Hindawi
International Journal of Corrosion
Volume 2024, Article ID 9427747, 11 pages
https://doi.org/10.1155/2024/9427747

Research Article

Walk-Through Corrosion Assessment of Slurry Pipeline Using Machine Learning

Abdou Khadir Dia,[1] **Axel Gambou Bosca**,[2] **and Nadia Ghazzali**[1]

[1]*Université du Québec à Trois-Rivières, Department of Mathematics and Computer Science, Trois-Rivières, Canada*
[2]*Québec Metallurgy Center, Trois-Rivières, Canada*

Correspondence should be addressed to Abdou Khadir Dia; abdou.khadir.dia@uqtr.ca

Received 31 July 2023; Revised 18 December 2023; Accepted 13 April 2024; Published 24 April 2024

Academic Editor: Michael J. Schütze

Copyright © 2024 Abdou Khadir Dia et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The study of pipeline corrosion is crucial to prevent economic losses, environmental degradation, and worker safety. In this study, several machine learning methods such as recursive feature elimination (RFE), principal component analysis (PCA), gradient boosting method (GBM), support vector machine (SVM), random forest (RF), K-nearest neighbors (KNN), and multilayer perceptron (MLP) were used to estimate the thickness loss of a slurry pipeline subjected to erosion corrosion. These different machine learning models were applied to the raw data (the set of variables), to the variables selected by RFE, and to the variables selected by PCA (principal components), and a comparative analysis was carried out to find out the influence of the selection and transformation of the data on the performance of the models. The results show that the models perform better on the variables selected by RFE and that the best models are RF, SVM, and GBM with an average RMSE of 0.017. By modifying the hyperparameters, the SVM model becomes the best model with an RMSE of 0.011 and an R-squared of 0.83.

1. Introduction

Slurry pumping technology is a well-established and favored method for transporting mineral concentrates through pipelines. These pipelines can be made from a variety of materials, including carbon steel, alloy steel, hardened steel, stainless steel, abrasion-resistant lined pipes, nonferrous pipes, and HDPE, with the choice depending on the application, material being transported, and cost [1]. Despite the excellent safety record and favorable economics of long-distance slurry pipeline systems compared to traditional bulk transport systems, pipe abrasion and erosion loss remain a significant concern. While nonferrous pipelines can extend the life of the transport system for mineral concentrates, carbon steel pipes are prone to internal corrosion, especially when dealing with abrasive or corrosive slurries. The pipeline structure and materials are continually being improved for various industries. For example, HDPE is extensively used for applications such as mine tailings due to its ultrahigh molecular weight and resistance to abrasiveness, making it more durable than carbon steel pipes. In addition, nonferrous materials are used to line the inside of steel pipes to protect against erosion and corrosion, and low wear resistance nonferrous pipes such as polyurethane, polybutylene, PVC, PP, ABS, and fiberglass pipe with internal ceramic chips are also available for slurry transport.

Despite the use of carbon steel pipes, the high wear conditions caused by the large quantities and abrasive nature of slurry can result in leaks or ruptures, leading to significant maintenance costs in the mining industry due to erosion corrosion, especially in long pipelines spanning hundreds of kilometers. Other industries, such as the oil and gas sector, have also reported erosion corrosion as one of the top five forms of damage mechanisms, posing challenges to machinery and equipment with short lifecycles [2, 3]. Therefore, to mitigate these risks, it is crucial to implement pipeline integrity detection and monitoring, including an understanding of defect progression, condition-based maintenance, and lifecycle management [4].

Over the years, several nondestructive testing methods have emerged for inspecting pipelines while in use, including ultrasonic inspection (UT), which uses high-frequency

sound waves to identify defects on materials or their surfaces. UT is effective at detecting cracks, crevices, metal losses, and other discontinuities at varying depths within samples due to the reflection, diffraction, and transmission characteristics of ultrasonic sound [4, 5]. A bulk wave ultrasonic thickness measurement technique for corrosion monitoring can be used by temporarily or permanently coupling a transducer to the outer surface of a pipe, and the wall thickness of the pipe can be determined based on the time difference between transducer excitation and reception of the reflected wave from the back-wall surface. Traditional inspection and maintenance practices based solely on experience are no longer sufficient, and pipeline operators now require quantitatively risk-based methodologies. To reduce the economic impact of failures and minimize their impact on the environment, health, and safety, analytical tools have been developed over the years [6].

After the emergence of big data techniques, machine learning (ML) has demonstrated significant benefits in modeling and data mining [7]. ML has been utilized in various corrosion-related issues, such as the modeling of CO_2 corrosion [8], automated image analysis to detect corrosion [9], modeling of corrosion defect growth in pipelines [10], material inspection [11], predicting corrosion rates in marine environments [12], determining the initiation time of embedded steel corrosion in reinforced concrete [13], and predicting electrochemical impedance spectra [14]. In exploring the predictability of corrosion rates in reinforced concrete, Ji and Ye conducted a comprehensive study employing various machine learning algorithms. The study revealed that electrical resistivity emerged as the most significant factor influencing the corrosion rate. Among the algorithms tested, support vector regression showed the highest predictability for estimating corrosion rates [15]. The study by Fang et al. addresses the critical gap in publicly available corrosion data for pipelines. Utilizing the OLI Studio Corrosion Analyzer, a tool grounded in rigorous first principles, the research simulated thousands of corrosion scenarios for both crude oil and natural gas pipelines. This simulation produced a vast dataset, which was then analyzed using two machine learning algorithms: random forest (RF) and CatBoost [16]. A variety of machine learning techniques have been employed to predict the rate of corrosion or identify the areas most affected by corrosion [17, 18]. According to Zhang et al., they trained six machine learning models on ultrasonic testing data to forecast the degree of corrosion based on ultrasonic characteristics [19]. The findings suggest that, except for the linear model, machine learning models can accurately and robustly forecast the corrosion degree despite the interference of outlier amplitude and training set size. In their study, Velázquez et al. examine various statistical and probabilistic methods that have been utilized in the literature to investigate corrosion issues and their practical applications [20]. Meanwhile, Wei et al. utilize an artificial neural network to establish a relationship model between the corrosion potential of low alloy steel in Sanya seawater and its influencing factors, allowing them to visualize the impact of different alloy elements on corrosion potential [21]. To estimate the corrosion defect depth growth of aged pipelines, Ossai adopts a data-driven machine learning approach, relying on techniques such as principal component analysis (PCA), particle swarm optimization (PSO), feed-forward artificial neural network (FFANN), gradient boosting machine (GBM), random forest (RF), and deep neural network (DNN), to estimate the growth of corrosion defect depth in aged pipelines [10]. Roy et al. use the gradient boosting regressor to predict corrosion resistance in alloys with multiple principal elements [22], while Zhao et al. suggest using rough set and decision tree methods to analyze pipeline soil corrosion [23]. To model experimental data of time-varying corrosion rates in mild steel specimens when corrosion inhibitors are added to the system at varying concentrations and dose schedules, Aghaaminiha et al. perform regression with several ML algorithms, ultimately finding random forest to be the best option [24]. Peng et al. propose a new hybrid intelligent algorithm that combines SVR, PCA, and CPSO to predict the corrosion rate of multiphase flow pipelines, utilizing PCA to reduce data dimensionality and CPSO to optimize hyperfine parameters in SVR [25]. Cicceri et al. have made significant contributions to wastewater treatment in two key studies. The first study focuses on the challenges of managing wastewater in urban and industrial areas, especially fluctuating water flows. It proposes a smart system for efficient water purification, featuring real-time monitoring of water quality and flow rates using a cyberphysical system approach and data from an environmental Internet of Things platform, tested in Briatico, Italy [26]. The second study introduces the Smart Wastewater Intelligent Management System (SWIMS), which advances intelligent wastewater management. SWIMS monitors and controls water flows and quality, using deep learning for anomaly detection and decision-making. This enhances wastewater treatment efficiency and was also implemented in Briatico, Italy, demonstrating the effectiveness of advanced technology in wastewater management. These studies highlight the role of intelligent systems in sustainable water management [27]. In the field of indoor and environmental air quality monitoring, two significant contributions stand out. Cicceri et al. developed the Smart and Healthy Intelligent Room System (SHIRS), a low-cost system for indoor air quality (IAQ) monitoring using edge computing. SHIRS uses machine learning (ML) to analyze environmental data for human presence detection. The effectiveness of this approach has been proven experimentally, supporting the use of Cloud-IoT frameworks in smart environments [28].

The literature review above highlights the increasing use of machine learning methods in the field of corrosion, which is attributed to the emergence of software solutions that reduce the need for extensive mathematical and statistical knowledge. However, the risk of obtaining false-positive results in a "black box" automated process cannot be ignored. The purpose of this paper is not to present a complete machine learning model for predicting corrosion erosion degradation in a slurry pipeline but rather to describe the methodology in detail to provide a corrosion assessment using machine learning as a starting point for the corrosion community. Full research papers often struggle to explain

the essential aspects of machine learning tools suitable for a specific dataset, as much emphasis is placed on results and discussion. Therefore, this paper is aimed at explaining every step and parameter needed to draw robust and trustworthy predictive models, including the effect of feature engineering methods like recursive feature elimination and principal component analysis, as well as data transformation. The data used in this publication was obtained from the periodic nondestructive evaluation of a pipeline, and five machine learning models were applied and compared, including RF, KNN, SVM, MLP, and GBM, on both unprocessed and feature-engineered data. Hyperparameter optimization using the grid search method improved the model's results and made it more robust.

2. Materials and Methods

2.1. Ultrasonic Monitoring. The issue of erosion corrosion in pipelines is significant in slurry pumping systems. To prevent risks and monitor structural health, corrosion models are constructed by quantitatively estimating the degradation. Conventionally, this involves placing coupons made of the same pipe material inside the pipe and measuring the resulting weight loss after exposure to the environment for a specified period [29]. However, this method is intrusive, costly, and time-consuming due to manual intervention. Alternatively, ultrasonic technology offers a nonintrusive way to monitor corrosion. An array of eight ultrasonic transducers with a diameter of 10 mm and a frequency of 5 MHz, smart-PIMS distributed by Procon Systems, Canada, was used to monitor a section of a $12''$ diameter and $24''$ length pipeline in pulse-echo mode, as shown in Figure 1, for long-term monitoring since 2021.

2.2. Data Collection. Data for this study were obtained from Agnico Eagle Mine Goldex. These data are physical measurements from a pipeline that is used to transport residue (pulp/slurry) from the concentrator to the Manitou Residue Park site owned by the MERN (Ministère de l'Énergie et des Ressources Naturelles) in Val-d'Or, Quebec. The data for our study was collected using ultrasonic transducers (smart-PIMS). These sensors were strategically placed at eight different positions along the pipeline to ensure comprehensive coverage and accuracy in data acquisition. The sensors were used to measure various parameters indicative of pipeline integrity, including wall thickness, and process parameters such as pulp temperature, pH, and pressure. While thickness measurements were taken on a daily basis, process variables were collected every five minutes. Once collected, the data from the sensors underwent a series of preprocessing steps to make it suitable for analysis. Initial cleaning was conducted to remove any outliers or noise that could potentially skew our results. This included filtering out any readings that fell outside the expected range for the given pipeline conditions. The data was then normalized to ensure that all input features contributed equally to the analysis. This step is crucial for the effective training of machine learning models. Table 1 lists the different variables and their descriptive statistics.

2.3. Feature Selection. In this study, the process of feature selection involved choosing important features that contribute significantly to thickness loss. To achieve this, we employed the recursive feature elimination (RFE) technique which involves building a model on the entire set of predictors and assigning an importance score to each predictor. Less important predictors are then eliminated, and the model is rebuilt with the remaining predictors, and importance scores are computed again [30]. RFE is particularly useful for certain models like the random forest [31]. By eliminating redundant or less informative features, RFE helps in reducing the complexity of the model, thereby minimizing the risk of overfitting. RFE focuses the model's learning on the most relevant features for thickness loss prediction, potentially enhancing its accuracy and predictive power. With fewer variables, the model becomes more interpretable, making it easier to understand and explain the factors most critical to pipeline corrosion. Along with RFE, we also used principal component analysis (PCA) as a dimensionality reduction method to extract important information from the data and represent it as a set of new orthogonal variables called principal components [32]. These components are derived in such a way that the first few retain most of the variation present in the original dataset. Although PCA is not a variable selection method, it can be used to enhance model performance. By transforming the data, PCA can reveal underlying structures that might not be apparent in the original feature space. PCA helps in dealing with multicollinearity among features, which can be a challenge in machine learning models. We identified principal components that explained a significant proportion of the variance in the dataset and used them as explanatory variables to compare models with the initial data.

2.4. Machine Learning Models

2.4.1. K-Nearest Neighbors (KNN). The K-nearest neighbor method is a distance-based supervised learning method [33]. It is a method easily scalable and has few hyperparameters. It is used for classification and regression problems. The classification or prediction of a new value is based on the values of the nearest neighbors that are determined using a distance between those values. The k value represents the number of neighbors; if it is equal to 1, for a classification problem, the predicted class is the class of the nearest neighbor and for a regression problem, the predicted value is the value of the nearest neighbor. If k is greater than 1, the predicted value is the average of the values of k neighbors for a regression problem. One of the most important steps of the KNN algorithm is the determination of the neighbors which is done through a distance calculation. One of the most used distances which is used in this paper is the Euclidean distance. Given two samples $X = \{x_1, x_2, \cdots\cdots, x_n\}$ and $Y = \{y_1, y_2, \cdots\cdots\}$, the Euclidean distance is calculated as follows:

$$D(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \quad (1)$$

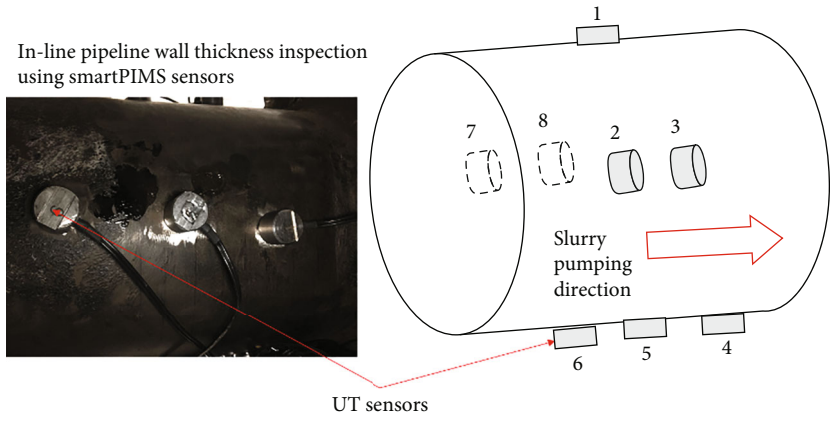

Figure 1: Ultrasonic transducers fixed at the pipe for continuous wall thickness monitoring.

Table 1: List of features.

	Minimum	Maximum	Mean	SD
Tonnage sag (t)	8.78	404.39	342.65	65.15
Flotation pulp temperature (°C)	13.92	34.92	23.07	4.68
Flotation pH	7.47	9.33	9.03	0.27
Residue flow (m^3/h)	18.54	495.97	437.59	90.22
% solid residue	0.02	46.13	23.54	10.17
TPH-calculated residue	0.02	301.95	141.31	68.20
Pressure at km 0 (Psi)	624.61	3439.54	2209.96	618.89
T° at km 0 (°C)	3.76	30.86	16.25	5.70
Flow rate (m^3/h) (Thompson River)	30.00	381.19	207.72	110.66
T° (Thompson River) (°C)	0.99	20.45	13.42	6.04
Flow rate (m^3/h) (sedimentation basin)	26.88	122.30	68.82	16.17
T° (sedimentation basin) (°C)	2.71	21.55	9.58	5.71
Flow rate (m^3/h) (South Park)	0.04	369.80	168.89	95.23
T° (South Park) (°C)	0.73	16.23	5.21	4.53
Pipe wall thickness (mm)	5.75	6.01	5.84	0.03

Determining the value of k is very important since a poor choice of k can lead to overfitting or underfitting. High deviations with low bias are often characterized by lower k values.

2.4.2. Random Forest (RF). Random forest is an ensemble learning method that consists of multiple decision trees. It was proposed by Ho in 1995 [34], and an extension was proposed by Breiman in 2001 [35]. It is an algorithm that can be used for both classification and regression problems and has been rapidly adopted because of its flexibility. For a classification problem, the predicted class is the class predicted by most trees. For a regression problem, the predicted value, represented in the following equation, is the average of the values predicted by the different trees.

$$\bar{h}(x) = \frac{1}{T}\sum_{t=1}^{T}\{h(x,\theta_t)\}. \quad (2)$$

The learning process of the random forest model starts with bagging by sampling a training dataset with replacement, also called bootstrap sampling, and k predictors for each tree. Then, a decision tree is trained on each sample. Finally, the prediction of the random forest model is the average of the values predicted by each tree.

2.4.3. Gradient Boosting Machine (GBM). Gradient boosting is a popular machine learning technique applied to classification tasks, known for its robustness and high performance compared to decision trees and random forest algorithms in certain cases. The method improves the accuracy of predictions by iteratively combining multiple "weak learners," which are simple models, to produce a "strong learner" with superior performance. On the other hand, the gradient boosting machine, developed by Friedman, was inspired by gradient boosting and is utilized for regression problems [36].

Let be a training sample $(x_i, y_i)\, i = 1, \cdots \cdots n$. We make the assumption that we have a set of base learners $\mathscr{B}$ and

our objective function can be expressed as a linear combination of these base learners, which is denoted $\mathrm{Lin}(\mathscr{B})$. The set of learners, $\mathscr{B}$, is defined as $B = \{b_\tau(x) \in \mathbb{R}\}$, where $\tau \in T$ represents the parameters of the learners. To predict the output for a given feature vector, x, we use an additive model represented by the following equation [37]:

$$f(x) \coloneqq \left(\sum_{m=1}^{M} \beta_m b_{\tau_m}(x) \right) \in \mathrm{lin}(\mathscr{B}), \tag{3}$$

where $b_{\tau_m}(x) \in \mathscr{B}$ is a weak learner and β_m is its corresponding additive coefficient.

The objective of GBM is to derive an accurate approximation of the function f that can effectively reduce the empirical loss [37]:

$$L^\star = \min_{f \in \mathrm{lin}(\mathscr{B})} \left\{ L(f) \coloneqq \sum_{i=1}^{n} \ell(y_i, f(x_i)) \right\}, \tag{4}$$

where $\ell(y_i, f(x_i))$ is a measure of the data fidelity for the i-th sample for the loss function ℓ.

The objective of the GBM method, as a numerical optimization algorithm, is to minimize the loss function by finding an additive model. The algorithm (as shown in Algorithm 1) initializes the model with a first estimation, typically a decision tree, and is aimed at minimizing the loss function. With each iteration, the algorithm calculates a model that best fits the residuals and adds it to the previous model to update the residuals. The algorithm stops after reaching the maximum number of iterations specified by the user.

2.4.4. Support Vector Machine. Support vector machine (SVM) is a supervised machine learning model used for regression and classification problems. It was first developed by Cortes and Vapnik [38, 39]. For regression problems, the name changes to support vector regression (SVR). The principle is almost the same as for classification problems except that for regression problems, the continuous variable must be predicted. The goal of the SVR algorithm is to find a hyperplane in an n-dimensional space that best fits the data. The hyperplane is the line that will help us predict the continuous value or the target value. The continuous function to be approximated can be written as in the following equation:

$$y = w \cdot x + b. \tag{5}$$

(1) SVR Linear. SVR formulates this function approximation problem as an optimization problem as presented in the following equations:

$$\min \frac{1}{2} \left\| w^2 \right\| + c \sum_{i=1}^{n} \xi_i + \xi_i^*, \tag{6}$$

subject to

$$\begin{cases} z_i - (w \cdot x + b) & \leq \varepsilon + \xi_i, \\ (w \cdot x + b) - z_i & \leq \varepsilon + \xi_i^*, \\ \xi_i \xi_i^* & \geq 0, \end{cases} \tag{7}$$

$$y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle x_i, x \rangle + b,$$

where $\|w^2\|$ indicates the size of the normal vector corresponding to the surface being approximated and the variables ξ_i and ξ_i^* are responsible for determining the allowable number of points outside the tube, while C acts as a regularization parameter that can be adjusted to give greater importance to minimizing either the error or the flatness of the solution in this problem involving multiple objectives. This information is cited from reference [40].

(2) Nonlinear SVR. In the nonlinear case (Figure 2), kernel functions are used to transform the data to allow for linear separation as in the following equations.

$$\min \frac{1}{2} \left\| w^2 \right\| + c \sum_{i=1}^{n} \xi_i + \xi_i^*, \tag{8}$$

subject to

$$\begin{gathered} y_i - w^T \varphi(x_i) \leq \varepsilon + \xi_i^* i = 1, \cdots, N, \\ w^T \varphi(x_i) - y_i \leq \varepsilon + \xi_i i = 1, \cdots, N, \\ \xi_i, \xi_i^* \geq 0 i = 1, \cdots, N, \\ w = \sum_{i=1}^{N_{SV}} (\alpha_i^* - \alpha_i) \varphi(x_i), \\ y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot \langle \varphi(x_i), \varphi(x) \rangle + b, \\ y = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) \cdot K(x_i, x) + b, \end{gathered} \tag{9}$$

where $K(x_i, x)$ is the kernel function.

2.4.5. Multilayer Perceptron. The MLP, a form of artificial neural network, is structured with multiple layers and operates through direct propagation from the input to output layer. The number of neurons in each layer varies, with the last layer being designated as the output layer. In the multilayer backpropagation perceptron, adjacent layers are interconnected, with the strength of these connections being determined by a coefficient that influences the destination neuron's response. The backpropagation algorithm is used to calculate these coefficients, which are essential to the network's functionality. Figure 3 shows a multilayer network. Each neuron i receives a series of signals from neurons j

Initialization: Initialize with $f^0(x) = 0$
For $m = 0, \cdots, M-1$ do:
Perform Updates:
(1) Compute pseudo residual: $r^m = -[\partial \ell(y_1, f^m(x_i))/\partial f^m(x_i)]_{i=1,\cdots,n}$.
(2) Find the parameters of the best weak-learner: $\tau_m = \arg\ \min_{\tau \epsilon \mathcal{T}} \sum_{i=1}^{n} (r_i^m - b_\tau(x_i))^2$.
(3) Choose the step-size η_m by line-search: $\eta_m = \arg\ \min_\eta \sum_{i=1}^{n} \ell(y_i, f^m(x_i) + \eta b_{\tau m}(x_i))$.
(4) Update the model $f^{m+1}(x) = f^m(x) + \eta_m b_{\tau m}(x)$.
Output. $f^M(x)$.

Algorithm 1: Algorithm gradient boosting machine (GBM) [37].

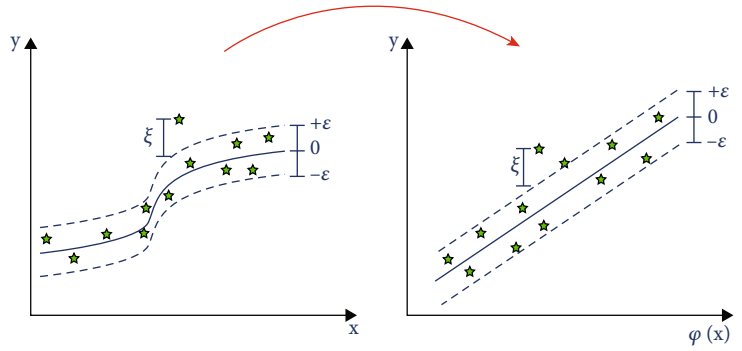

Figure 2: Nonlinear SVR.

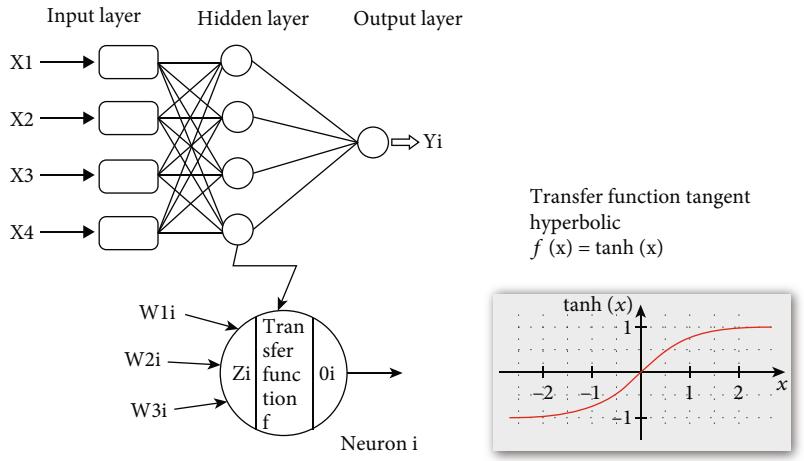

Figure 3: Multilayer perceptron [41].

located at the previous layers. The operation of the illustrated network is governed by the following equation [41].

$$Z_i = \sum_{j=1}^{n_j} W_{ij} X_j + b_i, \tag{10}$$

where n_j is the number of input neurons and X_j is the value of the signal transmitted by neuron j of the previous layer. The W_{ij} represent the respective weights of the connections between the neurons j of the previous layers and the neuron i of the current layer. The parameters b_i are bias values allowing a nonzero transfer function at the origin. The inputs X_j are weighted by the weights W_{ij}. Once the input is provided, neuron i transforms it and produces an output. In this case, Z_i and the output O_i, of a given neuron, are related by a transfer function of hyperbolic tangent form.

$$O_i = f(Z_i) = \frac{1 - e^{-2Z_i}}{1 + e^{-2Z_i}}. \tag{11}$$

The error made by the network at the output is calculated and then minimized. This is referred to as the error backpropagation method. The weights of the network W_{ij} are corrected to reduce the overall error $\underline{E}$. The gradient descent method is used to minimize the global error. It is represented by the following equation:

$$\underline{E} = \frac{1}{2}\sum_{k=1}^{n_k} (S_k - O_k)^2, \tag{12}$$

where S_k represents the estimated value, O_k is the observed value, and $\underline{E}$ is the overall error.

The steps of the error backpropagation algorithm are the following:

(1) Presentation of a training pattern to the network

(2) Comparison of the network output with the target output

(3) Compute the output error of each neuron in the network

(4) Compute, for each neuron, the output value that would have been correct

(5) Definition of the increase or decrease necessary to obtain this value (local error)

(6) Adjustment of the weight of each connection towards the lowest local error

(7) Assigning a blame to all previous neurons

(8) Repeat from step 4, on the previous neurons using the blame as error

3. Results and Discussion

3.1. Models with All Features. The prediction of thickness loss has been studied several times in the literature using different types of features and models. The features most often found in the literature are the chemical characteristics of the pipeline such as CO_2 partial pressure, corrosion inhibitor type [24], sulfate ion concentration, and chloride ion concentration [10]. In this study, other types of variables directly related to the pipeline (pH, residue flow, pressure, and calculated residual TPH) and variables external to the pipeline such as flow rate and temperature of the rivers and sedimentation basin that feed the pipeline were collected. Different set of models with all the variables were performed on the training data. Both the coefficient of determination (R^2) and root-mean-square error (RMSE) were employed for the model's predictive performance evaluation. They were defined by the following equations [7]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \overline{y_i})^2},$$
$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}, \tag{13}$$

where y_i, $\widehat{y_i}$, and $\overline{y_i}$ represent the measured value, the predicted value, and the average value of the corrosion rate, respectively. During the training process, a cross-validation technique (i.e., 5-fold repeated cross-validation method) was utilized to avoid random errors as much as possible [42]. The models were trained on 80% of the data, and the rest (20%) was used for validation. We utilized the R programming language for our machine learning implementations. Within R, we employed the "caret" package, which was instrumental in developing and tuning our gradient boosting machine (GBM), random forest (RF), support vector machine (SVM), and K-nearest neighbors (KNN) models. For the multilayer perceptron (MLP) model, we used the "neuralnet" package. These tools were chosen for their robustness and versatility in handling various machine learning tasks.

The results presented in Table 2 show slight differences in RMSE for the RF (0.017), GBM (0.018), and SVM (0.018) models. The KNN and MLP models perform less well with an RMSE of 0.02. The R^2 of the MLP model is low compared to the other models; i.e., the explanatory variables explain less the variation of the thickness loss. Table 3 shows pairwise statistical significance scores. The table's lower diagonal displays p values for the null hypothesis, indicating that the distributions are the same. Conversely, the upper diagonal shows the estimated difference between the distributions. It is evident from the table that there is no discernible difference between RF and GBM, and the differences between the distributions for RF, SVM, and KNN are minimal. The above results are those obtained with the training data. We will apply the validation data at the end when we have found the best model.

3.2. Models with Feature Selection. To enhance the model's performance, feature engineering is a crucial stage in modeling that involves selecting the most significant variables using various methods. According to [7] research, the gradient boosting decision tree (GBDT) method and Kendall correlation analysis were employed as feature engineering approaches.

During the second stage of thickness loss data modeling, we employed the recursive feature elimination technique to enhance the model's effectiveness. This approach involves fitting a model and removing the weakest feature or features until the designated number of features is achieved. The model implemented in this process is RF, which has a reliable built-in feature importance calculation mechanism. The purpose of this method is to eliminate any dependencies and collinearity that could potentially exist in the model. Figure 4 illustrates the change in RMSE concerning the number of selected variables in our model. The optimal number of variables is 10: tonnage sag, pulp temperature

Table 2: Result of the models with all variables.

Models	Training set	
	RMSE	R^2
SVM	0.018	0.667
GBM	0.018	0.672
RF	0.017	0.715
KNN	0.021	0.504
MLP	0.026	0.182

Table 3: Pairwise statistical significance scores.

	SVM	KNN	RF	GBM
SVM		-0.003	0.001	0.001
KNN	0.02		0.004	0.004
RF	0.02	0.00		0.00
GBM	1.00	0.01	1.00	

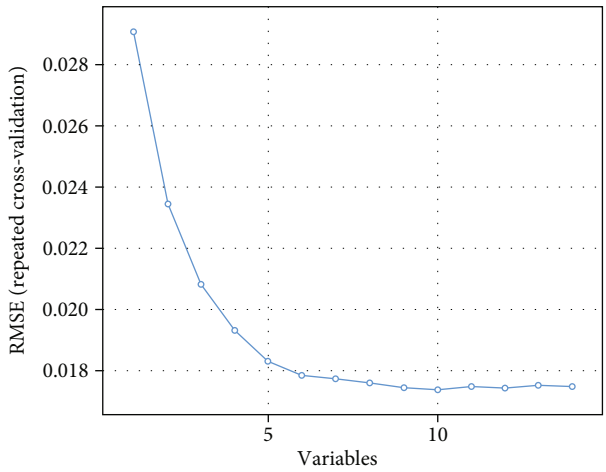

Figure 4: Recursive feature elimination.

Table 4: Result models with feature selection.

Models	Training set	
	RMSE	R^2
SVM	0.016	0.735
GBM	0.017	0.707
RF	0.017	0.725
KNN	0.027	0.274
MLP	0.027	0.074

flotation, % solid residue, TPH-calculated residue, pressure at km 0, T° at km 0, T° (Thompson River), Flow rate (sedimentation basin), T° (sedimentation basin), flow rate (South Park), and T° (South Park) with an RMSE of 0.017. These 10 variables were used to study the other models. The results (Table 4) show little variation in the performance metrics. The results of the significance test for the RF, SVM, and GBM models show that there is no difference between these models. This result is consistent with the findings of [16],

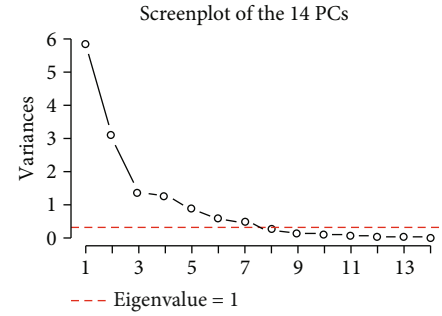

Figure 5: Screenplot of 14 PCs.

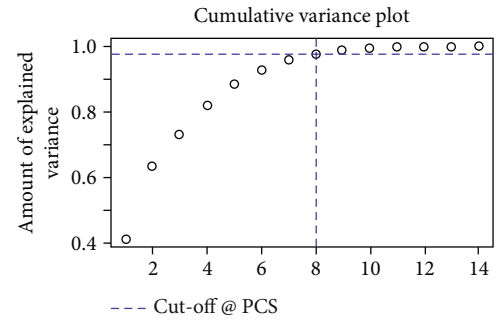

Figure 6: Cumulative variance plot.

who also reported the effectiveness of RF in similar contexts. However, these models are better than KNN and MLP. MLP work better in situations where the sample size is very large which is not the case for our study.

3.3. Models With Feature Selection by PCA. Principal component analysis (PCA) is sometimes used as a method of feature engineering using explanatory variables as the principal components that explain the greatest variation. It is a method that sometimes gives excellent results [10]. We performed PCA on our training data and selected the 8 principal components (PC) (Figures 5 and 6) that account for more than 95% of the variation in the data. These 8 principal components are then used as explanatory variables in our different models. The results (Table 5) vary slightly from those found previously. However, the KNN model performs better with the PCA transformation of the data.

3.4. Tuning Hyperparameters. Hyperparameter optimization [43, 44] or tuning involves selecting an optimal set of hyperparameters for a learning algorithm that maximizes the model's performance and minimizes a predefined loss function to produce accurate results with fewer errors. Grid search, also known as parameter sweep, has traditionally been the preferred method for hyperparameter optimization, which involves an exhaustive search through a manually specified subset of the algorithm's hyperparameter space. Grid search is guided by a performance metric, which is typically measured by cross-validation on the training set [44, 45]. For the SVM model, we aim to identify the optimal values for C and gamma. C represents the cost of constraint

Table 5: Result models with feature selection by PCA.

Models	Training set	
	RMSE	R^2
SVM	0.018	0.66
GBM	0.02	0.51
RF	0.019	0.61
KNN	0.019	0.55

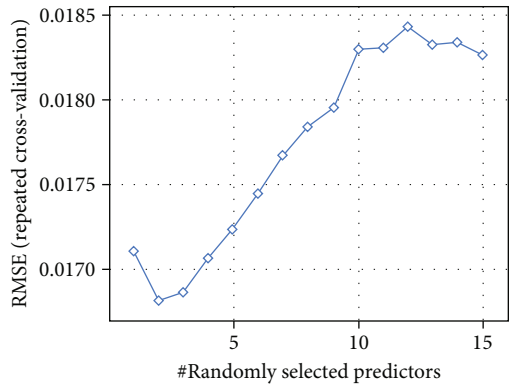

Figure 7: Hyperparameter tuning—random forest.

violation and is the regularization term constant in the Lagrange formulation. A low value may cause the model to incorrectly classify some training data, while a high value may lead to overfitting, which results in an analysis that is too specific for the current dataset and may not be suitable for future data. Gamma is the inverse of the influence radius of data samples chosen as support vectors. High values indicate a small radius of influence and small decision boundaries that do not consider relatively close data samples, leading to overfitting. Low values indicate a significant impact of distant data samples, causing the model to fail to capture the correct decision boundaries from the dataset. For the RF model, we aim to find the optimal values of max_features and n_estimators. Max_features represents the maximum number of features that random forest can attempt in an individual tree, while n_estimators refers to the number of trees built before taking the maximum voting or averages of predictions. The best hyperparameters for the RF model are max_features = 2 and n_estimators = 2500, resulting in an RMSE of 0.016 and an R^2 of 0.73. Figure 7 illustrates the variation of RMSE as a function of max_features with n_estimators = 2500.

Hyperparameter optimization of the SVM model produces more interesting results. Figure 8 shows the variation of the RMSE according to the hyperparameters. We can clearly see that the smallest RMSE (better performing model) is at the point cost = 5 and sigma = 0.05. With these hyperparameters, the RMSE is estimated at 0.015 and the R^2 at 0.76. These estimates are made on the training data.

The SVM model appears to be the best model with a lower RMSE than found in most of the review. To investigate the performance of the model in new data, we use the validation data (20% of the dataset) to predict thickness

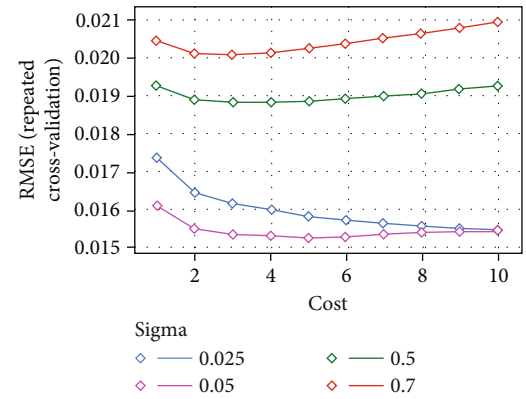

Figure 8: Hyperparameter tuning—SVM.

Table 6: Result on validation set.

Model	Validation set	
	RMSE	R^2
SVM	0.011	0.83

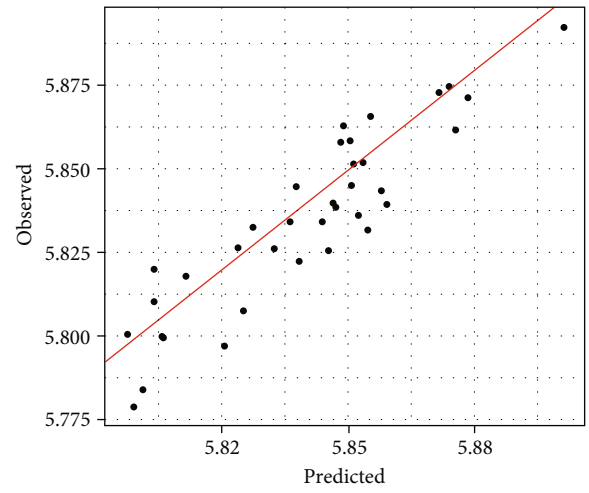

Figure 9: Pipeline wall thickness values: observed vs. predicted.

losses and estimate the RMSE. Table 6 shows the estimation results of the SVM model on the validation data. The model predicts the thickness loss well with a low RMSE evaluated at 0.011 and R^2 of 0.83. We can see in Figure 9 the small difference between the observed and predicted values of the thickness measurements by running the model on new data (validation data) with the best model (SVM) and the optimal hyperparameters $C = 5$ and sigma = 0.05.

Our results indicated that models such as RF, SVM, and GBM outperformed others, showing lower RMSE values, particularly after the application of RFE. The superior performance of these models can be attributed to several factors. Both RF and GBM are known for their ability to handle nonlinear relationships within data. Given the complex nature of erosion corrosion processes in pipelines, which often involve nonlinear interactions between various

factors, these models could capture these complexities more effectively than linear models. The application of RFE significantly improved model performance. This is likely because RFE helped in removing redundant or less significant features, allowing the models to focus on the most relevant predictors of thickness loss in the pipeline. SVM, after hyperparameter tuning, emerged as the best model. This improvement is likely due to SVM's flexibility in defining the margin of separation and its ability to handle high-dimensional spaces effectively, especially after feature selection.

While our study specifically addresses erosion corrosion in sewage pipelines, the application of machine learning models like SVM, RF, and GBM may yield different outcomes when considering other types of pipelines, such as oil, gas, or water supply lines. These differences can arise due to several factors as material composition, operational conditions, corrosive agents, and environmental factors. Given these variations, our machine learning models may require adjustments or retraining with relevant data from the specific pipeline type and corrosion conditions under study.

4. Conclusions

In the current situation, the majority of the internal pipeline wall is deteriorating due to both slurry erosion and corrosion, which result in the gradual removal of material from the surface due to the impact of solid particles suspended in the liquid phase. As stated earlier, the intent of this paper was not to present the full machine learning model for predicting corrosion erosion degradation in a slurry pipeline, which will be done in a subsequent publication, but rather to take a unique opportunity to describe the methodology in detail as a walk-through corrosion assessment of slurry pipeline using machine learning. Explaining the essential aspects of machine learning tools suitable for a specific dataset is often difficult in a full research paper as much focus needs to be on results and discussion. While the study combined several machine learning techniques to achieve better results, it was found that the SVM, RF, and GBM models perform better on the initial data. On the other hand, the KNN model performs better on the principal component data. The change in hyperparameters was important in this analysis, as the SVM model went from an RMSE of 0.016 to 0.011, remaining the best model for predicting pipeline thickness loss. These results are pivotal for professionals in pipeline corrosion management, offering actionable insights for planning risk-based inspection and corrosion mitigation strategies. However, we acknowledge a limitation in our research concerning the volume of data used. Future studies will aim to address this by collecting extensive datasets from various pipeline locations, thereby enabling more precise predictions across different scenarios.

Data Availability

Restrictions apply to the availability of these data. Data was obtained from Agnico Eagle Mine Goldex and are available from the authors with the permission of Agnico Eagle Mine Goldex.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

The authors are thankful of Agnico Eagle Goldex Mine and the Quebec Metallurgy Centre for their support. This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Québec Fonds de Recherche Nature et Technologies (FRQNT).

References

[1] EDDY Pump Corporation, "Slurry pipeline changes: what effect can it have on your operation?," 2023, https://eddypump.com/.

[2] R. J. Chung, J. Jiang, C. Pang, B. Yu, R. Eadie, and D. Y. Li, "Erosion-corrosion behaviour of steels used in slurry pipelines," *Wear*, vol. 477, article 203771, 2021.

[3] M. Jones and R. J. Llewellyn, "Erosion-corrosion assessment of materials for use in the resources industry," *Wear*, vol. 267, no. 11, pp. 2003–2009, 2009.

[4] Q. Ma, G. Tian, Y. Zeng et al., "Pipeline in-line inspection method, instrumentation and data management," *Sensors*, vol. 21, no. 11, 2021.

[5] T. D'Orazio, M. Leo, A. Distante, C. Guaragnella, V. Pianese, and G. Cavaccini, "Automatic ultrasonic inspection for internal defect detection in composite materials," *NDT and E International*, vol. 41, no. 2, pp. 145–154, 2008.

[6] S. P. Trasatti, "Risk-Based Inspection and Integrity Management of Pipeline Systems," in *Degradation Assessment and Failure Prevention of Pipeline Systems*, G. Bolzon, G. Gabetta, and H. Nykyforchyn, Eds., vol. 102 of Lecture Notes in Civil Engineering, Springer, Cham, 2021.

[7] Y. Diao, L. Yan, and K. Gao, "Improvement of the machine learning-based corrosion rate prediction model through the optimization of input features," *Materials & Design*, vol. 198, article 109326, 2021.

[8] S. Nesic, M. Nordsveen, N. Maxwell, and M. Vrhovac, "Probabilistic modelling of CO_2 corrosion laboratory data using neural networks," *Corrosion Science*, vol. 43, no. 7, pp. 1373–1392, 2001.

[9] W. T. Nash, C. J. Powell, T. Drummond, and N. Birbilis, "Automated corrosion detection using crowdsourced training for deep learning," *Corrosion*, vol. 76, no. 2, pp. 135–141, 2020.

[10] C. I. Ossai, "A data-driven machine learning approach for corrosion risk assessment—a comparative study," *Big Data and Cognitive Computing*, vol. 3, no. 2, p. 28, 2019.

[11] G. Sanchez, W. Aperador, and A. Cerón, "Corrosion grade classification: a machine learning approach," *Indian Chemical Engineer*, vol. 62, no. 3, pp. 277–286, 2020.

[12] L. Yan, Y. Diao, Z. Lang, and K. Gao, "Corrosion rate prediction and influencing factors evaluation of low-alloy steels in marine atmosphere using machine learning approach," *Science and technology of Advanced Materials*, vol. 21, no. 1, pp. 359–370, 2020.

[13] B. A. Salami, S. M. Rahman, T. A. Oyehan, M. Maslehuddin, and S. U. Al Dulaijan, "Ensemble machine learning model

for corrosion initiation time estimation of embedded steel reinforced self-compacting concrete," *Measurement*, vol. 165, article 108141, 2020.

[14] X. Gong, C. Dong, J. Xu, L. Wang, and X. Li, "Machine learning assistance for electrochemical curve simulation of corrosion and its application," *Materials and Corrosion*, vol. 71, no. 3, pp. 474–484, 2020.

[15] H. Ji and H. Ye, "Machine learning prediction of corrosion rate of steel in carbonated cementitious mortars," *Cement and Concrete Composites*, vol. 143, article 105256, 2023.

[16] J. Fang, X. Cheng, H. Gai, S. Lin, and H. Lou, "Development of machine learning algorithms for predicting internal corrosion of crude oil and natural gas pipelines," *Computers & Chemical Engineering*, vol. 177, article 108358, 2023.

[17] A. K. Dia, N. Ghazzali, and A. G. Bosca, "Unsupervised neural network for data-driven corrosion detection of a mining pipeline," in *The 35th International FLAIRS Conference*, Florida, May 2022.

[18] D. I. A. Abdou Khadir, *Modèles prédictifs par apprentissage automatique pour la classification et la régression : application en science des données*, Université du Québec à Trois Rivières, Trois Rivières, 2023.

[19] J. Zhang, M. Zhang, B. Dong, and H. Ma, "Quantitative evaluation of steel corrosion induced deterioration in rubber concrete by integrating ultrasonic testing, machine learning and mesoscale simulation," *Cement and Concrete Composites*, vol. 128, article 104426, 2022.

[20] J. C. Velázquez, E. Hernández-Sánchez, G. Terán, S. Capula-Colindres, M. Diaz-Cruz, and A. Cervantes-Tobón, "Probabilistic and statistical techniques to study the impact of localized corrosion defects in oil and gas pipelines: a review," *Metals*, vol. 12, no. 4, p. 576, 2022.

[21] X. Wei, D. Fu, M. Chen, W. Wu, D. Wu, and C. Liu, "Data mining to effect of key alloying elements on corrosion resistance of low alloy steels in Sanya seawater environmentAlloying elements," *Journal of Materials Science & Technology*, vol. 64, pp. 222–232, 2021.

[22] A. Roy, M. F. N. Taufique, H. Khakurel, R. Devanathan, D. D. Johnson, and G. Balasubramanian, "Machine-learning-guided descriptor selection for predicting corrosion resistance in multi-principal element alloys," *npj Materials Degradation*, vol. 6, no. 1, p. 9, 2022.

[23] Z. Zhao, M. Chen, H. Fan, and N. Zhang, "Data analysis and knowledge mining of machine learning in soil corrosion factors of the pipeline safety," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 9523878, 9 pages, 2022.

[24] M. Aghaaminiha, R. Mehrani, M. Colahan et al., "Machine learning modeling of time-dependent corrosion rates of carbon steel in presence of corrosion inhibitors," *Corrosion Science*, vol. 193, article 109904, 2021.

[25] S. Peng, Z. Zhang, E. Liu, W. Liu, and W. Qiao, "A new hybrid algorithm model for prediction of internal corrosion rate of multiphase pipeline," *Journal of Natural Gas Science and Engineering*, vol. 85, article 103716, 2021.

[26] G. Cicceri, R. Maisano, N. Morey, and S. Distefano, "A Novel Architecture for the Smart Management of Wastewater Treatment Plants," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 392–394, Irvine, CA, USA, 2021.

[27] G. Cicceri, R. Maisano, N. Morey, and S. Distefano, "SWIMS: the Smart Wastewater Intelligent Management System," in *2021 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 228–233, Irvine, CA, USA, 2021.

[28] G. Cicceri, C. Scaffidi, Z. Benomar et al., "Smart Healthy Intelligent Room: Headcount through Air Quality Monitoring," in *2020 IEEE International Conference on Smart Computing (SMARTCOMP)*, pp. 320–325, Bologna, Italy, 2020.

[29] J. C. Adamowski, F. Buiochi, M. Tsuzuki, N. Pérez, C. S. Camerini, and C. Patusco, "Ultrasonic measurement of micrometric wall-thickness loss due to corrosion inside pipes," in *2013 IEEE International Ultrasonics Symposium (IUS)*, pp. 1881–1884, Prague, Czech Republic, 2013.

[30] M. Kuhn and K. Johnson, *Feature engineering and selection: a practical approach for predictive models*, Chapman and Hall/CRC, 2019.

[31] V. Svetnik, A. Liaw, C. Tong, J. Christopher Culberson, R. P. Sheridan, and B. P. Feuston, "Random forest: a classification and regression tool for compound classification and QSAR modeling," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1947–1958, 2003.

[32] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[33] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[34] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282, Montreal, QC, Canada, 1995.

[35] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[36] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, 2001.

[37] H. Lu, S. Karimireddy, N. Ponomareva, and V. Mirrokni, "Accelerating gradient boosting machine," in *Proceedings of Machine Learning Research*, 2019.

[38] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[39] C. Cortes and V. Vapnik, "Support-vector networks," *Chemical Biology & Drug Design*, vol. 297, pp. 273–297, 2009.

[40] M. Awad and R. Khanna, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Springer Nature, 2015.

[41] M. Bélanger, N. El-Jabi, D. Caissie, F. Ashkar, and J. M. Ribi, "Estimation de la température de l'eau de rivière en utilisant les réseaux de neurones et la régression linéaire multiple," *Revue des Sciences de l'Eau*, vol. 18, no. 3, pp. 403–421, 2005.

[42] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 2, pp. 111–133, 1974.

[43] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," in *MIC 2015: The XI Metaheuristics International Conference*, 2015.

[44] M. Feurer and F. Hutter, "Hyperparameter optimization," in *Automated Machine Learning*, The Springer Series on Challenges in Machine Learning, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Springer, Cham, 2019.

[45] F. Tang, Y. Wu, and Y. Zhou, "Hybridizing grid search and support vector regression to predict the compressive strength of fly ash concrete," *Advances in Civil Engineering*, vol. 2022, Article ID 3601914, 12 pages, 2022.