



ADVANCED STRATEGY FOR PREDICTIVE MODELING AND EMPIRICAL ANALYSIS OF PIPELINE EROSION RATES

Xue Zi*, Yujie Song

College of Petroleum and Chemical Engineering, Dongying Vocational College, Shandong, China

ABSTRACT

Erosion frequently occurs in pipelines due to interactions between solid particles, the transported fluid, and the surrounding pipeline materials. This process can cause material deterioration within pipelines, potentially leading to pipeline failure and adverse impacts on public health, property, and the environment. Therefore, accurate prediction of pipe erosion rates is essential for ensuring safer, more cost-effective, and sustainable pipeline operations. This research presents a detailed investigation of pipe erosion, a common problem that presents significant safety and environmental hazards. The objective of this research is to develop a reliable erosion-rate prediction model to provide sustainable pipeline performance. The research uses a robust methodological framework in the form of both standalone and hybrid predictive models consisting of SVR and AdaBoost regression techniques. Data was collected systematically with respect to relevant parameters influencing erosion behavior; these parameters were also confirmed through rigorous validation testing. Through optimization algorithms such as AEO, GWO, and MFO, the model parameters were optimally determined in order to improve the predictive capability of the models. Detailed empirical analyses using advanced statistical tools illustrated the complexity of the relationships between input variables and their impact on the erosion rate. Comprehensive evaluation metrics underscored the supremacy of the SVR models, particularly SVR-MFO, in predicting erosion rates with unprecedented accuracy. Overall, the results presented in this paper provide pipeline engineers and stakeholders with valuable information regarding the optimization of models for predicting erosion to provide accurate, cost-effective, and safer operations of pipelines.

Keywords: erosion; machine learning (ML); support vector regression (SVR); AdaBoost; moth flame optimization (MFO); GWO; AEO.

Date submitted: 06.10.2025

Date accepted: 08.06.2026

© 2026 «OilGasScientificResearchProject» Institute. All rights reserved.

1. Introduction

The main piece of infrastructure for the transportation of gas and oil is pipelines[1, 2]. These pipelines frequently transport combustible and explosive substances under elevated temperatures and pressures. The implications of a leak can be catastrophic due to the potential for ignition, injuries and fatalities; irreversible damage to property; and severe environmental pollution. Ensuring the safety and ongoing functionality of these pipelines is crucial for a consistent and safe oil and gas distribution. Erosion, caused by the impact of solid particles on the walls of pipelines by high-velocity fluids, is one of the many ways damage can occur in pipelines. In oil and gas production, both fluids remain at high velocities and pressures and contain solid particles, creating a complex multiphase flow within pipelines known as gathering pipelines [3-5].

As fluids flow through gathering pipelines, the pipe wall thickness will gradually reduce, potentially leading to the formation of holes in the pipeline. Erosion-related degrada-

tion of the pipeline are further complicated where the flow direction changes (e.g., an elbow) [6, 7].

Fractures of pipelines, caused by erosion, will create leaks due to the high-pressure fluids that have ruptured the pipeline, thereby significantly threatening environmental and operational safety. To effectively design and maintain pipelines, accurate predictions of erosion under service conditions are essential. Maintaining erosion within manageable limits, can help prevent the most severe outcomes [8, 9].

There are numerous parameters that affect the severity of the erosion (e.g., flow velocity, impact angle, particle properties, and target material properties). Flow velocity, in particular, has long been considered the most important factor in determining erosion, due to its influence on the erosion behavior [10].

Generally, the relationship between erosion rate and flow velocity follows a power law (with flow velocity exponents ranging from 1.7 to 2.6). In addition, the impact angle and its effect on erosion behaviour depend on whether the material is brittle or ductile. At a 90° angle, erosion wear is most noticeable, and for delicate materials, erosion wear rises with an increase in impact angle. On the other hand, for ductile

*E-mail: zxdyzyxy@126.com

<http://dx.doi.org/10.5510/OGP20260301234>

materials, erosion rates increase initially and subsequently decrease with increasing impact angle, with peak erosion often occurring at angles between 20° and 40° [11, 12].

Several researchers have investigated the influence of particle characteristics affect erosion. Important properties include the amount of particles present, their size, shape, and strength. Laboratory tests confirm the results of field tests: as the amount of particles increases, their size, angularity and the strength increase, erosion intensity increases [13].

When taking into account the characteristics of the target, the material's ability to withstand erosion is mostly dependent on its mechanical characteristics. Numerous studies on erosion have focused on a variety of materials, like carbon steel, ceramics, brass, aluminum alloys, as well as composite materials [11, 14].

Relative to both target and particle properties, flow conditions exert a more pronounced effect on erosion wear. In their orthogonal erosion tests, Yoganandh et al. (2013) studied the effects of flow rate, angle of impact on a surface, concentration of particles, and size of particles. They concluded that the angle of impact and flow rate accounted for 21 and 60 % of the total erosion, thus, the greatest contribution to erosion is from these two variables. In addition, as flow increases, the energy transferred to the surface from particle impact increases as well, thus increasing the amount of erosion that occurs on the target surface as well [15].

To address the erosion challenges in pipelines, erosion models have been developed by pipeline engineers and scholars. Due to the complex nature of erosion, which results from interactions among fluid properties, particle characteristics, pipeline materials, design, flow patterns, and operating conditions, most contemporary models rely on empirical, mechanistic, and CFD approaches [16].

While experiments deliver the most dependable and realistic data, they are associated with significant labor and time expenses. Typical laboratory approaches used to conduct tests for erosion only control one or more of the variables in a laboratory environment that can create some of the same basic conditions that would exist in an actual project site. Therefore, laboratory tests cannot reproduce the very complex conditions that exist in the engineering domain. Laboratory results may also have limited validity and may not be relevant for engineering applications. With advances in computer technology, numerical simulation has become a common and cost-effective approach for predicting erosion [17].

Lu & Agrawal (2014) proposed an Eulerian-Granular approach according to CFD for assessing erosion wear in multiphase systems, demonstrating its superior accuracy compared to conventional Eulerian-Lagrangian and empirical approaches. This approach provides a full representation of the complex flow dynamics and sediment interaction occurring, which enhances the overall accuracy of erosion prediction. Thus, this is a foundational aspect of thorough multiphased erosion study efforts [18]. Yu et al. (2019) investigated erosion in multiphase pipelines utilized in the gas as well as oil sector, concentrating on combinations of solids and liquids or gas and solids. The study employed a CFD-based multi-fluid approach to examine wall erosion in a 90° elbow, considering particle turbulent kinetic energy for accurate erosion prediction. Experimental validation was achieved while exploring the relationship between erosion rate, as well as solid mass flow rate, and wall roughness. The research nota-

bly identified a crucial solid mass flow rate that causes noticeable particle impact damage [19]. Bilal et al. (2021) examined pipeline erosion in elbows using alternative geometries and CFD simulations. Their findings indicated that increasing elbow curvature reduces erosion rates significantly. Half of the erosion in a 90° elbow is observed in a 45° curve [20].

Due to the constantly changing nature of these types of flows and the resource intensity of CFD methods, accurate predictions for CFD erosion rates at multiphased interfaces are not always possible. Previous works often used static laboratory measurements and CFD results; therefore, these efforts did not have real-world relevance. However, recent advancements have seen ML and intelligent algorithms offering precise predictions of pipeline erosion. These techniques ensure safety, and as the oil and gas industry evolves, digital twins and pipeline intelligence will be integral in all stages. The vast amount of available data, improved algorithms, and computing capabilities, coupled with the rapid speed of ML, can offer timely, low-cost solutions. Over the past twenty years, ML has become a staple in the storage and oil and gas transportation industries [21]. Pandya et al. (2021) addressed the challenges of predicting solid particle erosion in oil and gas equipment, noting significant errors in traditional methods like CFD and empirical models. By integrating ML with CFD techniques and analyzing 90° elbows made from various materials, they introduced regression models, improving erosion rate prediction accuracy by 25%. Their ANN further optimized results, reducing errors from over 40% to less than 10% compared to the baseline model [22]. When Suresh et al. (2009) examined the PPS reinforced material's erosion behavior with different amounts of glass fiber, they discovered that the greatest rates of degradation occurred at a 30° impact angle. Although PPS is inherently brittle, its composites showed increased erosion rates with higher glass fiber content. The eroded surfaces' morphology was studied. Wear mechanisms were examined utilizing SEM. An ANN accurately predicted erosion rates, showing potential for future wear studies [23]. Karimi et al. (2020) researched several AI techniques (ML models), including Elastic Net, RF, and SVM-based, for erosion predictions. While their success has been limited in practical industrial adoption due to prior use of inconsistent testing approaches, the comprehensive training, evaluation, and validation under multiple conditions revealed that their approach can yield favorable results. Random Forest and SVM outperformed Elastic Net, with SVM aligning closely with both CFD simulations and the E/CRC's Sand Production Pipe Saver results. The study also delved into the AI model's capability to assess erosion uncertainty [24]. Employing ML, Chandra Pal et al. (2020) investigated gully erosion in West Bengal, India. They employed methods consisting of bagging, the BRT, as well as a special ensemble method that included K-fold cross-validation with BRT-bagging. Incorporating 20 factors related to gully erosion and drawing on data from 199 gully head cut points, they found the ensemble technique to be the most effective in forecasting gully erosion risks. These insights offer critical guidance for policymakers to address and mitigate the impacts of erosion [25]. Yang et al. (2021) examined erosion wear in fracturing pipelines, highlighting the combined effects of tensile stress and particle-induced damage. Through extensive experiments under various conditions, the study revealed the significant impact on the erosion rate.

Utilizing ML, the research found the random forest regression approach performed in predicting erosion in high-pressure pipelines [26]. Wang et al. (2022) studied pipeline erosion caused by solid particle impact, introducing an ML model optimized for gas–solid flows. For gas–liquid–solid flows, their proposed WOA–HKELM model aligns closely with experimental results, demonstrating high accuracy [21].

In this investigation, ML techniques were employed to predict pipeline erosion. The goal of the study was to improve overall predictive accuracy by utilizing SVR and AdaBoost with optimization via AEO, GWO, and MFO. Additionally, the study used comparative analysis of both single and hybrid-ML models to determine the most effective approach for improving pipeline erosion predictive accuracy.

The subsequent parts of this paper are structured as follows: Section 2 delves into the comprehensive procedure of the prediction techniques in use, detailing the proposed models, namely the SVR and AdaBoost, and elaborating on the working of the optimizers. The outcomes, as well as follow-up research on the models under consideration, are presented in Section 3, together with a variety of charts and tables for examination. The research wraps up with conclusions in section 4.

2. Methodology

This document provides a systematic account of the development of predictive models for pipe erosion using statistical methodologies and techniques. Careful collection of data on the various parameters that influence the erosion pattern was necessary in the initial stage of model development. Five variables – target material, tensile stress, flow velocity, impact angle, and solid concentration – were selected as input to the model based upon the absolute validity of their use with respect to the final output variable, erosion wear rate. While exploring the data collected in relation to pipe erosion, a complete analysis of the relationships between these input parameters was conducted. To examine the interactions among the selected parameters and their combined effect on pipeline erosion rate, various advanced statistical tools were employed, including correlation analysis and sensitivity analysis. Once the dataset was curated, representative samples of 80% of the data were used in the training phase of model-building; whereas, 20% of the data was reserved for validation/testing of the predictive models. Our modeling endeavors hinged on the state-of-the-art SVR and the AdaBoost regression techniques. To further enhance model performance, advanced optimization algorithms were applied to tune hyperparameters. The AEO, GWO, and MFO optimizers played a central role in this process. For AdaBoost, the max depth, learning rate, and the number of

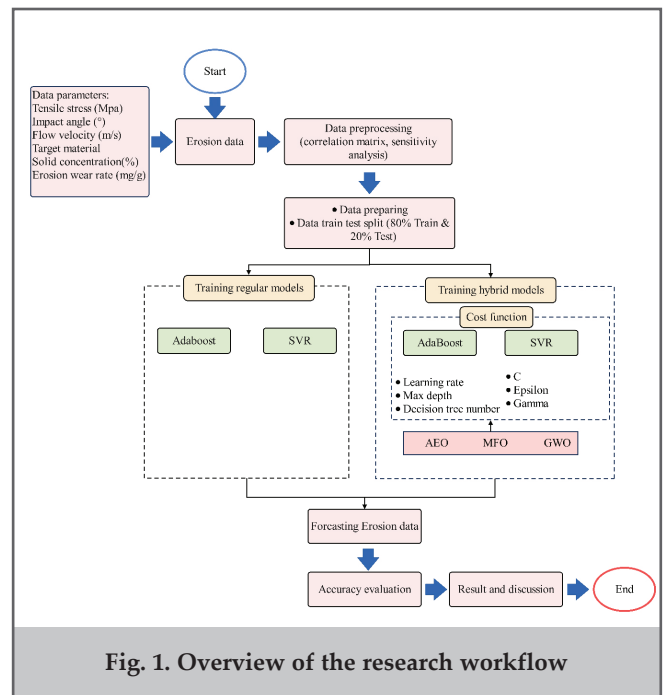


Fig. 1. Overview of the research workflow

decision trees were optimized, while C, epsilon, and gamma were fine-tuned for the SVR model. After finalizing the models, a comprehensive evaluation was conducted using multiple performance metrics to assess their effectiveness. The results, supported by informative visualizations, charts, and tables, are presented in the following sections. For the readers' convenience and to provide an overview of the proposed methodology, the entire process is summarized in a succinct flowchart, duly presented in figure1.

2.1. Data

The data set employed in this study is derived from previous research undertaken by Yang et al. [26]. Essentially, multiple elements play a crucial role in assessing the internal degradation of pipework systems. A detailed statistical overview of these elements is provided in table 1.

2.2. ML methods

In this section, the techniques used for predicting corrosion are briefly discussed. Models such as SVR and AdaBoost are utilized. The optimization methods incorporated include MFO, AEO, and GWO.

2.2.1. SVR

SVR is a widely used supervised machine learning technique for regression tasks. Its primary objective is to maximize the distance between the occurrences in the training data set [27]. SVR's ability to use curved decision limits to

Variables	count	mean	std	min	25%	50%	75%	max
Tensile stress (Mpa)	152	254.61	175.61	0	100	300	400	500
Impact angle (°)	152	43.03	23.46	15	30	30	60	90
Flow velocity (m/s)	152	19.31	2.54	7.5	20	20	20	22.50
Target material	152	1.47	1.02	0	1	1	2	3
Solid concentration(%)	152	11.18	4.52	5	10	10	10	30
Erosion wear rate (mg/g)	152	0.0026	0.0016	0.0002	0.0015	0.0022	0.0033	0.0082

capture non-linear relationships makes it particularly useful when working with complex data [28]. The foundational work on SVR was introduced by Vapnik et al. (1997) [29]. The subsequent equation defines SVR's mathematical representation:

$$f(x) = (w, \phi(x)) + b \quad (1)$$

The function $f(x)$ stands for a generic non-linear regression model. The function $\phi(x)$ represents a non-linear transformation applied to the input space x . The weight vector is represented by w , and the scalar bias by b [30]. A regularized risk function is also standardized to determine b and w 's values.

$$\min L(c) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n |\xi_i^* + \xi_i| \quad (2)$$

$$s.t. f(x_i) - y_i \leq \varepsilon + \xi_i \quad (3)$$

$$s.t. f(x_i) - f(x_i) \leq \varepsilon + \xi_i \quad (4)$$

$$s.t. \varepsilon \geq 0, \xi_i \geq 0, i = 1, 2, \dots, n \quad (5)$$

Slack variables are represented by the variables ξ_i^* and ξ_i . C is a penalty phrase that considers the severity of the error. The ε characterizes an insensitive loss function, highlighting the approximation's precision. To find the solution, the Lagrangian method can be used. Subsequently, the optimized regression model can be represented by the following equation:

$$f(x) = \sum_{i=1}^n (a_i^* - a_i) k(x_i, x') + b \quad (6)$$

In the study, the SVR was formulated utilizing the Gaussian kernel function, as described by equation (7).

$$k(x_i, x') = \exp\left(-\gamma \|x_i - x'\|^2\right) \quad (7)$$

where γ represents the Gaussian kernel function's parameter, in fact, the SVR model's prediction accuracy is greatly impacted by the parameters γ , C , and ε .

2.2.2. AdaBoost

In ML, AdaBoost is a sequential ensemble technique. It builds a robust model by combining several weak learners that are chosen at random from the dataset. These weak models are generated using various ML techniques. In each training cycle, specific weights are allocated to each data point, directing the hypothesis's learning [31-33]. If predictions are incorrect, the subsequent weak learner is given a higher weight to focus on these errors. This process continues until the algorithm achieves the desired level of accuracy. In regression tasks, predictions are not merely correct or incorrect but come with a measurable error. The ensemble model may make its final forecast utilizing the weighted average of all weak models or the median, rather than a binary outcome [34].

We obtain a system dataset S composed of pairs (x_i, y_i) with i ranging from 1 to N if we assume that each x_i is an instance from a set X and each y_i is a label from a set Y . The initial distribution (or weight) D is first configured by the algorithm as follows, taking into account the number of rounds indicated by M :

$$D_i^1 = \frac{1}{N}, \text{ for } i = \{1, \dots, N\} \quad (8)$$

For every iteration j from 1 to M , the AdaBoost method generates weak models or learners, represented by h_j , utilizing the training dataset and the current weight distribution D . The objective is to create a model that guarantees that ε_j is smaller than 0.5 in addition to minimizing the error ε_j . This ε_j is defined by equation (9), which is the weighted error for j th the model.

$$\alpha_j = \ln\left(\frac{1 - \varepsilon_j}{\varepsilon_j}\right), \quad \varepsilon_j = \sum_{i: h_j(x_i) \neq y_i} D_i^j \quad (9)$$

The confidence or weight α_j associated with the j th model is determined using equation (10).

$$\alpha_j = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_j}{\varepsilon_j}\right) \quad (10)$$

The weight distributions for the subsequent iteration are adjusted as follows:

$$D_i^{j+1} = e^{-y_i h_j(x_i) \alpha_j D_i^j} \quad (11)$$

$$D_i^{j+1} = \frac{D_i^{j+1}}{\sum_{i=1}^N D_i^{j+1}} \quad (12)$$

The models' weighted average, or h_j is calculated in the following way to determine the forecast for a fresh dataset:

$$H(x) = \sum_j \alpha_j h_j(x) \quad (13)$$

Due to its versatility, the AdaBoost technique can be used for both regression and classification tasks. However, only the AdaBoost algorithm for regression was used in the current study [35].

2.2.3. GWO

Grey wolves' hunting behavior as well as social structure served as the model for the GWO metaheuristic algorithm. Mirjalili introduced it in 2014 [36]. This algorithm emulates the wolves' hunting mechanisms. Wolves are split into four main groups according to the GWO framework: alpha (α), beta (β), delta (δ), and omega (ω).

The social hierarchy places the omega wolves at the bottom, whereas the alpha wolves have the highest level of authority within the pack. Beta wolves act in supporting roles to the alphas. During hunting exploration phases, the wolves' positions get adjusted according to the leading wolves' (α, β, δ) locations, described by X_α , X_β , and X_δ .

$$\bar{X}_1(t) = \bar{X}_\alpha(t) - \bar{A}_1 \cdot [\bar{C}_1 \cdot \bar{X}_\alpha(t) - \bar{X}(t)] \quad (14)$$

$$\bar{X}_2(t) = \bar{X}_\beta(t) - \bar{A}_2 \cdot [\bar{C}_2 \cdot \bar{X}_\beta(t) - \bar{X}(t)] \quad (15)$$

$$\bar{X}_3(t) = \bar{X}_\delta(t) - \bar{A}_3 \cdot [\bar{C}_3 \cdot \bar{X}_\delta(t) - \bar{X}(t)] \quad (16)$$

$$\bar{X}(t+1) = \frac{1}{3} \{ \bar{X}_1(t) + \bar{X}_2(t) + \bar{X}_3(t) \} \quad (17)$$

In this case, t is the number of iterations, and X is the grey wolf's location vector. The subsequent is A and C 's definition as coefficient vectors:

$$\bar{A} = 2\bar{a} \cdot \bar{r}_1 - \bar{a} \quad (18)$$

$$C = 2 \cdot \bar{r}_2 \quad (19)$$

In this case, $\bar{a}$ is a vector that decreases linearly from 2 to 0 during the iterations. The vectors r_1 and r_2 are random

and fall within the [0,1] range. For a comprehensive understanding and the actual implementation of this metaheuristic algorithm, refer to the website: <http://www.alimirjalili.com/GWO.html>.

2.2.3. MFO

T inspired by the natural navigational strategies used by moths during flight, the MFO algorithm was first presented in 2015 [37]. At night, moths see the moon's light as parallel rays and utilize it as a navigational aid. The MFO algorithm mimics moth behavior when attracted to lights and is an optimization technique for non-convex functions with more than one local maximum. Although moths are capable of flying at different angles to obtain this result, when attracted to the light source, moths will maintain a constant angle towards the light. Due to this behavior, moths create spirals, where, as they approach the light, they converge toward it in successively smaller circles. To identify regions that are more likely to hold the global optimum, the MFO algorithm is skilled at exhaustively examining the search space. Moths as well as flames are crucial elements of the MFO algorithm [38, 39].

Moths exist in a multi-dimensional space (1D, 2D, 3D, etc.) that has a rich set of alternative pathways, called the fitness landscape, which are represented in the matrix M and the OM array. The best places the moths have attained, however, are referred to as flames and are listed in matrix F . The OF array contains the fitness metrics corresponding them.

Each moth adjusts its position according to its corresponding flame, which can be mathematically represented with the subsequent equation:

$$\vec{M}_i = \vec{S}(\vec{M}_i, \vec{F}_j) \quad (20)$$

In this case $\vec{M}_i$ stands for the location of the i -th moth, $\vec{F}_j$ for the j -th flame following sorting, as well as $\vec{S}$ for the spiral function.

The spiral function has certain situations that it needs to meet:

1. The initial position vector of the $\vec{S}$ function must be established before the MFO algorithm can do the required calculations.
2. The optimal solution found throughout each cycle of the MFO algorithm should be preserved by the $\vec{S}$ function at the conclusion of each iteration.
3. The lower and upper bound vectors specify the range within which the function's amplitude falls.

Given these prerequisites, the equation for the spiral function is expressed as follows:

$$\vec{S}(\vec{M}_i, \vec{F}_j) = \vec{D}_i \cdot e^{bt} \cdot \cos(2\pi t) + \vec{F}_j \quad (21)$$

A logarithmic helix is represented by the constant b . The range of values that the variable t can take is [-1,1]. The distance between the i -th moth as well as the j -th flame is shown by the vector $\vec{D}_i$. Eq. (22) explains how to calculate $\vec{D}_i$.

$$\vec{D}_i = |\vec{F}_j - \vec{M}_i| \quad (22)$$

Eq. (21) has its constraints, as it solely illustrates the moth's flight path towards the flame. The MFO method may converge to a local optimum as a result of this feature. A dynamic update method for the flame is necessary to get around this issue. Reducing the number of flames in the fire-place, the time required to calculate solutions is decreased, allowing the MFO algorithm to be more efficient. Eq. (23) displays the flame's modification formula.

$$flame_{no} = round\left(N - k \cdot \frac{N-1}{T}\right) \quad (23)$$

Here, k is the current iteration number, N is the maximum number of flames, and T is the total number of iterations allowed. Once the termination conditions for the iterations are met, the optimal moth is identified and returned as the best achieved value.

2.2.3. AEO

An innovative optimization technique called the AEO draws inspiration from the dynamics and actions of natural ecosystems, as explained by [40]. The method simulates the flow of energy within an ecosystem as well as the interactions and behavior of living organisms, such as through production, decomposition, and consumption.

AEO operates in three main stages:

1. Production Stage: This is the initial phase, which primarily enhances the solution's capability in both the exploitation and exploration phases. It helps the algorithm to navigate and explore the potential solution spaces.
2. Consumption Stage: The AEO consumption stage demonstrates the exploration capability of the MFO algorithm as it travels through all possible paths to find the optimal solution.
3. Decomposition Stage: This stage is directed towards ensuring the algorithm zeroes in on a solution within the permissible solution areas, refining its search, and ensuring the viability of the solution.

Stage 1: Production

In the production phase of AEO, agents are generated, and a random number is assigned to each agent. Following this, each agent's fitness value is computed. The agent with the lowest or least favorable fitness value is termed the producer. The value or state of the producer can then be modified or updated based on the following mathematical equation:

$$x_i(t+1) = (1-a)x_n(t) + a \times x_r(t), \quad a = \left(1 - \frac{t}{t_{max}}\right) \quad (24)$$

In this context, t_{max} denotes the overall number of iterations, and t indicates the current iteration. The best solution, referred to as the decomposer, is denoted by x_n . The factor a acts as a weighting parameter, while r_1 , randomly selected from the range [0, 1] represents a random solution. x_r is an agent randomly selected from the population X , and its update is determined by a subsequent equation:

$$x_r(t) = \beta \times (UB - LB) + LB \quad (25)$$

The search space boundaries are delineated by UB and LB . β represents a random vector.

Stage 2: Consumption

In this phase, agents are referred to as consumers. These consumers can consume either the x_1 (known as the producer) or another random consumer with a lower energy level. There exist multiple consumer types, and each adopts a distinct strategy for its update. The main type, known as herbivores, updates its position as described below:

$$x_i(t+1) = x_i(t) + C \times (x_i(t) - x_j(t)) \quad (26)$$

C represents a consumption factor. This factor is determined through a basic random walk, inspired by the Lévy flight behavior.

The next category is termed as carnivore, and its update is defined by a subsequent equation:

$$x_i(t+1) = x_i(t) + C \times (x_i(t) - x_j(t)),$$

$$j = \text{randi}[2i-1], \quad i = 3, \dots, N \quad (27)$$

The Omnivore represents the third consumer category, and its value is updated using the subsequent equation.

$$x_i(t+1) = x_i(t) + C \times r_3 \times (x_i(t) - x_1(t)) +$$

$$+ C \times (1 - r_3) \times (x_i(t) - x_j(t)) \quad (28)$$

where j is a random number drawn from the range $[2, (i-1)]$ as well as r_3 is a random value selected from the interval $[0, 1]$.

Stage 3: Decomposition

The operators in this stage mimic the behaviors of living organisms that decompose the remnants of deceased agents. This process is formulated as follows:

$$x_i(t+1) = x_i(t) + D \times (e \times x_n(t) - h \times x_i(t)), \quad i = 1, \dots, N \quad (29)$$

$$D = 3u, \quad h = 2r_4 - 1, \quad e = r_4 \text{ randi}([12]) - 1, \quad (30)$$

$$u \in N(0, 1)$$

where D stands for the decomposition factor, and both h and e serve as weight coefficients.

2.2.6. Hyperparameter Tuning

To ensure optimal predictive performance, each optimization algorithm (AEO, GWO, and MFO) was used to tune the hyperparameters of the SVR and AdaBoost models within predefined search ranges. The SVR model parameters, namely the penalty coefficient C , the insensitive loss parameter ϵ , and the kernel width γ were optimized within the intervals $C \in [1, 200]$, $\epsilon \in [0.0001, 0.1]$, and $\gamma \in [0.01, 1]$. For AdaBoost. R2, the hyperparameters tuned included the learning rate $\in [0.01, 1]$, the maximum depth of the base estimator $\in [1, 10]$, and the number of estimators $\in [50, 300]$. Each optimization algorithm iteratively searched the space of possible solutions to minimize the prediction error (based on RMSE), ultimately converging to the most suitable hyperparameter values. The optimized hyperparameters obtained for all hybrid models are summarized in table 2.

These optimized hyperparameters were subsequently used to train the final hybrid models, whose performance is presented in the Results and Discussion section. The differences in optimal values across algorithms reflect the distinct search mechanisms of AEO, GWO, and MFO. Models tuned with MFO generally converged to more efficient parameter

sets, aligning with the superior predictive accuracy reported for SVR-MFO in later sections. This table ensures full transparency regarding the configuration of all hybrid models evaluated in this study.

2.3. PAWN sensitivity analysis index

By determining the changes caused by the removal of uncertainty from one or more inputs, density-based sensitivity indices quantify the sensitivity. To do this, the sensitivity to a variable x is measured by calculating the difference between the output probability distribution curve when all inputs are changed and when the variable x remains constant. In this article, a novel method named PAWN (named after the authors) [41] is employed to enhance the utilization of density-based methods. Since cumulative distributions are easy to approximate for determining the distance between the conditional and unconditional states, this method's main idea is to employ them. Consequently, invariance of the variable x simplifies computations for assessing the index under different conditions. The Kolmogorov-Smirnov statistics is used to determine the distance between the unconditional and conditional states [42].

Continuing, we represent the cumulative distribution function of the output variable y in the conditional state with the input variable x fixed at $x|F(x)$ as $F(y|x)$, as well as in the unconditional state as $F(y)$. Thus, $F(y)$ represents the events that occur by removing variations on y_i , and the distance of $F(y)$ from $F(y|x)$ signifies the representation of the effects of input variations, denoted by x_i , on the input x .

As previously stated, Eq. (31) utilizes the Kolmogorov-Smirnov statistic to determine the distance.

$$KS(X_i) = \max_y |F_y(y) - F_{y|x_i}(y)| \quad (31)$$

Since KS depends on the value at which the input x is regarded as constant, Eq. (32) defines the PAWN sensitivity index, represented by T_i , as the statistic (maximum or average) of all conceivable states that x can have.

$$T_i(X_i) = \text{stat}_{x_i} [KS(X_i)] \quad (32)$$

Based on this definition, T_i ranges from 0 to 1. A lower value of T_i suggests a reduced influence of X_i with $T_i=0$ signifying that the input X_i does not affect the output at all. This index meets all the criteria mentioned for a good index. In essence, this index is universal, countable, and model-independent. Moreover, the dependency on the KS conditions, when using its statistic (maximum or average) relative to x , is eliminated. This index is very easy to interpret since it is defined as a distance, and its numeric value is not dependent on the measurement unit of y . Calculating this index is simple, thus making its stability analysis feasible.

The PAWN sensitivity analysis was incorporated to ensure a systematic and quantitative assessment of each parameter's influence on erosion rate prior to model optimization. Given the non-linear structure of erosion mechanisms and the interaction between flow conditions, material behavior, and particle dynamics, a density-based sensitivity method enables capturing the variability of outputs when each input is fixed. This approach reduces model uncertainty, prevents overfitting, and ensures that only the most informative parameters drive the optimization process. Therefore, PAWN serves as a critical preprocessing step in building reliable hybrid predictive models.

Table 2
Optimal hyperparameters obtained for hybrid SVR and AdaBoost models

Model	C	ϵ	γ	Learning rate	Max depth	n_estimators
SVR-MFO	135.7	0.0013	0.42	—	—	—
SVR-AEO	122.4	0.0018	0.51	—	—	—
SVR-GWO	148.2	0.0021	0.47	—	—	—
AdaBoost-MFO	—	—	—	0.12	4	150
AdaBoost-AEO	—	—	—	0.15	3	180
AdaBoost-GWO	—	—	—	0.09	5	130

2.4. Model verification and evaluation

A variety of analytical techniques and performance metrics are used to confirm the suggested models' correctness. By evaluating the remaining errors, these indicators draw attention to the discrepancies between the actual and predicted values. *RMSE*, *MAE*, *MAPE*, R^2 , *A10*, along with *VAF*, are among the metrics that are being utilized. Table 3 displays the mathematical equations that correspond to these statistical metrics.

3. Results and discussion

In this section, the outcomes generated by both the single and hybrid modeling approaches implemented to forecast pipeline erosion are discussed. The standalone models under scrutiny encompass the SVR and AdaBoost techniques. Concurrently, the hybrid models represent a fusion of these individual algorithms, optimized using the trio of optimization techniques: GWO, AEO, and MFO. A number of charts, graphical representations, as well as organized tables are used to methodically present these empirical data. In the following sections of the article, each outcome will be thoroughly discussed and evaluated analytically.

For the selected input and output variables of the model, a carefully created correlation matrix is presented in figure 2. Target material, impact angle, flow velocity, tensile stress, as well as solid concentration are all included in the input parameters. At the same time, the erosion wear rate is the study's intended result or target variable. The color gradation on the chart spans from -1 (depicted in a profound blue shade) to +1 (light gray). Positive values indicate a direct relationship, whereas negative values represent an inverse correlation. It is clear from interpreting the graphic representation that the parameters Target material and Impact angle, exhibit a negative correlation with the target variable. This infers that an increase in these parameters, specifically as the material strength increases, leads to a reduction in erosion. Furthermore, as the angle increases and approaches a horizontal orientation, the propensity for erosion also decreases. In stark contrast, parameters such as tensile stress and flow velocity showcase a pronounced positive correlation with the target, insinuating that escalations in these metrics correspond to a surge in the erosion rate.

Metrics like the minimum, maximum, average, median, and coefficient of variation (CV) were employed to compute the index for PAWN sensitivity analysis, as demonstrated in figure 3. As elucidated in figure 3, and upon careful consideration of both the Mean and Median metrics, it becomes evident that the solid concentration parameter emerges as the most sensitive determinant. Subsequent to this, the parameters tensile stress and impact angle manifest relatively comparable levels of sensitivity. In a contrasting observation, the parameter pertaining to target material exhibit lower sensitivity compared to its counterparts.

As shown in figure 4, the empirical outcomes generated by the SVR and AdaBoost algorithms are systematically analysed. This visual representation encompasses a comparison of observed and calculated data in time-series format, augmented by scatter plots and complemented by an error metrics table. The scatter plot reveals a pronounced dispersion of data points along the $x=y$ axis, coupled with a decrease in clustering. This pattern underscores the suboptimal precision inherent in the SVR and AdaBoost methodologies.

Table 3		
Statistical evaluation indices		
Statistics	Criteria	Equation
<i>MAPE</i>	Mean absolute percentage error	$MAPE = \frac{1}{N} \sum_{i=1}^n \left \frac{y_i - \hat{y}_i}{y_i} \right $
<i>RMSE</i>	Root mean square error	$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}}$
<i>VAF</i>	Variance accounted for	$VAF = 100\% \times \frac{\sum_{i=1}^n (y_i - \bar{y})(f_i - \bar{f})}{\sum_{i=1}^n (y_i - \bar{y})^2}$
<i>MAE</i>	Mean absolute error	$MAE = \frac{\sum_{i=1}^n y_i - \hat{y}_i }{n}$
R^2	Coefficient of determination	$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$
<i>A10</i>	A10 Index	$A10 = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1 & \text{if } \frac{ \hat{y}_i - y_i }{y_i} \leq 0.1 \\ 0, & \text{otherwise} \end{cases}$



Fig. 2. The features' correlation matrix

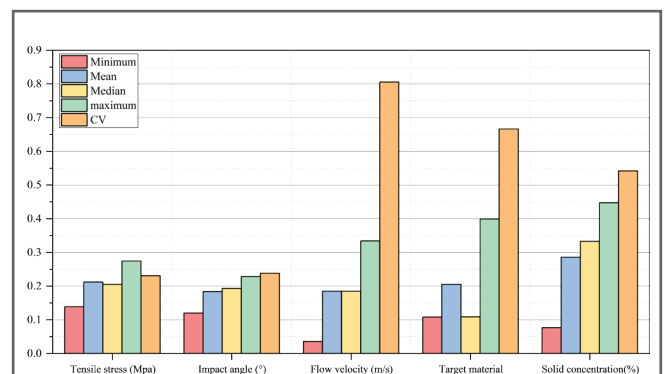


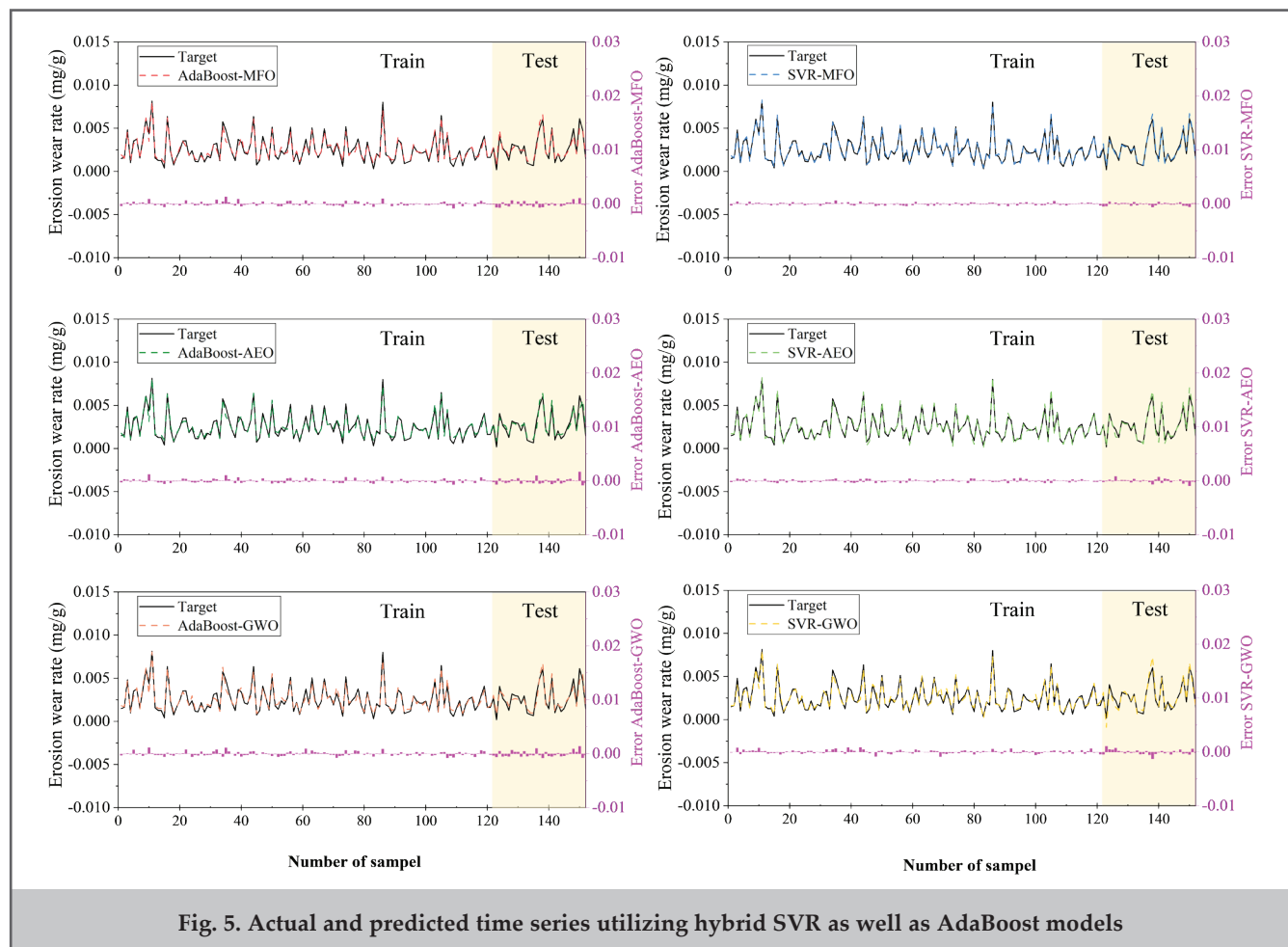
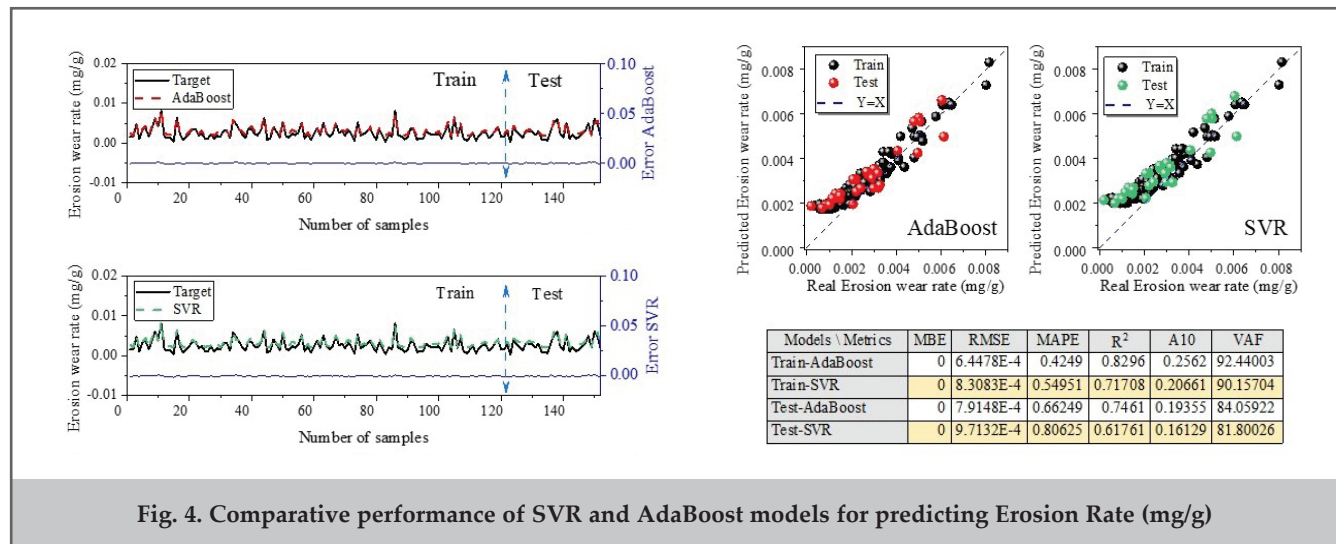
Fig. 3. Variable sensitivity analysis using the PAWN index

A detailed examination of the error metrics, particularly the R^2 coefficient, shows that the SVR model, registering an R^2 value of 0.6176, lags behind the AdaBoost algorithm, with an R^2 of 0.7461. This relative disparity in R^2 values is a testament to the inherent limitations of the algorithms when applied independently, pointing to the imperative for further algorithmic refinement and optimization.

As mentioned in the previous section, standalone algorithms did not meet our performance criteria when executed independently. To address this challenge, three optimization techniques, GWO, AEO, and MFO, were used to optimize the parameters inherent to these algorithms. A careful visual

depiction of the time-series data is shown in figure 5, which makes a distinction between computed and observed values during the training and testing stages. Moreover, the error trajectory is specified with a salient pink line for enhanced clarity. An analysis of the visual data, supplemented by the quantified error rates, highlights that the error magnitude in the hybrid SVR models is notably smaller in comparison to their AdaBoost counterparts. Notably, the SVR-MFO hybrid emerges as the most promising configuration, registering the narrowest error bandwidth.

To provide a more nuanced evaluation of the hybrid models' performance, scatter plots are depicted in figure 6.



The scatter plot of each hybrid model includes the R^2 coefficients for the training and testing stages. A detailed assessment of the ensuing R^2 values indicates that all models achieved satisfactory training performance. However, during the validation phase, the SVR-MFO hybrid emerges preeminently, registering an unparalleled performance metric. It warrants emphasis that an optimal model is often characterized by minimal deviation between the R^2 values

training and testing phases. Based on the empirical data, the hybrid SVR models manifest a diminished variance in their respective R^2 values. Notably, the SVR-MFO hybrid exhibits the least variance, underscoring its robustness. In stark contrast, the AdaBoost-GWO hybrid reveals the most pronounced discrepancy, signifying a suboptimal performance trajectory among the evaluated models.

In addition to the aforementioned methods for evalu-

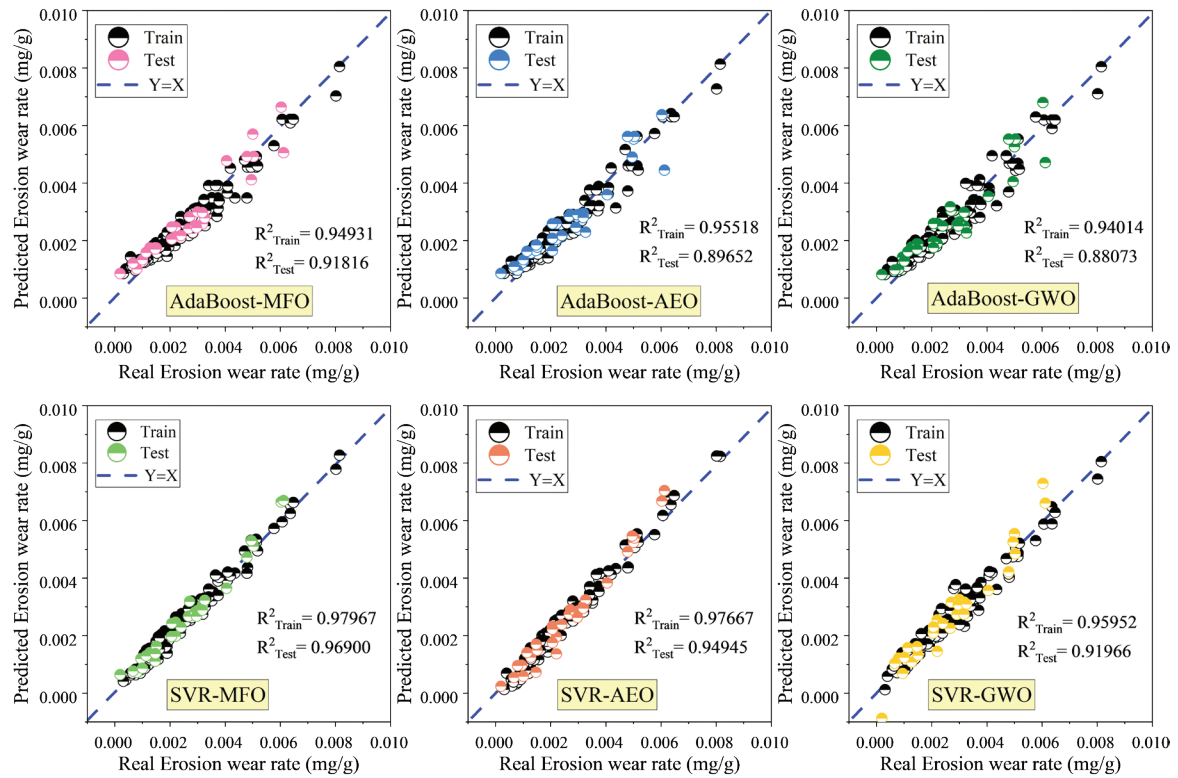


Fig. 6. SVR and AdaBoost hybrid models' observation-prediction scatter plot

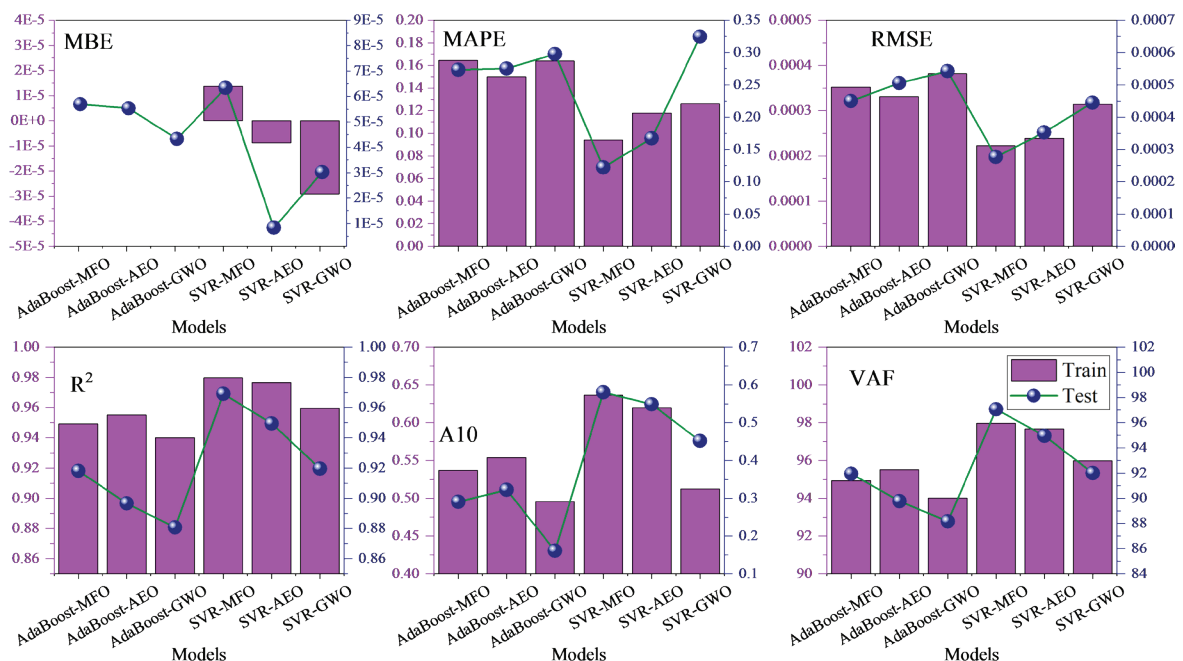
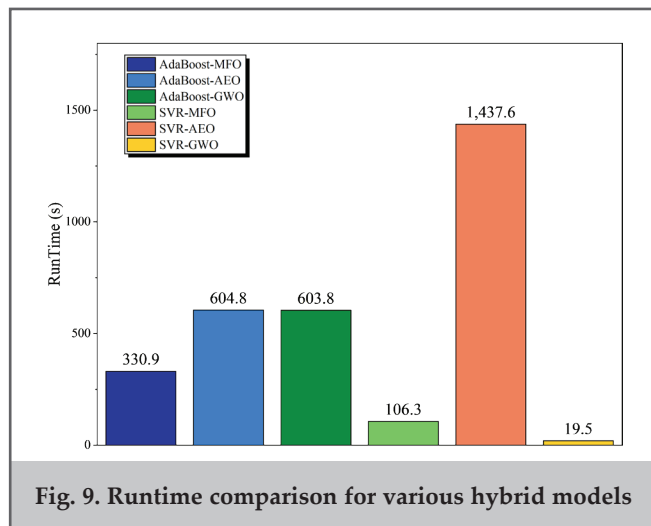
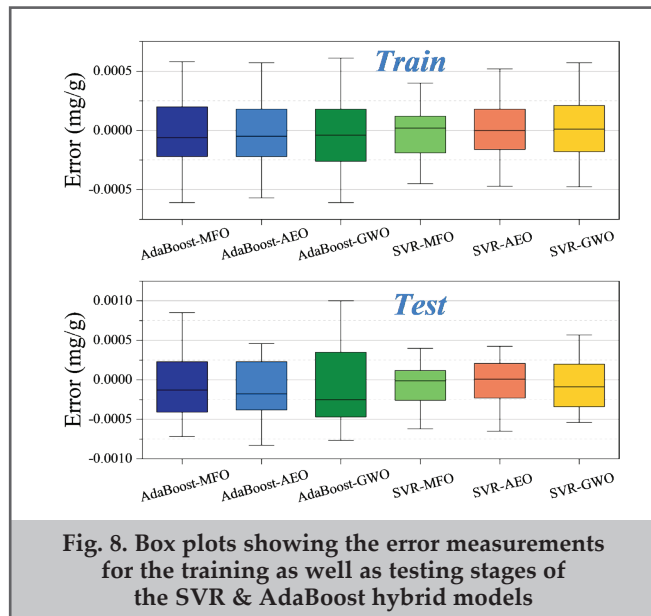


Fig. 7. Plots of error metrics for suggested hybrid models

ating model performance, several other metrics have been employed, the results of which are presented in figure 7. These metrics include RMSE, R^2 , MAPE, MBE, VAF, and A10. Results for the training and testing segments are provided.



The training phase is delineated using rectangular graphs, while the testing phase is illustrated with line graphs. Taking into account the two critical indices, R^2 and RMSE, it is evident that the SVR models exhibit a superior performance in comparison to the hybrid AdaBoost models. The hybrid AdaBoost-GWO model is responsible for the biggest error in the RMSE figure, while the SVR-MFO model records the lowest error. Similarly, in the R^2 graph, the highest value pertains to SVR-MFO, and the lowest is associated with AdaBoost-GWO. This trend is consistent across all charts and other metrics. The numerical values for each metric are reported in table 4 for a more thorough analysis of the indices and to determine the exact performance of each model.

The box plots of the hybrid models for the training and testing stages are presented in figure 8. This graphical representation provides insights into data dispersion and the error range intrinsic to each model. Within the training segment, it is evident that the SVR-MFO model exhibits the narrowest error amplitude, with its median line closely approximating zero. In the testing realm, the SVR-MFO model continues to demonstrate the lowest error magnitude, whereas the AdaBoost-GWO model manifests the most expansive error range among the evaluated models.

Figure 9 elucidates the runtime of the hybrid models. As depicted in the chart, the SVR-AEO model requires the most extended execution time, reaching 1437.6 seconds. Conversely, both the SVR-MFO and SVR-GWO models emerge as the most efficient in terms of runtime among the evaluated models, with SVR-GWO demonstrating the shortest runtime at 19.5 seconds. It is worth noting that runtime depends on various factors, including the search strategy and the convergence properties of the models, which significantly influence the model's operational duration.

Figure 10 illustrates the search process in the hybrid models to ascertain the optimal state. This quest, tracked over 500 iterations, has been documented leveraging two pivotal indices, namely exploitation and exploration. Drawing insights from the chart, it is evident that models SVR-MFO and SVR-GWO have swiftly converged to their respective optimal states, whereas the SVR-AEO model demonstrated a more protracted convergence trajectory. The contrasting runtime durations, notably the efficiency

Phase	Metric	AdaBoost-MFO	AdaBoost-AEO	AdaBoost-GWO	SVR-MFO	SVR-AEO	SVR-GWO
Train	MAE	0.00000000	0.00000000	0.00000000	0.00001376	-0.0000087	-0.0000291
	MAPE	0.16466	0.15000	0.16402	0.09418	0.11792	0.12630
	RMSE	0.00035	0.00033	0.00038	0.00022	0.00024	0.00031
	R^2	0.94932	0.95518	0.94015	0.97967	0.97668	0.95953
	A10	0.54	0.55	0.50	0.64	0.62	0.51
	VAF	94.93179	95.51807	94.01450	97.97479	97.67079	95.98777
Test	MAE	0.00005698	0.00005529	0.00004329	0.00006337	0.00000834	0.00003026
	MAPE	0.27324	0.27514	0.29786	0.12235	0.16706	0.32439
	RMSE	0.00045	0.00051	0.00054	0.00028	0.00035	0.00045
	R^2	0.91817	0.89653	0.88074	0.96901	0.94946	0.91967
	A10	0.29	0.32	0.16	0.58	0.55	0.45
	VAF	91.94859	89.77649	88.14984	97.06359	94.94837	92.00383

of SVR-MFO and SVR-GWO compared with SVR-AEO, can be attributed to their respective approaches in reaching the optimal state.

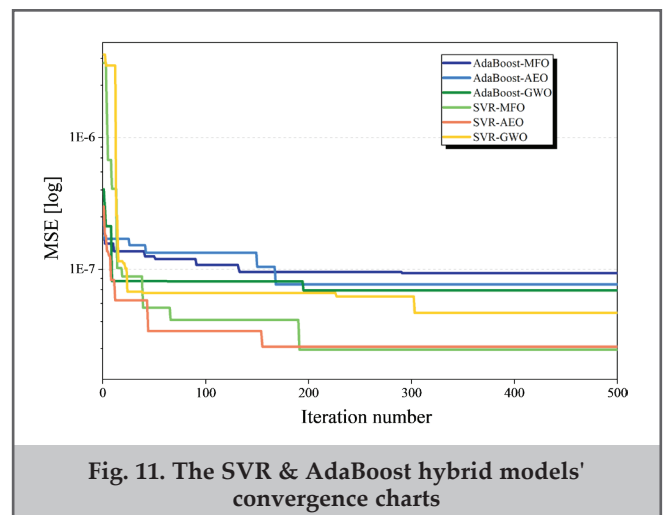
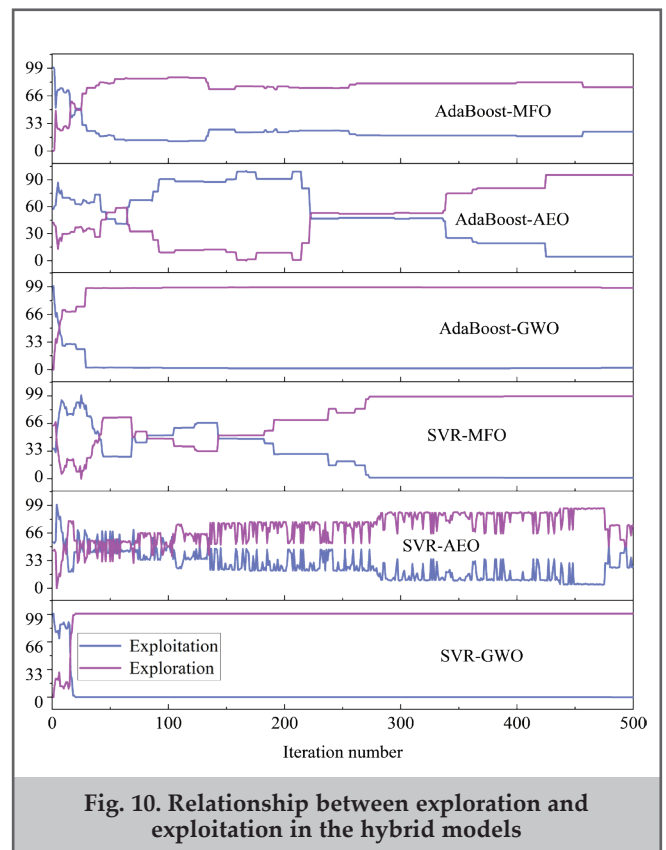
The hybrid models' convergence graphs are presented in figure 11. Convergence is measured using the Mean Squared Error (MSE) index, which is shown on the chart's vertical axis, with the number of iterations limited to 500. Drawing from the insights of figure 11, the hybrid SVR models demonstrate convergence characterized by a lower MSE index, among which the SVR-MFO hybrid distinctly registers the lowest MSE. In contrast, all three hybrid AdaBoost models exhibit a comparatively higher MSE, underscoring a less favorable convergence metric compared to the SVR models.

Nomenclature

A and C	Coefficient vectors
$A10$	A10 Index
$AdaBoost$	Adaptive Boosting Regression
AEO	Artificial Ecosystem-Based Optimization
ANN	Artificial neural network
BRT	Boosted regression tree
UB and LB	The boundaries of the search space
CFD	Computational Fluid Dynamics
D	Decomposition factor
$\bar{D}_i$	The distance between the i -th moth and the j -th flame
$\phi(x)$	Represents a non-linear transformation
GWO	Grey Wolf Optimizer
$HKELM$	Hybrid kernel extreme learning machine
$\bar{M}_i$	The position of the i -th moth
γ	Denotes the parameter of the Gaussian kernel function
$MAPE$	Mean Absolute Percentage Error
MFO	Moth Flame Optimization
X	The position vector of a grey wolf
PPS	Polyphenylene sulfide
R^2	Coefficient of Determination
$RMSE$	Root Mean Square Error
SVM	Support Vector Machine
$\bar{S}$	The spiral function
SVR	Support Vector Regression
C	Penalty term
T	Total allowable iterations
VAF	Variance Accounted For
WOA	Whale optimization algorithm
$\bar{F}_j$	Denotes the flame
ξ_i^* and ξ_i	Slack variables
ε	Insensitive loss function
AE	Mean Absolute Error
ML	Machine learning
x_n	Best solution
β	Random vector

Additional model relevance assessment techniques

To further strengthen the reliability of the hybrid models, additional validation approaches can complement the statistical metrics used in this study. Cross-validation can assess the robustness of the models across different subsets of the data, while bootstrapping-based uncertainty quantification provides insight into the stability of predictions under parameter variability. Furthermore, comparing model outputs with traditional mechanistic erosion correlations can offer an additional layer of verification. Integrating these techniques into future research can significantly reduce the risk of model misinterpretation in managerial decision-making.



Conclusions

In this research, various ML methodologies were investigated to predict the erosion rates. Several ML algorithms, consisting of SVR and AdaBoost, as well as hybrid models optimized using GWO, AEO, and MFO, were used. The application of ML facilitates faster decision-making and uncertainty management within a probabilistic framework. The main outcomes of this study are:

- Tensile stress, impact angle, flow velocity, target material, and solid concentration were identified and confirmed as important factors affecting erosion patterns.
- Advanced statistical tools like correlation matrix and sensitivity analysis revealed:
- Target material and impact angle showed negative correlations with erosion.
- Tensile stress and flow velocity presented positive correlations with the erosion rate.
- Optimization algorithms (AEO, GWO, MFO) played a crucial role in refining models through hyper-parameter tuning.
- Standalone models revealed limitations, thus emphasizing the need for hybrid modeling approaches.
- The hybrid models, particularly the SVR-MFO configuration, consistently outperformed others:
- Demonstrated a narrower error bandwidth.
- Exhibited lower variance between training and testing phases.
- Efficiently converged to optimal states.
- The SVR-AEO model had the longest execution time, while the SVR-GWO & SVR-MFO models were the most time-efficient.

The hybrid modeling framework developed in this study can be directly integrated into the design and evaluation of pipeline systems operating under complex geological or climatic conditions. By accurately estimating erosion rates under high stress, high-velocity, or particle-rich flow environments, the models, particularly the SVR-MFO hybrid, can guide the selection of pipeline materials, define safe operating envelopes, and support predictive maintenance planning. These capabilities are especially relevant for pipelines in Arctic regions, mountainous areas, or abrasive sediment-laden environments.

This research offers easily accessible and cost-effective tools for estimating erosion rates under different operational scenarios, assisting pipeline managers in ensuring more secure and sustainable pipeline operations.

References

1. Chen, C., Li, C., Reniers, G., Yang, F. (2021). Safety and security of oil and gas pipeline transportation: a systematic analysis of research trends and future needs using WoS. *Journal of Cleaner Production*, 279, 123583.
2. Nurlan, Z. (2022). Evaluate the depth of scouring of the bridge base using soft calculation techniques. *Advances in Engineering and Intelligence Systems*, 1(2), 1–7.
3. He, G., Lyu, X., Liao, K., et al. (2019). A method for fast simulating the liquid seepage-diffusion process coupled with internal flow after leaking from buried pipelines. *Journal of Cleaner Production*, 240, 118167.
4. Shin, S., Lee, G., Ahmed, U., et al. (2018). Risk-based underground pipeline safety management considering corrosion effect. *Journal of Hazardous Materials*, 342, 279–289.
5. Wee, S. K., Yap, Y. J. (2019). CFD study of sand erosion in pipeline. *Journal of Petroleum Science and Engineering*, 176, 269–278.
6. Zolfagharnasab, M. H., Salimi, M., Zolfagharnasab, H., et al. (2021). A novel numerical investigation of erosion wear over various 90-degree elbow duct sections. *Powder Technology*, 380, 1–17.
7. Wang, Q., Ba, X., Huang, Q., et al. (2022). Modeling erosion process in elbows of petroleum pipelines using large eddy simulation. *Journal of Petroleum Science and Engineering*, 211, 110216.
8. Josifovic, A., Roberts, J. J., Corney, J., et al. (2016). Reducing the environmental impact of hydraulic fracturing through design optimisation of positive displacement pumps. *Energy*, 115, 1216–1233.
9. Rezaei, A., Dindoruk, B., Soliman, M. Y. (2019). On parameters affecting the propagation of hydraulic fractures from infill wells. *Journal of Petroleum Science and Engineering*, 182, 106255.
10. Yang, S., Fan, J., Zhang, L., Sun, B. (2021). Performance prediction of erosion in elbows for slurry flow under high internal pressure. *Tribology International*, 157, 106879.
11. Abedini, M., Ghasemi, H. M. (2014). Synergistic erosion–corrosion behavior of Al–brass alloy at various impingement angles. *Wear*, 319(1–2), 49–55.
12. Nguyen, Q. B., Lim, C. Y. H., Nguyen, V. B., et al. (2014). Slurry erosion characteristics and erosion mechanisms of stainless steel. *Tribology International*, 79, 1–7.
13. Desale, G. R., Gandhi, B. K., Jain, S. C. (2009). Particle size effects on the slurry erosion of aluminium alloy (AA 6063). *Wear*, 266(11–12), 1066–1071.
14. Lu, B. T., Luo, J. L. (2015). Correlation between surface-hardness degradation and erosion resistance of carbon steel—Effects of slurry chemistry. *Tribology International*, 83, 146–155.
15. Yoganandh, J., Natarajan, S., Babu, S. P. K. (2013). Erosive wear behavior of nickel-based high alloy white cast iron under mining conditions using orthogonal array. *Journal of Materials Engineering and Performance*, 22(9), 2534–2541.

16. Liu, G., Ayello, F., Vera, J., et al. (2021). An exploration on the machine learning approaches to determine the erosion rates for liquid hydrocarbon transmission pipelines towards safer and cleaner transportations. *Journal of Cleaner Production*, 295, 126478.
17. Sheng, M., Huang, Z.-W., Tian, S.-C., et al. (2020). CFD analysis and field observation of tool erosion caused by abrasive waterjet fracturing. *Petroleum Science*, 17(3), 701–711.
18. Lu, Y., Agrawal, M. (2014). A computational-fluid-dynamics-based Eulerian-granular approach for characterization of sand erosion in multiphase-flow systems. *SPE Journal*, 19(4), 586–597.
19. Yu, W., Fede, P., Climent, E., Sanders, S. (2019). Multi-fluid approach for the numerical prediction of wall erosion in an elbow. *Powder Technology*, 354, 561–583.
20. Bilal, F. S., Sedrez, T. A., Shirazi, S. A. (2021). Experimental and CFD investigations of 45 and 90 degrees bends and various elbow curvature radii effects on solid particle erosion. *Wear*, 476, 203646.
21. Wang, Z., Chen, H., Wang, M., et al. (2022). Solid particle erosion prediction in elbows based on machine learning and swarm intelligence algorithm. *Journal of Petroleum Science and Engineering*, 218, 111042.
22. Pandya, D. A., Dennis, B. H., Russell, R. D. (2017). A computational fluid dynamics based artificial neural network model to predict solid particle erosion. *Wear*, 378, 198–210.
23. Suresh, A., Harsha, A. P., Ghosh, M. K. (2009). Solid particle erosion studies on polyphenylene sulfide composites and prediction on erosion data using artificial neural networks. *Wear*, 266(1–2), 184–193.
24. Karimi, S., Xu, B., Asgharpour, A., et al. (2020). Predicting solid particle erosion and uncertainty in elbows by artificial intelligence methods. In: *Fluids Engineering Division Summer Meeting, American Society of Mechanical Engineers*.
25. Pal, S. C., Arabameri, A., Blaschke, T., et al. (2020). Ensemble of machine-learning methods for predicting gully erosion susceptibility. *Remote Sensing (Basel)*, 12(22), 3675.
26. Yang, S., Zhang, L., Fan, J., Sun, B. (2021). Experimental study on erosion behavior of fracturing pipeline involving tensile stress and erosion prediction using random forest regression. *Journal of Natural Gas Science and Engineering*, 87, 103760.
27. Li, D., Simske, S. (2010). Example based single-frame image super-resolution by support vector regression. *Journal of Pattern Recognition Research*, 1(1), 104–118.
28. Sadaghat, B., Javadzade Khiavi, A., Naeim, B., et al. (2023). The utilization of a Naïve Bayes model for predicting the energy consumption of buildings. *Journal of Artificial Intelligence and System Modelling*, 1(1), 72–89.
29. Vapnik, V., Golowich, S., Smola, A. (1996). Support vector method for function approximation, regression estimation and signal processing. *Advances in Neural Information Processing Systems*, 9, 281–287.
30. Naeim, B., Akbarzadeh, M. R., Jahangiri, V. (2024). Machine learning-based prediction of seismic response of elevated steel tanks. *Structures*, 70, 107649.
31. Khajavi, E., Taghavi Khanghah, A. R., Javadzade Khiavi, A. (2025). An efficient prediction of punching shear strength in reinforced concrete slabs through boosting methods and metaheuristic algorithms. *Structures*, 74, 108519.
32. Taghavi Khanghah, A. R., Khajavi, E., Azizi, H., Alizade Novin, A. R. (2024). Radial basis function coupling with metaheuristic algorithms for estimating the compressive strength and slump of high-performance concrete. *Advances in Engineering and Intelligence Systems*, 3(4), 124–142.
33. Naeim, B., Javadzade Khiavi, A., Erfan Khajavi, E., et al. (2025). Machine learning approaches for fatigue life prediction of steel and feature importance analyses. *Infrastructures (Basel)*, 10(11), 295.
34. Gupta, K. K., Kalita, K., Ghadai, R. K., et al. (2021). Machine learning-based predictive modelling of biodiesel production - a comparative perspective. *Energies (Basel)*, 14(4), 1122.
35. Freund, Y., Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139.
36. Mirjalili, S., Mirjalili, S. M., Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61.
37. Zhao, X., Zhang, X., Cai, Zh., et al. (2019). Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients. *Computational Biology and Chemistry*, 78, 481–490.
38. Xu, Y., Chen, H., Asghar Heidari, A., et al. (2019). An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Systems with Applications*, 129, 135–155.
39. Heidari, A. A., Yin, Y., Mafarja, M., et al. (2019). Efficient moth-flame-based neuroevolution models. In: *Evolutionary machine learning techniques: algorithms and applications*. Springer.
40. Oliva, D., Abd Elaziz, M., Elsheikh, A. H., Ewees, A. A. (2019). A review on meta-heuristics methods for estimating parameters of solar cells. *Journal of Power Sources*, 435, 126683.
41. Pianosi, F., Wagener, T. (2015). A simple and efficient method for global sensitivity analysis based on cumulative distribution functions. *Environmental Modelling & Software*, 67, 1–11.
42. Stephens, M. A. (1992). Introduction to Kolmogorov (1933) on the empirical determination of a distribution. In: *Breakthroughs in statistics: Methodology and distribution*. Springer.