Received: 30 March 2022, Accepted: 10 May 2022, Published: 31 May 2022, Publisher: UTP Press, Creative Commons: CC BY 4.0

APPLICATION AND COMPARISON OF ENSEMBLE MODELS TO PREDICT CORROSION RATE FOR AMINE REGENERATION UNIT IN REFINERY

Sathiyamurthi a/l Raja Moorthi[1]*, Izzatdin A. Aziz[2]

[1]Asset Integrity Management Department, Engineering Division, Pengerang Refining Company Sdn. Bhd., Pengerang, Malaysia
[2]Center for Research in Data Science (CERDAS), University Teknologi PETRONAS, Malaysia

*Email: sathiyamurthi.raja@petronas.com

ABSTRACT

Monitoring the corrosion activity of Amine Regeneration Units (ARU) in oil and gas refineries is crucial to avoid unexpected failures and unplanned shutdowns. Various corrosion monitoring devices are installed to provide real time metal loss readings translated into corrosion rates. However, existing corrosion monitoring devices do not offer prediction capabilities, and a huge amount of supervised historical data is usually ignored due to its complex characteristics and incompleteness. By utilising ensemble models, machine learning can be adopted to generate value from the unattended historical data and provide prediction capabilities. This research work is mainly intended to process the historical data in Amine Regeneration Unit (ARU), implement Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM) ensemble models for corrosion rate prediction, and compare the two to identify the most accurate model. An experiment was conducted via simulation through programming software that utilises Python as the programming language to process data from an undisclosed oil and gas refinery in Malaysia. Evaluation of accuracy was performed using statistical methods, which include R-squared (R^2), Mean Average Error (MAE), Mean Squared Error (MSE), and Root-Mean-Square Error (RMSE). Upon model hyperparameter tuning and comparison between eight different models, the model utilising the XGBoost algorithm with K-Fold cross-validation was reported to provide the best accuracy in terms of R^2 score at 65.4%. Future improvement for this research work includes exploration of various other corrosion groups in ARU among multiple oil and gas refineries and experimentation on various boosting tree hyperparameter tuning.

Keywords: *Amine, corrosion rate, LightGBM, oil and gas, predictive analytics, refinery, XGBoost*

INTRODUCTION

The most common corrosion issues that threaten the integrity of piping and vessels in Amine Regeneration Units (ARU) are lean amine corrosion, rich amine corrosion, and ammonium bisulfide corrosion. All three corrosion mechanisms present thinning damage appearance and cause carbon steel pipes to experience thickness loss over time. Cracking damages through amine stress corrosion cracking that will not be emphasised in this research work. Our primary focus will be on rich amine corrosion's thinning damage mechanism.

The Amine Regeneration Unit (ARU) corrosion mechanisms are summarised into groups called the Corrosion Groups, as depicted in A.P. Institute [1]. Each Corrosion Group contains independent variables, corrosion-related parameters, and dependent variables, which are the corrosion rates generated by an ultrasonic thickness monitoring device called Permasense or Corrosion Probes. Permasense sensors produce current corrosion rate values based on historical wall thickness monitoring. Corrosion-related parameters include historical temperature, pressure, pH, velocities, data, etc., that influence the wall thickness loss and corrosion rates.

Alternatively, corrosion rates can be calculated using formulas or estimated based on available international standards such as American Petroleum Institute, API RP 571 and API RP 581standards. For example, the corrosion rate for rich amine corrosion can be estimated from a table containing estimated corrosion rates based on temperature, velocity, heat stable amine salts (HSAS), and acid gas loading values [2].

It is crucial to monitor the corrosion rates on steel pipes in the refinery as part of the quality control and quality assurance activity. Several existing corrosion monitoring devices such as Permasense, corrosion probes, and corrosion coupons measure current corrosion rates, but these readings often differ from each other resulting in ambiguity. Besides, large ARU datasets' unpredictable nature and vagueness make it difficult to understand and troubleshoot the extent of corrosion activity. Moreover, these monitoring tools are never designed to predict future corrosion rates. This is where machine learning plays an important role.

Machine learning is capable of learning based on new input data and provides more accurate and representative outputs. Continuous improvement allows prediction capabilities to mature and reflect actual conditions. Various trends and patterns from a huge amount of data are analysed to form the prediction. These analyses cannot be performed using Microsoft Excel alone due to its limitations, such as the inability to handle file sizes beyond a maximum of 2 gigabytes (GB), limited characters in tables or columns, and slow processing [3]. In fact, Microsoft Excel although contains some machine learning algorithms such as a linear regression model, which is not suitable for being a database, performing database manipulations, or handling huge datasets at speed.

The objectives of this research work are to perform data cleaning and pre-process historical data in Amine Regeneration Unit (ARU) and compare Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM) ensemble models for accurate corrosion rate prediction. The dataset from ARU is expected to be incomplete, inconsistent, and noisy. Applying data analytics through machine learning processes allows re-engineering these data to provide more representative, accurate, faster, and scalable results. The ensemble models, which are supervised learning algorithms, will best fit the continuous numerical datasets in ARU that contain a set of variable input parameters linking to defined output labels. Boosting algorithms such as Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM) will best suit the regression-based relationship of the huge datasets since they are capable of handling uncertainties as well. Besides, ARU dataset features are numerical and not categorical, making the above gradient boosting tree models suitable. They also support missing or sparse data, which is very common since downtime in the plant often leads to missing values. Another key feature of XGBoost and LightGBM is their ability to perform fast processing and analysis on huge sets of observations in training data. These attributes justify the selection of XGBoost and LightGBM as two of the most suitable ensemble models for comparison to represent the described characteristics of ARU datasets.

With machine learning, engineers will be able to predict and evaluate future corrosion rates and map the expected life of the asset against its design life. The valuable insight will also enable engineers to foresee failures or possible downtime that could disrupt the business and induce loss to the company. Engineers will be able to confidently perform informed decision-making supported by the predictive nature of machine learning. At the end of this research work, the most accurate boosting algorithm will be selected to represent the described characteristics of ARU datasets best.

LITERATURE REVIEW

Corrosion Issues in Amine Regeneration Unit (ARU)

Amine treating or regeneration units involve various corrosion mechanisms, especially if the structure, which includes piping and vessels, is constructed using carbon steel material. Corrosion is promoted by dissolved acid gases, high temperatures, and high concentration of corrosive species, i.e., HSAS (Heat Stable Amine Salt) [4]. The general appearance of the corrosion mechanism indicates an acidic attack on carbon steel internal walls, which may be controlled through proper process control or upgrading material to austenitic stainless steel.

Predicting the corrosion rates in advance will allow proactive management of process parameters to avoid catastrophic failures and avoid the need to spend extra on superior materials. According to Hatcher et al. [5], predicting corrosion rates pertinent to the particular processing condition is a must to prevent equipment failures and mitigate risks. Commonly, these predictions experimentally to produce a model for reference based on a set of process values. Lagad et al. [6] in their study to develop a software tool to predict rich amine corrosion rates for three amine solvents, discovered that quantitative data availability is low, hindering the ability to define guidelines for corrosion control in amine systems properly. The available historical data are most often noisy and of less value. Data analytics and machine learning may provide a solution to the issue.

Data Characteristics in ARU

Datasets in ARU are huge and inconsistent in nature. This is due to frequent operation downtime and transmitter errors. As such, many missing values exist that may hinder accurate evaluation. Selecting a machine learning algorithm to address inconsistent datasets is challenging. In a study by Yan Liu et al. [7], machine-learning techniques were applied to handle datasets with missing values for warfarin dose predictions. Light Gradient Boosting Machine (LightGBM) was implemented to learn from incomplete data directly on the selected single and cross-over variables for dosing prediction. It was found that LightGBM, being a boosting type of ensemble model, is very capable of exploring the variable interactions and learning from incomplete data. Ensemble learning is also proven to improve classification accuracy when dealing with missing values which are a common issue in many real-world datasets [8].

Machine Learning – Ensemble Model and Boosting Techniques

Various types of ensemble techniques, such as Adaptive Boosting (AdaBoost), Catboost, LightGBM, Extreme Gradient Boosting (XGBoost), and Gradient Boosting, were compared and evaluated [9]. There is a gap in terms of addressing the most accurate corrosion rate predictive algorithm for the vague and inconsistent nature of the ARU dataset. This research work will address the gaps accordingly and will focus on two main ensemble algorithms: Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM). These two algorithms were selected based on ARU dataset characteristics that are relational-based, numerical, non-categorical, continuous, huge in quantity, and inconsistent. The basis of selection is tabulated in Table 1.

Based on Table 1, XGBoost and LightGBM are selected for comparison in this research work. XGBoost is proven accurate and suitable for numerical datasets, whereas LightGBM is good at handling overfitting issues and verified to be the fastest boosting algorithm through literature reviews, compared to other boosting algorithms such as gradient boosting.

Table 1 Basis of selection of XGBoost and LightGBM

Boosting Techniques	Characteristics	Gap Analysis
AdaBoost	• It uses multiple weak models and assigns higher weights to those observations for which misclassification was observed [9] • Prediction accuracy of Gradient Boosting algorithm is better than traditional machine learning and AdaBoost algorithm – a study to classify patients with diabetes using diabetes risk factors [10]. • AdaBoost-DT model for predicting CO2 loading capacity of MEA, DEA, TEA, and MDEA aqueous solutions performs better than ANN and LSSVM models [11].	• Accuracy is lower than other booting techniques, especially Gradient Boosting.
XGBoost	• Not suitable for categorical features as they must be converted into numerical features during pre-processing [9] • Slower compared to LightGBM and CatBoost [9] • Data scientists widely use this technique to attain state-of-the-art results on various machine learning challenges [12]	• Has been used for corrosion-related research and has proven to be accurate.

Cont. **Table 1**

	• An optimal algorithm for corrosion failure risk prediction of pipelines [13]. • XGBoost was used to generate pseudo density logs for CSG wells in Surat Basin, Australia, and it generated good results and stable performance [14]. • In terms of hyperparameter tuning for XGBoost, Grid Search is said to be very popular, proven in a study that applied Artificial Neural Network (ANN) and XGBoost to improve data analytics in the oil and gas industry [15].	• Suitable for numerical datasets. • Proven in providing state-of-the-art results.
LightGBM	• More efficient compared to XGBoost since gradient-based one-side sampling (GOSS) is less time consuming compared to the histogram-based algorithm • Overfitting issues can be controlled by LightGBM [16]. • Research on Water Levels prediction of the Lower Columbia River revealed that the LightGBM model could achieve high prediction accuracy with the root-mean-square-error values [17].	• Fastest boosting algorithm compared to others • More efficient than XGBoost • Can manage overfitting issues • Can achieve high prediction accuracy.
CatBoost	• This technique performs best if dataset contains categorical features [9] • CatBoost is sensitive to hyper-parameter tuning [18].	• Not suitable for numerical dataset
Gradient Boosting	• Gradient Boosting models showed better predictive accuracy than other models, with GBDT being the most accurate [19]. • Accuracy is improved by using the loss function but is prone to overfitting issues.	• Prone to overfitting issues

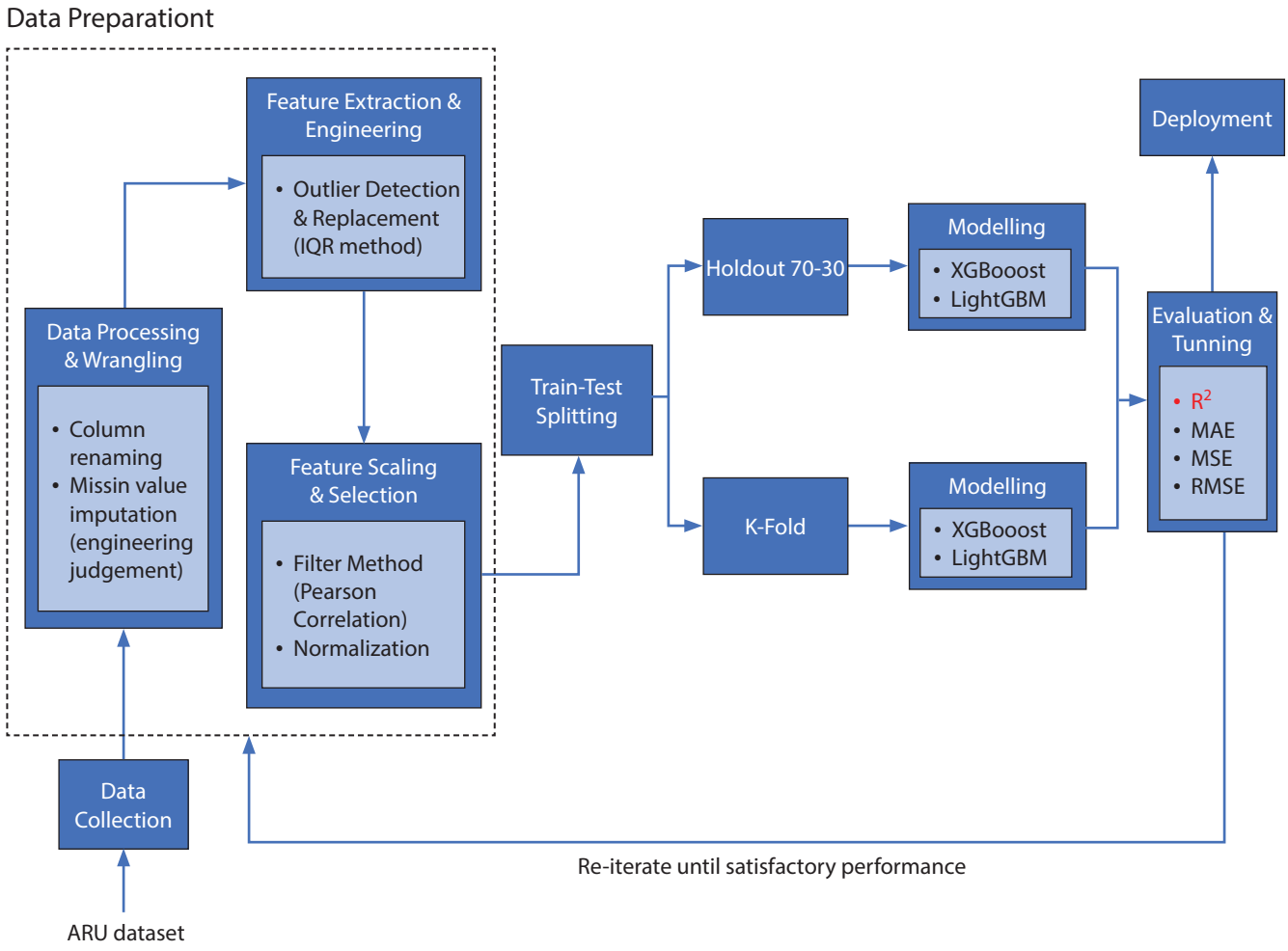

Figure 1 Research Workflow

RESEARCH WORKFLOW AND METHODOLOGY

The scope or sample for this research work was based on a selected oil and gas refinery in Malaysia, which will not be disclosed. Amine Regeneration Units (ARU) system and real-world industry datasets were used for the experiment. The Cross-Industry Standard Process for Data Mining (CRISP-DM) was adopted as a reference framework. The software utilised was Spyder IDE 5.1.5, which employs Python version 3.8.3 as the programming language. Figure 1 illustrates the detailed research workflow.

Data Collection

Dataset for this research work was retrieved from an undisclosed refinery company's Plant Management Software which links various transmitters and monitoring devices across the plant in Amine Regeneration Unit (ARU). It is linked to Microsoft Excel and stored in a comma-separated file (.csv) extension file. The .csv file was then loaded into Spyder IDE 5.1.5 for the next steps.

Data Preparation

This section includes three areas which are data processing and wrangling, feature extraction and engineering, and feature scaling and selection. The raw dataset is prepared for modelling in this section.

Data Processing and Wrangling

This process involves cleaning and unifying messy and complex datasets for easy access and analysis. The steps are summarised as follows,

1. The initial data frame contains 915 rows and 9 columns.
2. The columns were renamed according to python syntax to allow easy referencing. The renamed columns are "Time", "Velocity_01", "Velocity_02", "Ammonia", "Dissolved_H2S", "MDEA_conc", "CR1003", "CR1010", and "CR1011".
3. Next, the negative values from the columns "Velocity_01" and "Velocity_02, are replaced with zero using the *where* function from *numpy*. As per engineering judgement, these negative values indicate no flow in the structure and represent 0 m/s. An instrumentation error causes the value to drop below zero.
4. The "No Data" string values in all other columns, which include "Ammonia", "Dissolved_H2S", and "MDEA_conc", are replaced with respective mean values for each column using the functions *replace* and *mean* as a form of imputation because these parameters are generally stable in terms of values with minimal spiking. Note that columns "Ammonia" and "MDEA_conc" do not contain any data and will be omitted from the analysis.
5. As for the remaining "CR1003", "CR1010", and "CR1011" columns, the "No Data" string values were replaced using the *bfill* function, which performs backwards fill with values that are present in the pandas data frame. These three columns will represent the dependent variables which is the corrosion rate. The transmitter returns "No Data" due to errors within the network. Replacing the non-numeric values with 0 mm/yr will be a conservative approach assuming no corrosion activity is present. Thus, the *bfill* method was selected as it represents the actual or logical corrosion rate that would have been transmitted if there were no errors within the network.

Feature Extraction and Engineering

In this section, the dataset is visualised by each feature to understand the requirements for re-engineering. Then, each feature is scanned for any outliers and replaced with the appropriate values. This step is performed to increase the accuracy of our model. The steps involved are outlined as,

1. *matplotlib.pyplot* library was imported for figure generation. A line chart is plotted for each feature to observe any irregularities.
2. Based on the visualisation of the features, it was observed that "Velocity_01" may contain outliers that can affect the accuracy of the model being developed. The outliers are then replaced with the median value since the median represents the center point of the data.
3. The Inter-Quartile Range (IQR) method was selected for outlier detection. Any values beyond the lower and upper bounds are detected as outliers.
4. The dependent variables, "CR1003", "CR1010", and "CR1011" columns, were aggregated using the *max* function prior to outlier detection. The new aggregated data frame was named "Y_df_agg". The *max* function was selected to represent the

worst-case scenario for the dependent variable (corrosion rate).

Feature Scaling and Selection

The filter technique was utilised for this section and the Pearson correlation factor was used to benchmark the correlation of the independent variables to the dependent variable. A Pearson correlation is a number between –1 and 1, indicating the extent to which two variables are linearly related. Figure 2 represents the Pearson correlation heatmap and based on the results, it can be concluded that the two most correlated variables are "Velocity_01" and "Velocity_02". A comparison between training the model with two features and three were performed in terms of computational time.

Normalisation was selected to be performed compared to standardisation. Normalisation is a scaling technique whereby values are shifted, rescaled, and ranged between 0 and 1. It is also known as Min-Max scaling. Normalisation is also selected because the data distribution does not follow Gaussian distribution as per Figure 3 and Figure 4. "Ammonia" and "MDEA_conc" did not contain any data and were removed from analysis in the following steps.

Modeling

The details of the models developed are tabulated in Table 2. The target was to develop the best model that could represent the ARU dataset. The prepared dataset was split into training and testing data using

Figure 2 Pearson Correlation of Independent Variables to "Aggregated_CR"

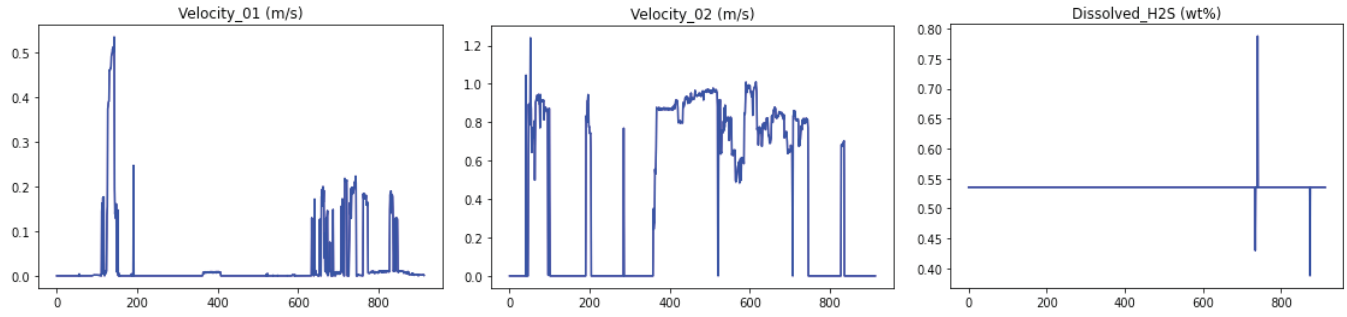

Figure 3 Feature Distribution (Independent Variables) prior to Outlier Detection

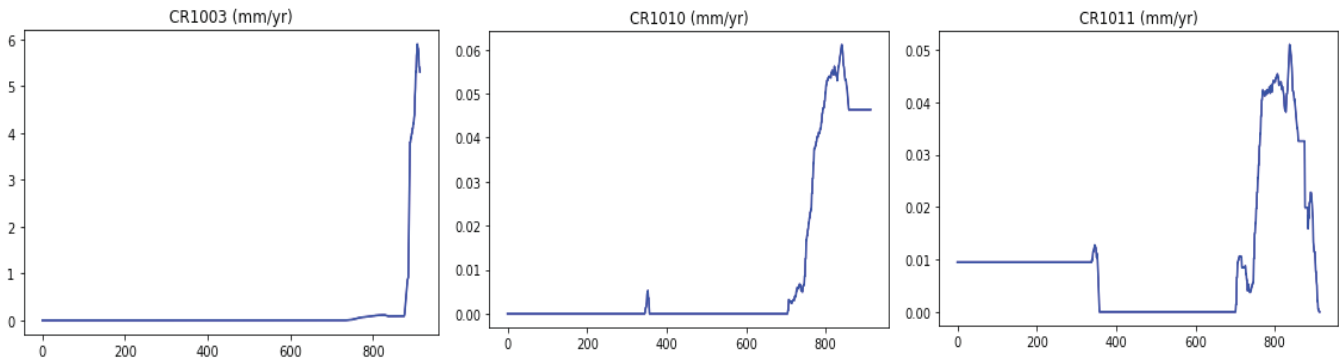

Figure 4 Feature Distribution (Dependent Variables) prior to Outlier Detection

two different approaches, which are the holdout method and the K-Fold cross-validation method. For the holdout method, the split was finalised at 70:30, whereby 70% of the dataset will be used to train the model and 30% will be used for testing and evaluation. As for K-Fold cross-validation, the folds or "k" is set at 5. Figure 5 represents the K-Fold technique.

K-Fold will reduce bias and variance as each data point gets in a validation set once and gets in a training set "k"-1 times.

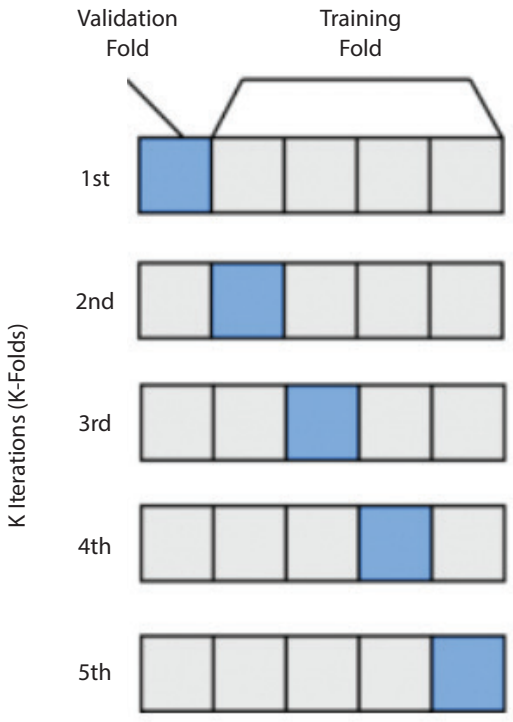

Figure 5 K-Fold Cross-Validation (CV=5) Illustration

Table 2 Model names and description

Model Name	Algorithm	Train-Test Split Method
xgboost1	Extreme Gradient Boosting (XGBoost)	Holdout – 70:30
xgboost2		Holdout – 70:30 (Tuned)
xgboost3		K-Fold (5 folds)
xgboost4		K-Fold (5 folds) – (Tuned)
lgboost1	Light Gradient Boosting (LightGBM)	Holdout – 70:30
lgboost2		Holdout – 70:30 (Tuned)
lgboost3		K-Fold (5 folds)
lgboost4		K-Fold (5 folds) – (Tuned)

Evaluation and Tuning

The data evaluation is performed using statistical methods, which are R-squared (R^2), Mean Average Error (MAE), Mean Squared Error (MSE), and Root-Mean-Square Error (RMSE). R^2 was used as the primary tool to benchmark the accuracy of all models. The coefficient of determination, R^2, is proved to be more informative and truthful in dealing with regression data compared to MAE, MSE, and RMSE which possess interpretability limitations [20]. R^2 scores reflect the performance of the original and synthetic data in a regression model.

Three hyperparameters were tuned in this research for both XGBoost and LightGBM models. They are *learning_rate*, *max_depth*, and *n_estimators*.

GridSearchCV from *sklearn.model_selection* was imported as one of the powerful functions to finetune the hyperparameters through its exhaustive search for optimum parameters. This is done for both XGBoost

and LightGBM models. Each model was then retrained using the optimum hyperparameters generated and evaluated with all four statistical methods.

RESULTS AND DISCUSSION

Dataset Preparation and Visualization

Outliers were detected using the IQR approach and replaced with the median value for "Velocity_01" in Figure 6. There were no outliers detected for other variables. Outliers are bad for boosting algorithms, including XGBoost and LightGBM, because boosting builds each tree on previous trees' residuals/errors. Outliers will have much larger residuals than non-outliers thus, gradient boosting will focus a disproportionate amount of its attention on those points.

Figure 7 illustrates the normalised aggregated corrosion rate used as labels for our model training. Based on the pattern of corrosion rate compared to "Velocity_01", it appears that the corrosion rate spikes as the velocity increases. This is evident in the last approximately 200 sets of observations.

Feature Selection

The final prepared dataset contained three independent variables ("Velocity_01", "Velocity_02", and "Dissolved_H2S") and a single dependent variable (aggregated corrosion rate). It was important to decide on the relevancy of the variables or features to the label. The filter method was implemented utilising the Pearson correlation factor to benchmark the correlation between independent variables to the dependent

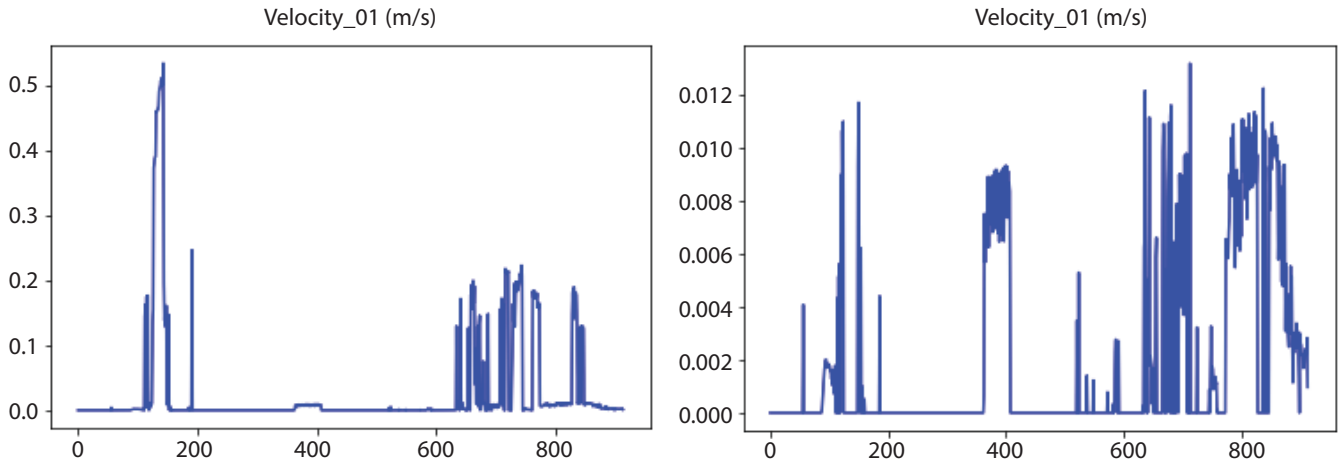

Figure 6 Velocity_01 values after outlier removal and replacement

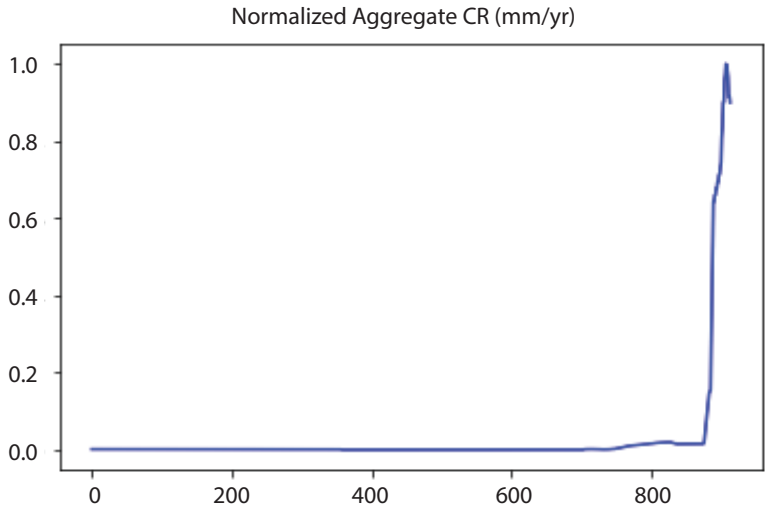

Figure 7 Normalized Aggregated Corrosion Rate - Labels for Supervised Learning

Table 3 Pearson Correlation Factor - Feature Selection (Filtering)

Variables	Pearson Correlation
Velocity_01	0.044794686
Velocity_02	-0.192155565
Dissolved_H2S	-0.001638142
Aggregated_CR	1

variable. Based on Table 3, it is evident that "Velocity_01" has a positive correlation, whereas "Velocity_02" and "Dissolved H2S" have negative correlations with "Aggregated_CR". A negative correlation is an indication that both variables move in the opposite direction. Since the direction is not a major concern, the absolute Pearson correlation factor values were taken and "Dissolved_H2S" was removed due to having the lowest correlation.

Reducing features during modelling improves computational time without impacting the model's accuracy. Table 4 summarises the computational time for some models developed. Based on the table, it is evident that the computation time improved by reducing the number of features from three to two.

Data Splitting

Two approaches were experimented for this research: Holdout and K-Fold cross-validation. Both the approaches were used to model XGBoost and LightGBM algorithms, respectively. Based on Figure 7, it is observed that the label (aggregated corrosion rate) pattern is skewed to the left. Thus, the holdout approach will not be as suitable as it will be biased toward the current pattern. The K-Fold (k=5) approach is expected to be more accurate and representative since each data point gets to be in a validation set exactly once and in a training set "k"-1 times.

Data Modelling and Evaluation

R-squared (R^2), Mean Average Error (MAE), Mean Squared Error (MSE), and Root-Mean-Square Error (RMSE) were the statistical methods used for the evaluation of all eight models developed. The R^2 score was used as the primary benchmarking criterion. Figure 8 illustrates the improvement of the model (XGBoost) in terms of predicted values (represented by "Tuned Prediction"). The actual values from the testing data are denoted by the blue line.

A summary of the accuracies for all models is available in Table 5. It includes both the holdout and K-Fold approaches.

It is noted that the K-Fold cross-validation approach is better than the holdout approach for both XGBoost and LightGBM modelling. This is evident based on Table 5. The R^2 score improved by 11.9% for the XGBoost model with K-Fold cross-validation compared to the holdout approach. On the other hand, the R^2 score improved by 2.5% for the LightGBM model with K-Fold cross-validation compared to the holdout approach. From Table 5, the two best-performing models can be shortlisted as "xgboost4" and "lgboost4," with 65.4% and 64.5% R^2 scores, respectively. Table 6 compares the statistical scoring accordingly. Model "lgboost4" did not perform better than model "xgboost4," although its computational time is faster by 62.4%.

Faster computation does not mean better accuracy. LightGBM is proven to be faster because the algorithm splits the tree leaf-wise with the best fit, whereas other boosting algorithms, such as XGBoost, split the tree depth-wise or level-wise. Splitting the tree level-wise takes a longer time to arrive at optimum results.

Table 4 Comparison of Best Computational Time for Selected Models - Feature Reduction

Model	Train-Test Split Method	Computation time (sec)		
		3 Features	2 Features	Improvement (%)
xgboost4	K-Fold (5 folds) – (Tuned)	3.18918395	2.97192144	6.812479614
lgboost1	Holdout - 70:30	0.072498322	0.05829906	19.58563536
lgboost3	K-Fold (5 folds)	1.923957825	1.43927574	25.19192869
lgboost4	K-Fold (5 folds) – (Tuned)	1.157142878	1.11723042	3.44922507

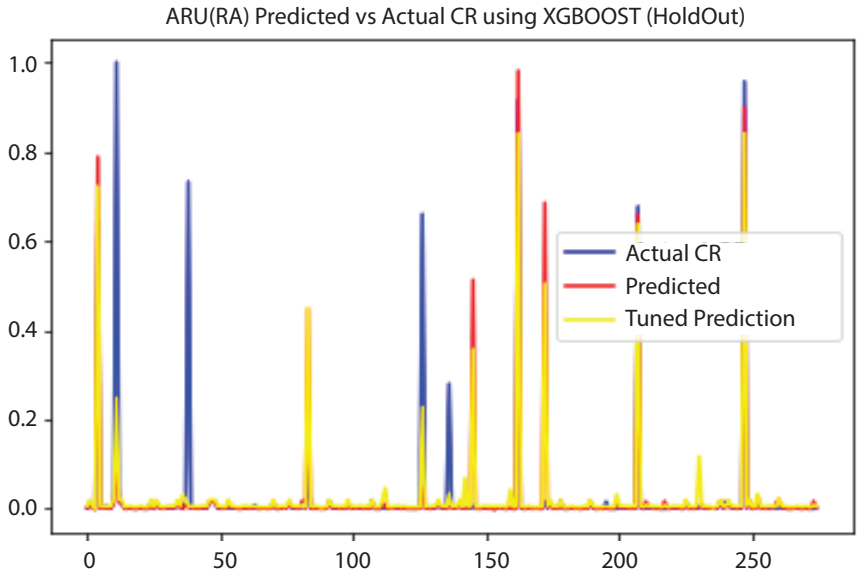

Figure 8 Prediction models "xgboost1" vs "xgboost2" for ARU dataset

Table 5 Summary of accuracies between all models

Accuracy (%)	Models							
	xgboost1 (XG_ HoldOut)	xgboost2 (XG_ HoldOut_ Tuned)	xgboost3 (XG_ KFold)	xgboost4 (XG_ Kfold_ Tuned)	lgboost1 (LG_ HoldOut)	lgboost2 (LG_ HoldOut_ Tuned)	lgboost3 (LG_KFold)	lgboost4 (LG_Kfold_ Tuned)
R^2	42.027	58.423	46.788	65.382	60.635	62.932	64.002	64.515
MAE	1.720	1.860	2.096	2.098	1.785	1.757	2.044	2.028
MSE	0.955	0.685	1.006	0.667	0.649	0.611	0.687	0.689
RMSE	9.775	8.278	9.604	7.836	8.055	7.816	7.991	7.966

Accuracy depends on the suitability of the model to the dataset. In this research work, the ARU dataset seems to prefer the XGBoost algorithm compared to the LightGBM algorithm when K-Fold cross-validation is used. This is contrary to most literature that claims the LightGBM algorithm to be far better and capable of outperforming existing boosting algorithms in terms of accuracy and computational speed. In summary, the best model that represents the ARU dataset based on the coefficient of determination and R^2 scoring is model "xgboost4," which is developed using the XGBoost algorithm with K-Fold cross-validation. The best accuracy obtained is 65.4%, based on the R^2 score.

Table 6 Comparison between the best LightGBM and XGBoost tuned models

Accuracy (%)	Models		Improvement (%)
	xgboost4 (XG_KFold Tuned)	lgboost4 (LG_KFold Tuned)	
R^2	65.38170685	64.51520544	-1.325296412
MAE	2.097639143	2.028058545	-3.317090975
MSE	0.666744839	0.688568605	3.273181095
RMSE	7.836057022	7.965926745	1.657335086
Computation Time (s)	2.97192144	1.11723042	62.40713483

CONCLUSION

There is a need to perform forecasting of corrosion rates in ARU to prevent failures and mitigate risks. Machine learning provides the means to perform data analytics to predict future corrosion rates as opposed to existing corrosion monitoring devices. Ensemble models, which are supervised learning algorithms, will best fit the continuous, numerical, and inconsistent datasets in ARU. Boosting algorithms, namely, Extreme Gradient Boosting (XGBoost) and Light Gradient Boosting (LightGBM), are shortlisted to compare corrosion rate prediction accuracy.

The research work adopted CRISP-DM framework and the activities include data collection from an undisclosed oil and gas refinery in Malaysia, data processing, wrangling, feature extraction, engineering, feature scaling, selection, data modelling, evaluation, tuning, and model deployment. Model evaluation was performed using statistical methods, which include R-squared (R^2), Mean Average Error (MAE), Mean Squared Error (MSE), and Root-Mean-Square Error (RMSE). A total of eight models were developed and compared for accuracy, primarily based on the R^2 score. The training and testing data were split based on two approaches: Holdout (70:30) and K-Fold cross-validation (splits at 5) before model training. The approaches were modelled separately for each algorithm. Hyperparameter tuning was performed by focusing on three defined parameters: learning_rate, max_depth, and n_estimators.

Initial accuracies were reported to be as low as 42% (based on R^2 score) and 60% for XGBoost (holdout) and LightGBM (holdout) algorithms, respectively. Further tuning of models and utilisation of the K-Fold cross-validation approach improved the accuracy remarkably. The overall best model that represents the ARU dataset based on R^2 scoring was reported to be model "xgboost4," which was developed using the XGBoost algorithm with K-Fold cross-validation. The best and improved accuracy reported based on R^2 scoring was 65.4%. The selection contradicts most literature that claims the LightGBM algorithm to be far better and capable of outperforming existing boosting algorithms in terms of accuracy and computational speed. Evidently, accuracy depends on the suitability of the model to the dataset, especially for the ARU dataset, which possesses unique characteristics.

REFERENCES

[1] A.P. Institute, "ANSI/API RECOMMENDED PRACTICE 571: Damage Mechanisms Affecting Fixed Equipment in the Refining Industry", *American Petroleum Institute,* 3rd Edition, pp. 345-367, 2020.

[2] "API Recommended Practice 581, Risk-Based Inspection Methodology", *Americal Petroleum Institute,* pp. 350-400, 2019.

[3] Microsoft, "Data Model specification and limits," *support.microsoft.com*, [Online]. Available: https://support.microsoft.com/en-us/office/data-model-specification-and-limits-19aa79f8-e6e8-45a8-9be2-b58778fd68ef#:~:text=Maximum%20file%20size%20for%20rendering,(GB)%20maximum%20(2).

[4] F. Namazi & H. Almasi, "Amine Corrosion in Gas Sweetening Plant: Causes and Minimiation on Real Case Study", *Nace International,* NACE-2016-7187, pp. 1-10, 2016.

[5] N.A. Hatcher, C.E. Jones, G.S. Weiland & R.H. Weiland, "Predicting corrosion rates in amine and sour water systems", *eptq,* pp. 1-5, 2014. https://www.ogtrt.com/files/publications/74/gas_14_ogrt.pdf

[6] V.V. Lagad, M.S. Cayard & S. Srinivasan, "Prediction And Assessment Of Rich Amine Corrosion Under Simulated Refinery Conditions," *Nace International,* NACE-10183, pp. 1-10, 2010.

[7] Y. Liu, J. Chen, Y. You, A. Xu, P. Li, Y. Wang, J. Sun, Z. Yu, F. Gao & J. Zhang, "An ensemble learning based framework to estimate warfarin maintenance dose with cross-over variables exploration on incomplete data set", *Computers in Biology and Medicine,* 131, 104242, pp. 1-10, 2021.

[8] C.T. Tran, M. Zhang, P. Andrea, B. Xue & L. T. Bui, "Multiple Imputation and Ensemble Learning for Classification with Incomplete Data", *Intelligent and Evolutionary Systems*, 8, pp. pp 401-415, 2016.

[9] S. Rahman, M. Awais, K.M. Ghori, M. Raza, S. Yaqoob & M. Irfan, "Performance Analysis of Boosting Classifiers in Recognising Activities of Daily Living", *International Journal of Environmental Research and Public Health,* 8, 17(3), pp. 1-17, 2020.

[10] P. Bahad & P. Saxena, "Study of AdaBoost and Gradient Boosting Algorithms for Predictive Analytics," in

International Conference on Intelligent Computing and Smart Communication, Springer, Singapore. Doi:10.1007/978-981-15-0633-8_22S2019.

[11] H. Saghafi & M. Arabloo, "Modeling of CO_2 solubility in MEA, DEA, TEA, and MDEA aqueous solutions using AdaBoost-Decision Tree and Artificial Neural Network", *International Journal of Greenhouse Gas Control,* 58, pp. 256-265, 2017.

[12] T. Chen & C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[13] R.K.Mazumder, A.M.Salman & Y. Li, "Failure risk analysis of pipelines using data-driven machine learning algorithms", *Structural Safety,* 89, pp. 1-10, 2021.

[14] R. Zhong, R. Johnson Jr & Z. Chen, "Generating pseudo density log from drilling and logging-while-drilling data using extreme gradient boosting (XGBoost)", *International Journal of Coal Geology,* 220,1, 2020.

[15] R. Simanjuntak & D. Irawan, "Applying Artificial Neural Network and XGBoost to Improve Data Analytics in Oil and Gas Industry"m *Indonesian Journal of Energy,* 4, 1, pp. 26-35, 2021.

[16] H. Yang, L. Lu & K. Tsai, "Machine Learning Based Predictive Models for CO2 Corrosion in Pipelines With Various Bending Angles", *Society of Petroleum Engineers,* 2020.

[17] M. Gan, S. Pan, Y.P. Chen & C. Cheng, "Application of the Machine Learning LightGBM Model to the Prediction of the Water Levels of the Lower Columbia River", *Journal of Marine Science and Engineering,* 9, 5, pp. 1-22, 2021.

[18] J. T. Hancock & T. M. Khoshgoftaar, "CatBoost for big data: an interdisciplinary review," *Journal of Big Data, 7, 1, 94* pp. 1-45, 2020.

[19] L. Yan, Y. Diao, Z. Lang & K. Gaoa, "Corrosion rate prediction and influencing factors evaluation of low-alloy steels in marine atmosphere using machine learning approach", *Science and Technology of Advanced Materials* 21,1, pp. 359–370, 2020.

[20] D. Chicco, M. J. Warrens & G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation", *PeerJ Computer Science,* 7, pp. 1-30, 2021.