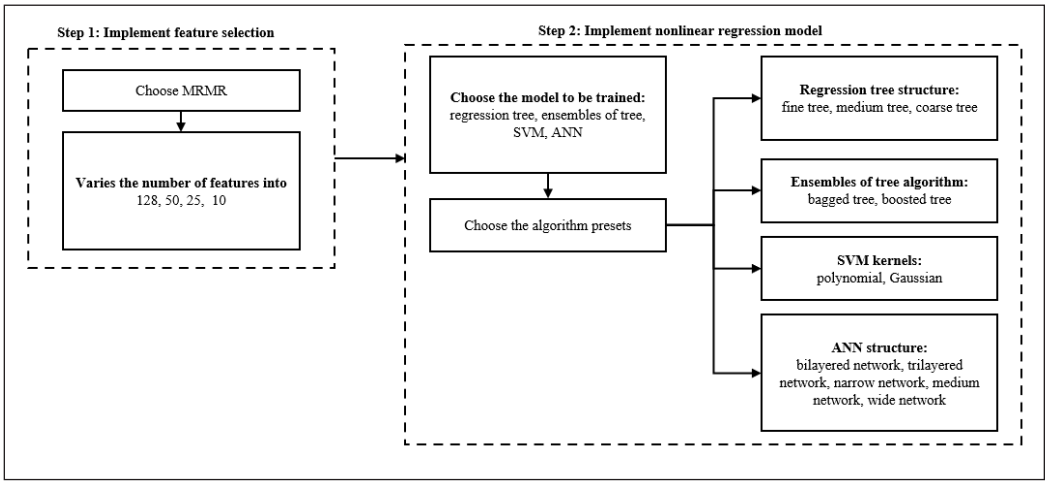

Figure 2. Structure of the research framework
Note. MRMR = Minimum redundancy maximum relevance; SVM = Support vector machine; ANN = Artificial neural network

Implementation of Feature Selection

The MRMR algorithm is a widely used feature selection method that identifies the most relevant features for predicting the target output while simultaneously minimizing redundancy among the selected features. By balancing relevance and redundancy, MRMR ensures that the chosen features provide highly informative and non-overlapping inputs to the model, ultimately enhancing the model's predictive accuracy and interpretability (Peng et al., 2005). This study adopts MRMR mainly due to its characteristics, which balance feature importance while still removing redundancy. Other techniques, such as recursive feature elimination (RFE), may have a high computational cost for large datasets (Ding et al., 2022) .

Prior to model training, feature selection is conducted using the MRMR algorithm. This step helps evaluate the impact of feature selection on model performance and stability, providing insight into the effectiveness of different feature sets. For each algorithm, feature subsets are selected in varying sizes—128, 50, 25, and 10 parameters, allowing the model to be trained with predetermined numbers of features. This approach enables a thorough assessment of the consistency and effectiveness of model performance based on different feature set sizes, supporting a robust evaluation of the role of feature selection in enhancing model reliability and generalizability.

Implementation of Nonlinear Regression Models

Models that correlate dependent and independent variables in a nonlinear manner to predict numerical outcomes effectively capture complex relationships, providing more accurate predictions of continuous data compared to linear models (Liang et al., 2022). The models selected for this study are RT, SVR, an ensemble of trees, and ANN. These models were chosen based on their suitability for handling nonlinear datasets, relatively low computational complexity, efficient processing time, and availability in the Matrix Laboratory (MATLAB). RT offers clear interpretability and can capture nonlinear relationships with minimal data preprocessing. SVR is particularly effective in high-dimensional datasets and maintains strong generalization performance due to its robustness against overfitting. An ensemble of trees combines multiple trees to enhance predictive performance, thereby improving accuracy, reducing overfitting, and increasing model robustness. Finally, ANN is widely applied in machine learning for its powerful capabilities in capturing complex and nonlinear patterns within data. It can also generalize well to unseen data. The following section provides a detailed discussion of the theory and function of each model.

RT

An RT is a decision tree used to predict continuous dependent variables. It functions by iteratively splitting the dataset into subsets based on the values of input features, aiming to partition the data into distinct regions. The process begins at the root node and continues through a series of splits until reaching the terminal leaves (Loh, 2011). Figure 3 illustrates the detailed structure of the RT splits. The tree structure shows how the CF is partitioned

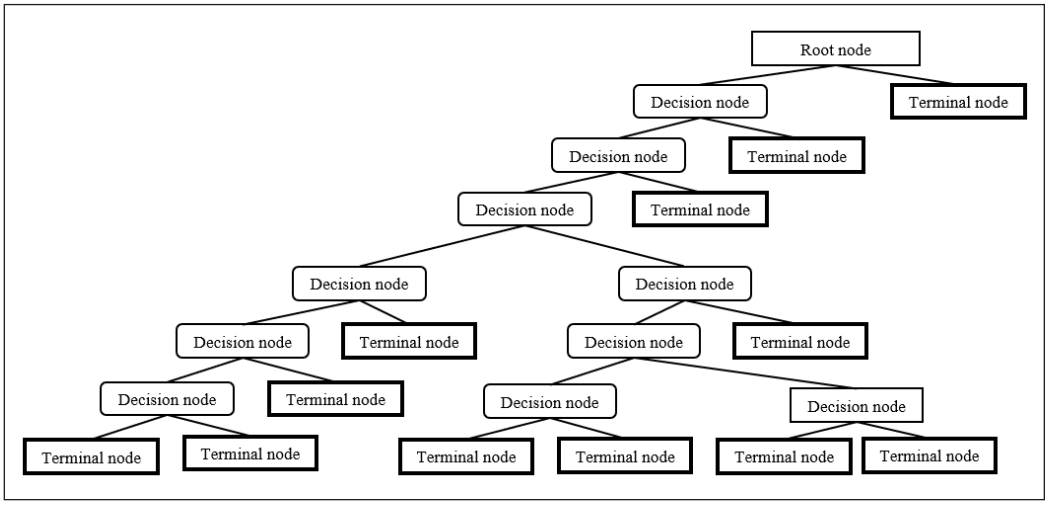

Figure 3. The regression tree structure with 12 terminal nodes

in Equation 3, where each data point (x_i) are segregated into the left and right sides based on the most relevant features (x_j) and their threshold (t), with the primary objective of minimizing prediction error.

The collected operational parameters serve as the input features, denoted as X, while the target values represent the outputs of the dataset.

$$X_{left} = \{x_i | x_j \leq t\}, X_{right} = \{x_i | x_j > t\} \quad [3]$$

The tree structure continues splitting nodes until a stopping criterion is met. Equation 4 represents the computation of predictions by applying new input features (x) within a specific region (R). The prediction, $y(x)$, is calculated as the average output over individual trees, expressed as the sum of outputs from each tree (y_i):

$$y(x) = \frac{1}{|R|} \sum y_i \quad [4]$$

There are three main types of RT structures based on the number of terminal nodes: fine, medium, and coarse trees. Fine trees create a structure with four terminal nodes, capturing complex patterns in the data but often resulting in overfitting, which can reduce the model's generalizability. Medium trees create a structure with 12 terminal nodes, providing a balance between complexity and generalization. Coarse trees, with 36 terminal nodes, are less prone to overfitting but may be less accurate in capturing patterns within the data (Cai et al., 2021).

Ensemble of Trees

Ensemble tree methods are regression algorithms that combine multiple decision tree models to achieve greater accuracy than a single predictive model (Ghiasi & Zendehboudi, 2021) . The methods applied in this paper are the bagged tree and the least-squares boosting tree (LSBoost). The bagged tree method employs bootstrap aggregation, also known as bagging, to enhance predictive performance and reduce variance by averaging the results of multiple decision trees. It creates an ensemble of trees by training multiple trees independently on random subsets of the training data. Equation 5 describes the final prediction $y(x)$, where the predictions from each individual tree, $y_m(x)$, are averaged. The random subsets are generated through bootstrap sampling, where samples are partitioned randomly with replacement, and M is the number of trees trained. Each tree is trained to minimize prediction error on its respective sample.

$$y(x) = \frac{1}{M} \sum_{m=1}^{M} y_m(x) \quad [5]$$

Boosted trees utilize the LSBoost algorithm to minimize the residual sum of squares between predicted and actual outputs, thereby enhancing model accuracy. This method is particularly suitable for weak learners because it improves the overall model performance. Each new model is trained sequentially to correct the errors made by previous models. While this algorithm reduces bias, it can be prone to overfitting.

Equation 6 represents the ensemble models $F_m(x)$, where $F_o(x)$ is the initial model that is calculated as the mean of the target values. The learning rate, denoted as v, partially influences the contribution of each weak learner. The function $h_m(x)$ represents the weak learner, which computes the difference between the actual CF and the current model predictions.

$$F_m(x) = F_0(x) + \sum_{m=1}^{M} v h_m(x) \qquad [6]$$

SVR

Support vector machines (SVM) have an extension known as SVR, which creates an optimal hyperplane based on the support vectors. Figure 4 illustrates the mechanism of SVR, where the hyperplane is a multidimensional surface that separates the data points, and the support vectors are the data points nearest to the hyperplane (Marafino et al., 2014). The goal of this algorithm is to maximize the margin that distinctly separates the data points. Equation 7 describes the function of parallel hyperplanes ($y(x)$).

$$f(\mathrm{x}) = w\mathrm{x} + b \qquad [7]$$

where w representing the margin and b is a constant.

The decision boundary is defined as $f(x) = 0$, which completely separates the two classes. Data points with $f(x) > 0$ belong to the red class, while data points with $f(x) < 0$ belong to the green class. Kernel functions are used to transform the data into a higher-dimensional space, aiming to model a nonlinear relationship between the input features and the CF. The kernels can be categorized into Gaussian SVM and polynomial SVM. The variations of kernels implemented in this study include quadratic, cubic, fine Gaussian, medium Gaussian, and coarse Gaussian.

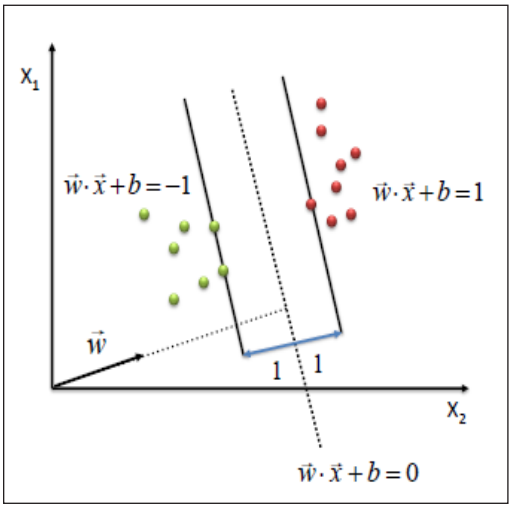

Figure 4. The illustration of the support vector regression mechanism (Marafino et al., 2014)
Note. $\vec{w}$ = Weight vector; $\vec{x}$ = Data point vector; b = Bias

ANN

An ANN is an algorithm inspired by the human brain's mechanisms, enabling it to handle non-linear relationships effectively. Figure 5 illustrates the architecture of a neural network, where the input layer contains the attributes of the input features, with each neuron representing a single feature. The output layer contains the neuron that holds the target value, which in this case is the CF values. In the case of a continuous target value or regression learner, the output layer consists of only one neuron. Between the input and output layers is the hidden layer, where the network learns to improve prediction accuracy. As the network expands, its capacity to learn complex relationships increases; however, this also raises the risk of overfitting. Backpropagation is employed to minimize the error between predicted and actual outputs (Comito & Pizzuti, 2022).

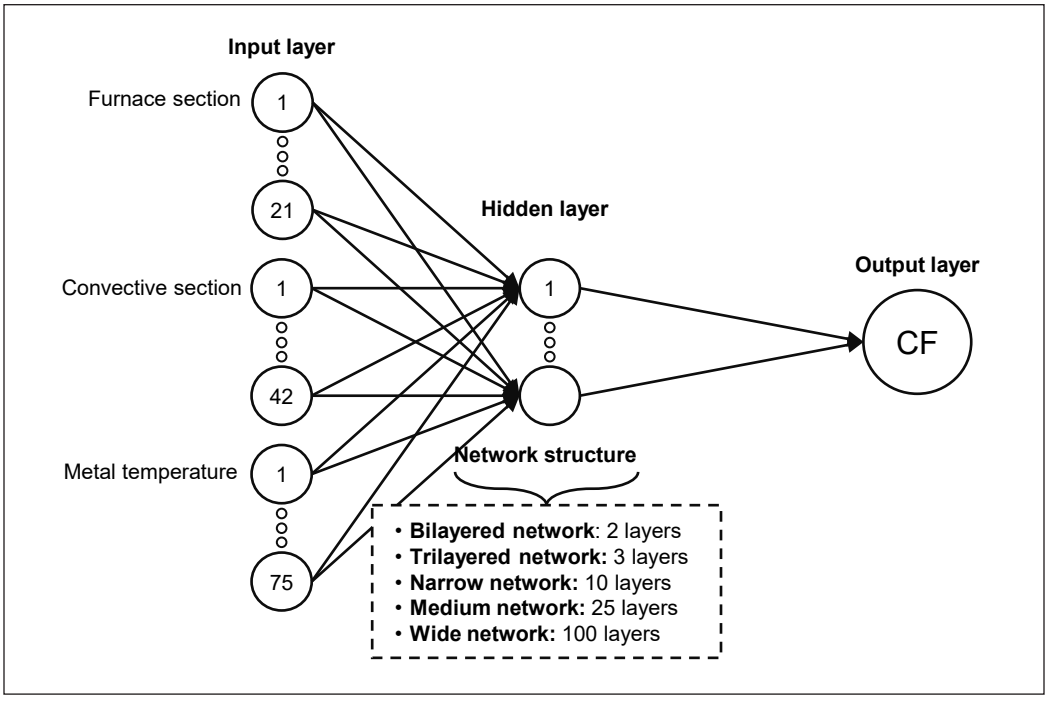

Figure 5. Artificial neural network
Note. CF = Cleanliness factor

Model Validation

After training the model, the prediction accuracy of the training and testing datasets has been analyzed using R^2 and MSE as key metrics. R^2 indicates the extent to which the variance in the dependent variable can be predicted based on the independent variable, with values closer to 1 indicating better model performance. Meanwhile, MSE measures

the average squared difference between predicted and actual values; thus, a smaller MSE indicates better performance. The equations for R^2 and MSE as shown in Equations 8 and 9.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad [8]$$

where y is the real plant data set or actual value, $\hat{y}$ is the prediction data set, and i is the number of data until n, the total number of samples.

$$R^2 = 1 - \frac{R_{RSS}}{R_{TSS}} \quad [9]$$

where R_{RSS} is the residual sum of squares between actual and predicted values and R_{TSS} is the sum of squared differences between actual values and the mean of actual values.

In addition, prediction speed and training time are also considered to provide insight into model efficiency. Prediction speed refers to the number of predictions the model can make per second, with higher values indicating better performance. Training time, on the other hand, is the amount of time required to train the model, with more complex models generally taking longer to train.

For this study, the analysis focuses on identifying the model with the highest R^2, lowest MSE, fewest selected features, highest prediction speed, and lowest training time. This criterion is established to find the best nonlinear regression models that are compatible with feature selection methods and capable of creating a robust nonlinear regression model.

RESULTS AND DISCUSSION

This section presents the simulation results and analyzes the performance of various regression models, including RT, SVR, ensembles of trees, and ANN. Each model is evaluated in detail, focusing on its strengths and weaknesses in predicting outcomes. The section concludes with a comparative analysis of nonlinear regression models, comparing the effectiveness of these models based on evaluation metrics.

Performance of the RT Model

The overall RT model demonstrates excellent performance on both datasets, as shown in Table 10. Across all trees, the MSE values remain consistently low, ranging from 0.001 to 0.002, indicating that the models fit the training data very well. The medium tree model exhibits the best performance, with R^2 values ranging from 0.973 to 0.958 for training and from 0.979 to 0.958 for testing.

In comparison, the fine tree model shows a gradual decline in performance, with R^2 values ranging from 0.973 to 0.953 for training and from 0.978 to 0.956 for testing. Lastly,

Table 10
Performance results of the regression tree

Preset	No. of features	MSE Training)	R^2 (Training)	MSE (Test)	R^2 (Test)
Fine tree	138	0.001	0.972	0.001	0.975
Fine tree	128	0.001	0.973	0.001	0.975
Fine tree	50	0.001	0.971	0.001	0.978
Fine tree	25	0.001	0.970	0.001	0.978
Fine tree	10	0.001	0.957	0.001	0.956
Medium tree	138	0.001	0.973	0.001	0.979
Medium tree	128	0.001	0.973	0.001	0.979
Medium tree	50	0.001	0.971	0.001	0.976
Medium tree	25	0.001	0.972	0.001	0.976
Medium tree	10	0.001	0.958	0.001	0.958
Coarse tree	138	0.001	0.969	0.001	0.975
Coarse tree	128	0.001	0.969	0.001	0.976
Coarse tree	50	0.001	0.968	0.001	0.968
Coarse tree	25	0.001	0.967	0.001	0.973
Coarse tree	10	0.002	0.953	0.002	0.954

Note. MSE = Mean squared error

the coarse tree model delivers the lowest performance, with R^2 values ranging from 0.969 to 0.953 for training and from 0.975 to 0.954 for testing.

The worst-performing model is the coarse tree with 10 selected features, achieving R^2 values of 0.953 for training and 0.954 for testing, alongside MSE values of 0.002 for both datasets. In contrast, the medium tree model with 10 features shows significantly better performance, attaining an R^2 value of 0.958 and an MSE value of 0.001 for both datasets. This indicates that the medium tree model achieves optimal performance while maintaining computational efficiency with a smaller subset of features.

Performance of the Ensemble of Trees

The simulation results presented in Table 11 indicate that the bagged trees exhibit exceptional performance on both datasets, with R^2 values ranging from 0.973 to 0.983 for training and from 0.976 to 0.990 for testing, along with MSE values of 0.001 for training and between 0 and 0.001 for testing. This performance suggests that the model is highly effective at fitting the dataset with minimal variation across feature subsets.

In comparison, the performance of the boosted trees is lower, with R^2 values ranging from 0.874 to 0.891 for training and from 0.872 to 0.899 for testing, as well as MSE values of 0.004 for training and between 0.003 and 0.004 for testing. This indicates that the boosted tree is less effective for this specific dataset, although it still performs adequately.

Table 11
Performance results of the ensemble of trees

Preset	No. of features	MSE (Training)	R^2 (Training)	MSE (Test)	R^2 (Test)
Bagged tree	138	0.001	0.983	0	0.990
Bagged tree	128	0.001	0.983	0	0.990
Bagged tree	50	0.001	0.982	0	0.989
Bagged tree	25	0.001	0.981	0	0.988
Bagged tree	10	0.001	0.973	0.001	0.976
Boosted tree	138	0.004	0.891	0.003	0.899
Boosted tree	128	0.004	0.891	0.003	0.898
Boosted tree	50	0.004	0.89	0.003	0.896
Boosted tree	25	0.004	0.887	0.004	0.893
Boosted tree	10	0.004	0.874	0.004	0.872

Note. MSE = Mean squared error

Overall, the boosted tree model with 10 features shows the worst performance, achieving R^2 values of 0.874 for training and 0.872 for testing, along with an MSE value of 0.004 for both datasets. In contrast, the bagged trees model with 10 features demonstrates excellent performance, achieving R^2 values of 0.973 for training and 0.976 for testing, with an MSE value of 0.001 for both datasets.

Performance of the SVR

For this study, two types of kernels are used: Gaussian and polynomial kernels. The Gaussian kernels consist of fine Gaussian, medium Gaussian, and coarse Gaussian, while the polynomial kernels comprise quadratic and cubic kernels. Based on the results documented in Table 12, Gaussian kernels demonstrate better performance compared to polynomial kernels. This is evident from the R^2 values ranging from 0.798 to 0.990 for training and from 0.785 to 0.982 for testing, along with MSE values ranging from 0.001 to 0.008 on both datasets.

In comparison, polynomial kernels exhibit a drastic performance reduction, with R^2 values ranging from -236.941 to 0.898 for training and from -35.709 to 0.966 for testing, alongside MSE values ranging from 0.003 to 56.522 for training and from 0.001 to 1.282 for testing. Despite the polynomial kernels showing better performance within the test datasets, the significant differences between the training and testing results suggest potential issues of overfitting or underfitting when feature selection is applied.

Among Gaussian kernels, the medium Gaussian kernel shows excellent performance, with R^2 values ranging from 0.965 to 0.985 for training and from 0.966 to 0.938 for testing, along with MSE values ranging from 0 to 0.001 for training and from 0.001 to 0.002 for testing. Following that, the coarse Gaussian kernel achieves R^2 values ranging

Table 12
Performance results of the support vector regression

Preset	No. of features	MSE (Training)	R^2 (Training)	MSE (Test)	R^2 (Test)
Fine Gaussian	138	0.007	0.798	0.008	0.785
Fine Gaussian	128	0.006	0.836	0.006	0.823
Fine Gaussian	50	0.004	0.878	0.005	0.867
Fine Gaussian	25	0.004	0.880	0.004	0.873
Fine Gaussian	10	0.004	0.893	0.004	0.887
Medium Gaussian	138	0	0.989	0.001	0.966
Medium Gaussian	128	0	0.990	0.001	0.964
Medium Gaussian	50	0	0.990	0.001	0.958
Medium Gaussian	25	0.001	0.984	0.002	0.951
Medium Gaussian	10	0.001	0.970	0.002	0.938
Coarse Gaussian	138	0.001	0.979	0.001	0.982
Coarse Gaussian	128	0.001	0.980	0.001	0.982
Coarse Gaussian	50	0.001	0.979	0.001	0.982
Coarse Gaussian	25	0.001	0.971	0.001	0.970
Coarse Gaussian	10	0.002	0.943	0.002	0.937
Quadratic	138	0.007	0.795	0.001	0.966
Quadratic	128	0.128	-2.781	0.002	0.950
Quadratic	50	0.003	0.898	0.007	0.787
Quadratic	25	0.198	-4.838	0.022	0.374
Quadratic	10	0.006	0.835	0.053	-0.527
Cubic	138	0.185	-4.451	0.048	-0.364
Cubic	128	6.109	-178.741	0.053	-0.508
Cubic	50	56.522	-1662.100	0.067	-0.919
Cubic	25	8.087	-236.941	0.015	0.570
Cubic	10	2.706	-78.625	1.282	-35.709

Note. MSE = Mean squared error

from 0.980 to 0.943 for training and from 0.982 to 0.937 for testing, with MSE values ranging from 0.001 to 0.002 for both datasets. Lastly, the fine Gaussian kernel delivers lower performance, with R^2 values ranging from 0.798 to 0.893 for training and from 0.785 to 0.887 for testing, alongside MSE values ranging from 0.004 to 0.008 for both datasets.

The significance of feature selection in improving model accuracy is evident throughout the evaluations, particularly in the SVR models with fine Gaussian kernels. The models

incorporating feature selection consistently achieve higher R^2 values, indicating a better fit and enhanced predictive performance. These findings highlight the critical role of feature selection in boosting the efficiency and reliability of machine learning models.

The worst-performing model is the cubic kernel with 25 features selected, achieving R^2.values of -236.941 for training and 0.570 for testing, and MSE values of 8.087 for training and 0.015 for testing. In comparison, the best-performing model is the medium Gaussian kernel with 10 features selected, achieving R^2 values of 0.970 for training and 0.938 for testing, with MSE values of 0.001 for training and 0.002 for testing.

Performance of the ANN

Overall, all network structures perform exceptionally well compared to other algorithms, as shown in Table 13. This is evident with R^2 values ranging from 0.91 to 0.99 for training, 0.817 to 0.993 for testing, and MSE values ranging from 0 to 0.006 in both datasets. Among the various network structures, it is particularly challenging to select the most suitable one. Despite this, the worst-performing network structure is the trilayered network, which has R^2 values ranging from 0.91 to 0.986 for training, 0.817 to 0.984 for testing, and MSE values ranging from 0 to 0.006 in both datasets.

The most accurate models for each algorithm, regardless of the number of feature subsets, have been identified. The bilayered network with 128 selected features achieves R^2 values of 0.987 for training and 0.966 for testing, with MSE values of 0 for training and 0.001 for testing. The trilayered network with 128 selected features achieves R^2 values of 0.987 for training and 0.817 for testing, with MSE values of 0 for training and 0.006 for testing. The narrow network with 128 features achieves R^2 values of 0.989 for training and 0.988 for testing, with MSE values of 0 for both datasets. The medium network with 128 features achieves R^2 values of 0.988 for training and 0.993 for testing, along with MSE values of 0 for both datasets as well. Lastly, the wide network with 128 features achieves R^2 values of 0.99 for training and 0.993 for testing, with MSE values of 0 for both datasets.

Despite the models' superior accuracy, they are not considered the optimal choice due to their reliance on numerous variables, which increases the risk of overfitting and reduces the models' generalizability. Overall, all network structures exhibit a great balance in performance, but their effectiveness gradually declines as the number of features is reduced to 10.

The worst-performing model is the trilayered model with 10 selected features, achieving R^2 values of 0.91 for training and 0.978 for testing, along with MSE values of 0.003 for training and 0.001 for testing. In comparison, the best-performing model is the medium neural network with 10 selected features, achieving R^2 values of 0.961 for training and 0.988 for testing, with MSE values of 0.001 for both datasets.

Table 13
Performance results of the artificial neural network

Preset	No. of features	MSE (Training)	R^2 (Training)	MSE (Test)	R^2 (Test)
Bilayered	138	0.001	0.979	0.004	0.884
Bilayered	128	0	0.987	0.001	0.966
Bilayered	50	0.001	0.979	0	0.991
Bilayered	25	0.001	0.967	0.001	0.972
Bilayered	10	0.002	0.944	0.001	0.984
Trilayered	138	0	0.986	0.001	0.984
Trilayered	128	0	0.987	0.006	0.817
Trilayered	50	0.001	0.984	0.003	0.918
Trilayered	25	0.001	0.973	0.001	0.971
Trilayered	10	0.003	0.910	0.001	0.978
Narrow	138	0.001	0.983	0.001	0.959
Narrow	128	0	0.989	0	0.988
Narrow	50	0.001	0.978	0.001	0.970
Narrow	25	0.001	0.973	0.001	0.969
Narrow	10	0.001	0.957	0.001	0.978
Medium	138	0	0.987	0.001	0.964
Medium	128	0	0.988	0	0.993
Medium	50	0.001	0.982	0.001	0.978
Medium	25	0.001	0.967	0.001	0.984
Medium	10	0.001	0.961	0.001	0.980
Wide	138	0	0.986	0.001	0.959
Wide	128	0	0.990	0	0.993
Wide	50	0.001	0.985	0	0.992
Wide	25	0.001	0.976	0	0.987
Wide	10	0.002	0.946	0.001	0.975

Note. MSE = Mean squared error

Comparative Analysis of Nonlinear Regression Models

The best model from each algorithm has been selected for evaluation, as summarized in Table 14 for detailed comparison and analysis. The final training step compared model performance based on the highest R^2 value, lowest MSE, and fewest selected features, providing a robust measure of each model's predictive capability in relation to feature selection.

The bagged tree model performed exceptionally well in terms of overall accuracy, achieving the highest R^2 value of 0.973 (training) and 0.976 (test), along with low MSE values of 0.001 for both datasets. In Figure 6, there are a few outliers in the high-value

Table 14
Summary of the best-performing model

Model	No. of features	MSE (Training)	R^2 (Training)	MSE (Test)	R^2 (Test)
Bagged tree	10	0.001	0.973	0.001	0.976
Medium NN	10	0.001	0.961	0.001	0.980
Medium Gaussian SVR	10	0.001	0.970	0.002	0.938
Medium tree	10	0.001	0.958	0.001	0.958

Note. MSE = Mean squared error; NN = Neural network; SVR = Support vector regression

range, but they are minimal, indicating that the bagged tree model provides highly accurate predictions with only a small margin of error. The regression line equation of this model is $y = 0.9675x + 0.0378$, and achieves the highest R^2 value among all, indicating that the predicted values explain 97.3% of the variance in the actual values. The slope of 0.9675, which is very close to 1, implies that the prediction scales well with the actual values. Meanwhile, the small positive interception of 0.03781 may have introduced a very slight upward bias. Overall, the bagged tree model achieves high accuracy and low bias, making it the most reliable model in this comparison.

The second-best performer is the medium neural network (NN), with R^2 values of 0.961 (training) and 0.98 (test), and low MSE values of 0.001 for both datasets. In Figure 7, a few outliers appear around value 6, which are much larger than the true response, though most predictions cluster near the diagonal line. This suggests that the medium NN model demonstrates strong predictive capability but may handle more outliers than the bagged tree model. The regression line equation of this model is $y = 0.9874x + 0.01478$, the lower variance of 96.1% compared to the bagged tree shows it is slightly less reliable. Its slope is closest to 1, and the interception is minimal. However, a drop in R^2 suggests slightly more errors in generalization and prediction variability.

The third performer is the medium Gaussian SVR, with R^2 values of 0.97 (training) and 0.938 (test), and MSE values of 0.001 (training) and 0.002 (test). In Figure 8, predictions generally scatter near the diagonal line but not as closely, indicating that the medium Gaussian SVR model has good prediction accuracy but may be more prone to error. The regression line equation of this model is $y = 0.9637x + 0.0418$, with a strong R^2 value is nearly on par with the bagged tree. The slope is a little lower with a value of 0.9637, meaning predictions increase slightly less rapidly compared to actual values. The slightly higher interception suggests a small, consistent overestimation. While it is a strong model, its slightly lower slope and higher intercept are showing an inferior performance as compared to the bagged tree model.

The final performer is the medium tree, with R^2 values of 0.958 for both datasets, and MSE values of 0.001 for both datasets. This performance comparison suggests that these

models effectively reduce computational complexity while maintaining strong predictive accuracy and efficiency. In Figure 9, more outliers appear compared to other models, suggesting that the fine tree model struggles with outliers more than the others, leading to increased errors. The regression line equation of this model is $y = 0.9667x + 0.03852$, with the lowest R^2 value among the four models, suggesting it is less effective at capturing the variability of the target variable. The slope and intercept are closely similar to the bagged tree, but the reduced R^2 value implies more prediction error and weaker model generalization. This model performs decently well but is the least preferable among the four.

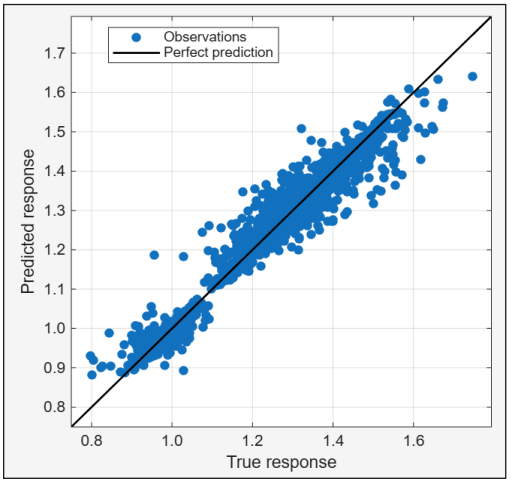

Figure 6. Response of bagged trees with 10 selected features

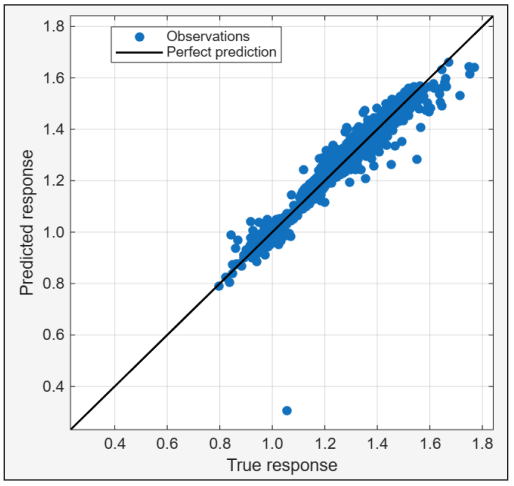

Figure 7. Response of the medium artificial neural network with 10 selected features

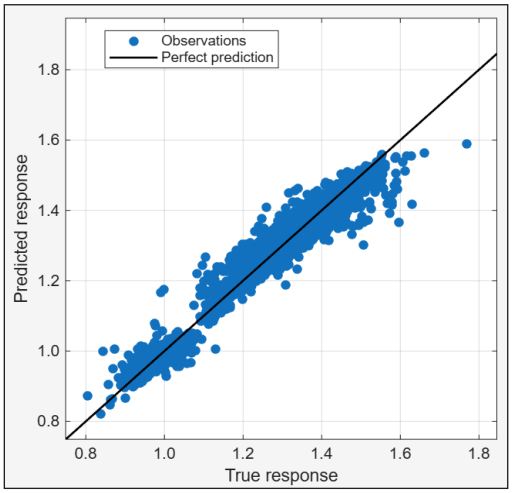

Figure 8. Response of the medium Gaussian support vector regression with 10 selected features

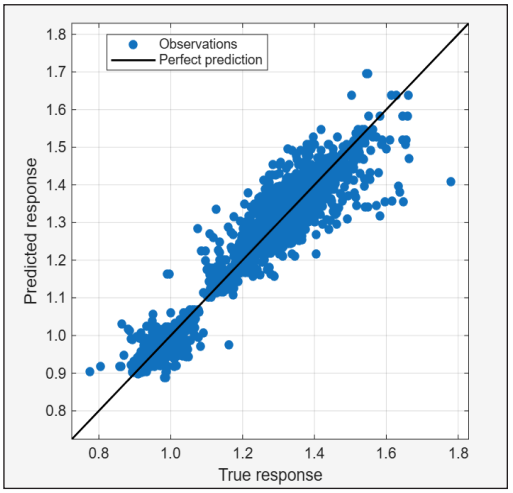

Figure 9. Response of the medium tree with 10 selected features

Overall, the bagged tree model is designated as the best model due to its balance of high predictive accuracy, efficient use of a smaller, well-selected feature set, and strong capability to minimize deviations caused by outliers. Moreover, the bagged tree model consistently demonstrates excellent performance across both datasets, even with smaller feature subsets.

CONCLUSION

This research effectively analyzed nonlinear regression models to predict the CF within the convective sections of the boiler, highlighting the models' practicality and performance in these crucial areas. The study identified bagged trees, a medium neural network, a medium Gaussian SVR, and a medium tree, all with 10 features selected to be the top-performing machine learning models that required fewer feature subsets, which is essential for optimizing boiler performance. For example, the bagged trees achieve an R^2 of 0.99 without feature selection and 0.976 with only 10 selected features, resulting in a 93% reduction from the original input. This reduction not only enhances computational speed and reduces memory usage but also improves model interpretability and operational feasibility.

Comparatively, while NNs and SVR provided slightly higher accuracy, they demanded more computational resources, especially in the training and tuning phases. On the other hand, tree-based models such as bagged trees and medium trees offered a balanced trade-off between performance and processing efficiency, allowing them to be more suitable for real-time deployments. Moreover, implementing these models in operational environments exposes them to challenges such as real-time sensor data, managing model complexity during control system integration, and maintaining operational parameter settings. Therefore, careful consideration must be given to sensor calibration and data preprocessing pipelines to ensure robust integration.

Future research should focus on integrating reinforcement learning and online monitoring machine learning techniques to adjust predictions based on live operational data dynamically. By pinpointing these features more accurately, the study can support the development of a more targeted and efficient soot-blowing mechanism. This would enhance operational precision, leading to improved boiler efficiency, cost savings, and extended equipment lifespan.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2024/TK07/UPM/02/3). Special thanks are extended to Sultan Azlan Shah Power Station, Tenaga Nasional Berhad, for providing data and consultation for this project.

REFERENCES

Bezerra, F. E., de Oliveira Neto, G. C., Cervi, G. M., Mazetto, R. F., de Faria, A. M., Vido, M., Lima, G. A., de Araújo, S. A., Sampaio, M., & Amorim, M. (2024). Impacts of feature selection on predicting machine failures by machine learning algorithms. *Applied Sciences*, *14*(8), 3337. https://doi.org/10.3390/app14083337

Breeding, C., Tandra, D., & Shah, S. (2010). Boiler cleaning using ISB (Intelligent Soot Blowing) system integration: Recent developments and case study. In *ASME 2010 Power Conference* (pp. 89–96). American Society of Mechanical Engineering. https://doi.org/10.1115/power2010-27322

Cai, H., Qin, W., Wang, L., Hu, B., & Zhang, M. (2021). Hourly clear-sky solar irradiance estimation in China: Model review and validations. *Solar Energy*, *226*, 468–482. https://doi.org/10.1016/j.solener.2021.08.066

Comito, C., & Pizzuti, C. (2022). Artificial intelligence for forecasting and diagnosing COVID-19 pandemic: A focused review. *Artificial Intelligence in Medicine*, *128*, 102286. https://doi.org/10.1016/j.artmed.2022.102286

Ding, X., Yang, F., & Ma, F. (2022). An efficient model selection for linear discriminant function-based recursive feature elimination. *Journal of Biomedical Informatics*, *129*, 104070. https://doi.org/10.1016/j.jbi.2022.104070

Ghiasi, M. M., & Zendehboudi, S. (2021). Application of decision tree-based ensemble learning in the classification of breast cancer. *Computers in Biology and Medicine*, *128*, 104089. https://doi.org/10.1016/j.compbiomed.2020.104089

Jemai, J., & Zarrad, A. (2023). Feature selection engineering for credit risk assessment in retail banking. *Information*, *14*(3), 200. https://doi.org/10.3390/info14030200

Kumari, S. A., & Srinivasan, S. (2019). Ash fouling monitoring and soot-blow optimization for reheater in thermal power plant. *Applied Thermal Engineering*, *149*, 62–72. https://doi.org/10.1016/j.applthermaleng.2018.12.031

Li, P., Li, K., Zhou, Y., Li, Q., Shi, Z., & Zhong, W. (2023). An effective strategy for monitoring slagging location and severity on the waterwall surface in operation coal-fired boilers. *Energies*, *16*(24), 7925. https://doi.org/10.3390/en16247925

Li, Q., Yan, P., Liu, J., Shi, Y., & Yu, D. (2020). Prediction of pollution state of heating surface in coal-fired utility boilers. *IEEE Access*, *8*, 206132–206145. https://doi.org/10.1109/access.2020.3036840

Liang, D., Frederick, D. A., Lledo, E. E., Rosenfield, N., Berardi, V., Linstead, E., & Maoz, U. (2022). Examining the utility of nonlinear machine learning approaches versus linear regression for predicting body image outcomes: The U.S. Body Project I. *Body Image*, *41*, 32–45. https://doi.org/10.1016/j.bodyim.2022.01.013

Loh, W.-Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews Data Mining and Knowledge Discovery*, *1*(1), 14–23. https://doi.org/10.1002/widm.8

Madejski, P., Janda, T., Taler, J., Nabagło, D., Węzik, R., & Mazur, M. (2018). Analysis of fouling degree of individual heating surfaces in a pulverized coal fired boiler. *Journal of Energy Resources Technology*, *140*(3), 032003. https://doi.org/10.1115/1.4037936

Marafino, B. J., Davies, J. M., Bardach, N. S., Dean, M. L., & Dudley, R. A. (2014). N-gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit. *Journal of the American Medical Informatics Association*, *21*(5), 871–875. https://doi.org/10.1136/amiajnl-2014-002694

Menn, N., & Chudnovsky, B. (2021). FTR-based expert system for power generation units. In A. J. Tallón-Ballesteros (Ed.), *Modern management based on Big Data II and Machine Learning and Intelligent Systems III* (Vol. 341, pp. 548-556). IOP Press. https://doi.org/10.3233/faia210287

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(8), 1226–1238. https://doi.org/10.1109/tpami.2005.159

Shi, Y., Zhang, Z., Li, M., Wen, J., & Cui, F. (2021). Soot blowing optimization for coal-fired boilers with ash accelerated deposition model based on gamma process. In *40th Chinese Control Conference* (pp. 6617-6622). IEEE. https://doi.org/10.23919/ccc52363.2021.9549924

Shi, Y., Li, M., Wang, J., Wen, J., Cui, F., & Qiao, G. (2022). Prediction of ash deposition on heating surfaces of coal-fired power plant boiler based on dynamic neural network. In *34th Chinese Control and Decision Conference* (pp. 779–783). IEEE. https://doi.org/10.1109/ccdc55256.2022.10034385

Tenaga Nasional Berhad. (2024). *Ultra-supercritical boiler power plant*. TNB.

Thota, S., & Syed, M. B. (2024). Analysis of feature selection techniques for prediction of boiler efficiency in case of coal based power plant using real time data. *International Journal of Systems Assurance Engineering and Management*, *15*, 300–313. https://doi.org/10.1007/s13198-022-01725-y

Wei, W., Cheng, S., & Sun, F. (2020). Research on formation mechanism of typical low-temperature fouling layers in coal-fired boilers. *Fuel*, *261*, 116215. https://doi.org/10.1016/j.fuel.2019.116215

Wen, J., Shi, Y., Pang, X., & Jia, J. (2022). Optimal soot blowing and repair plan for boiler based on HJB equation. *Optimization*, *71*(16), 4603–4622. https://doi.org/10.1080/02331934.2021.1954922