Proceedings of the IISE Annual Conference & Expo 2022
K. Ellis, W. Ferrell, J. Knapp, eds.

Evaluation of Machine Learning Algorithms in Predicting Corrosion Rates on Oil Refinery Piping

Tiffany Meeks[a], Abdulsalam Alqarni[a,b], Sandra Ponce[c], Heidi Fehr[d], Hajar Razip[e], Om Prakash Yadav[a,f], Mohd Hilmi Hasan[e]

[a] North Dakota State University, Fargo, ND, USA
[b] King Khalid University, Abha, Asir, Saudi Arabia
[c] Texas A&M University, College Station, TX, USA
[d] University of Nevada, Las Vegas, NV, USA
[e] Universiti Teknologi PETRONAS, Seri Iskandar, Perak Darul Ridzuan, Malaysia
[f] North Carolina A&T State University, Greensboro, NC, USA

Abstract

Maintenance on piping failures due to degrading materials are time consuming, costly, and result in decreased productivity to fix. However, with growing technology it is possible to use machine learning models to predict when maintenance would be needed prior to such failures. With machine learning models and time series data analysis being a key aspect in the growing field of predictive maintenance, data from sensors in pipelines on corrosion were analyzed in Python to create a predictive model that would accurately predict plant equipment failure. Both classical and machine learning models were analyzed to use on the time series data of corrosion rates and Integrity Operating Window (IOW) tags collected from oil refinery piping. Models such as ARIMA and ARMA were comparatively the best at predicting the corrosion rates. On the other hand, machine learning models such as Random Forest (RF) and Support Vector Machine (SVM) were not as accurate, although RF was one of the more accurate machine learning models tested. Analyzing the corrosion rates yielded more accurate predictions than analyzing the IOW tags. Analyzing the IOW tags with machine learning models yielded percentage errors of 99% and above. However, ARIMA was found to have the highest success rate in predicting the corrosion rates obtained from the data.

Keywords

Corrosion Rate, Machine Learning Models, Corrosion Prediction, Time-Series Forecasting, Classical Modeling.

1. Introduction

Failure of pipelines due to degrading materials is a common issue dealt with by infrastructure, specifically oil refineries. Maintenance on these systems can be costly, time consuming, and can cause a decrease in productivity. However, with increasing technology and the ability to evaluate equipment's need for maintenance has proven to increase productivity, save time, along with many other benefits. This predictive maintenance through collaboration with machine learning would prove to be very beneficial to pipelines in oil refineries. As pipelines are the transportation network of oil refineries and they have limited accessibility for maintenance. This makes it highly important to be able to maintain them and avoid failures that could cause delays. Predictive maintenance would need to be able to identify and alert someone of the optimal time to perform the maintenance to ensure longest life and productivity of the pipes/piping and pipelines. In order to develop a system that is capable of predicting the optimal time to repair a pipe/piping and/or pipeline network, an understanding of the degradation rate is necessary. The degradation rate is the process by which, in this case a pipe, loses its functionality with respect to time and external strenuous conditions at a continuous rate [1]. There are many prediction modeling techniques used to predict the rate of degradation and the remaining useful life (RUL). Linear models are widely used to predict corrosion rate, especially when dealing with limited time-series data points [2]. The power-law model is widely accepted to describe the corrosion growth mechanism and expect the corrosion depth over time in the pipeline [3]. The de Waard-Milliams model is the most commonly used to predict internal corrosion rate, which is commonly known as CO2 corrosion rate [4]. Several probabilistic models consider probability as a primary factor are also proposed in predicting the corrosion growth process over time in pipelines and are reviewed in [5]. As the industry continues to progress through "The

Fourth Industrial Revolution," also known as Industry 4.0, Many engineers and researchers are focusing on the combination of physical data and digital systems and equipment. This is especially true in predictive maintenance modeling (PdM). Technological advances in the past decade have let researchers and engineers be able to analyze large amounts of data and predict important degradation of plant equipment. PdM often uses machine learning (ML) to recognize patterns in data and predict when equipment needs to be fixed before failure actually occurs. Many common models of ML used in PdM are Random Forest Regression (RF), Artificial Neural Network (ANN), Convolution Neural Network (CNN), Support Vector Machine (SVM), Decision Tree (DT), etc. A comparative study [6] compares the performance of three artificial intelligence models such as ANN, RF, and SVM in predicting the internal corrosion rate in the pipeline and found that ANN outperforms others.

This paper aims to analyze the data given and come up with a model to predict plant equipment failure of pipelines. Sensors along oil refinery pipelines in Malaysia collected data over a ten-month span in 2019, from March to December, on corrosion rates and properties that influence corrosion; such as temperature, pH, flow velocity, salt, iron, chloride, and basic sediment and water. With the given data, different time series forecasting models are used to find the most accurate model. This comparative study selects only a few corrosion rate sensors and considers both classical ways of analyzing time series data and other machine learning (ML) models. Classic time series analysis algorithms we used involved: Autoregression (AR), Autoregressive Moving Average (ARMA), and Autoregressive Integrated Moving Average (ARIMA). For ML models we looked at: RF, SVM, Gradient Boosting Regression (GBR), and K-Nearest Neighbor Regression (KNN). As we analyzed the results of these models, we learned which predicted the designated test section more accurately than others.

2. Method and Testing

2.1 Software

In this paper, we used Python, a programming language, and JupyterLab to perform all steps in the machine learning lifecycle such as data preprocessing, feature engineering and model development. Python was chosen due to its ease of learning, simplicity and the wealth of resources for data science libraries. Jupyterlab is a computational notebook that allows us to run codes and document our process in a single web page. Both are popular tools in the data science community.

2.2 Performance Measure

The accuracy of the models was measured using mean absolute percentage error (MAPE) and mean absolute error (MAE). MAPE was used to determine how accurate the forecast model was, and MAE was used to show the average of all errors [7]. We also calculated the percentage error using a Root Mean Square Estimate (RMSE) divided by the standard deviation of the data. To find the RMSE, we used cross evaluation on the data to get a mean RMSE.

2.3 Data Preparation & Cleaning

When preparing the data for analysis we went through a couple of steps:

1. Read in the data from the excel files (both metadata and dataset)
2. Parsed the dates and index columns
3. Selected a corrosion group (CG) from the existing cg in the metadata file and created an object containing only the items from that CG
4. Converted the values in the dataset file to numeric values
5. Created objects for both the corrosion rate tags and the Integrity Operating Window (IOW) tags
6. Selected a target tag from the data
7. Dropped the null values in the target tag and selected a time frame
8. Split the data into train and test sets. There were two different ways to create train and test groups in order to analyze the data: custom_train_test_split (seen in section A) and the train_test_split function from the sklearn library. The first algorithm was used for splitting the data for the classical methods, where the training and testing sets were adjusted to 3 weeks and 1 week, respectfully. The second way was used to split data for the ML methods, where 70% of the data would be used for training and the remaining 30% for testing. Figure 1 demonstrates an example of a graph after completing these steps with the first way.

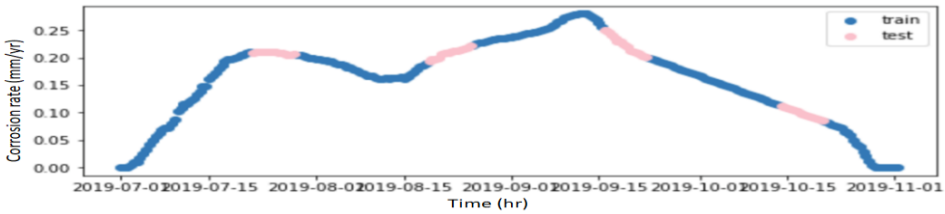

Figure 1: Train and Test Sets on PU1-CT-1129 Graph

2.4 Classical Models

The first method examined was AR, which studies previous data and uses that knowledge to predict the next value in the series [8]. Next, ARMA was considered, it is a combination of AR and Moving Average (MA), hence the name. It observes and uses errors from the past to calculate future steps [8]. Lastly, ARIMA was used, which is very similar to ARMA except it includes Integration (I), which differences data in order to make it stationary [8].

2.5 ML Models

Some of the ML models we looked at were Linear Regression (LR), Decision Tree (DT), Support Vector Machine (SVM), RF, KNN, and GBR. To briefly go over each model that we had looked at in our testing. LR is a model that uses the process of finding a straight line that best approximates the set of points. DT models use a system of roots and nodes, breaking the dataset into smaller subtrees to analyze. This results with a tree with decision nodes. SVM modeling is based on finding a hyperplane that best divides the dataset into two classes. It then conducts optimization to pick an optimal line from the distance of the closet values to the hyperplane. RF is a model that uses a collection of arbitrary decision trees; it aims to predict the remaining useful life of an object. KNN model approximates association between variables and averages the observations based on a similarity measure. GBR modeling uses a weak learner (such as decision or regression trees) to add the trees using a gradient descent procedure to minimize the loss.

3. Results

3.1 Corrosion Rate

Both classical and ML models were utilized to predict the time series data of corrosion rates for the first processing unit (PU1). Figure 2 and Figure 3 display the outcome of these classical and ML methods, respectively. From these images and Table 1, it was noted that the classical models provided a better prediction for the first 4 weeks of the PU1-CT-1129 graph, with ARMA having the best result and lowest error of 0.0083.

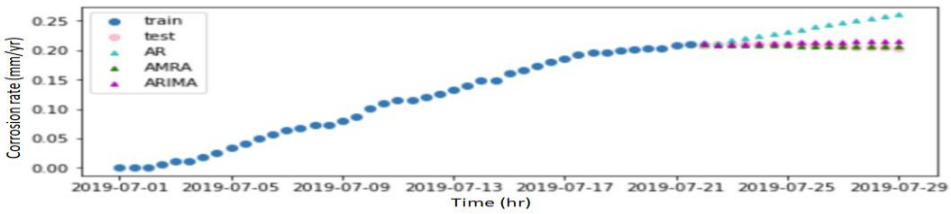

Figure 2: Classical Forecast of PU1-CT-1129.1 Graph

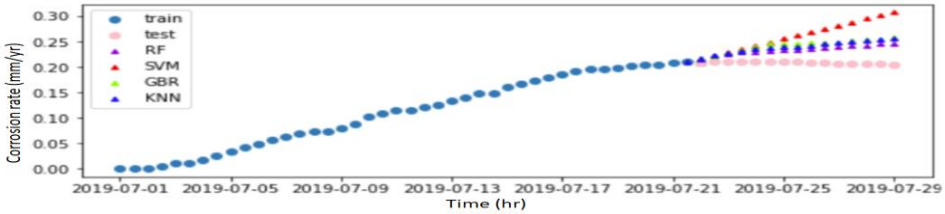

Figure 3: ML Forecast of PU1-CT-1129.1 Graph

Similarly, Figure 4 and Figure 5 illustrate the different prediction methods for the second 4 weeks of the PU1-CT-1129 graph. The best model here was ARIMA, with the MAPE error being 0.016, as seen in Table 1.

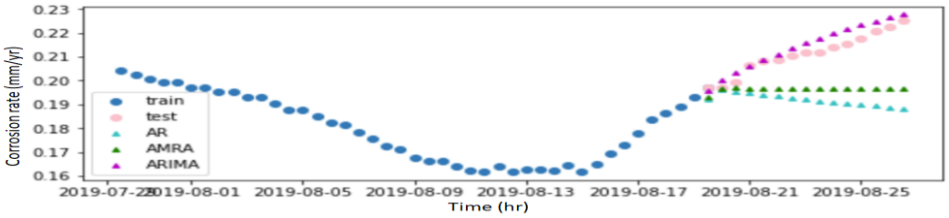

Figure 4: Classical Forecast of PU1-CT-1129.2 Graph

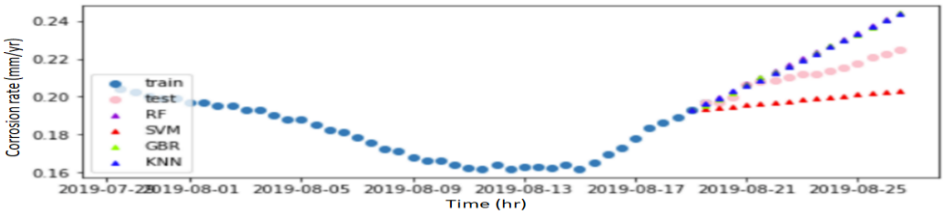

Figure 5: ML Forecast of PU1-CT-1129.2 Graph

Since the classical methods were the best for the first two trials, we conducted several more tests on corrosion rate data to see if this pattern would continue. Parameters such as the length of the train and test sets were changed from 3 and 1 week, respectively, to 6 and 2 weeks, still maintaining the 3:1 ratio. The reason for this was to observe whether adding more data would improve the model. Figure 6 and Figure 7 demonstrate this by showing the first 8 weeks of PU1-CT-1188. In these graphs ARIMA was the best method with an error of 0.1633.

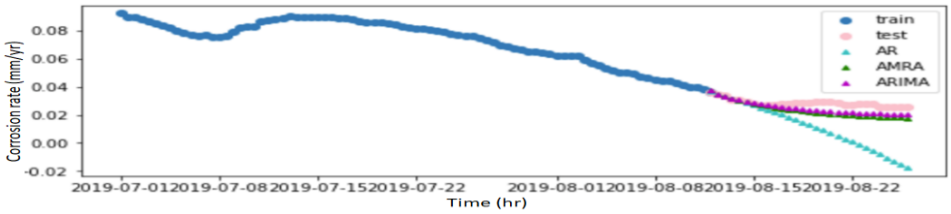

Figure 6: Classical Forecast of PU1-CT-1188.1 Graph

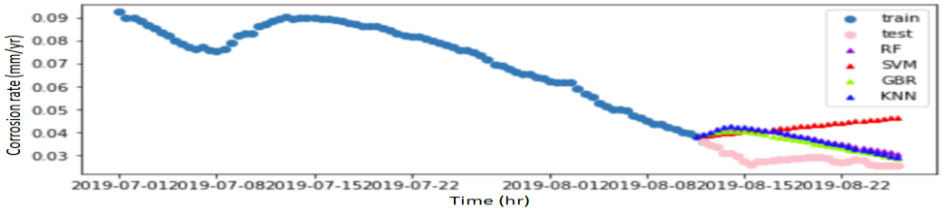

Figure 7: ML Forecast of PU1-CT-1188.1 Graph

After all methods were tested on three PU1 models with different train and test lengths and different time stamps, the MAPE errors were summarized in Table 1. The green highlights signify which classical method had the lowest error and the yellow highlights indicate which ML method had the lowest error. From the table, we observed that ARIMA had the highest success rate of 43.75% out of the classical methods. The second highest success rate was ARMA with

a 37.5% accuracy, and third was AR with an accuracy of 18.75%. Out of the ML models, RF had a success rate of 31.25%, SVM and KNN both were 25% accurate, and GBR was 18.75% accurate. We also noticed how the classical method always had a lower error than ML method, except for two times: PU1-CT-1173 and PU1-CT-1188.2. This suggests that the classical methods were more accurate compared to the ML ones.

Table 1: Comparison of Data using MAPE

	AR	ARMA	ARIMA	RF	SVM	GBR	KNN
PU1-CT-1101	0.0176	0.0180	0.0654	0.0379	0.1099	0.0708	0.0388
PU1-CT-1103	0.1756	0.1370	0.0114	0.1676	0.2391	0.2067	0.1717
PU1-CT-1105	0.0546	0.0187	0.0367	0.0369	0.0326	0.0371	0.0347
PU1-CT-1129	0.4391	0.3631	0.2510	0.5999	0.5602	0.5850	0.4951
PU1-CT-1129.1	0.1118	0.0083	0.0230	0.1065	0.1946	0.1383	0.1298
PU1-CT-1129.2	0.1004	0.0742	0.0160	0.0405	0.0638	0.0380	0.0377
PU1-CT-1129.3	0.1246	0.1093	0.0634	0.2349	0.1159	0.1937	0.2033
PU1-CT-1129.4	5.43E-08	0.1393	0.1330	6.15E-08	1.94E-07	5.81E-08	8.62E-08
PU1-CT-1161	0.1940	0.0138	0.0626	0.1351	0.1603	0.1679	0.1557
PU1-CT-1162	0.1821	0.0193	0.1109	0.1434	0.1320	0.1420	0.1378
PU1-CT-1173	0.0011	0.0370	3.44E-05	1.18E-07	2.61E-08	1.20E-07	2.03E-07
PU1-CT-1179	0.0808	0.0921	0.0198	0.0566	0.0497	0.0421	0.0622
PU1-CT-1180	1.3494	0.8670	1.9274	17.2086	1.4595	2.1814	3.3926
PU1-CT-1188.1	3.7579	0.2554	0.1633	0.2327	0.3256	0.1982	0.2281
PU1-CT-1188.2	0.1284	0.4034	0.2688	0.2830	0.2752	0.3098	0.1279
PU1-CT-1189	0.0681	0.0542	0.0703	0.0683	0.0659	0.0576	0.0564

It is important to note that not all of the data was suitable for predicting since some corrosion rate processing units had irregular or non-existing data such as the model in Figure 8. The lines are jagged, and it would be difficult to predict the next point in the sequence. Therefore, we focused on graphs that were more continuous.

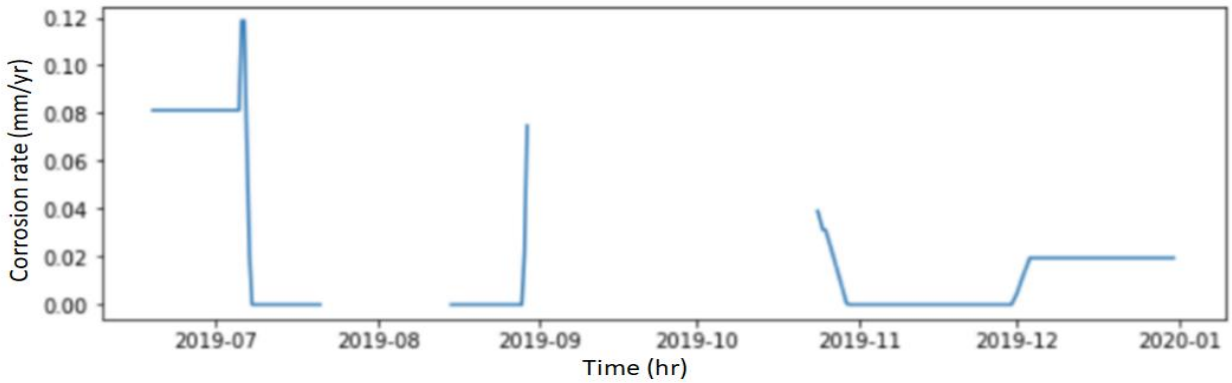

Figure 8: PU1-CT-1164 Graph

3.2 IOW

We also looked at how the IOW tags could be used to predict corrosion rates. Focusing on the IOW tags corresponding with the tag PU3-CT-1010 from the 3rd processing unit, we used LR, DT, RF, GBR, and KNN regression models. In this research, it was found that each model had a percentage error higher than 100% (seen in Table 2).

Table 2: Comparison Data for IOW ML Model

	Percentage Error
LR	2.243627
DT	1.143244
RF	0.997076
GBR	1.334363
KNN	1.061953

This data shows the importance of having enough data to train the model with. The data from the corresponding IOW tag did not offer enough data for any of the models to properly be trained and tested. We suggest gathering more data, test more feature engineering, and also test more ML models with more hyperparameter tuning in order to improve the IOW models.

4. Conclusion

In our research, we were able to look at how valuable ML can be in PdM. Through assessing the corrosion rate tags, we found that of both the classical and the ML modeling, ARIMA had the highest success rate. Meaning the classic ways of assessing time series forecasting was the most accurate way to predict future data points. Through assessing the IOW, we saw that there was not enough data to create low error models. The next steps we could continue in our research to produce more accurate results would be to collect more data, test more feature engineering, assess the data with more ML models, and do more hyperparameter tuning.

Acknowledgements

We wish to acknowledge the following universities and foundations: National Science Foundation (NSF) International Research Experience for undergraduate Students (IRES) under the grant number OISE-1952493, North Dakota State University College of Engineering, University of Nevada – Las Vegas College of Engineering, Texas A&M University College of Engineering, Universiti Teknologi PETRONAS College of Engineering, and the National Science Foundation (NSF) EPSCoR RII Track-2 Program under the grant number OIA-2119691. The findings and opinions expressed in this article are those of the authors only and do not necessarily reflect the views of the sponsors.

References

[1] N. Gorjian, L. Ma, M. Mittinity, P. Yarlagadda, and Y. Sun. “A Review on Degradation Models in Reliability Analysis.” Proceedings of the 4th World Congress on Engineering Asset Management (WCEAM 2009), 369-384, 2009.

[2] F. A. V. Bazán and A. T. Beck, “Stochastic process corrosion growth models for pipeline reliability,” Corrosion Science, vol. 74, pp. 50–58, Sep. 2013.

[3] M. Mahmoodian, Reliability and maintainability of in-service pipelines. 2018.

[4] R. Nyborg, “Overview of CO2 corrosion models for wells and pipelines,” in NACE - International Corrosion Conference Series, 2002, vol. 2002-April.

[5] H. R. Vanaei, A. Eslami, and A. Egbewande, “A review on pipeline corrosion, in-line inspection (ILI), and corrosion growth rate models,” International Journal of Pressure Vessels and Piping. 2017.

[6] W. M. A. Mohammad Zubir, I. Abdul Aziz, and J. Jaafar, “Evaluation of machine learning algorithms in predicting CO2 internal corrosion in oil and gas pipelines,” in Advances in Intelligent Systems and Computing, 2019, vol. 859.

[7] S. Glen. “Mean Absolute Percentage Error (MAPE).” Statisticshowto.com. Accessed June 28, 2021. [Online]. Available: https://www.statisticshowto.com/mean-absolutepercentage-error-mape/

[8] J. Brownlee. “11 Classical Time Series Forecasting Methods in Python (Cheat Sheet).” Machinelearningmastery.com. Accessed July 20, 2021. [Online]. Available: https://machinelearningmastery.com/time-seriesforecasting-methods-in-python-cheat-sheet/