



# Machine learning and LSSVR model optimization for gasification process prediction

Wei Cong<sup>1</sup>

Received: 10 February 2024 / Accepted: 18 July 2024  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2024

## Abstract

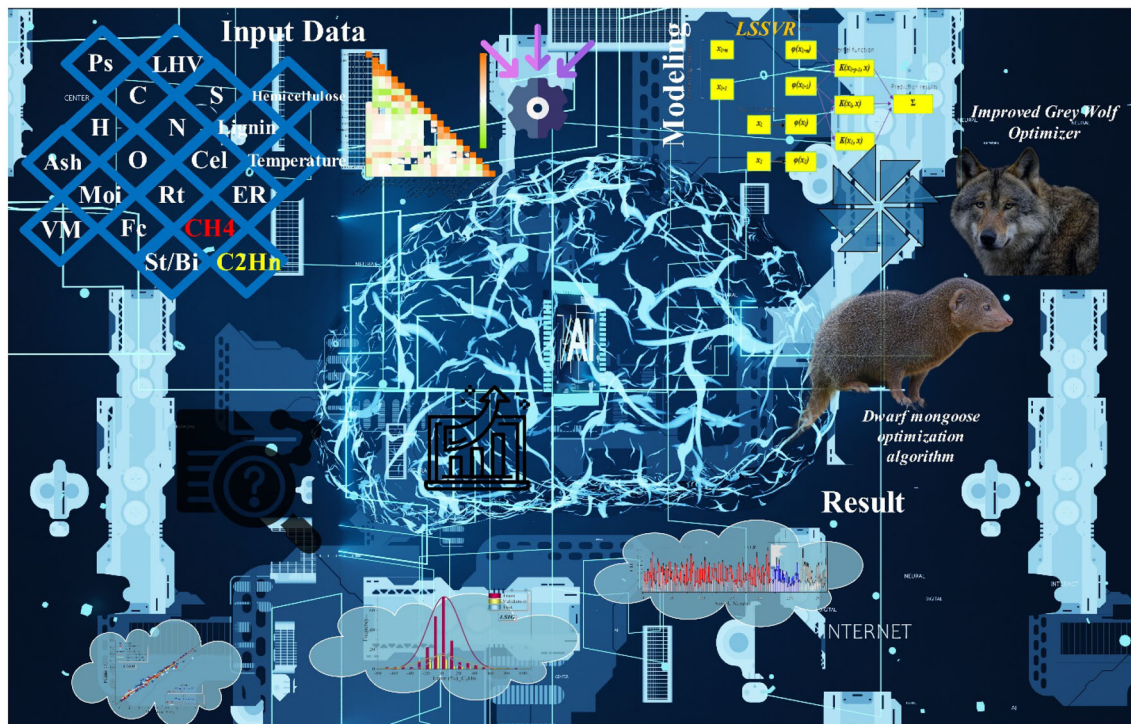
Gasification stands as a transformative thermochemical process, ingeniously converting carbon-rich substances like methane ( $\text{CH}_4$ ) and a spectrum of hydrocarbons, including ethylene ( $\text{C}_2\text{H}_n$ ), into a versatile synthesis gas (syngas). This dynamic blend predominantly comprises hydrogen ( $\text{H}_2$ ) and carbon monoxide ( $\text{CO}$ ), presenting a potent feedstock for diverse industrial applications. In recent years, the focus on sustainable energy has intensified due to concerns about climate change, energy security, and dwindling fossil fuel reserves. Biomass energy has emerged as a promising alternative, offering the potential for a global circular economy and carbon neutrality, thanks to its abundant resources and reliable energy production. This article introduces two hybrid models that combine Least Square Support Vector Regression (LSSVR) with Dwarf Mongoose Optimization (DMO) and the Improved Grey Wolf Optimization Algorithm (IGWO). These models utilize nearby biomass data to predict the elemental compositions of  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$ . The assessment of both individual and hybrid models has demonstrated that integrating LSSVR with these optimizers significantly improves the accuracy of  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$  predictions. According to the findings, the LSDM model emerges as the top performer for predicting both  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$ , achieving impressive  $R^2$  values of 0.988 and 0.985, respectively. Moreover, the minimal RMSE values of 0.367 and 0.184 for  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$  predictions respectively affirm the precision of the LSDM model, rendering it a suitable option for practical real-world applications. Accurate predictions enable the design of systems that efficiently convert a wide range of feedstocks into valuable syngas, which can be employed to generate heat, electricity, fuels, and chemicals. By understanding and optimizing gasification processes, it becomes possible to minimize emissions of pollutants, reduce waste, and mitigate greenhouse gas emissions through carbon capture and utilization technologies.

---

✉ Wei Cong  
kgy\_congwei@163.com

<sup>1</sup> School of Computer Science, Xijing University, Xi'an  
710123, Shaanxi, China

## Graphical abstract



**Keywords** Biomass gasification · Least square support vector regression · Dwarf mongoose optimization · Improved grey wolf optimization algorithm · Machine learning

## Abbreviations

CH <sub>4</sub>	Methane	RMSE	Root Mean Square Error
C <sub>2</sub> H <sub>n</sub>	Ethylene	MAE	Mean Absolute Error
syngas	Synthesis gas	MRAE	Mean Relative Absolute Error
H <sub>2</sub>	Hydrogen	RSR	RMSE-observations standard deviation ratio
CO	Carbon monoxide	$\psi$	Mapping function
LSSVR	Least Square Support Vector Regression	$\varepsilon$	Precision
SVM	Support Vector Machine	$\delta$ and $\delta_i^*$	Slack variables
DMO	Dwarf Mongoose Optimization	$c$	Trade-off parameter
IGWO	Improved Grey Wolf Optimization Algorithm	$\Omega$	Regulatory factor
RE	Renewable energy	$\alpha_i$	Lagrangian multiplier
ML	Machine learning	Q and e	Lagrangian function
AI	Artificial intelligence	$\lambda$	Kernel trick
VM	Volatile matter content	$v^2$	Square of the kernel bandwidth
Fc	Fixed carbon	K	Number of input patterns
LHV	Lower heating value	d	Number of dimensions in the underlying problem
O	Oxygen content	n	Size of the population.
NO	Nitrogen content	$Var_{Size}$	Problem's dimensions
SB	Steam-to-biomass ratio	$Var_{Min}$	Minimum limit of limits
ER	Equivalence Ratio	$Var_{Max}$	Maximum limit of limits
Rt	Residence time	$peep$	Acoustic signa
R <sup>2</sup>	Coefficient of Determination	$bs$	Count of individuals
		$\varphi$	Uniform distribution
		$\vec{M}$	Driving force

*rand* Random value

## 1 Introduction

### 1.1 Background

Rising global energy demand is a persistent concern across social, political, and business domains, expected to continue for decades (Vo Thanh et al. 2024; Vo Thanh et al. 2024). The current energy supply falls short due to climate change and technological shifts, with a heavy reliance on non-renewable fossil fuels like coal, natural gas, and oil. These fossil fuels constitute a large part of global energy consumption, contributing to rising atmospheric carbon dioxide and global warming (Velvizhi et al. 2022; Raud et al. 2019). The adoption of renewable energy (RE) has garnered widespread support as a feasible substitute for fossil fuels and as a strategy to alleviate the harmful consequences of emissions resulting from the combustion of fossil fuels. Renewable energy sources, including solar, wind, hydropower, biomass-biofuels, and other options, have generated significant interest in the pursuit of creating more environmentally friendly and sustainable energy systems with lower carbon emissions (Narnaware and Panwar 2022).

Biomass-derived from agricultural leftovers or forests has the potential to be utilized for the production of various materials and bioenergy that are essential in contemporary societies. When compared to other available resources, biomass stands out as one of the most abundant and widely distributed resources globally. Consequently, biomass holds the promise of serving as a locally and globally accessible renewable energy source (Hakeem et al. 2016). Bioenergy, classified as a renewable energy source, stands as a prospective alternative to conventional fossil fuels. Furthermore, the conversion of bioenergy from waste materials supports the progression of sustainable waste management practices, aligning with the principles of a circular economy and the achievement of carbon neutrality (Ewees et al. 2024; Chen et al. 2021). While renewable energy sources suffer from issues of irregular availability, fluctuation, and unpredictability, the progress in energy storage technologies has notably offered successful resolutions to these problems, ultimately enhancing energy security (Hassan et al. 2022; Malka et al. 2022).

Gasification plays a pivotal role in biomass utilization, providing substantial flexibility in utilizing a variety of feedstock materials and producing diverse output products (Ahrenfeldt et al. 2013). The biomass gasification process involves transforming solid or liquid organic compounds into both a gas or vapor phase and a solid phase. The resulting gas phase, commonly referred to as syngas, possesses significant heating potential and can serve for power generation or the

production of Zhang et al. 2024; Qian et al. 2013). Biomass gasification is regarded as a means to expand the utilization of biomass for energy generation, enabling the broader adoption of biomass as an energy source (Wu et al. 2014). Numerous valuable products can be obtained through the gasification of biomass, such as syngas, heat, power, biofuels, fertilizer, and biochar. Syngas can undergo further processing, mainly through the Fischer–Tropsch method, to produce substances like dimethyl ether, methanol, and various chemical feedstocks. Typically, biomass feedstocks are categorized into four primary groups: herbaceous biomass, marine biomass, woody biomass, and manure. The design of the gasifier is typically tailored to yield a specific product, with the choice and optimization of the feedstock material being a critical parameter (McKendry 2002). Gasifiers can be categorized into four main types: moving beds (including downdraft and updraft designs), circulating beds, and fluidized bed gasifiers. These types are distinguished based on how the biomass and gas interact within the gasifier (Vo Thanh et al. 2020; Basu 2010; Ciferno and Marano 2002).

### 1.2 Related works

The adoption of ML models (Xie et al. 2013; Sadaghat et al. 2024), including ANNs (Jain et al. 1996; Yao 1999; Goh 1995; Eberhart and Shi 1998) and SVM (Awad and Khanna 2015; Rodriguez-Galiano et al. 2015; Zhang and Chapter 2020; Wang 2005), in gasification research is experiencing notable traction (Ascher et al. 2022a; Wang et al. 2021). ML methodologies serve as indispensable instruments for fine-tuning diverse thermochemical processes within bioenergy, notably gasification (Wang et al. 2022; Tang et al. 2021). By leveraging ML models, the efficiency and sustainability of gasification processes can be substantially heightened, simultaneously addressing a spectrum of environmental, economic, and social considerations. Numerous studies have underscored the efficacy of ML models in accurately predicting and optimizing various facets of gasification processes. The innovative application of ML techniques in gasification research has yielded significant advancements across various domains. George et al. (George et al. 2018) utilized an ANN model to achieve remarkable accuracy in predicting syngas production from a lab-scale bubbling fluidized bed gasifier. Ascher et al. (Ascher et al. 2022b) further advanced this frontier by developing a sophisticated ANN model that attained an impressive  $R^2$  value of 0.9310, notably predicting gasification technology performance. This breakthrough bears immense potential for revolutionizing fields like energy production and waste management. Shenbagaraj et al. (Shenbagaraj et al. 2021) introduced the ANN models' capabilities, demonstrating their proficiency in forecasting a broader spectrum of gasification outputs, including syngas composition and yield, with remarkable precision (Han et al.

2008). Additionally, the application of SVM algorithms has shown promise, particularly in estimating optimal oxygen-steam ratios in biomass and coal gasification processes, as demonstrated by the work of Bahadar et al. (Bahadar et al. 2022). Their SVM model exhibited strong alignment with actual data, showcasing its potential to optimize gasification processes effectively. Moreover, Bahadar et al.'s (Tang et al. 2021) investigation into generating hydrogen-rich syngas in co-gasification processes involving coal and biomass highlights the pivotal role of ML in enhancing hydrogen-rich syngas production across diverse gasification scenarios. Further contributions to the field include the work of Baruah et al. (2017), who employed an ANN model to predict gas yields with exceptional precision from a downdraft gasifier. Similarly, Serrano and Castello (2020) leveraged ANN modeling to accurately predict tar production during the gasification of woody biomass. Meanwhile, Ceylan et al. (2021) delved into the application of several ML algorithms to forecast critical factors in gasification processes. Their findings revealed that the Random Forest model outperformed others in predicting product gas generation and gas heating value, underscoring the significant potential of ML in advancing gasification procedure.

As demonstrated in the literature review, numerous researchers have undertaken the task of predicting gasification processes, employing various models for this purpose. However, it is noteworthy that, to date, the LSSVR model has not been utilized for gasification process prediction.

### 1.3 Objective

The objective of this article is to fill a notable gap in the existing literature by introducing the implementation of the LSSVR model for predicting gasification processes, specifically focusing on  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$ . This research endeavors to contribute novel insights by employing a machine learning approach, specifically LSSVR-based models, to forecast syngas generation in biomass gasification. The study innovatively integrates two optimization techniques, namely the DMO and the IGWO, aimed at augmenting model performance. The selection of these algorithms stems from their demonstrated efficacy in optimizing complex processes and their potential to enhance predictive accuracy in gasification modeling. To assess the efficacy of these optimization techniques and conduct a comparative analysis, the study employs five statistical metrics, prioritizing precision in prediction outcomes. Accurate forecasting of  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$  enables optimization of gasification conditions to maximize syngas production while minimizing unwanted byproducts, thus enhancing overall process efficiency. Additionally, understanding the concentrations of these components allows for the estimation of the energy

content of syngas, aiding in assessing the economic viability of gasification operations. Moreover, monitoring  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$  levels is crucial for mitigating environmental concerns, as methane is a potent greenhouse gas, and higher hydrocarbons can contribute to air pollution.

### 1.4 Research organization

In the initial segment of this study, the introduction is subdivided into four key subsections: background, related work, objectives, and research organization. Moving forward, the second section delves into explanations pertaining to the dataset, the explanation of the algorithms for optimization as well as the model, and the definition of the performance evaluation metrics. In the third section, the results of the study, comprising hyperparameters' outcomes, tables, and figures based on the metrics, are presented. Additionally, section four, titled "Discussion," is divided into three subsections: one for explanations of the advantages and disadvantages of algorithms, the second for comparison between the results of this study and previous studies, and the third for addressing the limitations of the study. In section five, titled "Summary and Conclusion," a succinct summary of the study alongside its significant conclusions are provided. Finally, section six serves as the conclusive segment, containing the references cited throughout the article.

## 2 Methodological details

### 2.1 Dataset overview

This research focuses primarily on predicting the elemental compositions of  $\text{CH}_4$  and  $\text{C}_2\text{H}_n$ . To accomplish this, the researchers utilize the LSSVR machine learning regression method, along with the integration of the Dwarf Mongoose Optimization Algorithm and the Improved Grey Wolf Optimizer in the training process. The outcomes of the gasification process are contingent upon the gasifier's design, biomass characteristics, key operational factors such as gasifying agent, air-to-fuel ratio, temperature, pressure, and the nature of bed materials, including the presence or absence of catalytically active substances (Molino et al. 2016).

The process of collecting data involved several stages, which encompassed aligning parameters with the dataset's structure, defining particle size as the smallest dimension, and excluding data related to feedstock type and shape. Notably, volatile matter content (VM) and fixed carbon (Fc) were considered dependent variables that have associations with other compositional parameters. The decision to exclude feedstock with lower heating value (LHV), oxygen content (O), and nitrogen content (N) was based on their strong correlations with other input parameters. The steam-to-biomass ratio (SB)

was identified as a factor affecting gasification efficiency, while the Equivalence Ratio (ER) impacted the trade-off between syngas production and combustion (Antonopoulos et al. 2012). Residence time (Rt) emerged as a crucial factor determining gasification efficiency, with values ranging from 16.3 min to 403 min in the study.

Table 1 and Fig. 1 offer statistical insights and correlation into the input variables as well as the two target variables (CH<sub>4</sub> and C<sub>2</sub>H<sub>n</sub>) collected and documented within the database.

## 2.2 Introduction to machine learning models and optimizers

### 2.2.1 Least square support vector regression (LS-SVR)

Vapnik (Vapnik 1995) developed the Support Vector Machine (SVM), a powerful supervised learning method used for function approximation, nonlinear classification, and density estimation in nonlinear classification. When it comes to Support Vector Machine for Regression (SVR), its objective is to discover a linear regression function within a higher-dimensional space in a manner that minimizes the function's slope and ensures minimal deviation from the training data (Ayubi Rad and Ayubirad 2017). In this scenario, the provided training data consists of { $x_i, y_i$ }  $i = 1, 2, \dots, l$ . The definition of the regression function is as follows:

$$f(x) = Q^S x + d \text{ with } Q, x \in R^n \text{ \& } d \in R \tag{1}$$

Typically, in order to take into account the nonlinear relationship between the input and output variables, the SVR algorithm projects the input data into a higher-dimensional space. The mapping function that converts the input  $x$  from the input space to the feature space is represented by the  $\psi$ . Consequently, the following formulation may be used to create the linear regression function inside the feature space:

$$f(x) = Q^S \psi(x) + d \tag{2}$$

To determine  $f(x)$  such that the difference between the function's output and the true target value for all training dataset points is reduced to a value less than  $\epsilon$ , the method of Epsilon SVR ( $\epsilon$ -SVR) is applied. In this case, the challenge of identifying the regression function is represented by the convex optimization problem outlined by the following constraints:

$$\min \frac{1}{2} \|Q\|^2 \begin{cases} y_i - Q^S \psi(x_i) - d \leq \epsilon \\ Q^S \psi(x_i) + d - y_i \leq \epsilon \end{cases} \tag{3}$$

Recognizing that achieving convex optimization and finding an  $f$  function with  $\epsilon$  precision may not always be

attainable, the introduction of  $\delta$  and  $\delta_i^*$  as slack variables are employed. Thus, the following is a new structure that may be used for the convex optimization problem:

$$\min \frac{1}{2} \|Q\|^2 + c \sum_{i=1}^l (\delta_i + \delta_i^*) \tag{4}$$

$$\begin{cases} y_i - Q^S \psi(x_i) - d \leq \epsilon + \delta_i \\ Q^S \psi(x_i) + d - y_i \leq \epsilon + \delta_i^* \\ \delta_i, \delta_i^* \geq 0 \end{cases}$$

here, the  $c$ , with  $c \geq 0$ , serves as the trade-off parameter that balances the smoothness of the function  $f$  against the tolerance level for deviations exceeding  $\epsilon$ .

The Least Squares Support Vector Machine (LS-SVM) is an improved variation of the SVM algorithm designed for tasks involving classification and regression analysis. LS-SVM deals with the computational challenges of SVM by transforming quadratic optimization problems into a set of linear equations, simplifying the solving process (Okkan and Serbes 2012; Pham et al. 2019). More precisely, By tackling the following optimization problems, LS-SVM simplifies the problem:

$$\min \frac{1}{2} \|Q\|^2 + \frac{\Omega}{2} \sum_{i=1}^l e_i^2 \tag{5}$$

Such that  $y_i = Q^S \psi(x_i) + d + e_i,$

The parameter  $\Omega$  serves as a regulatory factor, and it symbolizes how LSSVR achieves a trade-off between minimizing training errors and enhancing the function's smoothness. This balance is achieved by using a Lagrangian function that has the following structure:

$$LF = \frac{1}{2} \|Q\|^2 + \frac{\Omega}{2} \sum_{i=1}^l e_i^2 - \sum_{i=1}^l \alpha_i (y_i - Q^S \psi(x_i) - d - e_i), \tag{6}$$

where,  $\alpha_i$  represents the Lagrangian multiplier. To solve the optimization problem, both the primal and dual variables need to be equated to zero.

$$\frac{\partial LF}{\partial Q} = 0 \rightarrow Q = \sum_{i=1}^l \alpha_i \psi(x_i), \tag{7}$$

$$\frac{\partial LF}{\partial d} = 0 \rightarrow \sum_{i=1}^l \alpha_i = 0, \tag{8}$$

$$\frac{\partial LF}{\partial e_i} = 0 \rightarrow \alpha_i = \Omega e_i, \tag{9}$$

**Table 1** Statistical properties for dataset variables of CH<sub>4</sub>, C<sub>2</sub>H<sub>n</sub>

Components	Unit	Properties				
		Min	Max	Median	Avg	St. Dev
Ps [mm]	[mm]	0.250	70.0	5.0	5.60	6.770
LHV	[MJ/kg wb]	11.50	42.90	18.10	18.60	6.680
C	[%daf]	40.10	86.03 0	49.40	51.7 0	9.360
H	[%daf]	3.79 0	14.230	6.120	6.93 0	2.030
N	[%daf]	0.040	7.320	0.5 0	1.160	1.890
S	[%daf]	0.010	1.610	0.330	0.430	0.450
O	[%daf]	4.620	53.40	43.10	41.10	6.830
Ash	[%db]	0.270	44.0	1.910	7.550	12.10
Moi	[%wb]	0.20	27.0	8.50	9.180	5.110
VM	[%db]	56.0	89.10	79.90	77.40	8.380
Fc	[%db]	9.070	23.80	15.0	15.10	2.750
Cel	[%db]	29.60	46.20	39.10	39.20	1.60
Hemicellulose	[%db]	14.0	29.60	21.0	21.10	1.580
Lignin	[%db]	14.0	27.80	23.6 0	23.70	1.290
Temperature	[°C]	553.0	105.0	800.0	801.0	80.60
Rt	[min]	16.30	403.0	100.0	111.0	80.60
St/Bi	[wt/wt]	0.0	4.040	1.210	1.200	0.630
ER	–	0.090	0.870	0.30	0.30	0.10
CH <sub>4</sub>	[vol.% db]	0.250	16.0	4.050	5.190	3.280
C <sub>2</sub> H <sub>n</sub>	[vol.% db]	0.0	9.50	1.750	1.960	1.400

$$\frac{\partial LF}{\partial \alpha_i} = 0 \rightarrow Q^S \psi(x_i) + d + e_i - y_i = 0. \tag{10}$$

where,  $Q$  and  $e$  are replaced within the Lagrangian function to create a linear system in the following manner:

$$\begin{bmatrix} 0 & 1^S \\ 1 & \lambda + \Omega^{-1}I \end{bmatrix} \begin{bmatrix} d \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \tag{11}$$

where:

$$\alpha = [\alpha_1 + \alpha_2, \dots, \alpha_L]^S,$$

$$e = [e_1 + e_2, \dots, e_L]^S,$$

$$\lambda = VV^S,$$

$$V = [\psi(x_1), \psi(x_2), \dots, \psi(x_L)]^S,$$

$$y = [y_1 + y_2, \dots, y_L]^S,$$

$$1 = [1 + 1, \dots, 1]^S.$$

The matrix  $\lambda$  can be constructed using the kernel approach, as demonstrated by Eq. (12), leading to the reformulation of the solution for Eq. (11) as expressed in Eq. (13).

$$\lambda_{ij} = \psi(x_i)^S \psi(x_j)^S = K(x_i, x_j). \tag{12}$$

$$c = \frac{1^T (\lambda + \Omega^{-1}I)^{-1} y}{1^T (\lambda + \Omega^{-1}I)^{-1} I} \tag{13}$$

$$\alpha = (\lambda + \Omega^{-1}I)^{-1} (y - d1). \tag{14}$$

The RBF ( $v^2$ ) can serve as a kernel function representing an infinite-dimensional function space. Additionally, with the values of  $d$  and  $\alpha$  that have been derived, the optimal regression function can be formulated in the following:

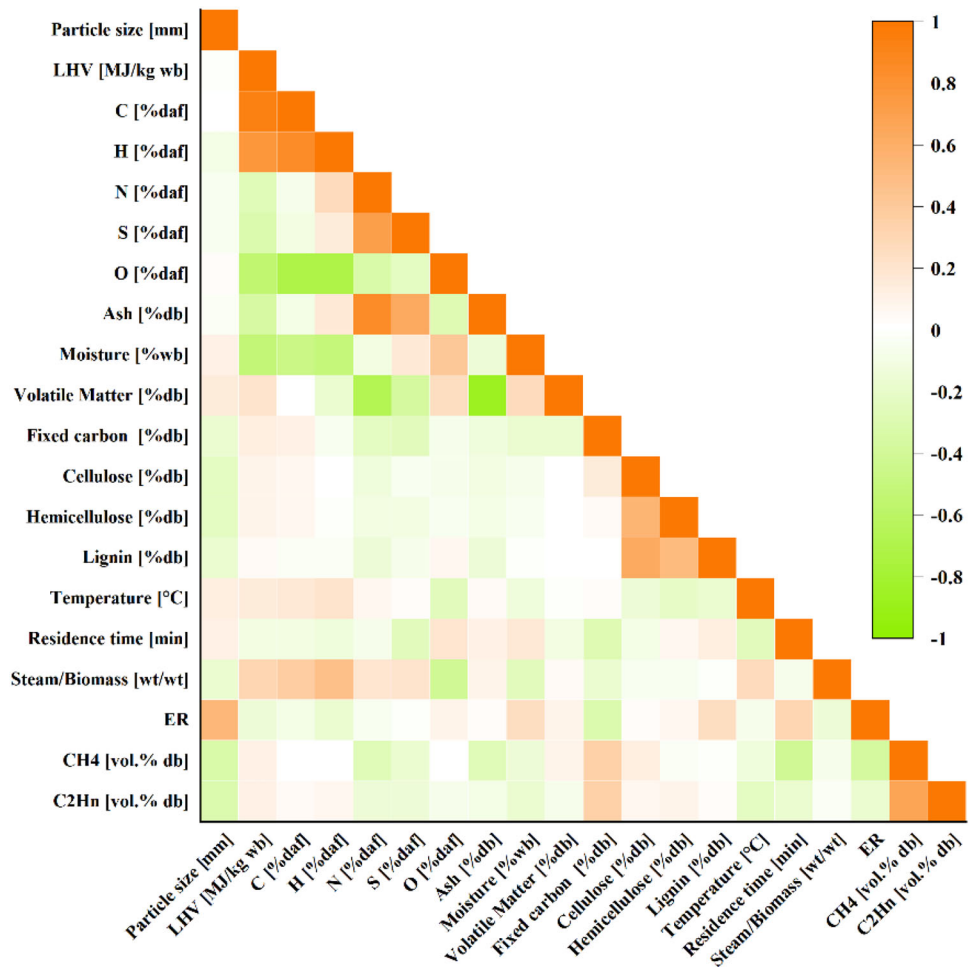
$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) + d, \tag{15}$$

$$K(x_i, x) = \exp\left(-\frac{\|x_i - x\|^2}{2v^2}\right) \tag{16}$$

$v^2$  represents the square of the kernel bandwidth.

In previous studies, several ideal values for  $\Omega$  and  $v^2$  have been suggested. For instance,  $1/K$  (where  $K$  is the number of input patterns) and 1 for  $\Omega$  and  $v^2$  are proposed by Hsu et al. (Hsu et al. 2003). Aiyer et al. (Aiyer et al. 2014) introduced a trial-and-error method, which can be quite time-intensive,

**Fig. 1** Matrix of correlation for the relationship between the variables



especially when estimating  $\Omega$  and  $v^2$ , as these parameters can take on any positive real values. Therefore, optimization techniques can be employed to fine-tune these parameters for LSSVR.

**2.2.2 Dwarf mongoose optimization (DMO)**

The population-based, stochastic metaheuristic Dwarf Mongoose Optimization (DMO) method draws its inspiration from the dwarf mongoose’s reported social and foraging behaviors, as noted by Helogale (Agushaka et al. 2022).

DMO starts the problem-solving process by selecting the first set of possible solutions from the mongoose colony. It commences with the generation of an initial candidate solution population. These initial solutions are created randomly, adhering to the specified upper and lower limits as per the requirements of the specific problem under consideration.

$$k = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,d-1} & x_{1,d} \\ x_{2,1} & x_{2,2} & \dots & x_{2,d-1} & x_{2,d} \\ \vdots & \vdots & & x_{1,1} \vdots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,d-1} & x_{n,d} \end{bmatrix} \quad (17)$$

$d$  represents the number of dimensions in the underlying problem.

$n$  indicates the size of the population.

Equation (18) computes the positional attribute of individual elements in a population, which is represented by  $x_{i,j}$ :

$$x_{i,j} = \text{unifrnd}(Var_{Min}, Var_{Max}, Var_{Size}) \quad (18)$$

$Var_{Size}$  pertains to the problem’s dimensions and the associated value ranges.

where,  $Var_{Min}$  and  $Var_{Max}$  represent the minimum and maximum limits, respectively.

The function  $\text{unifrnd}$  functions as a random number generator, creating numbers distributed uniformly.

The DMO algorithm adheres to the conventional metaheuristic strategy, which comprises two clear-cut phases: exploitation and exploration. The initial phase involves an exhaustive and meticulous examination of the specified region by each mongoose, known as the intensification process. On the other hand, the second method entails a less systematic exploration of fresh resources like food supplies or resting spots, referred to as diversification. The operation

of the DMO algorithm is characterized by two discernible phases, both guided by three essential social components: the alpha group, scout group, and babysitters. These components have a pivotal role in orchestrating the actions of the solution population, ensuring the algorithm effectively explores and exploits the search space.

- Alpha Group

The selection of the alpha female ( $\alpha$ ) to lead the unit of family is determined using Eq. (19).

$$\alpha = \frac{fit_i}{\sum_{i=1}^n fit_i} \tag{19}$$

$n$  depicts the existing count of mongooses forming the alpha group.

$peep$  denotes the term for an auditory cue produced by an alpha or dominant female mongoose.

$bs$  indicates the count of individuals with the mongoose group responsible for nurturing and overseeing young offspring.

Equation (20) calculates a positive correlation observed between the sleeping mound and the availability of abundant nutritional elements:

$$X_{i+1} = X_i + \varphi * peep \tag{20}$$

$\varphi$  determines a value that follows a uniform distribution within the interval of  $[-1,1]$ .

Each time the algorithm runs, the size and quality of the sleeping mound are evaluated, as shown by mathematical Eq. (21):

$$sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}|, |fit_i|\}} \tag{21}$$

When an inactive accumulation that was previously not in use is identified, a statistical metric is computed using Eq. (22):

$$\rho = \frac{\sum_{i=1}^n sm_i}{n} \tag{22}$$

- Scout Group

Following the fulfillment of the requirements to be a member of a babysitter exchange program, the next step of the procedure is scouting. This stage involves evaluating the location of a potential resting place in relation to the availability of a certain food supply. Taking into account that mongooses do not typically reuse their sleeping places, the scouting team’s task is to find a new mound to aid in the further progress of their explorations. The behavior of the mongoose exhibits a unique pattern that involves scouting and foraging within the parameters of the DMO algorithm. This behavior is grounded on the idea that expanding the distance covered while foraging increases the probability of coming across a new resting place. This mechanism is mathematically illustrated through the utilization of Eqs. (23–25):

$$X_{i+1} = \begin{cases} X_i - CF * phi * rand * [X_i - \vec{M}] & if \rho_{i+1} > \rho_i \\ X_i + CF * phi * rand * [X_i - \vec{M}] & else \end{cases} \tag{23}$$

$$CF = \left(1 - \frac{iter}{Max_{iter}}\right)^{\left(2 \frac{iter}{Max_{iter}}\right)} \tag{24}$$

$$\vec{M} = \sum_{i=1}^n \frac{X_i \times sm_i}{X_i} \tag{25}$$

$\vec{M}$  depicts the driving force that propels the mongoose’s motion toward a recently established resting mound.  $rand$  stands for a random value uniformly distributed in the interval  $[-1,1]$ .

- Babysitters Group

The group responsible for watching for the small offspring stays watchful over them while the scouting and foraging team searches for a suitable location for food and rest. As some group members choose to defer their foraging or scouting tasks until they meet the requirements for the exchange program, the number of qualified applicants for the babysitter exchange decreases. The DMO algorithm’s pseudo-code is provided by Algorithm 2.



**Algorithm 2:** Pseudo-Code of DMO Algorithm

```

Set the parameters of the algorithm:
Generate
for iter = 1: max_iter
Compute the fitness of the mongoose
Set time counter C
Determine the alpha  $\alpha = \frac{fit_i}{\sum_{i=1}^n fit_i}$ 
Obtain a candidate food position
 $X_{i+1} = X_i + \varphi * peep$ 
Guess new fitness of  $X_{i+1}$ 
Guess sleeping mound
 $sm_i = \frac{fit_{i+1} - fit_i}{\max\{|fit_{i+1}, fit_i\}}$ 
Compute the sleeping mound average value  $\rho = \frac{\sum_{i=1}^n sm_i}{n}$ 
Compute the movement vector  $\vec{M} = \sum_{i=1}^n \frac{X_i \times sm_i}{X_i}$ 
Exchange babysitters if  $C \geq L$ 
Set bs position
compute fitness  $fit_i \leq \alpha$ 
Simulate the scout mongoose's next position.
 $X_{i+1} = \begin{cases} X_i - CF * phi * rand * [X_i - \vec{M}] & \text{if } \rho_{i+1} > \rho_i \\ X_i + CF * phi * rand * [X_i - \vec{M}] & \text{else} \end{cases}$ 
Modernize the best solution so far.
end For
return the best solution
end
    
```

**2.2.3 Improved grey wolf optimization algorithm (IGWO)**

• Overview of Grey Wolf Optimization Algorithm

The social structure and hunting behaviors of grey wolves inspire the Grey Wolf Optimizer (GWO) algorithm (Wang et al. 2022). It involves three prominent leader wolves, labeled as  $\alpha$ ,  $\beta$ , and  $\delta$ , which act as top solutions, guiding a group of  $\omega$  wolves in search of the global solution. The wolf hunting process consists of three main stages: encircling, hunting, and engaging with the prey.

• Encircling:

The mathematical representation of how grey wolves encircle their prey is depicted in Eqs. (26) and (27):

$$D = |C \times X_p(t) - X(t)| \tag{26}$$

$$X(t + 1) = X_p(t) - A \times D \tag{27}$$

$X_p$  depicts the location of the target.  
 $t$  indicates the present iteration.

$C$  and  $A$  determine the vectors of coefficients determined using Eqs. (28) and (29):

$$A = 2 \times A \times r_1 - a(t) \tag{28}$$

$$C = 2 \times r_2 \tag{29}$$

The vector's components "a" demonstrate a linear reduction from 2 to 0 as the iterations progress, in accordance with the instructions provided in Eq. (30):

$$a(t) = 2 - (2 \times t) / MaxIter \tag{30}$$

- Hunting:

The mathematical depiction of wolves' hunting behavior is based on the assumption that  $\alpha$ ,  $\beta$ , and  $\delta$  have superior knowledge about the prey's location. As a result, the  $\omega$  wolves are required to follow their movements. Equation (31) describes and clarifies this hunting behavior.

$$\begin{aligned} D_\alpha &= |C_1 \times X_\alpha - X(t)| \\ D_\beta &= |C_2 \times X_\beta - X(t)| \\ D_\delta &= |C_3 \times X_\delta - X(t)| \end{aligned} \tag{31}$$

$C_1$ ,  $C_2$ , and  $C_3$  are computed using the calculations outlined in Eq. (32):

$$\begin{aligned} X_{i1}(t) &= X_\alpha(t) - A_{i1} \times D_\alpha(t) \\ X_{i2}(t) &= X_\beta(t) - A_{i2} \times D_\beta(t) \\ X_{i3}(t) &= X_\delta(t) - A_{i3} \times D_\delta(t) \end{aligned} \tag{32}$$

$X_\alpha$ ,  $X_\beta$ , and  $X_\delta$  denote the positions of the three leading solutions at iteration  $t$ .

The values for  $D_\alpha$ ,  $D_\beta$ , and  $D_\delta$  are precisely determined by the equation mentioned earlier.

$$X(t + 1) = \frac{X_{i1}(t) + X_{i2}(t) + X_{i3}(t)}{3} \tag{33}$$

The hunting process in this algorithm ends when the prey stops moving, a transition controlled by "a" changing from 2 to 0 throughout the course of the iterations, progressively decreasing over time to balance exploration and exploitation. As per guidance (Emary et al. 2017), half of the iterations focus on exploration, while the remaining half shift into the exploitation phase. During exploitation, wolves adjust their positions to random locations between their current positions and the prey's location.

- Improved Grey Wolf Optimization Algorithm (IGWO)

In GWO,  $\alpha$ ,  $\beta$ , and  $\delta$  guide  $\omega$  wolves to promising areas in the search space, but this can lead to being trapped in local optima and reduced population diversity, causing premature convergence. To address these issues, an improved variant, the IGWO, has been introduced. IGWO introduces new search strategies and updates steps with a dashed line border. It operates in three phases: initialization, movement, selection, and updating.

- Initializing phase:

This step involves utilizing Eq. (34) to randomly place all  $N$  wolves in the search space within the preset range  $[l_i, u_j]$ :

$$X_{i,j} = l_j + rand_j[0,1] \times (u_j - l_j), i \in [1, N], j \in [1, D] \tag{34}$$

In each algorithm iteration, the position of the  $i_{th}$  wolf is represented by a vector  $X_i(t) = \{x_{i1}, x_{i2}, \dots, x_{iD}\}$  in the search space. These wolves are organized into a matrix called *Pop*, with  $D$  columns and  $N$  rows, where  $N$  is the population size, and  $D$  is the problem's dimension. The function  $f(X_i(t))$  is used to calculate the fitness value of  $X_i(t)$ .

- Movement phase:

Grey wolves not only engage in group hunting but also exhibit individual hunting behavior (MacNulty et al. 2007). This inspired the enhancement of GWO with the Dimension Learning-Based Hunting (DLH) strategy. In DLH, wolves learn from their neighbors and become potential candidates for new positions  $X_i(t)$ . This introduces two distinct candidates, with the canonical GWO and DLH strategies contributing to their selection.

- The canonical GWO search strategy:

$\alpha$ ,  $\beta$ , and  $\delta$  are the top three wolves in the population in GWO. Using Eqs. (30–32) the coefficients  $a$ ,  $A$ , and  $C$  are computed linearly. Using Eqs. (33–34), prey encircling is calculated from the locations of  $X_\alpha$ ,  $X_\beta$ , and  $X_\delta$ . Equation (35) is used to determine the initial candidate for a wolf's new position, which is represented as  $X_{i-GWO}(t + 1)$ . On the other hand, to tackle the convergence problems of GWO, a new search approach called Dimension Learning-Based Hunting (DLH) is presented. The new position of a wolf in DLH,  $X_{i-DLH}(t + 1)$ , is calculated for every dimension utilizing Eq. (38), in which the wolf gains knowledge from its neighbors and a randomly chosen wolf from *Pop*. Furthermore, DLH produces an additional candidate for the new position, designated as  $X_{i-DLH}(t + 1)$ . This is achieved by computing a radius,  $R_i(t)$ , which depends on the Euclidean

distance between the current position of  $X_i(t)$  and the candidate's position  $X_{i-GWO}(t + 1)$ , as described in Eq. (36).

$$R_i(t) = \|X_i(t) - X_{i-GWO}(t + 1)\| \tag{35}$$

The neighbors of  $X_i(t)$ , represented as  $N_i(t)$ , are determined using Eq. (21) with respect to the radius  $R_i(t)$ .  $D_i$  is the Euclidean distance between  $X_i(t)$  and  $X_j(t)$ .

$$N_i(t) = \{X_j(t) | D_i(X_i(t), X_j(t)) \leq R_i(t), X_j(t) \in Pop\} \tag{36}$$

Once the neighborhood around  $X_i(t)$  is established, a learning process from multiple neighbors is performed utilizing Eq. (37). The  $d_{th}$  dimension of  $X_{i-DLH,d}(t + 1)$  is determined in this equation by adding the  $d_{th}$  dimensions of a randomly selected neighbor,  $X_{n,d}(t)$ , from  $N_i(t)$ , and a randomly selected wolf,  $X_{r,d}(t)$ , from the population, Pop.

$$X_{i-DLH,d}(t + 1) = X_{i,d} + rand \times (X_{n,d}(t) - X_{r,d}(t)) \tag{37}$$

In the selection and updating phase, the initial choice of the superior candidate is made by evaluating the fitness values of two candidates,  $X_{i-GWO}(t + 1)$  and  $X_{i-DLH}(t + 1)$ , through Eq. (38):

$$X_i(t + 1) = \begin{cases} X_{i-GWO}(t + 1) & \text{if } f(X_{i-GWO}) < f(X_{i-DLH}) \\ X_{i-DLH}(t + 1) & \text{otherwise} \end{cases} \tag{38}$$

In conclusion, the search is continued until the predefined number of iterations (*Maxiter*) is reached after this procedure is completed for each individual. The iteration counter (*iter*) is then increased by one.

### 2.3 Assessment indices

Commonly, a comprehensive evaluation of prediction model performance involves employing a set of model selection criteria that includes multiple goodness-of-fit and error indices. This approach helps mitigate the limitations of individual criteria, which on their own may lead to incorrect decisions (Dodo et al. 2022). It is ordinary to employ the following metrics in assessing the performance of a model.

- Coefficient of Determination ( $R^2$ ):

The Coefficient of Determination, commonly represented as  $R^2$ , is a statistical measure employed in regression analysis to quantify the degree to which a model elucidates the variance or variability observed in the dependent variable. This metric operates within a range spanning from 0 to 1, where

higher values signify an improved alignment of the model with the underlying data. Specifically, an  $R^2$  value of 0 indicates that the model does not account for any variance, while an  $R^2$  value of 1 signifies a perfect congruence between the model and the data. It is indicated by the following formula:

$$R^2 = \left( \frac{\sum_{i=1}^n (t_i - \bar{w})(v_i - \bar{v})}{\sqrt{[\sum_{i=1}^n (v_i - \bar{v})^2][\sum_{i=1}^n (t_i - \bar{w})^2]}} \right)^2 \tag{39}$$

- Error evaluation metrics (MAE, RMSE):

MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) are statistical metrics that provide a means to quantify the error between actual and predicted values, yet they differ in their sensitivity to the magnitude of errors. RMSE places more emphasis on larger errors, rendering it suitable when one seeks to penalize significant deviations substantially. Conversely, MAE treats all errors with equal weight and is particularly robust in the presence of outliers in the dataset. The selection between RMSE and MAE should be contingent upon the specific objectives and characteristics of the predictive modeling problem at hand. These measures are formally expressed in Eqs. (40) and (41) as elaborated below:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (v_i - w_i)^2} \tag{40}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n (v_i - w_i) \tag{41}$$

- Mean Relative Absolute Error (MRAE):

MRAE, which stands for Mean Relative Absolute Error, is a statistical measure employed to evaluate the precision of predictive models or the degree of error in forecasting. The average relative absolute error between the expected and actual numbers is quantified. MRAE is typically expressed as a percentage and serves as a reflection of the average relative discrepancy in the model's predictions. It offers valuable insights into how closely the model's forecasts align with the actual data while taking into account the relative scale of the errors. A lower MRAE signifies superior accuracy, whereas a higher MRAE indicates a less accurate model. The formula for computing MRAE is as stated below:

$$MRAE = \frac{1}{n} \sum_{i=1}^n \frac{|w_i - \bar{w}_i|}{|w_i|} \tag{42}$$

- RMSE-observations standard deviation ratio (RSR):

**Table 2** Results of hyperparameter

Target	Models	C	Gama
CH4	LSDM	1	0.01
	LSIG	179.92582	0.395188
C2HN	LSDM	1	0.01
	LSIG	240.3931	0.589175

The RSR (RMSE-observations standard deviation ratio) is a statistical metric for evaluating predictive models, especially in regression analysis. It combines RMSE and the standard deviation of observed data to gauge a model's accuracy while considering data variability. RSR normalizes RMSE by dividing it by the standard deviation, offering a context-aware assessment. Lower RSR values indicate a better model fit to data, while higher values indicate a poorer fit. Equation (43) indicates RSR:

$$RSR = \frac{[\sqrt{\sum_{i=1}^n (w_i - v_i)^2}]}{[\sqrt{\sum_{i=1}^n (w_i - \bar{w})^2}]} \quad (43)$$

Within each equation:  $n$ : number of samples,  $v_i$ : represents the specific predicted value.  $\bar{v}$ : denotes the average of the predicted values,  $w_i$ : signifies the experimentally measured value,  $\bar{w}$ : stands for the mean of the experimentally measured values.

## 3 Results

### 3.1 Hyperparameters' results

In contrast to parameters, hyperparameters are predefined specifications not directly derived from the dataset. Maximizing model performance heavily depends on fine-tuning hyperparameters, a process demanding extensive experimentation and the application of advanced optimization approaches. Presented in Table 2 are the detailed hyperparameter configurations corresponding to the LSDM and LSIG models, highlighting the critical parameters C and Gamma across both output variables, CH4 and C2Hn. For instance, in the LSDM model, the C hyperparameter maintains a constant value of 1 for both CH4 and C2Hn, underscoring a uniform setting for model optimization. In contrast, the LSIG model exhibits variations in C, with values of 179.926 for CH4 and 240.393 for C2Hn, indicating nuanced adjustments tailored to each output variable.

### 3.2 Results of evaluation metrics

This article demonstrates the use of enhanced Machine Learning models, specifically Least Square Support Vector Regression (LSSVR), to predict the composition of  $CH_4$  (methane) and  $C_2H_n$  (hydrocarbons) in the context of biomass gasification. The research incorporates two optimization algorithms, DMO (Dwarf Mongoose Optimization) and the Improved Grey Wolf Optimization Algorithm (IGWO), to create hybrid models known as LSDM and LSIG. The main objectives of this research are to identify the most robust optimization algorithm and to evaluate the predictive performance of both individual and combined hybrid models. This comparative evaluation employs five assessment metrics:  $R^2$ , RMSE, MAE, MRAE, and RSR. The following sections provide a comprehensive analysis of model performance, with dedicated segments for the prediction of  $CH_4$  and  $C_2H_n$ .

#### • $CH_4$ syngas forecasting

According to Table 3, it is evident that for the LSSVR, LSIG, and LSDM models in the three phases of training, validation, and testing, the  $R^2$  value is highest for the LSDM model, approaching 1. In comparison, the LSIG and LSSVR models take the second and third positions, respectively. For example, in the LSDM model, the  $R^2$  values for the mentioned phases are 0.988, 0.974, and 0.978.

RMSE and MSE represent error values based on their definitions. When these values are lower, it indicates better model performance. A thorough examination of Table 3 reveals that the LSDM model demonstrates the best performance, while the LSIG and LSSVR models rank second and third, respectively. In the Train phase of the LSDM model, the lowest RMSE is 0.367, and the lowest MSE is 0.171.

A lower MRAE indicates a greater precision, whereas a higher MRAE shows a less precise model. Therefore, it is concluded that the LSDM, with a value of 0.149 in the Training sector, has the best accuracy.

RSR is a metric used to assess the precision of a predictive model in comparison to real values within a scale from zero to one. When a parameter gradually approaches zero, it signifies a strong level of agreement. In this situation, it is evident that LSDM demonstrates the best performance with a value of 0.111 in the Training phase, the LSIG model follows in second place with a value of 0.975, and the LSSVR model ranks last with a value of 0.207.

A scattered representation of the average variety between measured and predicted values is shown in Fig. 2. Detailed dispersion points are derived from RMSE and  $R^2$ , which primarily controls data distribution so that the lower amount reaches a higher density in RMSE. A greater  $R^2$  value

**Table 3** Result of developed models for CH<sub>4</sub>

Phase	Index values	Models		
		LSSVR	LSIG	LSDM
Train	RMSE	0.687	0.523	0.367
	R <sup>2</sup>	0.967	0.975	0.988
	MAE	0.551	0.444	0.171
	RSR	0.207	0.158	0.111
	MRAE	0.333	0.296	0.149
Validation	RMSE	0.825	0.576	0.523
	R <sup>2</sup>	0.952	0.963	0.974
	MAE	0.664	0.476	0.357
	RSR	0.292	0.204	0.185
	MRAE	0.377	0.467	0.215
Test	RMSE	1.022	0.664	0.716
	R <sup>2</sup>	0.960	0.966	0.978
	MAE	0.785	0.536	0.485
	RSR	0.300	0.195	0.210
	MRAE	0.670	0.797	0.546

suggests that the line fits the data more closely. General comparisons show that the value of  $R^2$  is relatively high for all three models. According to the scatter data, the LSDM model with lower RMSE and higher  $R^2$  values in all three phases (train, validation, test) of the prediction model appear most appropriate.

Figure 3. presents a comparison between the measured and predicted values for CH<sub>4</sub>, confirming that a smaller difference between these two values indicates superior model performance. As shown in the diagram, the maximum difference values for the LSSVR, LSIG, and LSDM models occurred during the Test phase.

Figure 4 shows two graphs comparing the number of samples versus the error percentage of CH<sub>4</sub> and the error percentage versus its frequency. The error rate differs with the number of samples, as shown in the LSSVR charts. The error of this model ranges from -40 to 100 percent, with a high frequency between -20 and 0. For the LSIG and LSDM hybrid models, the error percentage compared to the number of samples is superior compared to the basic model. In LSIG, the error range is -80 to 100 percent, and the highest frequency is -10 to 0, while for the LSDM model, the error range is -20 to 60, and the highest frequency is -5 and 0.

#### • C<sub>2</sub>H<sub>n</sub>syngas forecasting

After reviewing the data presented in Table 4, it is clear that the LSDM model demonstrates superior performance compared to the LSSVR and LSIG models in terms of  $R^2$  values for all three phases: training, validation, and testing.

Specifically, the LSDM model achieved  $R^2$  values of 0.985, 0.979, and 0.973, respectively.

RMSE and MSE serve as metrics for measuring errors, and lower values signify improved model performance. A detailed analysis of Table 5 underscores that the LSDM model exhibits superior performance, with the LSSVR and LSIG models occupying the second and third positions, respectively. In the training phase of the LSDM model, the most favorable RMSE is 0.184, and the most favorable MSE is 0.046.

A lower MRAE signifies greater precision, while a higher MRAE reflects a less precise model. Consequently, it can be deduced that the LSDM, with a value of 0.274 in the validation sector, exhibits the highest level of accuracy.

RSR, a metric employed for evaluating the precision of a predictive model concerning real values on a scale ranging from zero to one, indicates a strong level of agreement as a parameter approaches zero. In this context, it is clear that the LSDM exhibits the most favorable performance, boasting a value of 0.129 in the training phase.

In Fig. 5, the comparisons indicate that the LSDM model, which boasts lower RMSE and higher  $R^2$  values in all three phases of the prediction model (training, validation, testing) with values of 0.985, 0.977, and 0.973 (for  $R^2$ ), and 0.184, 0.284, and 0.337 (for RMSE), respectively, is the most appropriate option for forecasting C<sub>2</sub>H<sub>n</sub>.

Figure 6. provides a visual comparison of the measured and predicted values for C<sub>2</sub>H<sub>n</sub>, affirming that a reduced disparity between these values reflects a higher level of model performance. As depicted in the graph, the largest differences

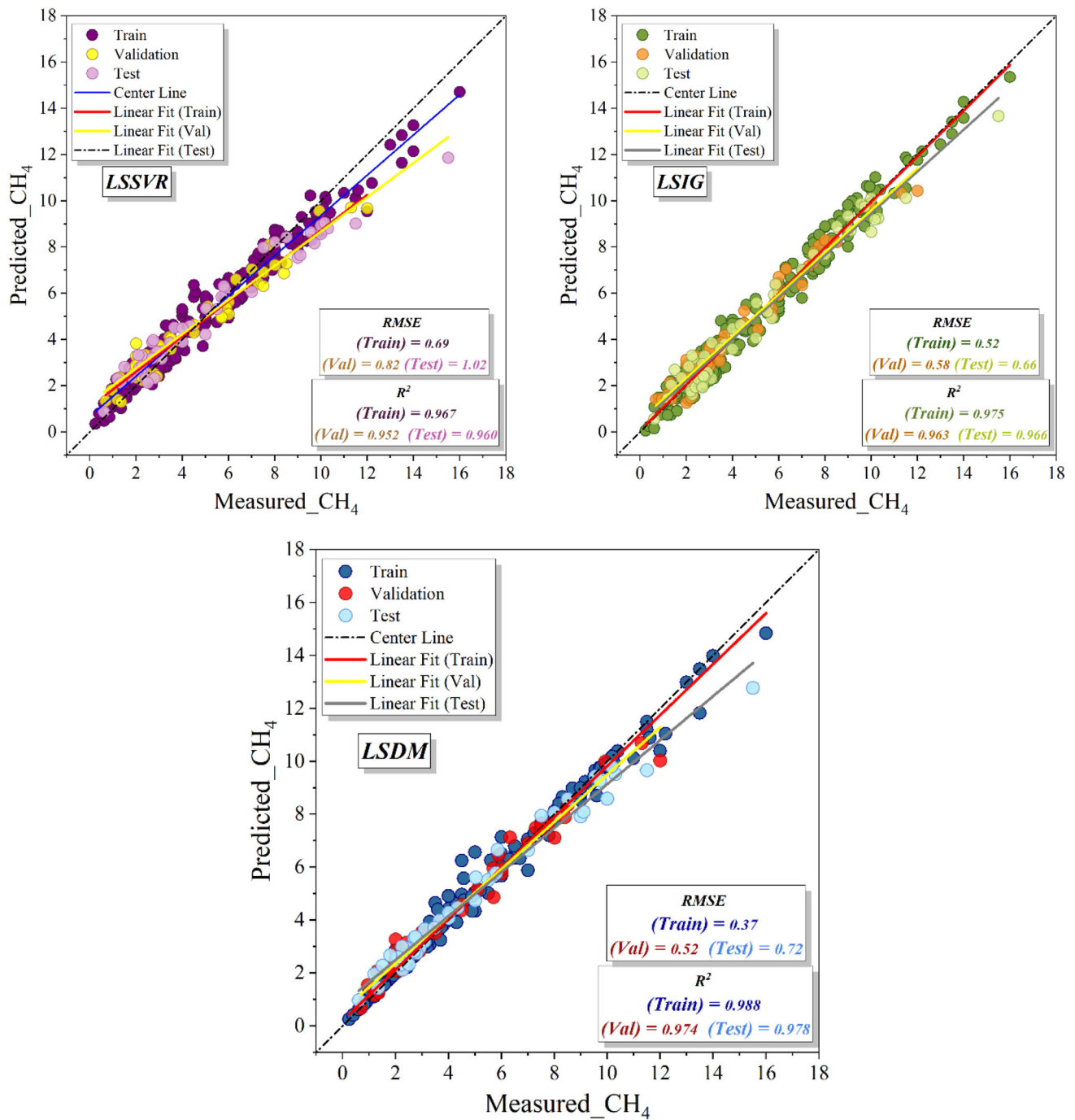


Fig. 2 Scatter plot for correlation between the predicted and measured  $\text{CH}_4$

for the LSSVR, LSIG, and LSDM models were observed in the training phase.

Figure 7 displays two graphs that illustrate the relationship between the number of samples and the error percentage for  $\text{C}_2\text{H}_n$ , as well as the error percentage versus its frequency. The error rate varies with the number of samples, particularly evident in the LSSVR charts, where the model's error ranges from -80 to 100 percent, with a notable frequency between

-20 and 10. Conversely, for the LSIG and LSDM hybrid models, the error percentage in relation to the number of samples outperforms the basic model. In LSIG, the error range is -70 to 100 percent, with the highest frequency occurring around -20 to 15, while for the LSDM model, the error range is -30 to 70, and the highest frequency is approximately -10 to 5.

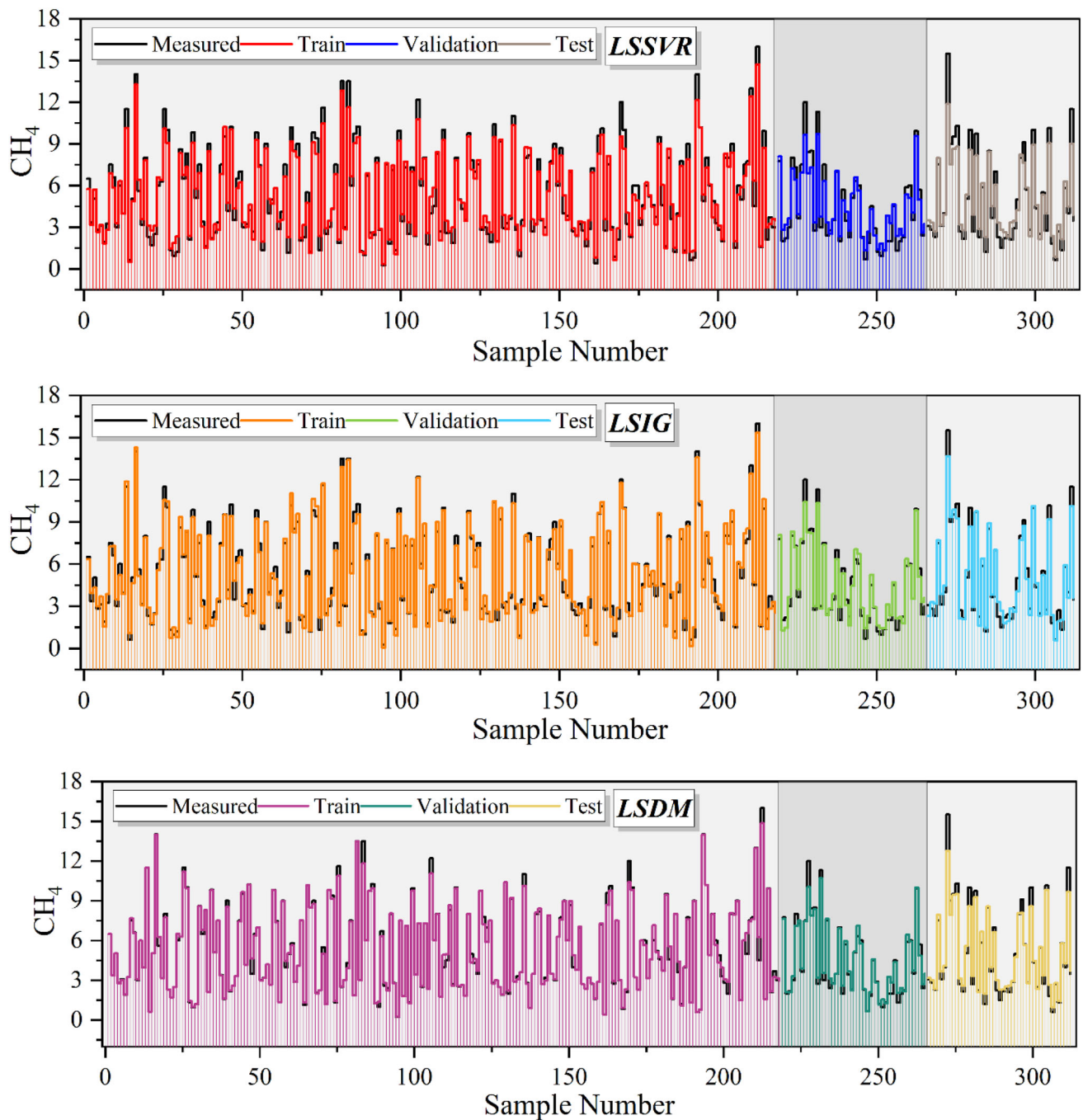


Fig. 3 Comparison of measured and predicted values

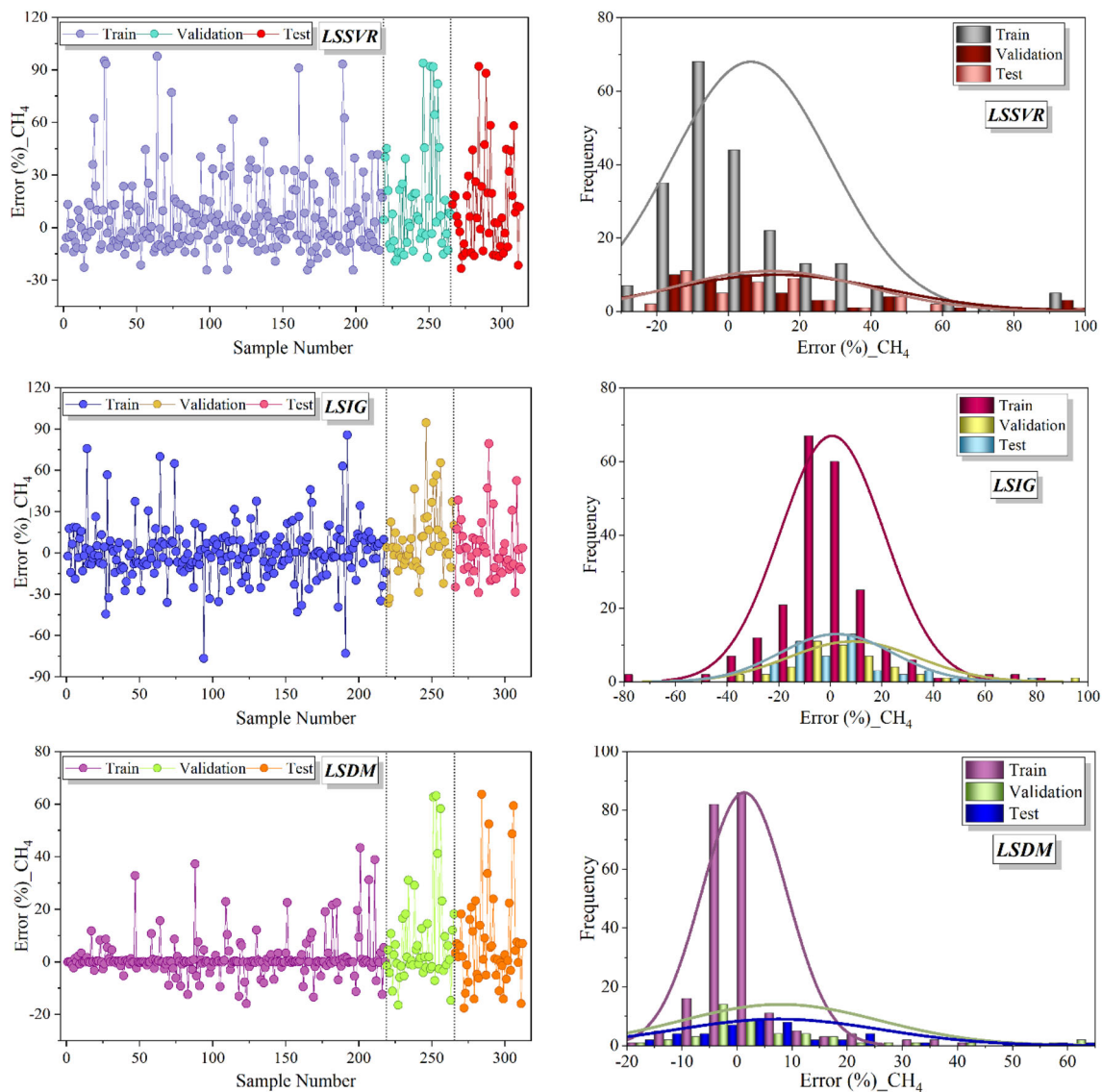
## 4 Discussion

### 4.1 Advantages and disadvantages of model and algorithms

#### 4.1.1 Advantages

The LSSVR is a versatile method that excels in both linear and nonlinear regression tasks. It harnesses the kernel

trick, enabling it to operate effectively in high-dimensional spaces while handling small datasets efficiently. One of its key strengths lies in its robustness against overfitting, thanks to its regularization parameter. Inspired by the social behavior of dwarf mongooses, the DMO algorithm offers a novel approach to optimization. It is adept at solving single-objective and multi-objective optimization problems with simplicity in implementation and understanding.



**Fig. 4** Error percentage of developed models by Line-symbol as well as Histogram Distribution plot for CH<sub>4</sub>

DMO effectively explores search spaces and avoids getting trapped in local optima, making it a valuable tool across various domains. The IGWO algorithm builds upon the foundation of the original Grey Wolf Optimization (GWO) by enhancing its performance. It exhibits faster convergence rates and employs a more efficient search mechanism by incorporating multiple wolf packs. IGWO proves effective for tackling both single-objective and multi-objective optimization problems, providing robust solutions to complex optimization challenges.

#### 4.1.2 Disadvantages

The LSSVR excels in handling both linear and nonlinear regression tasks but can be memory-intensive for large datasets due to the storage requirements for support vectors.

Additionally, it is computationally expensive during the training phase and demands careful selection of hyperparameters such as the regularization parameter and kernel function for optimal performance.

The DMO algorithm offers a unique perspective inspired by social behavior but may experience slower convergence rates compared to more established optimization algorithms. It is sensitive to parameter tuning, including factors like population size and mutation rate, which can affect its efficiency. DMO also lacks extensive theoretical groundwork and empirical validation compared to more widely studied algorithms in the optimization field.

The IGWO algorithm shares common traits with many metaheuristic algorithms. It requires meticulous parameter fine-tuning for achieving optimal performance across different problem domains. Efficiency may vary based on



**Table 4** Result of developed models for C<sub>2</sub>H<sub>n</sub>

Phase	Index values	Models		
		LSSVR	LSIG	LSDM
Train	RMSE	0.293	0.235	0.184
	R <sup>2</sup>	0.963	0.974	0.985
	MAE	0.235	0.186	0.046
	RSR	0.206	0.165	0.129
	MRAE	1.233	1.121	0.398
Validation	RMSE	0.347	0.272	0.284
	R <sup>2</sup>	0.950	0.960	0.977
	MAE	0.289	0.223	0.209
	RSR	0.276	0.216	0.226
	MRAE	0.486	0.421	0.274
Test	RMSE	0.380	0.327	0.337
	R <sup>2</sup>	0.953	0.962	0.973
	MAE	0.255	0.220	0.216
	RSR	0.277	0.239	0.246
	MRAE	1.427	1.302	0.858

**Table 5** Comparison between the published and presented investigation

Target	Index value	Articles		
		Present study (LSDM)	Ascher et al. (ANN) (Ascher et al. 2022b)	Ascher et al. (GBR) (Ascher et al. 2022)
CH <sub>4</sub>	RMSE	0.3670	0.1800	0.850
	R <sup>2</sup>	0.9880	0.7990	0.900
C <sub>2</sub> H <sub>n</sub>	RMSE	0.1840	0.1880	0.400
	R <sup>2</sup>	0.9850	0.5200	0.870

problem characteristics and parameter settings, highlighting the need for careful implementation. Despite its effectiveness, IGWO's theoretical analysis remains relatively limited compared to traditional optimization algorithms with more established theoretical frameworks.

#### 4.1.3 Specifications

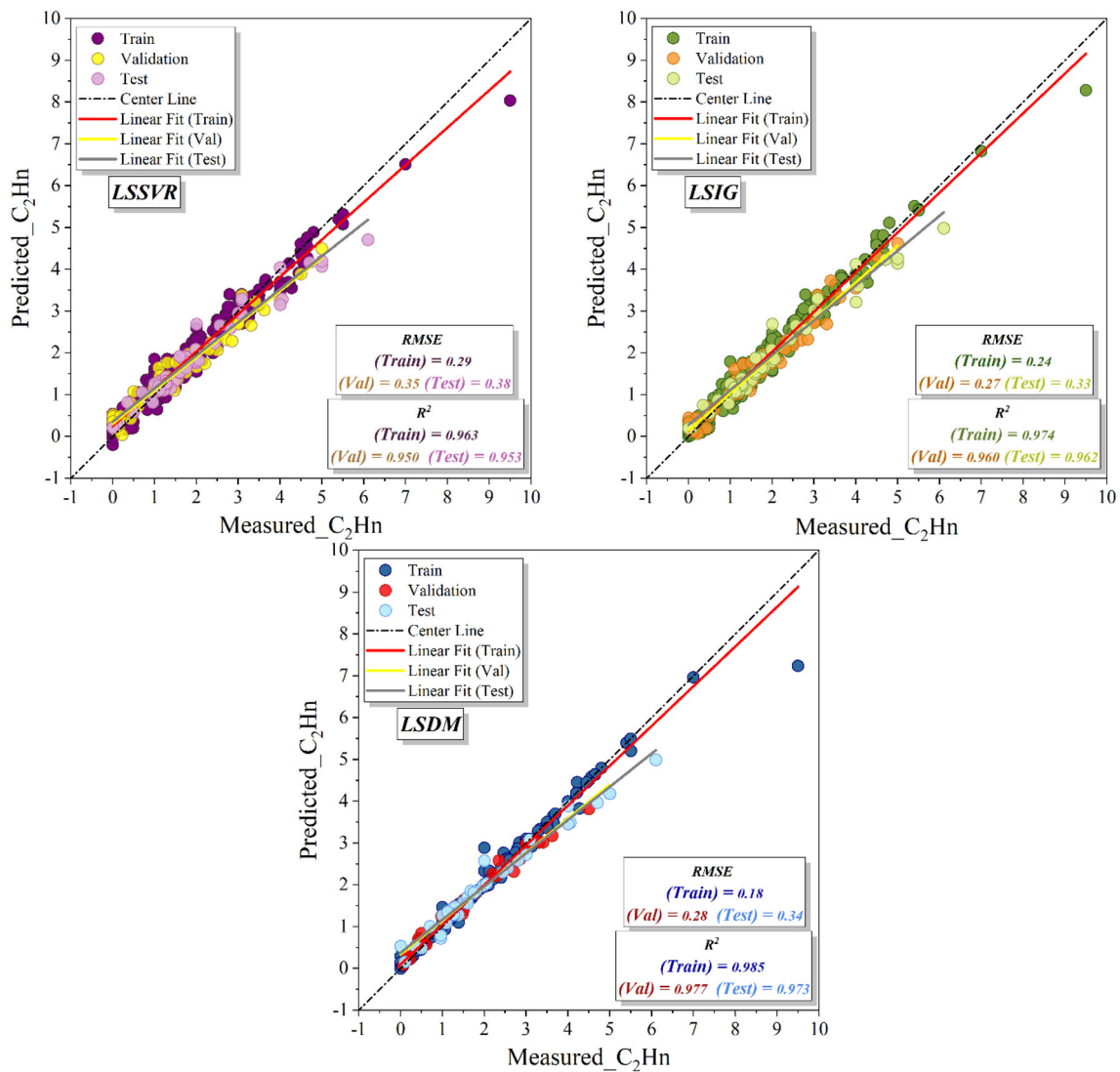
The LSSVR technique is rooted in support vector machines (SVM) principles, aiming to minimize the empirical risk while incorporating a regularization term to prevent overfitting. Its performance is commonly assessed utilizing metrics like coefficient of determination (R-squared) or mean squared error (MSE), highlighting its efficacy in regression tasks across various domains.

The DMO algorithm draws inspiration from the cooperative foraging behavior observed in dwarf mongooses. It leverages social interactions and leader–follower dynamics to guide the optimization process, striking a balance between exploration (seeking new solutions) and exploitation (refining existing solutions). This approach ensures robust

performance in both single-objective and multi-objective optimization scenarios, emphasizing simplicity and effectiveness in implementation. The IGWO algorithm enhances the original GWO by simulating the hierarchical and hunting behaviors of grey wolf packs more effectively. It represents solutions using vectors and utilizes a dynamic search mechanism to explore the solution space comprehensively. IGWO's convergence criteria typically involve a maximum number of iterations or a threshold for improvement, ensuring efficient convergence and optimal solutions for complex optimization problems.

#### 4.2 Comparison between the present study and existing publications

A few ranges of studies focused on the prediction of the gasification process such as CH<sub>4</sub> and C<sub>2</sub>H<sub>n</sub>. For example, Ascher et al. in two different studies investigate predicting these values via implementing machine learning models. The first study was conducted by Ascher et al. (Ascher et al. 2022b) based on the ANN model, achieved R<sup>2</sup> values of

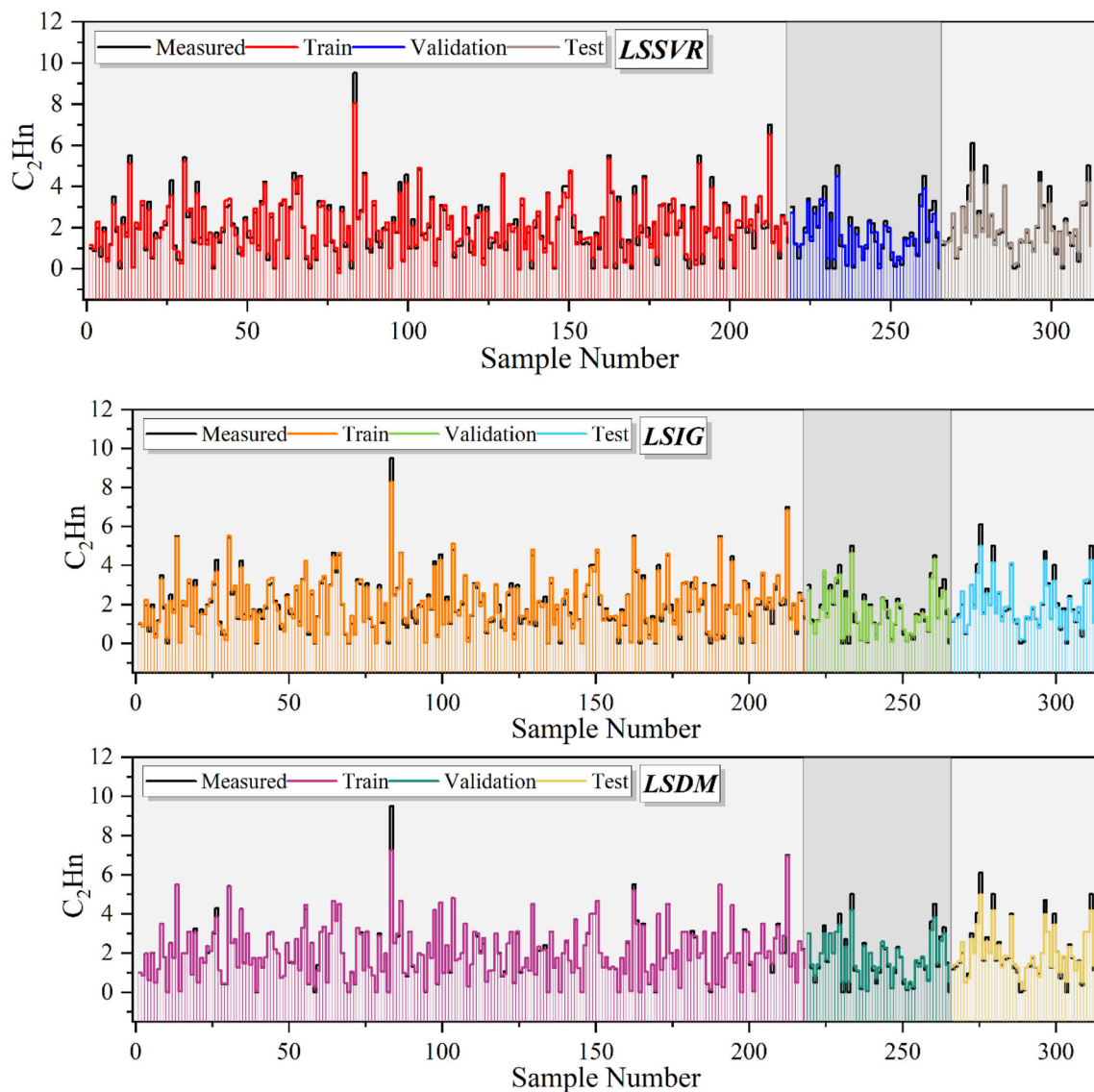


**Fig. 5** A scatter plot showing the correlation between the expected and measured  $C_2H_n$

0.52 and 0.799 for  $C_2H_n$  and  $CH_4$ , along with RMSE values of 0.118 and 0.18 for  $C_2H_n$  and  $CH_4$ . The second study by Ascher et al. (Ascher et al. 2024) based on the GBR model, Achieved  $R^2$  values of 0.9 and 0.87 for  $CH_4$  and  $C_2H_n$ , along with RMSE values of 0.4 and 0.85 for  $C_2H_n$  and  $CH_4$ . In this study, the LSSVR model was employed as the foundational framework, undergoing enhancement through the integration of DMO and IGWO algorithms. Upon evaluating the results obtained, it was found that the integration of DMO into the LSSVR model demonstrated exceptional applicability, yielding an  $R^2$  value of 0.988 and 0.985, as well as an RMSE of 0.184 and 0.367 for  $C_2H_n$  and  $CH_4$ , surpassing the performance of the other two models in this study.

### 4.3 Limitations of the study

While the study demonstrates promising results in predicting syngas production from biomass gasification, there are several limitations that should be acknowledged. Firstly, the models were evaluated based on specific input variables and performance criteria, which may not capture the full complexity of biomass gasification processes. Additionally, the study focuses primarily on  $CH_4$  and  $C_2H_n$  compositions, overlooking other important components of syngas. Future research could explore the prediction of a broader range of syngas components and consider the impact of varying process conditions and feedstock characteristics. Future works could investigate alternative optimization methods or hybrid approaches to further improve predictive performance. Additionally, experimental validation of the models using real-world biomass gasification data would strengthen



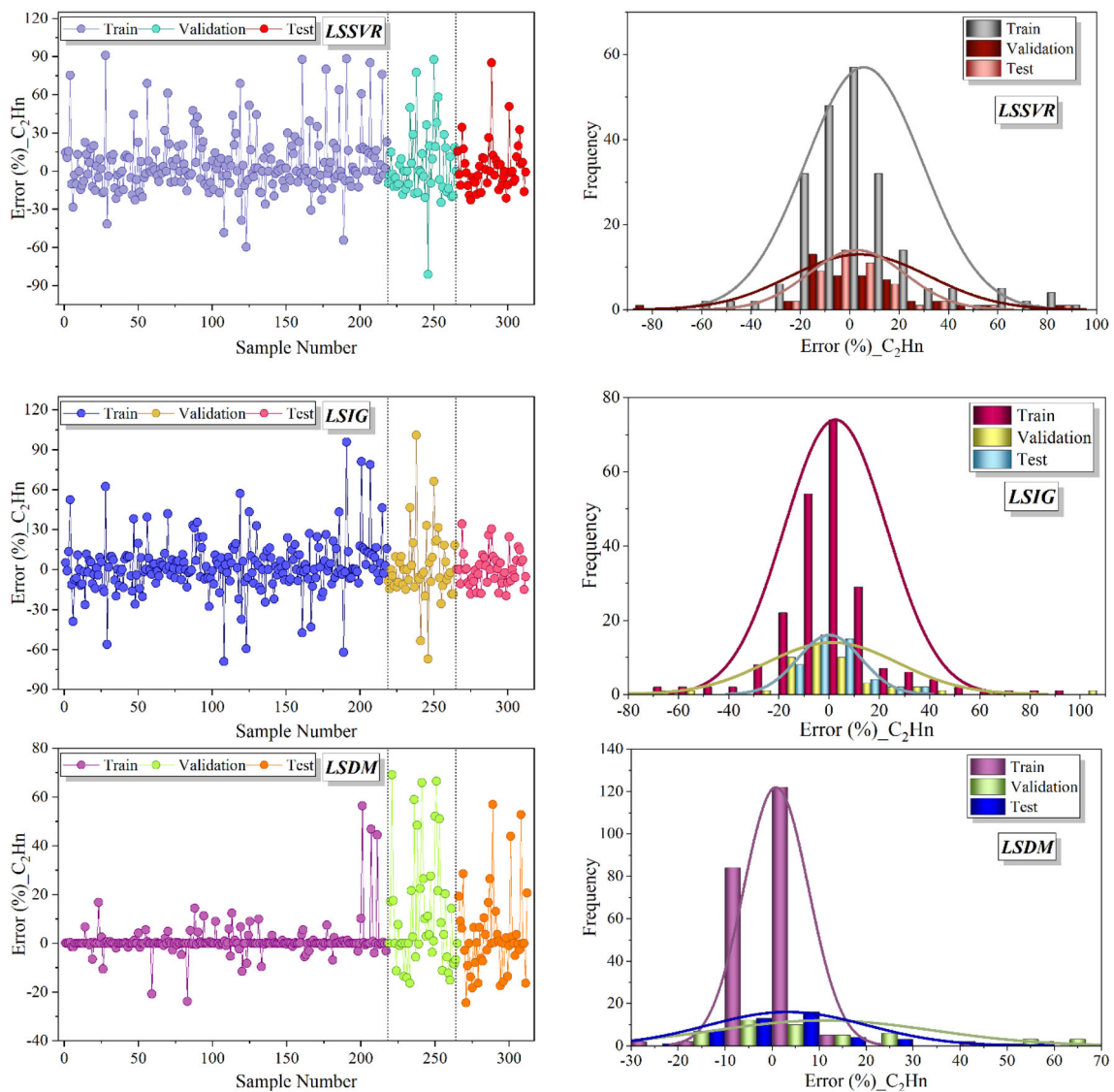
**Fig. 6** Comparison of measured and predicted values

their applicability and reliability. Overall, addressing these limitations and conducting further research could provide a more comprehensive understanding of biomass gasification processes and enhance the development of predictive models for sustainable energy production.

## 5 Summary and conclusion

Mounting global energy demand and environmental concerns related to fossil fuels, including natural gas, coal, and oil, are driving the widespread adoption of renewable energy sources like solar, wind, hydropower, and biomass-biofuels to address carbon emissions and establish sustainable energy systems. Biomass, derived from agricultural leftovers and

forests, offers a promising and widely available resource for the production of materials and bioenergy, supporting waste management and renewable energy sustainability, with the aid of advanced energy storage technologies to enhance energy security. Gasification is a crucial process in biomass utilization, offering flexibility in converting various feedstock materials into syngas with significant heating potential for power generation and biofuel production, thereby expanding biomass usage as a source of energy. This article presents the LSSVR single model and two hybrid models, LSIG (LSSVR optimized with IGWO) and LSDM (LSSVR optimized with DMO), which employ input variables to predict elemental compositions ( $CH_4$  and  $C_2H_n$ ). These models were assessed using established performance criteria. The



**Fig. 7** The Histogram Distribution plot for  $C_2H_n$  and the Line-symbol developed model's error % are presented

results demonstrate that both introduced optimizers, particularly LSDM, significantly enhance the accuracy of  $CH_4$  and  $C_2H_n$  predictions when combined with LSSVR. Notably, LSDM achieved outstanding performance with the highest  $R^2$  values of 0.988 for  $CH_4$  and 0.985 for  $C_2H_n$ , indicating its exceptional accuracy. Furthermore, LSDM's minimum RMSE values of 0.367 for  $CH_4$  and 0.184 for  $C_2H_n$  underscore its precision. Additionally, a comparative analysis revealed that LSDM was more accurate in predicting  $CH_4$  and  $C_2H_n$ . In summary, this research highlights the potential of optimization methods for forecasting syngas production and provides valuable insights into biomass gasification.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s41939-024-00552-x>.

**Author contributions** The author contributed to the study's conception and design. Data collection, simulation and analysis were performed by "Wei Cong".

**Funding** This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

**Data availability** No datasets were generated or analysed during the current study.

## Declarations

**Conflict of interest** The authors declare no competing interests.

## Appendix A

The code of LSSVR-DMO and LSSVR-IGWO:

## 1. LSSVR:

```

def cost_function(pop, X_tr, X_te, Y_tr, Y_te):
    net = LSSVR(C = pop[0], kernel = 'rbf', gamma = pop[1]/100)
    net.fit(X_tr, Y_tr)
    preds_tr = net.predict(X_tr)
    preds_te = net.predict(X_te)
    y = np.hstack([Y_tr, Y_te])
    preds = np.hstack([preds_tr, preds_te])
    MSE = met.mean_squared_error(y, preds)
    RMSE = np.sqrt(MSE)
    POP = structure(cost = RMSE, pre = preds, Net = net)
    return POP.cost, POP.pre, POP.Net
def roulettewheelselection(P):
    r = np.random.rand()
    C = np.cumsum(P)
    for j in range(len(C)):
        if r <= C[j]:
            i = j
            break
    return i

```

## 2. Dwarf Mongoose Optimization Algorithm:

```

X = np.loadtxt('input.txt')
Y = np.loadtxt('target.txt')
size_Y = len(Y)
size_tr = round(size_Y * 0.7)
size_te = size_Y - size_tr
X_tr = X[:size_tr]
X_te = X[size_tr:]
Y_tr = Y[:size_tr]
Y_te = Y[size_tr:]

Y_tr = np.array(Y_tr).reshape(size_tr)
Y_te = np.array(Y_te).reshape(size_te)

nVar =          # % Number of Decision Variables
VarMin =        # % Decision Variables Lower Bound
VarMax =        # % Decision Variables Upper Bound
# %% ABC Settings
MaxIt =         # % Maximum Number of Iterations
nPop =          # % Population Size (Family Size)
nBabysitter = 5 # % Number of babysitters
nAlphaGroup = nPop - nBabysitter # % Number of Alpha group
nScout = nAlphaGroup # % Number of Scouts
L = round(0.6 * nVar * nBabysitter) # % Babysitter Exchange Parameter
peep = 2        # % Alpha female 嗷 vocalization

# % Empty Mongoose Structure
empty_mongoose = structure(cost = None, pre = None, Net = None, Position
                          = None)

```

```

# % Initialize Population Array
pop = empty_mongoose * nAlphaGroup
BestSol = empty_mongoose
newpop = empty_mongoose
# % Initialize Best Solution Ever Found
BestSol.Cost = np.inf
tau = np.inf
Iter = 0
sm = np.ones(nAlphaGroup) * np.inf
# % Create Initial Population
for i in range(nAlphaGroup):
    pop[i].Position = np.random.uniform(VarMin, VarMax, nVar)
    # pop(i).Cost = F_obj(pop(i).Position);
    pop[i].Cost, pop[i].pre, pop[i].Net
        = cost_function(pop[i].Position, X_tr, X_te, Y_tr, Y_te)
    if pop[i].Cost <= BestSol.Cost:
        BestSol = pop[i]
# % Abandonment Counter
C = np.zeros(nAlphaGroup)
CF = (1 - Iter/MaxIt) ** (2 * Iter/MaxIt)
# % Array to Hold Best Cost Values
BestCost = np.zeros(MaxIt)
# %% DMOA Main Loop
for it in range(MaxIt):
    # % Alpha group
    F = np.zeros(nAlphaGroup)
    cost = []
    for i in range(nAlphaGroup):
        cost.append(pop[i].Cost)
        MeanCost = np.mean(cost);
    for i in range(nAlphaGroup):
        # % Calculate Fitness Values and Selection of Alpha
        F[i] = np.exp(-pop[i].Cost/MeanCost) # % Convert Cost to Fitness
        P = F/sum(F)
        # % Foraging led by Alpha female
    for m in range(nAlphaGroup):
        # % Select Alpha female
        i = roulettewheelselection(P);
        # % Choose k randomly, not equal to Alpha
        K = []
        for j in range(nAlphaGroup):
            if j < i:
                K.append(j)
            elif j > i:
                K.append(j)
            k = K[np.random.randint(0, len(K))]

```

```

    # % Define Vocalization Coeff.
    phi = (peep/2) * np.random.uniform(-1, +1, nVar)
    # % New Mongoose Position
    newpop.Position = pop[i].Position + phi * (pop[i].Position - pop[k].Position)
    for j in range(nVar):
        newpop.Position[j] = max(newpop.Position[j], VarMin)
        newpop.Position[j] = min(newpop.Position[j], VarMax)
        newpop.Position[j] = int(newpop.Position[j])
    # % Evaluation
    # newpop.Cost = F_obj(newpop.Position);
    newpop.Cost, newpop.pre, newpop.Net
        = cost_function(newpop.Position, X_tr, X_te, Y_tr, Y_te)
    # % Comparision
    if newpop.Cost <= pop[i].Cost:
        pop[i] = newpop
    else:
        C[i] = C[i] + 1
    # % Scout group
    for i in range(nScout):
        # % Choose k randomly, not equal to i
        for j in range(nAlphaGroup):
            if j < i:
                K.append(j)
            elif j > i:
                K.append(j)
        k = K[np.random.randint(0, len(K))]

    # % Define Vocalization Coeff.
    phi = (peep/2) * np.random.uniform(-1, +1, nVar);
    # % New Mongoose Position
    newpop.Position = pop[i].Position + phi * (pop[i].Position - pop[k].Position);
    for j in range(nVar):
        newpop.Position[j] = max(newpop.Position[j], VarMin)
        newpop.Position[j] = min(newpop.Position[j], VarMax)
        newpop.Position[j] = int(newpop.Position[j])
    # % Evaluation
    # newpop.Cost = F_obj(newpop.Position);
    newpop.Cost, newpop.pre, newpop.Net
        = cost_function(newpop.Position, X_tr, X_te, Y_tr, Y_te)
    # % Sleeping mould
    sm[i] = (newpop.Cost - pop[i].Cost)/max(newpop.Cost, pop[i].Cost);
    # % Comparision
    if newpop.Cost <= pop[i].Cost:
        pop[i] = newpop
    else:
        C[i] = C[i] + 1

```

```

# % Babysitters
for i in range(nBabysitter):
    if C[i] >= L:
        pop[i].Position = np.random.uniform(VarMin, VarMax, nVar)
        # pop(i).Cost = F_obj(pop(i).Position);
        pop[i].Cost, pop[i].pre, pop[i].Net
            = cost_function(pop[i].Position, X_tr, X_te, Y_tr, Y_te)
        C[i] = 0
# % Update Best Solution Ever Found
for i in range(nAlphaGroup):
    if pop[i].Cost <= BestSol.Cost:
        BestSol = pop[i]
# % Next Mongoose Position
newtau = np.mean(sm)
for i in range(nScout):
    M = (pop[i].Position * sm[i])/pop[i].Position
    if newtau > tau:
        newpop.Position
            = pop[i].Position - CF * phi * np.random.rand() * (pop[i].Position
            - M)
    else:
        newpop.Position
            = pop[i].Position + CF * phi * np.random.rand() * (pop[i].Position
            - M);
        tau = newtau
# % Update Best Solution Ever Found
for i in range(nAlphaGroup):
    if pop[i].Cost <= BestSol.Cost:
        BestSol = pop[i]

```

### 3. Improved Gray Wolf Optimization:

```

def IGWO(dim, N, Max_iter, lb, ub, fobj):
    lu = np.array([lb * np.ones(dim), ub * np.ones(dim)])
    Alpha_pos = np.zeros(dim)
    Alpha_score = np.inf # change this to -inf for maximization problems
    Beta_pos = np.zeros(dim)
    Beta_score = np.inf # change this to -inf for maximization problems
    Delta_pos = np.zeros(dim)
    Delta_score = np.inf # change this to -inf for maximization problems

```



```
Positions = initialization(N, dim, ub, lb)
Positions = boundConstraint(Positions, Positions, lu)
Fit = np.apply_along_axis(fobj, 1, Positions)
pBestScore = Fit.copy()
pBest = Positions.copy()
neighbor = np.zeros((N, N))
Convergence_curve = np.zeros(Max_iter)
iter = 0
while iter < Max_iter:
    for i in range(len(Positions)):
        fitness = Fit[i]
        if fitness < Alpha_score:
            Alpha_score = fitness
            Alpha_pos = Positions[i].copy()
        if fitness > Alpha_score and fitness < Beta_score:
            Beta_score = fitness
            Beta_pos = Positions[i].copy()
        if fitness > Alpha_score and fitness > Beta_score and fitness
            < Delta_score:
            Delta_score = fitness
            Delta_pos = Positions[i].copy()
        a = 2 - iter * (2 / Max_iter)
        X_GWO = np.zeros_like(Positions)
    for i in range(len(Positions)):
        for j in range(dim):
            r1 = np.random.rand()
            r2 = np.random.rand()
            A1 = 2 * a * r1 - a
            C1 = 2 * r2
            D_alpha = abs(C1 * Alpha_pos[j] - Positions[i, j])
```

---

```

X1 = Alpha_pos[j] - A1 * D_alpha
r1 = np.random.rand()
r2 = np.random.rand()
A2 = 2 * a * r1 - a
C2 = 2 * r2
D_beta = abs(C2 * Beta_pos[j] - Positions[i,j])
X2 = Beta_pos[j] - A2 * D_beta
r1 = np.random.rand()
r2 = np.random.rand()
A3 = 2 * a * r1 - a
C3 = 2 * r2
D_delta = abs(C3 * Delta_pos[j] - Positions[i,j])
X3 = Delta_pos[j] - A3 * D_delta
X_GWO[i,j] = (X1 + X2 + X3) / 3
X_GWO = boundConstraint(X_GWO, Positions, lu)
Fit_GWO = np.apply_along_axis(fobj, 1, X_GWO)
radius = np.linalg.norm(Positions - X_GWO, axis = 1)
dist_Position = np.linalg.norm(Positions[:, np.newaxis] - Positions, axis = 2)
neighbor = dist_Position <= radius[:, np.newaxis]
X_DLH = np.zeros_like(Positions)
for t in range(len(Positions)):
    idx = np.where(neighbor[t])[0]
    random_idx_neighbor = np.random.choice(idx, size = dim)
    for d in range(dim):
        X_DLH[t, d]
            = Positions[t, d] + np.random.rand()
            * (Positions[random_idx_neighbor[d], d] - Positions[t, d])
X_DLH = boundConstraint(X_DLH, Positions, lu)
Fit_DLH = np.apply_along_axis(fobj, 1, X_DLH)
tmp = Fit_GWO < Fit_DLH

```

---

```

tmpFit = tmp * Fit_GWO + (1 - tmp) * Fit_DLH
tmpPositions
    = tmp[:, np.newaxis] * X_GWO + (1 - tmp[:, np.newaxis]) * X_DLH
tmp = pBestScore <= tmpFit
pBestScore = tmp * pBestScore + (1 - tmp) * tmpFit
pBest = tmp[:, np.newaxis] * pBest + (1 - tmp[:, np.newaxis])
    * tmpPositions

Fit = pBestScore
Positions = pBest

Convergence_curve[iter] = Alpha_score
iter += 1
return Alpha_score, Alpha_pos, Convergence_curve

```

## References

- Agushaka JO, Ezugwu AE, Abualigah L (2022) Dwarf mongoose optimization algorithm. *Comput Methods Appl Mech Eng* 391:114570
- Ahrenfeldt J, Thomsen TP, Henriksen U, Clausen LR (2013) Biomass gasification cogeneration—A review of state of the art technology and near future perspectives. *Appl Therm Eng* 50:1407–1417
- Aiyer BG, Kim D, Karingattikkal N, Samui P, Rao PR (2014) Prediction of compressive strength of self-compacting concrete using least square support vector machine and relevance vector machine. *KSCE J Civ Eng* 18:1753–1758
- Antonopoulos I-S, Karagiannidis A, Gkouletsos A, Perkoulidis G (2012) Modelling of a downdraft gasifier fed by agricultural residues. *Waste Manag* 32:710–718
- Ascher S, Sloan W, Watson I, You S (2022b) A comprehensive artificial neural network model for gasification process prediction. *Appl Energy* 320:119289
- Ascher S, Wang X, Watson I, Sloan W, You S (2022) Interpretable machine learning to model biomass and waste gasification. *Bioresour Technol* 364:128062. <https://doi.org/10.1016/j.biortech.2022.128062>
- Ascher S, Watson I, You S (2022a) Machine learning methods for modelling the gasification and pyrolysis of biomass and waste. *Renew Sustain Energy Rev* 155:111902
- Awad M, Khanna R. Support Vector Regression BT - Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers. In: Awad M, Khanna R, editors., Berkeley, CA: Apress; 2015, p. 67–80. [https://doi.org/10.1007/978-1-4302-5990-9\\_4](https://doi.org/10.1007/978-1-4302-5990-9_4).
- Ayubi Rad M, Ayubirad MS (2017) Comparison of artificial neural network and coupled simulated annealing based least square support vector regression models for prediction of compressive strength of high-performance concrete. *Sci Iran* 24:487–496
- Bahadar A, Kanthasamy R, Sait HH, Zwawi M, Algarni M, Ayodele BV et al (2022) Elucidating the effect of process parameters on the production of hydrogen-rich syngas by biomass and coal Co-gasification techniques: a multi-criteria modeling approach. *Chemosphere* 287:132052
- Baruah D, Baruah DC, Hazarika MK (2017) Artificial neural network based modeling of biomass gasification in fixed bed downdraft gasifiers. *Biomass Bioenerg* 98:264–271
- Basu P. Biomass gasification and pyrolysis: practical design and theory. Academic press; 2010.
- Ceylan Z, Ceylan S (2021) Application of machine learning algorithms to predict the performance of coal gasification process. Elsevier, *Appl. Artif. Intell. Process Syst. Eng.*, pp 165–186
- Chen W-H, Farooq W, Shahbaz M, Naqvi SR, Ali I, Al-Ansari T et al (2021) Current status of biohydrogen production from lignocellulosic biomass, technical challenges and commercial potential through pyrolysis process. *Energy* 226:120433
- Ciferno JP, Marano JJ. Benchmarking biomass gasification technologies for fuels, chemicals and hydrogen production. US Dep Energy Natl Energy Technol Lab 2002.
- Dodo UA, Ashigwuike EC, Emechebe JN, Abba SI (2022) Prediction of energy content of biomass based on hybrid machine learning ensemble algorithm. *Energy Nexus* 8:100157
- Eberhart RC, Shi Y. Evolving artificial neural networks. *Proc. Int. Conf. neural networks brain*, vol. 1, PRC; 1998, p. PL5--PLI3.
- Emary E, Zawbaa HM, Grosan C (2017) Experienced gray wolf optimization through reinforcement learning and neural networks. *IEEE Trans Neural Networks Learn Syst* 29:681–694
- Ewees AA, Vo Thanh H, Al-qaness MAA, Abd Elaziz M, Samak AH. Smart predictive viscosity mixing of CO<sub>2</sub>-N<sub>2</sub> using optimized dendritic neural networks to implicate for carbon capture utilization and storage. *J Environ Chem Eng* 2024;12:112210. <https://doi.org/10.1016/j.jece.2024.112210>.
- George J, Arun P, Muralleedharan C (2018) Assessment of producer gas composition in air gasification of biomass using artificial neural network model. *Int J Hydrogen Energy* 43:9558–9568

- Goh ATC (1995) Back-propagation neural networks for modeling complex systems. *Artif Intell Eng* 9:143–151
- Hakeem KR, Jawaid M, Rashid U. *Biomass and bioenergy*. Springer; 2016.
- Han P, Li DZ, Wang Z (2008) A study on the biomass gasification process model based on least squares SVM. *Energy Conserv Technol* 1:3–7
- Hassan MA, Bailek N, Bouchouicha K, Ibrahim A, Jamil B, Kuriqi A et al (2022) Evaluation of energy extraction of PV systems affected by environmental factors under real outdoor conditions. *Theor Appl Climatol* 150:715–729
- Hsu C-W, Chang C-C, Lin C-J. *A practical guide to support vector classification* 2003.
- Jain AK, Mao J, Mohiuddin KM (1996) Artificial neural networks: a tutorial. *Computer (long Beach Calif)* 29:31–44
- MacNulty DR, Mech LD, Smith DW (2007) A proposed ethogram of large-carnivore predatory behavior, exemplified by the wolf. *J Mammal* 88:595–605
- Malka L, Daci A, Kuriqi A, Bartocci P, Rrapaj E (2022) Energy storage benefits assessment using multiple-choice criteria: the case of Drini River Cascade. *Albania Energies* 15:4032
- McKendry P (2002) Energy production from biomass (part 1): overview of biomass. *Bioresour Technol* 83:37–46
- Molino A, Chianese S, Musmarra D (2016) Biomass gasification technology: The state of the art overview. *J Energy Chem* 25:10–25
- Narnaware SL, Panwar NL (2022) Biomass gasification for climate change mitigation and policy framework in India: a review. *Bioresour Technol Reports* 17:100892
- Okkan U, Serbes ZA (2012) Rainfall–runoff modeling using least squares support vector machines. *Environmetrics* 23:549–564
- Pham QB, Yang T-C, Kuo C-M, Tseng H-W, Yu P-S (2019) Combining random forest and least square support vector regression for improving extreme rainfall downscaling. *Water* 11:451
- Qian K, Kumar A, Patil K, Bellmer D, Wang D, Yuan W et al (2013) Effects of biomass feedstocks and gasification conditions on the physiochemical properties of char. *Energies* 6:3972–3986
- Raud M, Kikas T, Sippula O, Shurpali NJ (2019) Potentials and challenges in lignocellulosic biofuel production technology. *Renew Sustain Energy Rev* 111:44–56
- Rodríguez-Galiano V, Sanchez-Castillo M, Chica-Olmo M, Chica-Rivas M (2015) Machine learning predictive models for mineral prospectivity: an evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geol Rev* 71:804–818
- Sadaghat B, Ebrahimi SA, Souri O, Yahyavi Niar M, Akbarzadeh MR (2024) Evaluating strength properties of Eco-friendly Seashell-Containing Concrete: Comparative analysis of hybrid and ensemble boosting methods based on environmental effects of seashell usage. *Eng Appl Artif Intell* 133:108388. <https://doi.org/10.1016/j.engappai.2024.108388>
- Serrano García D, Castelló D. Tar prediction in bubbling fluidized bed gasification through artificial neural networks 2020.
- Shenbagaraj S, Sharma PK, Sharma AK, Raghav G, Kota KB, Ashokkumar V (2021) Gasification of food waste in supercritical water: an innovative synthesis gas composition prediction model based on Artificial Neural Networks. *Int J Hydrogen Energy* 46:12739–12757
- Tang Q, Chen Y, Yang H, Liu M, Xiao H, Wang S et al (2021) Machine learning prediction of pyrolytic gas yield and compositions with feature reduction methods: Effects of pyrolysis conditions and biomass characteristics. *Bioresour Technol* 339:125581
- Vapnik VN. *The nature of statistical learning theory*. Theory 1995.
- Velvizhi G, Balakumar K, Shetti NP, Ahmad E, Pant KK, Aminabhavi TM (2022) Integrated biorefinery processes for conversion of lignocellulosic biomass to value added materials: Paving a path towards circular economy. *Bioresour Technol* 343:126151
- Vo Thanh H, Dai Z, Du Z, Yin H, Yan B, Soltanian MR et al (2024) Artificial intelligence-based prediction of hydrogen adsorption in various kerogen types: Implications for underground hydrogen storage and cleaner production. *Int J Hydrogen Energy* 57:1000–1009. <https://doi.org/10.1016/j.ijhydene.2024.01.115>
- Vo Thanh H, Sugai Y, Sasaki K (2020) Application of artificial neural network for predicting the performance of CO<sub>2</sub> enhanced oil recovery and storage in residual oil zones. *Sci Rep* 10:18204. <https://doi.org/10.1038/s41598-020-73931-2>
- Vo Thanh H, Zhang H, Dai Z, Zhang T, Tangparitkul S, Min B (2024) Data-driven machine learning models for the prediction of hydrogen solubility in aqueous systems of varying salinity: Implications for underground hydrogen storage. *Int J Hydrogen Energy* 55:1422–1433. <https://doi.org/10.1016/j.ijhydene.2023.12.131>
- Wang S, Wen Y, Shi Z, Zaini IN, Jönsson PG, Yang W (2022) Novel carbon-negative methane production via integrating anaerobic digestion and pyrolysis of organic fraction of municipal solid waste. *Energy Convers Manag* 252:115042
- Wang K, Zhang J, Shang C, Huang D (2021) Operation optimization of Shell coal gasification process based on convolutional neural network models. *Appl Energy* 292:116847
- Wang L. *Support vector machines: theory and applications*. vol. 177. Springer Science & Business Media; 2005.
- Wu Y, Yang W, Blasiak W (2014) Energy and exergy analysis of high temperature agent gasification of biomass. *Energies* 7:2107–2122
- Xie G, Wang S, Zhao Y, Lai KK (2013) Hybrid approaches based on LSSVR model for container throughput forecasting: a comparative study. *Appl Soft Comput* 13:2232–2241
- Yao X (1999) *Evolving Artificial Neural Networks*. Proc IEEE 87:1423–1447
- Zhang F, O'Donnell LJ. Chapter 7 - Support vector regression. In: Mechelli A, Vieira SBT-ML, editors., Academic Press; 2020, p. 123–40. <https://doi.org/10.1016/B978-0-12-815739-8.00007-9>.
- Zhang H, Wang P, Rahimi M, Vo Thanh H, Wang Y, Dai Z et al (2024) Catalyzing net-zero carbon strategies: Enhancing CO<sub>2</sub> flux Prediction from underground coal fires using optimized machine learning models. *J Clean Prod* 441:141043. <https://doi.org/10.1016/j.jclepro.2024.141043>

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.