Energy Proceedings
Vol 48, 2024
ISSN 2004-2965

Evaluation of Hyperparameter Optimization in Machine Learning Models for CH_4 and H_2 Production Prediction based on Supercritical Water Gasification#

Chiagoziem C. Ukwuoma[1,2], Dongsheng Cai[1,2*], Mmesoma P. Chukwuemeka[3], Blessing O. Ayeni[4], Ibrahim Kunle Adefarati[5], Huan Yang[1,2], Olusola Bamisile[1,2], Qi Huang[1,2]

1 College of Nuclear Technology and Automation Engineering, Chengdu University of Technology, Sichuan P.R., 610059, China

2 Sichuan Engineering Technology Research Center for Industrial Internet Intelligent Monitoring and Application, Chengdu University of Technology, Sichuan P.R., 610059, China

3 Department of Applied Biochemistry, Chukwuemeka Odumegwu Ojukwu University, Anambra State, Nigeria

4 Department of Medical Laboratory Science, University of Benin, Edo State, Nigeria

5 Geological Resources and Geological Engineering, College of Environment and Civil Engineering Chengdu University of Technology, Sichuan P.R., 610059, China

(Corresponding Author: caidongsheng@cdut.edu.cn)

ABSTRACT

Renewable energy continues to rise, with supercritical water gasification (SCWG) emerging as a potential biofuel production technique. Conventional approaches for estimating CH_4 and H_2 generation in complex systems frequently fail due to the complicated and dynamic nature of these processes. Machine learning (ML) has emerged as a disruptive technology in various industries, including energy, where it is used to optimize operations and improve prediction accuracy. Conventional techniques lack the versatility and scalability of ML models, resulting in less accurate and efficient prediction capabilities. This gap emphasizes the importance of incorporating machine learning into the energy domain, notably for optimization and prediction in SCWG processes. Furthermore, for any machine learning model, determining the appropriate hyperparameter setting has a direct and significant influence on its performance. In this study, we investigate the influence of three distinct types of hyperparameter optimization techniques on CH_4 and H_2 production prediction based on supercritical water gasification. Grid Search Optimization, Random Search Optimization, and Bayesian Optimization were analyzed utilizing six machine learning models: Ridge Lasso, Elastic Net, Decision Tree, Random Forest, and XGBoost, as well as two ensemble models: linear-based and tree-based models. The dataset from supercritical water gasification of Yimin lignite was used based on three evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Coefficient of Determination (R^2). The results were evaluated based on 5-fold cross-validation and results show that the Random Forest stood out with an R^2 of 0.995, RMSE of 0.109, and MAE of 0.085 for the CH_4 production prediction while recording an R^2 of 1.00, RMSE of 0.112 and MAE of 0.088 for H_2 production prediction. The analysis carried out in this study shows that the choice of optimization technique does not significantly impact the performance of the deployed models, which indicates that the hyperparameter space is relatively well-behaved for this CH_4 and H_2 Production Prediction based on Supercritical Water Gasification, and thus even simpler optimization strategies like Random Search can perform nearly as well as more sophisticated ones like Bayesian Optimization. The result implies that if computation time or resources are a factor, Random Search would be a more efficient approach without a significant trade-off in model performance.

Keywords: Hyperparameter Optimization, Supercritical Water Gasification, Hydrogen Production, Methane Production, Machine Learning

NOMENCLATURE

Abbreviations	
SCWG	Supercritical Water Gasification
CH_4	Methane
H_2	Hydrogen
ML	Machine Learning

This is a paper for the 10th Applied Energy Symposium: Low Carbon Cities & Urban Energy Systems (CUE2024), May. 11-13, 2024, Shenzhen, China.

MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
R^2	Coefficient Of Determination
SCWFB	Supercritical Water Fluidized Bed
CO	Carbon Monoxide
CO_2	Carbon Dioxide
SCW	Supercritical Water
GS	Grid Search
RS	Random Search
BO	Bayesian Optimization
MM	Millimeter
MPa	Megapascal
RPM	Revolutions Per Minute
CE%	Conversion Efficiency Percentage
WT%	Weight Percentage
^OC	Degrees Celsius

1. INTRODUCTION

Gasification is a thermodynamic method that turns organic wastes such as coal and biomass into gaseous products such as CO, CO_2, CH_4, and H_2 by reacting them with gasifying agents such as air, oxygen, steam, or supercritical water (SCW) at temperatures ranging from 500 to 1400 degrees Celsius[1]. Supercritical water gasification (SCWG) is a new method for effectively converting wet biomass or organic wastes into methane or hydrogen-rich gaseous products[2][3]. However, SCWG boiler metals may be subjected to highly destructive attacks during the conversion of diverse sources of biomass. Coal gasification is a green and economical technique for producing energy, electricity, and chemicals with little greenhouse gas emissions[4]. It turns coal into natural gas by employing water, air, or oxygen as combustion agents. Synthetic gas is mostly made up of CO, CH_4, H_2, and other gases, with hydrogen being a major element[5][6]. Coal gasification is a key method for environmentally friendly coal consumption, providing benefits in electric power generation, hydrogen-rich syngas synthesis, and CO_2 emission mitigation.

Machine Learning (ML) techniques have transformed computerized models by incorporating them into various fields, ranging from meteorological assessment to medical diagnostics. The effectiveness of these models is determined by the ML technique, training techniques, regularization methods, and optimum hyperparameters. ML models include two types of parameters: model parameters, which are defined by the training dataset, and hyperparameters, which may be changed by the user before running the model. Hyperparameters include neural network weights, learning rates, and regularization and kernel parameters. Different ML techniques need distinct sets of hyperparameters, and the majority of models include hyperparameters. The approach for determining the ideal hyperparameter configuration for a given algorithm trained on a certain dataset is known as Hyperparameter Optimization. Once the hyperparameters have been established, the algorithm uses the data to learn the model's parameters[7].

In this paper, the role of hyperparameter optimizations in ML models for CH4 and H2 production prediction from Supercritical Water Gasification was investigated specifically Grid search (GS) optimization, Random search (RS) Optimization, and Bayesian optimization (BO). The results were validated using a 5-fold cross-validation.

2. RELATED WORKS

Coal remains a crucial source of energy in contemporary society; however, traditional coal combustion methods have significantly degraded energy utilization quality and caused serious environmental pollution. Supercritical water gasification (SCWG) emerges as an effective technology for the clean and efficient transformation of coal, utilizing supercritical water to convert coal's hydrogen and carbon into H_2 and CO_2, thereby aligning with pollution control and CO_2 reduction goals[4]–[6][8]. Lu et al. successfully demonstrated a supercritical water fluidized bed (SCWFB) system for hydrogen production from biomass gasification using glucose and actual biomass under continuous operation at 873 K, showcasing its substantial advantages and promising potential[9]. Additionally, Li et al.[10] investigated the influence of key operational parameters like temperature and pressure on gasification within an SCWFB reactor, finding that higher temperatures favor hydrogen production, though pressure showed minimal effect. Further, Bei et al.[11] performed a numerical analysis on ethanol gasification, indicating significant impacts of wall temperature on the carbon gasification rate and yield.

Recent advancements include the application of ML techniques to predict CH_4 and H_2 production via SCWG. Li et al.[12] developed an ML framework to predict syngas composition in SCWG systems, achieving high accuracy which aided in process optimization and catalyst selection to maximize H_2 and minimize CO_2 production. Zhao et al.[13] utilized various ML models to enhance H_2 yield from SCWG, finding that oxygen-rich feedstocks enhance H_2 recovery. Furthermore, Sharma et al.[14] developed a model to identify optimal

gasification pathways for biomass energy, emphasizing ML's role in optimizing output while considering environmental and operational constraints. Dang et al.[15] focused on optimizing syngas production from steam gasification by analyzing various operating conditions and biomass compositions, highlighting the role of ML in enhancing process efficiency. Marcantonio et al.[16] integration of response surface methodology with kinetic modeling further illustrates how ML can enhance predictions and optimize processes in hydrogen-rich syngas production.

Recent literature has explored hyperparameter optimization in ML, discussing its theoretical and practical aspects. for instance, Liu et al.[17] highlighted the importance of learning rate schedules, Yu et al.[18] provided an exhaustive guide on hyperparameter optimization techniques, and Kelterborn et al.[19] discussed optimization strategies including gradient descent and Newton's method. Turner et al.'s[20] study from the Black-Box Optimization emphasized the efficacy of BO over RS, while Ali et al.[7] investigated algorithms for reducing computational complexity in ML, demonstrating the impact of different strategies on training efficiency. These studies collectively advance our understanding of both hyperparameter optimization and the application of ML in energy and environmental sciences, offering significant insights and tools applicable across various domains.

3. MATERIAL AND METHODS

This section presents the Analyzed models, deployed hyperparameter optimization techniques, Dataset and Data Augmentation, and evaluation metrics.

3.1 Analyzed Models

This paper investigated the effect of hyperparameter optimization in several ML models including three linear-based models, three Tree-based models, and two ensemble models. The linear-based models include Ridge regression, Lasso Regression, and Elastic Net. Ridge regression is a model that favors mostly data that suffer from multicollinearity. Multicollinearity occurs when least squares estimates are unbiased, but their variances are large so they may be far from the actual value. Mathematically, ridge regression is illustrated thus;

$$\mathrm{RSS} = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \hat{\mathrm{y}}_\mathrm{i})^2 = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \mathrm{X_i^T}\beta)^2 \qquad (1)$$

Where, y_i is the observed response, $\hat{y}_i$ is the predicted response, X_i is the feature vector for the ith observation, β is the coefficient vector. Lasso regression is a type of linear regression that uses shrinkage (data values are shrunk towards a central point, like the mean). It is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination. Mathematically, the lasso is represented as thus;

$$\mathrm{Lasso}(\beta) = \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \sum_{\mathrm{j}=1}^{\mathrm{p}} \mathrm{x_{ij}}\,\beta_\mathrm{j})^2 + \lambda \sum_{\mathrm{j}=1}^{\mathrm{p}} |\beta_\mathrm{j}| \qquad (2)$$

Where, y_i is the observed output, x_{ij} is the jth predictor for the ith observation, β_j is the coefficient for the jth predictor, λ is the regularization parameter controlling the amount of shrinkage applied to the coefficients. The model integration of the Lasso and Ridge principle is referred to as Elastic Net. It's particularly useful when there are correlations between parameters and is mathematically represented thus;

$$\begin{aligned}\text{Elastic Net} = {} & \sum_{\mathrm{i}=1}^{\mathrm{n}} (\mathrm{y_i} - \sum_{\mathrm{j}=1}^{\mathrm{p}} \mathrm{x_{ij}}\,\beta_\mathrm{j})^2 \\ & + \lambda_1 \sum_{\mathrm{j}=1}^{\mathrm{p}} |\beta_\mathrm{j}| + \lambda_2 \sum_{\mathrm{j}=1}^{\mathrm{p}} \beta_\mathrm{j}^2\end{aligned} \qquad (3)$$

Where, y_i is the observed output, x_{ij} is the jth predictor for the ith observation, β_j is the coefficient for the jth predictor. λ_1 is the regularization parameter for the L1 penalty (similar to Lasso), λ_2 is the regularization parameter for the L2 penalty (similar to Ridge).

For the tree models, the Decision tree, Random Forest, and XGBoost were considered. Decision trees are a supervised ML technique used for both classification and regression tasks. They work by recursively partitioning the feature space into regions, making decisions at each node based on the features of the data.

$$\mathrm{Infor.\,Gain} = \mathrm{Imp(Par)} - \sum_{\mathrm{i}=1}^{\mathrm{nn}} \frac{\mathrm{N_i}}{\mathrm{N}} \times \mathrm{Imp(Chi_i)} \qquad (4)$$

Where infor. Means information, Imp(Par) means the Impurity of the parent node before the split, N means the parent node's total number of samples, $\mathrm{N_i}$ means ith child node number of samples, n denotes the number of child nodes, and $\mathrm{Imp(Chi_i)}$ denotes the impurity of the ith child node after the split. Random Forest is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mean/median (for regression) prediction of the individual trees. It's designed to address overfitting and improve accuracy compared to individual decision trees. The random forest prediction $\hat{Y}_{RF}(x)$ for a new sample, x can be represented as:

$$\hat{\mathrm{Y}}_{\mathrm{RF}}(\mathrm{x}) = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{n}} \hat{\mathrm{Y}}_\mathrm{i}(\mathrm{x}) \qquad (5)$$

Where, $Y_i(x)$ is the prediction of the ith decision tree, n is the total number of decision trees in the Random Forest ensemble. XGBoost is an ensemble

learning technique that utilizes the gradient boosting framework to produce highly accurate models. It constructs an ensemble of trees by iteratively adding new trees to minimize the objective function. At each iteration, it computes the gradient and the Hessian of the loss function concerning the current ensemble of trees, then fits a new tree to the negative gradient of the loss function. The mathematical representation of the prediction of the t^{th} tree in the ensemble can be represented as:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta . h_t(x_i) \quad (6)$$

Where, $\hat{y}_i^{(t)}$ is the predicted value of the i^{th} instance by the t^{th} tree, $\hat{y}_i^{(t-1)}$ is the predicted value of the i^{th} instance by the ensemble of $t-1$ trees, $h_t(x_i)$ is the prediction of the t^{th} tree for the t^{th} instance, η is the learning rate, controlling the step size during the gradient descent optimization. The ensemble techniques involved the integration of linear-based models and tree-based models which are mathematically represented thus; Let M_1, M_2, and M_3 represent three individual linear or tress-based regression models, The final prediction of the ensemble is the weighted average of prediction from all the three models shown in Equation 7 where x denotes the predictions of the models.

$$\text{Ensemble} = \frac{1}{3}\left(M_1(x), M_2(x), M_3(x)\right) \quad (7)$$

3.2 Deployed Hyperparameter Optimization Techniques

In the field of ML, the challenge of hyperparameter optimization or tuning involves identifying the best hyperparameters for a learning algorithm[21]. A hyperparameter, distinct from model parameters, is utilized to oversee the learning process. The objective of hyperparameter optimization is to determine a set of hyperparameters that produces an optimal model, minimizing a pre-specified loss function on a given set of independent data[22]. The function evaluates the loss associated with a set of hyperparameters[22]. To estimate the generalization performance of the model, cross-validation is commonly employed, enabling the selection of hyperparameter values that enhance performance[23]. This paper focuses on three primary methods of hyperparameter optimization: GS, RS, and BO.

3.2.1 Grid Search (GS) Optimization

GS involves a comprehensive search through a specified subset of the hyperparameter space, guided by a performance metric, usually determined by cross-validation on the training data or through the assessment of a validation set[10][11]. Due to the potentially vast and complex space of hyperparameters, which may include continuous or unbounded parameters, setting limits and discretizing these values are necessary steps before applying GS. While GS is affected by the curse of dimensionality, it benefits from parallel execution since the evaluations of hyperparameter settings are independent[23].

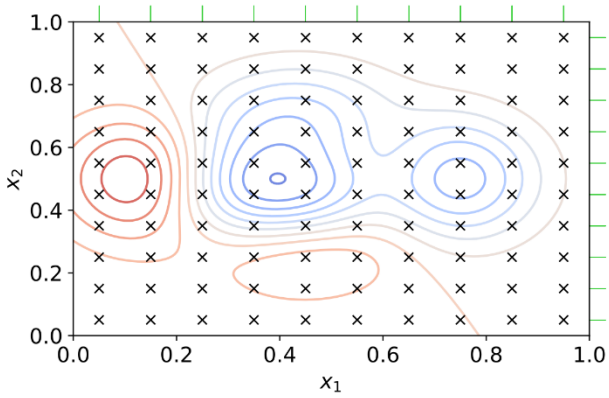

Fig. 1 Grid Search (GS) Sample

Fig. 1 shows a GS methodology evaluating a combination of two distinct hyperparameters, examining 10 distinct values for each. Consequently, this process tests a total of 100 combinations to ascertain the most effective one. The results are visually represented through contour lines, where blue contours denote areas of strong performance and red contours indicate areas with suboptimal outcomes.

3.2.2 Random Search (RS) Optimization

RS diverges from grid search by randomly selecting combinations of hyperparameters instead of exhaustively searching them. This approach applies to both discrete and continuous hyperparameter spaces and offers advantages in terms of exploration, particularly for hyperparameters that significantly impact model performance. RS can efficiently investigate a broader range of values and is noted for its simplicity and the ability to incorporate prior knowledge through the specified sampling distributions. It is also inherently parallel and remains a critical baseline for evaluating newer methods in hyperparameter optimization[9][12]. Fig. 2 shows a sample of RS across different combinations of values for two hyperparameters. 100 different random choices are evaluated. The green bars show that more individual values for each hyperparameter are considered compared to a GS.

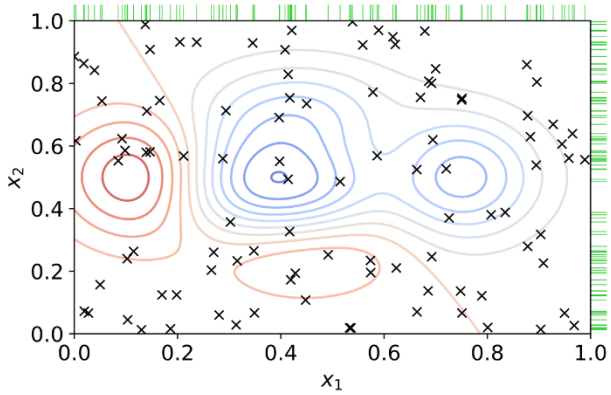

Fig. 2 Random Search (RS) Sample

3.2.3 Bayesian Optimization (BO)

BO represents a sophisticated global optimization technique for optimizing noisy black-box functions. In hyperparameter optimization, BO constructs a probabilistic model that maps hyperparameter values to the objective measured on a validation set. By iteratively selecting promising hyperparameter configurations based on the current model and updating the model with new data, this method seeks to maximize the information about the function's behavior, especially the location of the optimum. It balances the exploration of uncertain outcomes with the exploitation of hyperparameters likely to be near the optimum. Empirically, BO has demonstrated superior performance in achieving more effective results with fewer evaluations compared to GS and RS [13][14]. designations.

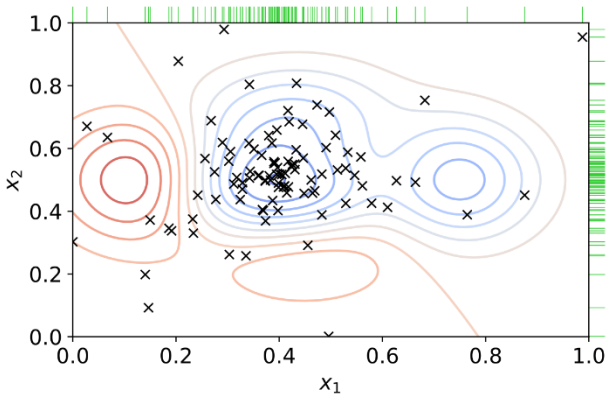

Fig. 3 Bayesian optimization (BO)

As shown in Fig. 3, techniques like BO intelligently navigate the potential space of hyperparameter choices by determining the next combination to investigate, guided by insights gained from previous observations described.

3.3 Dataset and Data Augmentation

This study utilized the dataset provided by Hui et al.[29] for its experimental analysis. The experiment was conducted using a cylindrical micro quartz tube, measuring 200mm in length and 1.5mm in diameter, serving as the reactor. This reactor was capable of withstanding temperatures up to 1000°C and pressures up to 45MPa. A mixture of Yimin coal and deionized water was prepared and loaded into the reactor. The tube was subsequently melted and sealed using a hydrogen flame. Yimin lignite was pulverized into a fine powder, and the resultant gaseous by-products were subjected to analysis through gas chromatography and thermal conductivity detectors. High-purity argon served as the carrier gas. Additionally, a thermogravimetric analyzer was employed to measure the RPM parameter j. The efficiency of carbon gasification (CE) in supercritical water was assessed to monitor the gasification process. The experimental results, varying by temperature, concentration, and residence time, are presented in Table 1.

3.4 Evaluation Metrics

This paper uses three commonly used evaluation metrics for regression tasks: RMSE, R^2, and MAE. RMSE and MAE measure prediction accuracy, while R^2 indicates the model's ability to explain the dependent variable's variance. Mathematically, RMSE is a measure of the average deviation of predictions from actual values, calculated by dividing the square root of the average of the squared differences between predicted and actual values.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

R^2 measures the proportion of variance in the dependent variable predicted from the independent variables.

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

MAE measures the average absolute difference between predicted and actual values.

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{10}$$

4. RESULTS

The correlation heatmap which shows the correlation coefficients between variables (variables related to the production of H_2 and CH_4) is presented in Fig. 4. It is observed that Temperature is highly positively correlated with CE% (Conversion Efficiency), H_2 (the production of hydrogen), and CH_4 (the production of

methane), with values of 0.87, 0.81, and 0.88, respectively indicating that as the temperature increases, these variables tend to increase as well. CE% has a very strong positive correlation with H_2 and CH_4, with correlation coefficients of 0.98 and 0.95, respectively indicating that better conversion efficiency is closely associated with higher production of both hydrogen and methane. H_2 and CH_4 are also strongly positively correlated with each other (0.88). This indicates a relationship where an increase in the production of one is associated with an increase in the production of the other. The Concentration(wt%) variable shows very little to no correlation with any other variables, as seen by the coefficients close to zero.

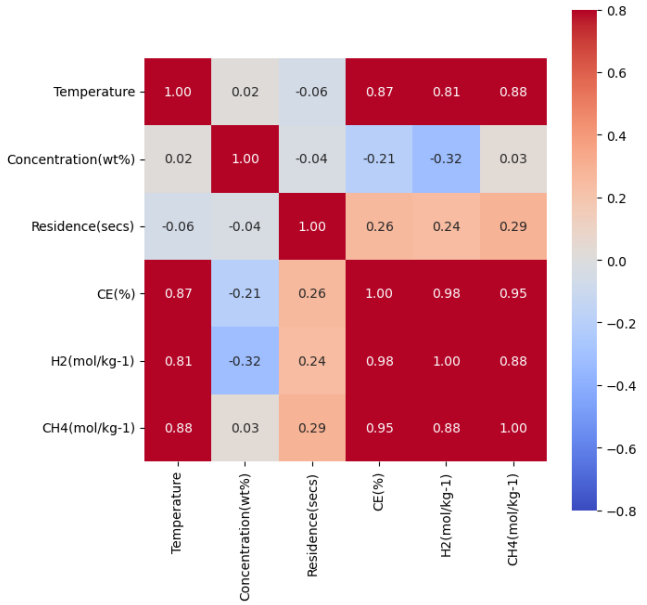

Fig. 4 Correlation heatmap showing the relationship between variables in the dataset

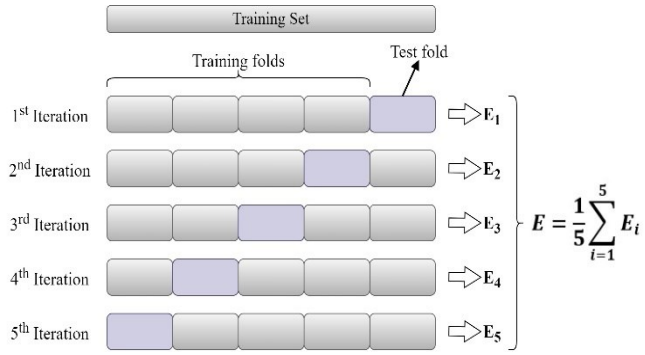

Fig. 5 Five-fold cross-validation techniques

The five-fold cross-validation approach is used in this work to assess the performance of the various models, as shown in Fig. 5. It entails dividing the dataset into five equal parts, iterating over each fold, training the model on the training set, and assessing it on the test set. This approach lowers bias, gives variance insights, assures effective data utilization, and assesses model generalization.

4.1 CH_4 Production Prediction

The GS Optimization Techniques results for CH_4 production prediction are shown in Table 2. The results show that Ridge is the best linear-based model with consistent performance across all metrics. Lasso has higher errors and lower R^2 scores indicating its biases in capturing the complexity of the dataset as well as Ridge. Tree-based models have low errors and high R^2 scores, but there may be concerns about overfitting due to their nature. The ensemble of linear-based models has better performance than Lasso and Elastic Net. However, the ensemble of tree-based models has higher errors and lower R^2 scores. The best-performing models are Random Forest and XGBoost, which benefit from ensemble learning that combines multiple decision trees to improve performance and robustness. Ridge regression is the best linear-based model indicating its ability in dealing with multicollinearity by shrinking coefficients. For model selection, Random Forest is the best model due to its strong performance across all metrics and robustness against overfitting.

RS optimization techniques for predicting CH_4 production using 5 K-folds are shown in Table 3. Ridge has moderate errors in prediction and a high R^2 score, explaining a significant proportion of the data's variance. Lasso has higher errors and a lower R^2 score, indicating that it may not capture the dataset's complexity effectively. Elastic Net shows a performance between Ridge and Lasso, combining features of both Ridge and Lasso regularization. Decision Tree exhibits lower errors and a high R^2 score but is sensitive to the specific data it's trained on, leading to variance across different training sets. Random Forest has the lowest average RMSE and highest R^2 scores, proving its robustness and performance in predicting CH_4 production. XGBoost has a slightly higher RMSE and lower R^2 but still performs well. In this paper, Random Forest remains the strongest model, indicating ensemble methods effectively reduce variance and bias in the dataset while Ridge Regression is the best among linear models. The BO techniques are used to evaluate the performance of various ML models across 5 K-folds as shown in Table 4. Linear-based models, such as Ridge and Decision Tree, show relatively low RMSE and MAE, with high R^2 scores, indicating stable performance. Lasso and Elastic Net show higher RMSE and MAE, indicating they may not capture the dataset's complexity effectively.

Table 1. Statistical analysis of the deployed dataset

	Temp	Conc(wt%)	Reside(secs)	CE(%)	CH_4(mol/kg-1)	H_2(mol/kg-1)
count	1000.000	1000.000	1000.000	1000.0000	1000.0000	1000.000
mean	0.880353	0.589600	0.472230	0.503629	3.773400	6.95337
Std	0.095676	0.330335	0.269607	0.197545	1.497359	5.17247
Min	0.764706	0.200000	0.090682	0.231032	1.180000	0.75000
25%	0.764706	0.200000	0.175541	0.323956	2.460000	2.73000
50%	0.882353	0.600000	0.425957	0.508312	3.780000	5.67500
75%	1.000000	1.000000	0.674709	0.614628	4.885000	9.59250
Max	1.000000	1.000000	1.000000	1.000000	6.640000	22.50000

Table 2. GS optimization techniques prediction for CH_4 production

Metrics	K-Folds	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.261	0.435	0.346	0.177	0.120	0.117	0.150	0.325
	2	0.268	0.428	0.346	0.125	0.109	0.121	0.137	0.283
	3	0.295	0.449	0.370	0.120	0.113	0.171	0.156	0.335
	4	0.310	0.392	0.301	0.111	0.105	0.122	0.128	0.269
	5	0.291	0.423	0.348	0.108	0.106	0.119	0.132	0.286
	Average	**0.285**	**0.425**	**0.342**	**0.128**	**0.111**	**0.130**	**0.141**	**0.300**
R^2	1	0.968	0.908	0.942	0.985	0.993	0.993	0.989	0.949
	2	0.969	0.922	0.949	0.993	0.995	0.994	0.992	0.966
	3	0.960	0.918	0.944	0.994	0.995	0.988	0.990	0.954
	4	0.960	0.926	0.956	0.994	0.995	0.993	0.992	0.965
	5	0.964	0.929	0.952	0.995	0.996	0.994	0.993	0.967
	Average	**0.964**	**0.921**	**0.949**	**0.992**	**0.995**	**0.992**	**0.991**	**0.960**
MAE	1	0.205	0.347	0.284	0.112	0.092	0.092	0.116	0.248
	2	0.205	0.347	0.290	0.098	0.085	0.094	0.111	0.223
	3	0.212	0.351	0.309	0.093	0.087	0.109	0.124	0.262
	4	0.241	0.308	0.249	0.087	0.085	0.094	0.107	0.204
	5	0.225	0.348	0.284	0.085	0.084	0.089	0.101	0.227
	Average	**0.217**	**0.340**	**0.283**	**0.095**	**0.087**	**0.096**	**0.112**	**0.233**

Table 3. RS optimization techniques prediction for CH_4 production

Metrics	K-Folds	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.260	0.435	0.349	0.137	0.118	0.125	0.152	0.326
	2	0.268	0.428	0.350	0.123	0.106	0.126	0.136	0.284
	3	0.295	0.449	0.366	0.125	0.112	0.175	0.159	0.333
	4	0.310	0.392	0.303	0.113	0.104	0.127	0.128	0.270
	5	0.291	0.423	0.351	0.123	0.106	0.118	0.134	0.286
	Average	**0.285**	**0.425**	**0.344**	**0.124**	**0.109**	**0.134**	**0.142**	**0.300**
R^2	1	0.968	0.908	0.941	0.991	0.993	0.992	0.989	0.948
	2	0.969	0.922	0.948	0.994	0.995	0.993	0.992	0.965
	3	0.960	0.918	0.946	0.994	0.995	0.988	0.990	0.955
	4	0.960	0.926	0.956	0.994	0.995	0.992	0.992	0.965
	5	0.964	0.929	0.951	0.994	0.996	0.994	0.993	0.967
	Average	**0.964**	**0.921**	**0.948**	**0.993**	**0.995**	**0.992**	**0.991**	**0.960**
MAE	1	0.204	0.347	0.284	0.106	0.091	0.096	0.118	0.247
	2	0.205	0.347	0.291	0.099	0.083	0.095	0.110	0.224
	3	0.212	0.351	0.305	0.096	0.086	0.110	0.127	0.259
	4	0.241	0.308	0.247	0.090	0.084	0.099	0.105	0.202
	5	0.225	0.348	0.290	0.097	0.082	0.088	0.103	0.226
	Average	**0.217**	**0.340**	**0.283**	**0.098**	**0.085**	**0.098**	**0.113**	**0.232**

Decision Tree has low RMSE and MAE values, with high R^2 scores, but may suffer from overfitting as a single model. Random Forest shows the best performance among all models, with the lowest RMSE and highest R^2 scores, indicating it effectively captures the dataset's patterns and generalizes well.

XGBoost shows slightly higher RMSE and lower R^2 compared to Random Forest, an advanced gradient-boosting algorithm. Key observations show that Random Forest maintains its superiority in performance for predicting CH_4 production, while Decision Tree shows strong performance on individual folds but may be less preferable due to its susceptibility to overfitting. Ridge Regression outperforms other linear models, indicating beneficial regularization for the dataset. Bayesian Optimization has found good hyperparameter settings for these models, leading to strong predictive performances

4.2 H_2 Production Prediction

Table 5 presents the results of a GS optimization for predicting H_2 production using various ML models. Ridge showed moderately high RMSE and MAE values, while Lasso had the highest RMSE and MAE values. Elastic Net performed slightly better but still had considerable errors. Decision Tree had the lowest RMSE and MAE values, indicating high predictive accuracy. Random Forest showed similar performance to Decision Tree but with slightly higher RMSE and MAE. XGBoost had higher RMSE and MAE values but still high R^2 scores, indicating good but not optimal performance. Tree-based models outperformed linear-based models for predicting H_2 production, with Decision Trees and Random Forest showing nearly perfect R^2 scores, indicating an excellent fit or potential overfitting. XGBoost performed well but not as well as Random Forest or Decision Tree.

The analysis of different models based on random search optimization techniques for H_2 production prediction as shown in Table 6 revealed that Ridge regression has a moderately high average RMSE, indicating it might not capture all the complexities in the data. Decision Tree and Elastic Net have the highest average RMSE among linear models, indicating that the combination of L1 and L2 regularization is not optimal for this dataset. Tree-based models, such as Decision Trees and Random Forest, have the lowest average RMSE and perfect R^2 scores, indicating excellent performance and generalization capability. XGBoost shows better performance than individual decision trees but is not as strong as Random Forest. The random search technique has successfully identified robust hyperparameters for the Random Forest model.

BO for H_2 production prediction is shown in Table 7. Linear-based models showed moderately high average RMSE, while tree-based models showed lower RMSE and higher R^2 scores. Decision Tree, Random Forest, and XGBoost were the most effective models, with the ensemble of linear models showing a high RMSE. Random Forest was the best model, showing it can handle the complexity of the dataset effectively. Tree-based models generally outperformed linear models, indicating the non-linear relationship in the data. XGBoost was strong but not as effective as Random Forest, suggesting further tuning. Linear models did not show significant improvement, implying non-linear models are more suitable for this dataset. Bayesian optimization worked well for tuning the Random Forest model.

4.3 Optimization Techniques Comparison

When comparing the optimization techniques (GS, RS, BO) for CH_4 and H_2 production prediction as shown in Table 8 and Table 9, it is noticeable that the differences in performance metrics for each model type are minimal indicating that all three techniques arrive at similar quality hyperparameters for the models, with Random Forest consistently emerging as the top-performing model across all three techniques. Decision Tree and XGBoost also show good performance but are consistently outperformed by Random Forest. Linear models, while they do not perform as well as tree-based models, still show reasonable performance and might be preferred for their simplicity and interpretability in certain scenarios.

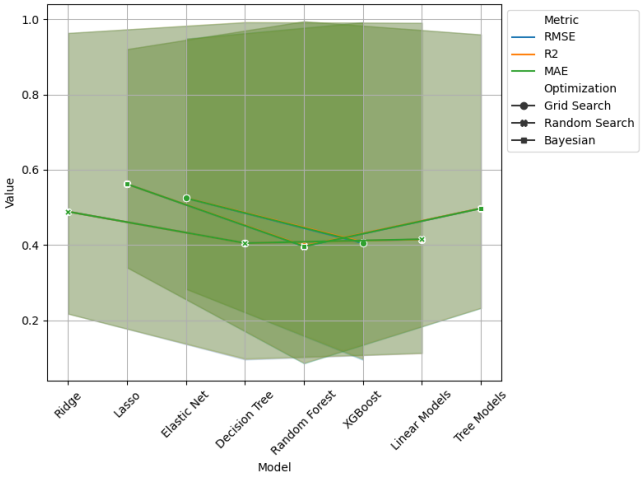

Fig. 6 CH_4 Production Optimization Comparison

The choice of optimization technique does not significantly impact the performance of the models,

Table 4. BO techniques prediction for CH_4 production

Metrics	K-Folds	Tree Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.261	0.435	0.346	0.137	0.117	0.117	0.150	0.326
	2	0.268	0.428	0.290	0.112	0.107	0.122	0.138	0.284
	3	0.295	0.449	0.309	0.125	0.112	0.162	0.156	0.333
	4	0.310	0.392	0.249	0.109	0.104	0.118	0.128	0.270
	5	0.291	0.423	0.348	0.138	0.104	0.114	0.132	0.286
	Average	**0.285**	**0.425**	**0.342**	**0.125**	**0.109**	**0.127**	**0.141**	**0.300**
R^2	1	0.968	0.908	0.942	0.991	0.993	0.993	0.989	0.948
	2	0.969	0.922	0.949	0.995	0.995	0.994	0.992	0.965
	3	0.960	0.918	0.944	0.994	0.995	0.989	0.990	0.955
	4	0.960	0.926	0.956	0.994	0.995	0.993	0.992	0.965
	5	0.964	0.929	0.952	0.992	0.996	0.995	0.993	0.967
	Average	**0.964**	**0.921**	**0.949**	**0.993**	**0.995**	**0.993**	**0.991**	**0.960**
MAE	1	0.205	0.347	0.284	0.105	0.090	0.091	0.117	0.247
	2	0.205	0.347	0.290	0.088	0.083	0.094	0.111	0.224
	3	0.212	0.351	0.309	0.097	0.087	0.108	0.124	0.259
	4	0.241	0.308	0.249	0.089	0.085	0.091	0.107	0.202
	5	0.225	0.348	0.284	0.105	0.082	0.085	0.101	0.226
	Average	**0.217**	**0.340**	**0.283**	**0.097**	**0.085**	**0.094**	**0.112**	**0.232**

Table 5. GS optimization techniques prediction for H_2 production

Metrics	K-Folds	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.763	0.989	1.048	0.124	0.118	0.286	0.333	1.007
	2	0.781	0.912	0.949	0.160	0.105	0.136	0.317	0.888
	3	0.822	0.943	1.005	0.113	0.110	0.322	0.324	0.937
	4	0.891	0.750	0.808	0.114	0.109	0.196	0.249	0.762
	5	0.856	0.878	0.892	0.119	0.117	0.122	0.288	0.876
	Average	**0.823**	**0.894**	**0.941**	**0.126**	**0.112**	**0.212**	**0.302**	**0.894**
R^2	1	0.970	0.967	0.963	0.999	1.000	0.997	0.996	0.966
	2	0.974	0.966	0.963	0.999	1.000	0.999	0.996	0.967
	3	0.977	0.972	0.968	1.000	1.000	0.997	0.997	0.972
	4	0.977	0.979	0.975	1.000	1.000	0.999	0.998	0.978
	5	0.978	0.972	0.971	0.999	0.999	0.999	0.997	0.972
	Average	**0.975**	**0.971**	**0.968**	**0.999**	**1.000**	**0.998**	**0.997**	**0.971**
MAE	1	0.609	0.748	0.753	0.093	0.089	0.125	0.265	0.744
	2	0.653	0.744	0.741	0.102	0.085	0.100	0.247	0.725
	3	0.646	0.760	0.782	0.093	0.088	0.167	0.270	0.749
	4	0.714	0.588	0.620	0.093	0.090	0.123	0.191	0.576
	5	0.700	0.739	0.715	0.093	0.089	0.096	0.240	0.720
	Average	**0.664**	**0.716**	**0.722**	**0.095**	**0.088**	**0.122**	**0.243**	**0.703**

which could indicate that the hyperparameter space is relatively well-behaved for this problem, and thus even simpler optimization strategies like RS can perform nearly as well as more sophisticated ones like BO. These results imply that if computation time or resources are a factor, RS could be a more efficient approach without a significant trade-off in model performance. Figure 6 and Figure 7 illustrate the prediction comparison via the different optimization techniques.

Table 6. RS optimization techniques prediction for H_2 production

Metrics	K-Folds	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.767	0.989	1.147	0.319	0.115	0.274	0.332	1.007
	2	0.782	0.912	0.973	0.117	0.111	0.170	0.319	0.888
	3	0.825	0.943	1.074	0.135	0.110	0.267	0.325	0.937
	4	0.891	0.750	0.872	0.110	0.112	0.203	0.246	0.762
	5	0.854	0.878	0.934	0.126	0.123	0.133	0.294	0.876
	Average	**0.879**	**0.894**	**1.000**	**0.161**	**0.114**	**0.209**	**0.303**	**0.894**
R^2	1	0.969	0.967	0.956	0.997	1.000	0.997	0.996	0.966
	2	0.974	0.966	0.961	0.999	0.999	0.999	0.996	0.967
	3	0.977	0.972	0.964	0.999	1.000	0.998	0.997	0.972
	4	0.977	0.979	0.971	1.000	1.000	0.999	0.998	0.978
	5	0.824	0.972	0.968	0.999	0.999	0.998	0.997	0.972
	Average	**0.975**	**0.971**	**0.964**	**0.999**	**1.000**	**0.998**	**0.997**	**0.971**
MAE	1	0.614	0.748	0.784	0.115	0.086	0.128	0.265	0.744
	2	0.652	0.744	0.741	0.092	0.088	0.121	0.249	0.725
	3	0.645	0.760	0.804	0.100	0.087	0.153	0.272	0.749
	4	0.713	0.588	0.644	0.088	0.091	0.139	0.187	0.576
	5	0.701	0.739	0.710	0.098	0.093	0.105	0.248	0.720
	Average	**0.665**	**0.716**	**0.737**	**0.099**	**0.089**	**0.129**	**0.244**	**0.703**

Table 7. BO techniques prediction for H_2 production

Metrics	K-Folds	Tree Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	1	0.764	0.989	1.048	0.125	0.114	0.294	0.339	1.007
	2	0.781	0.912	0.949	0.121	0.106	0.149	0.314	0.888
	3	0.825	0.943	1.005	0.141	0.108	0.348	0.347	0.937
	4	0.891	0.750	0.808	0.113	0.111	0.213	0.241	0.762
	5	0.857	0.878	0.892	0.113	0.124	0.122	0.291	0.876
	Average	**0.824**	**0.894**	**0.941**	**0.123**	**0.113**	**0.225**	**0.306**	**0.894**
R^2	1	0.969	0.967	0.963	0.999	1.000	0.997	0.996	0.966
	2	0.974	0.966	0.963	0.999	1.000	0.999	0.996	0.967
	3	0.977	0.972	0.968	0.999	1.000	0.996	0.996	0.972
	4	0.977	0.979	0.975	1.000	1.000	0.998	0.998	0.978
	5	0.978	0.972	0.971	1.000	0.999	0.999	0.997	0.972
	Average	**0.975**	**0.971**	**0.968**	**0.999**	**1.000**	**0.998**	**0.997**	**0.971**
MAE	1	0.611	0.748	0.753	0.097	0.089	0.143	0.267	0.744
	2	0.653	0.744	0.741	0.090	0.083	0.103	0.247	0.725
	3	0.646	0.760	0.782	0.104	0.086	0.175	0.286	0.749
	4	0.713	0.588	0.620	0.093	0.090	0.129	0.185	0.576
	5	0.700	0.739	0.715	0.088	0.092	0.096	0.244	0.720
	Average	**0.665**	**0.716**	**0.722**	**0.094**	**0.088**	**0.129**	**0.246**	**0.703**

Table 8. Optimization Techniques Performance Comparison for H_2 Production

Metrics	Optimization techniques	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	GS Optimization	0.823	0.894	0.941	0.126	0.112	0.212	0.302	0.894
	RS Optimization	**0.879**	**0.894**	**1.000**	**0.161**	**0.114**	**0.209**	**0.303**	**0.894**
	BO	0.824	0.894	0.941	0.123	0.113	0.225	0.306	0.894
R2	GS Optimization	0.975	0.971	0.968	0.999	1.000	0.998	0.997	0.971
	RS Optimization	0.975	0.971	0.964	0.999	1.000	0.998	0.997	0.971
	BO	0.975	0.971	0.968	0.999	1.000	0.998	0.997	0.971
MAE	GS Optimization	0.664	0.716	0.722	0.095	0.088	0.122	0.243	0.703
	RS Optimization	0.665	0.716	0.737	0.099	0.089	0.129	0.244	0.703
	BO	0.665	0.716	0.722	0.094	0.088	0.129	0.246	0.703

Table 9. Optimization Techniques Performance Comparison for CH_4 Production

Metrics	Optimization techniques	Linear Based Model			Tree Based Model			Ensemble	
		Ridge	Lasso	Elastic Net	Decision Tree	Random Forest	XGBoost	Linear Models	Tree Models
RMSE	GS Optimization	0.285	0.425	0.342	0.128	0.111	0.130	0.141	0.300
	RS Optimization	0.285	0.425	0.344	0.124	0.109	0.134	0.142	0.300
	BO	0.285	0.425	0.342	0.125	0.109	0.127	0.141	0.300
R^2	GS Optimization	0.964	0.921	0.949	0.992	0.995	0.992	0.991	0.960
	RS Optimization	0.964	0.921	0.948	0.993	0.995	0.992	0.991	0.960
	BO	0.964	0.921	0.949	0.993	0.995	0.993	0.991	0.960
MAE	GS Optimization	0.217	0.340	0.283	0.095	0.087	0.096	0.112	0.233
	RS Optimization	0.217	0.340	0.283	0.098	0.085	0.098	0.113	0.232
	BO	0.217	0.340	0.283	0.097	0.085	0.094	0.112	0.232

These results imply that if computation time or resources are a factor, Random Search could be a more efficient approach without a significant trade-off in model performance. Figure 6 and Figure 7 illustrate the prediction comparison via the different optimization techniques.

5. CONCLUSIONS

Machine Learning models have been strongly investigated in several fields including hydrogen production prediction due to their signification performance. However, to achieve optimal performance of machine learning models, hyperparameter optimization is crucial. Hyperparameter Optimization is the technique for determining the ideal hyperparameter configuration for a given algorithm trained on a certain dataset. Once the hyperparameters have been established, the algorithm uses the data to learn the model's parameters. This study investigated the role of hyperparameter optimizations in ML models for CH_4 and H_2 production prediction from Supercritical Water Gasification was investigated specifically Grid search (GS) optimization, Random search (RS) Optimization, and Bayesian optimization (BO). The results were validated using a 5-fold cross-validation. The following observations were made in this study;

- From the correlation heatmap, it is observed that Temperature is highly positively correlated with CE%, production of H_2, and the production of CH_4, with values of 0.87, 0.81, and 0.88, respectively indicating that as the temperature increases, these variables tend to increase as well.
- CE% has a very strong positive correlation with H2 and CH_4, with correlation coefficients of 0.98 and 0.95, respectively indicating that better conversion efficiency is closely associated with higher production of both hydrogen and methane.
- H_2 and CH_4 are also strongly positively correlated with each other (0.88). This indicates a relationship where an increase in the production of one is associated with an increase in the production of the other.
- For model selection, Random Forest is the best model for H_2 and CH_4 Production prediction due to its strong performance across all metrics and robustness against overfitting.
- The choice of optimization technique does not significantly impact the performance of the models, which could indicate that the hyperparameter space is relatively well-behaved for this problem, and thus even simpler optimization strategies like RS can perform nearly as well as more sophisticated ones like BO.

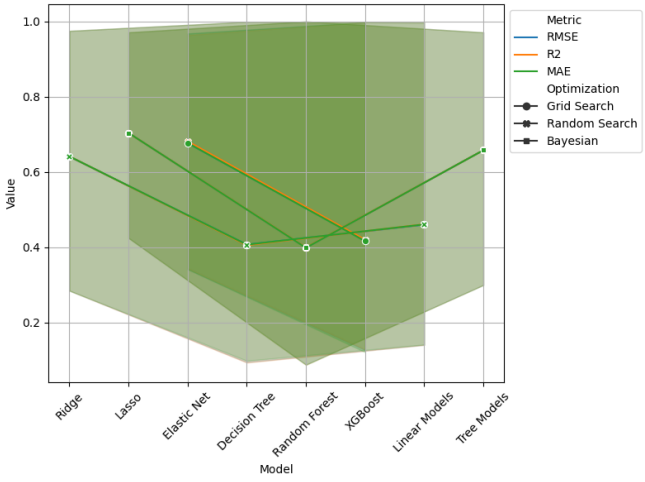

Fig. 7 H_2 Production Optimization Comparison

In conclusion, these results imply that if computation time or resources are a factor, RS could be a more efficient approach without a significant trade-off in model performance. Future research could explore advanced hyperparameter optimization techniques like genetic algorithms or reinforcement learning to enhance

CH_4 and H_2 production predictions from supercritical water gasification models. Additionally, incorporating multi-objective optimization might help balance prediction accuracy and computational efficiency. Further studies could also examine the impact of different data preprocessing methods on model performance.

ACKNOWLEDGEMENT

The authors gratefully acknowledge the support of the Natural Science Foundation of Sichuan (24NSFSC1803) and the National Natural Science Foundation of China (NFSC, Grant No. 52007025).

REFERENCE

[1] B. Buragohain, P. Mahanta, and V. S. Moholkar, "Biomass gasification for decentralized power generation: The Indian perspective," *Renew. Sustain. Energy Rev.*, vol. 14, no. 1, pp. 73–92, 2010.

[2] K. Kang, R. Azargohar, A. K. Dalai, and H. Wang, "Systematic screening and modification of Ni-based catalysts for hydrogen generation from supercritical water gasification of lignin," *Chem. Eng. J.*, vol. 283, pp. 1019–1032, 2016.

[3] S. N. Reddy, S. Nanda, A. K. Dalai, and J. A. Kozinski, "Supercritical water gasification of biomass for hydrogen production," *Int. J. Hydrogen Energy*, vol. 39, no. 13, pp. 6912–6926, 2014.

[4] H. H. Schobert and C. Song, "Chemicals and materials from coal in the 21st century," *Fuel*, vol. 81, no. 1, pp. 15–32, 2002.

[5] Y. Hu, M. Gong, X. Xing, H. Wang, Y. Zeng, and C. C. Xu, "Supercritical water gasification of biomass model compounds: A review," *Renew. Sustain. Energy Rev.*, vol. 118, p. 109529, 2020.

[6] B. Lu, Z. Ge, Y. Chen, J. Shi, and H. Jin, "Study on supercritical water gasification reaction and kinetic of coal model compounds," *Fuel Process. Technol.*, vol. 230, p. 107210, 2022.

[7] Y. A. Ali, E. M. Awwad, M. Al-Razgan, and A. Maarouf, "Hyperparameter search for machine learning algorithms for optimizing the computational complexity," *Processes*, vol. 11, no. 2, p. 349, 2023.

[8] H. Jin, C. Fan, L. Guo, X. Zhao, C. Cao, and W. Wei, "Pore structure and its evolution in char during supercritical water gasification process," *J. Porous Media*, vol. 22, no. 2, 2019.

[9] Y. J. Lu, H. Jin, L. J. Guo, X. M. Zhang, C. Q. Cao, and X. Guo, "Hydrogen production by biomass gasification in supercritical water with a fluidized bed reactor," *Int. J. Hydrogen Energy*, vol. 33, no. 21, pp. 6066–6075, 2008.

[10] Y. Li, L. Guo, X. Zhang, H. Jin, and Y. Lu, "Hydrogen production from coal gasification in supercritical water with a continuously flowing system," *Int. J. Hydrogen Energy*, vol. 35, no. 7, pp. 3036–3045, 2010.

[11] L. Bei, Z. Ge, C. Ren, D. Su, Z. Ren, and L. Guo, "Numerical study on supercritical water partial oxidation of ethanol in a continuous reactor," *Energy*, vol. 249, p. 123750, 2022.

[12] J. Li, L. Pan, M. Suvarna, and X. Wang, "Machine learning aided supercritical water gasification for H2-rich syngas production with process optimization and catalyst screening," *Chem. Eng. J.*, vol. 426, p. 131285, 2021.

[13] S. Zhao, J. Li, C. Chen, B. Yan, J. Tao, and G. Chen, "Interpretable machine learning for predicting and evaluating hydrogen production via supercritical water gasification of biomass," *J. Clean. Prod.*, vol. 316, p. 128244, 2021.

[14] A. Sharma and R. Nath, "H 2-rich syngas production from gasification involving kinetic modeling: RSM-utility optimization and techno-economic analysis," *RSC Adv.*, vol. 13, no. 15, pp. 10308–10321, 2023.

[15] Q. Dang, X. Zhang, Y. Zhou, and X. Jia, "Prediction and optimization of syngas production from a kinetic-based biomass gasification process model," *Fuel Process. Technol.*, vol. 212, p. 106604, 2021.

[16] V. Marcantonio, L. Di Paola, M. De Falco, and M. Capocelli, "Modeling of Biomass Gasification: From Thermodynamics to Process Simulations," *Energies*, vol. 16, no. 20, p. 7042, 2023.

[17] L. Liu *et al.*, "On the variance of the adaptive learning rate and beyond," *arXiv Prepr. arXiv1908.03265*, 2019.

[18] T. Yu and H. Zhu, "Hyper-parameter optimization: A review of algorithms and applications," *arXiv Prepr. arXiv2003.05689*, 2020.

[19] C. Kelterborn, M. Mazur, and B. V Petrenko, "Gravilon: Applications of a new gradient descent method to machine learning," *arXiv Prepr. arXiv2008.11370*, 2020.

[20] R. Turner *et al.*, "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020," in *NeurIPS 2020 Competition and Demonstration Track*, 2021, pp. 3–26.

[21] M. Feurer and F. Hutter, "Hyperparameter optimization," *Autom. Mach. Learn. Methods, Syst. challenges*, pp. 3–33, 2019.

[22] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv Prepr. arXiv1502.02127*, 2015.

[23] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *J. Mach. Learn. Res.*, vol. 13, no. 2, 2012.

[24] C.-W. Hsu, C.-C. Chang, C.-J. Lin, and others, "A practical guide to support vector classification." Taipei, 2003.

[25] D. Chicco, "Ten quick tips for machine learning in computational biology," *BioData Min.*, vol. 10, no. 1, p. 35, 2017.

[26] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. De Feitas, "Bayesian optimization in a billion dimensions via random embeddings," *J. Artif. Intell. Res.*, vol. 55, pp. 361–387, 2016.

[27] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.

[28] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," *Adv. Neural Inf. Process. Syst.*, vol. 25, 2012.

[29] H. Jin, X. Zhao, S. Guo, C. Cao, and L. Guo, "Investigation on linear description of the char conversion for the process of supercritical water gasification of Yimin lignite," *Int. J. Hydrogen Energy*, vol. 41, no. 36, pp. 16070–16076, 2016.