

**SUPERVISED MACHINE LEARNING MODELS  
FOR PREDICTING THE MAXIMUM DEPTH OF  
CORROSION DEFECTS BASED ON  
HISTORICAL IN-LINE INSPECTION DATA**

BY

**EYAD ABDULLAH KHALIFAH ALSHAYE**

A Thesis Presented to the  
**COLLEGE OF GRADUATE AND INTERDISCIPLINARY STUDIES**  
**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**  
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the  
Requirements for the Degree of

**MASTER OF SCIENCE**

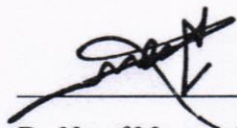
**In**

**MATERIAL SCIENCE AND ENGINEERING**

**December 2024**

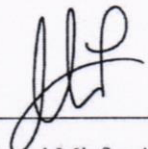
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS  
DHAHRAN- 31261, SAUDI ARABIA  
**COLLEGE OF GRADUATE AND INTERDISCIPLINARY STUDIES**

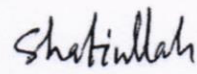
This thesis, written by Eyad Abdullah Alshaye under the direction of his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN MATERIALS SCIENCE & ENGINEERING.

  
\_\_\_\_\_  
Dr. Nayef Mesnad Alsaifi  
Department Chairman

  
\_\_\_\_\_  
Prof. Suliman S. Al-Homidan  
Dean of Graduate Studies

\_\_\_\_\_  
Date

  
\_\_\_\_\_  
Dr. Atif Saeed Misfer AlZahrani  
(Advisor)

  
\_\_\_\_\_  
Dr. Md Shafiullah  
(Co-Advisor)

  
\_\_\_\_\_  
Dr. Abduljabar Qassam Al-Sayoud  
(Member)



© Eyad Abdullah Alshaye

2024

To my beloved parents and family, whose unwavering support and encouragement have been the foundation of my journey. This thesis is dedicated to you.

## **ACKNOWLEDGMENTS**

I would like to express my deepest gratitude to Allah, whose blessings and guidance are beyond measure.

I am profoundly thankful to my parents, who instilled in me the values of hard work and a passion for pursuing science. Their unwavering support has been a constant source of motivation throughout my academic journey. I also extend my heartfelt appreciation to my family, whose encouragement and love have been invaluable during this time.

I would like to sincerely thank my research committee members, Dr. Atif AlZahrani, Dr. Md Shafiullah, and Dr. Abduljabar Al-Sayoud. Their contributions cannot be overstated; they have guided me through the entire research process, from formulating my ideas to reviewing my work and manuscripts. This thesis would not have reached its current form without their insightful feedback and support.

Additionally, I would like to express my gratitude to the Materials Science and Engineering Department, its faculty, and my fellow researchers. They have fostered a research-focused atmosphere that has greatly enriched my learning experience and inspired me to strive for excellence.

Finally, I would like to thank KFUPM for their assistance and support in facilitating postgraduate studies for part-time students. Their commitment to education has made a significant difference in my academic journey.

# TABLE OF CONTENTS

|                                                                                                                                                            |     |
|------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| ACKNOWLEDGMENTS .....                                                                                                                                      | III |
| TABLE OF CONTENTS.....                                                                                                                                     | V   |
| LIST OF TABLES.....                                                                                                                                        | VII |
| LIST OF FIGURES.....                                                                                                                                       | IX  |
| LIST OF ABBREVIATIONS.....                                                                                                                                 | XI  |
| ABSTRACT .....                                                                                                                                             | XIV |
| ملخص الرسالة .....                                                                                                                                         | XVI |
| CHAPTER 1 INTRODUCTION.....                                                                                                                                | 1   |
| 1.1 In-Line Inspection .....                                                                                                                               | 1   |
| 1.2 Service-Based Corrosion Prediction Machine Learning Models .....                                                                                       | 4   |
| 1.3 Contribution Of This Work.....                                                                                                                         | 8   |
| CHAPTER 2 DATA ACQUISITION AND DESCRIPTION .....                                                                                                           | 12  |
| 2.1 Data Acquisition .....                                                                                                                                 | 12  |
| 2.2 Data Description.....                                                                                                                                  | 13  |
| CHAPTER 3 SUPERVISED MACHINE LEARNING MODELS FOR PREDICTING THE<br>MAXIMUM DEPTH OF CORROSION DEFECTS BASED ON HISTORICAL IN-LINE<br>INSPECTION DATA ..... | 20  |

|                                                                                                                                                                         |               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| <b>3.1 Methods .....</b>                                                                                                                                                | <b>20</b>     |
| 3.1.1 Data Cleaning and Pre-Processing .....                                                                                                                            | 20            |
| 3.1.2 Machine Learning Models.....                                                                                                                                      | 24            |
| 3.1.3 Proposed Method .....                                                                                                                                             | 25            |
| <b>3.2 Results and Discussions.....</b>                                                                                                                                 | <b>27</b>     |
| 3.2.1 Models Screening.....                                                                                                                                             | 27            |
| 3.2.2 Final Model .....                                                                                                                                                 | 31            |
| <br><b>CHAPTER 4 TREE-BASED MACHINE LEARNING MODELS FOR PREDICTING THE<br/>MAXIMUM DEPTH OF CORROSION DEFECTS BASED ON HISTORICAL IN-LINE<br/>INSPECTION DATA .....</b> | <br><b>37</b> |
| <b>4.1 Methods .....</b>                                                                                                                                                | <b>37</b>     |
| 4.1.1 Data Cleaning and Pre-Processing .....                                                                                                                            | 37            |
| 4.1.2 Machine Learning Models.....                                                                                                                                      | 38            |
| 4.1.3 Proposed Method .....                                                                                                                                             | 40            |
| <b>4.2 Results and Discussions.....</b>                                                                                                                                 | <b>41</b>     |
| 4.2.1 Initial Models .....                                                                                                                                              | 41            |
| 4.2.2 Final Model .....                                                                                                                                                 | 48            |
| <br><b>CHAPTER 5 RECOMMENDATIONS AND LIMITATIONS.....</b>                                                                                                               | <br><b>56</b> |
| <br><b>CHAPTER 6 CONCLUSIONS .....</b>                                                                                                                                  | <br><b>58</b> |
| <br><b>REFERENCES.....</b>                                                                                                                                              | <br><b>60</b> |
| <br><b>VITAE .....</b>                                                                                                                                                  | <br><b>64</b> |

# LIST OF TABLES

|                                                                                                                                                                                                |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Table 1:</b> A summary of the machine learning models developed to predict defect maximum depth and corrosion rate.....                                                                     | 6  |
| <b>Table 2:</b> The number of data points (defects) obtained from ILI runs for each inspection year [15].....                                                                                  | 13 |
| <b>Table 3:</b> Summary statistics for the target variable (Maximum depth) for all ILI runs as well as the training dataset. All quantities have a unit of millimeter.....                     | 16 |
| <b>Table 4:</b> Recommended pre-processing methods for different models, the packages used for implementation, and the number of tuning parameters [23] .....                                  | 23 |
| <b>Table 5:</b> Naive forecasting benchmark value for all consecutive years in the ILI dataset... ..                                                                                           | 25 |
| <b>Table 6:</b> A summary table showing the best-performing sub-model from all the trained models ranked by RMSE. It also shows the training time in seconds required to train each model..... | 29 |
| <b>Table 7:</b> The results of Boosted decision trees (the best-performing model) on the full training data as well as on the testing data along with the model hyperparameter values .....    | 33 |
| <b>Table 8:</b> Summary statistics of the predictions of Boosted decision trees (the best-performing model) on the testing set .....                                                           | 35 |
| <b>Table 9:</b> Software packages and versions used in machine learning model development and hyperparameter tuning .....                                                                      | 40 |

|                                                                                                                                                                                                                                                                  |    |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Table 10:</b> Hyperparameters used for each model, along with the best value for each hyperparameter. Square brackets indicate the range considered during tuning. A dash (-) indicates that the corresponding parameter is not applicable for the model..... | 43 |
| <b>Table 11:</b> Evaluation metrics for each model, along with their relative performance compared to the Naive forecasting benchmark.....                                                                                                                       | 47 |
| <b>Table 12:</b> Training time, in seconds, for each model developed in this study .....                                                                                                                                                                         | 47 |
| <b>Table 13:</b> Statistical summary of the final model predictions on the test set. Q1 and Q3 denote the first and third quartiles, respectively .....                                                                                                          | 51 |

# LIST OF FIGURES

|                                                                                                                                                                                                                                                                                                                      |    |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Figure 1:</b> (a) a schematic diagram showing the structure and components that make the In-Line Inspection tool. (b) ILI tool inserted into a pipeline [7] .....                                                                                                                                                 | 2  |
| <b>Figure 2:</b> Schematic illustrations of parameters associated with pipeline defects during ILI run [10] .....                                                                                                                                                                                                    | 3  |
| <b>Figure 3:</b> Comparison of machine learning approaches for predicting pipeline corrosion                                                                                                                                                                                                                         | 9  |
| <b>Figure 4:</b> Histograms of the number of defects for equally spaced maximum depth intervals for each ILI run. Each bin represents a 0.07 mm maximum depth interval .....                                                                                                                                         | 15 |
| <b>Figure 5:</b> Histograms showing (from top to bottom): defect location relative to the pipeline origin, defect location relative to the pipe joint, circumferential location of the defect, and the defect's length and width for all inspection runs .....                                                       | 18 |
| <b>Figure 6:</b> Heat map illustrating the correlation between the variables utilized in this study .....                                                                                                                                                                                                            | 19 |
| <b>Figure 7:</b> Overview of the research method.....                                                                                                                                                                                                                                                                | 27 |
| <b>Figure 8:</b> Comparison of (RMSE) across all models developed in this research. The marker points indicate the RMSE for each grid search combination during hyperparameter tuning. Additionally, the RMSE of the Naïve forecasting benchmark, as well as the final model, are indicated by horizontal lines..... | 31 |
| <b>Figure 9:</b> A plot of the Error Residuals Vs the model predictions on the testing set. Darker colors indicate a higher density of points .....                                                                                                                                                                  | 35 |

|                                                                                                                                                                                                                                       |    |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Figure 10:</b> A histogram showing the distribution of Error residuals for the model predictions on the testing set. There are 67 bins; each one is approximately 0.1 mm in width .....                                            | 36 |
| <b>Figure 11:</b> RMSEs for all hyperparameter tuning trials conducted for each model developed in this study. The trials are sorted by RMSE values.....                                                                              | 46 |
| <b>Figure 12:</b> Parallel coordinate plot of the final model hyperparameters. RMSE is represented in two ways: by the last column on the right and through the color gradient.....                                                   | 48 |
| <b>Figure 13:</b> Error residuals of the final model predictions on the test set plotted against the corresponding model predictions .....                                                                                            | 50 |
| <b>Figure 14:</b> Histogram of the error residuals from the final model predictions on the test set, with each bin representing an interval of 0.01 mm.....                                                                           | 51 |
| <b>Figure 15:</b> Feature importance derived from SHAP analysis of the final model on the training data for each ILI run .....                                                                                                        | 53 |
| <b>Figure 16:</b> Feature importance derived from SHAP analysis of the final model on the training data, with feature values indicated by color .....                                                                                 | 54 |
| <b>Figure 17:</b> The first four levels of the first tree in the final model developed using all training data. The plot displays observations with the highest and lowest residuals, with 1,000 observations for each category ..... | 54 |

## LIST OF ABBREVIATIONS

|             |   |                                                   |
|-------------|---|---------------------------------------------------|
| <b>ANN</b>  | : | Artificial Neural Network                         |
| <b>API</b>  | : | American Petroleum Institute                      |
| <b>ASNT</b> | : | American Society for Nondestructive Testing       |
| <b>BART</b> | : | Bayesian Additive Regression Trees                |
| <b>CART</b> | : | Classification And Regression Trees               |
| <b>CNN</b>  | : | Convolution Neural Network                        |
| <b>CPU</b>  | : | Central Processing Unit                           |
| <b>CR</b>   | : | Corrosion Rate                                    |
| <b>CUDA</b> | : | Compute Unified Device Architecture               |
| <b>CV</b>   | : | Cross Validation folds                            |
| <b>DART</b> | : | Dropouts meet multiple Additive Regression Trees  |
| <b>DNN</b>  | : | Deep Neural Network (synonymous with <b>MLP</b> ) |
| <b>EDA</b>  | : | Exploratory Data Analysis                         |
| <b>GPU</b>  | : | Graphics Processing Unit                          |

|                      |   |                                                      |
|----------------------|---|------------------------------------------------------|
| <b>ILI</b>           | : | In-line Inspection                                   |
| <b>KNN</b>           | : | K-Nearest Neighbours                                 |
| <b>L.mm</b>          | : | Defect maximum length in millimeters                 |
| <b>LightGBM</b>      | : | Light Gradient-Boosting Machine                      |
| <b>MAE</b>           | : | Mean Absolute Error                                  |
| <b>MAPE</b>          | : | Mean Absolute Percentage Error                       |
| <b>MARS</b>          | : | Multivariate Adaptive Regression Splines.            |
| <b>MD.mm</b>         | : | Feature maximum depth in millimeters.                |
| <b>MFL</b>           | : | Magnetic Flux Leakage                                |
| <b>MLP</b>           | : | Multilayer Perceptron (synonymous with <b>DNN</b> ). |
| <b>PLS</b>           | : | Partial Least Squares.                               |
| <b>POF</b>           | : | Pipeline Operator Forum                              |
| <b>Q1</b>            | : | First quartile or 25th percentile                    |
| <b>Q3:</b>           | : | Third quartile or 75 <sup>th</sup> percentile        |
| <b>R<sup>2</sup></b> | : | Coefficient of determination                         |
| <b>RBF</b>           | : | Radial Basis Function.                               |
| <b>RMSE</b>          | : | Root Mean Squared Error                              |

|                 |   |                                                                                    |
|-----------------|---|------------------------------------------------------------------------------------|
| <b>RNNs</b>     | : | Recurrent Neural Networks                                                          |
| <b>SHAP</b>     | : | SHapley Additive exPlanations                                                      |
| <b>SIPO.deg</b> | : | The most significant circumferential location of the defect measured in degrees    |
| <b>SIPRD.m</b>  | : | Distance in meters from the most significant point of the defect to the pipe joint |
| <b>SPD.m</b>    | : | Distance from the defect starting point to the pipeline origin in meters           |
| <b>SVM</b>      | : | Support Vector Machines.                                                           |
| <b>TPE</b>      | : | Tree-structured Parzen Estimator                                                   |
| <b>W.mm:</b>    |   | Defect maximum width in millimeters                                                |
| <b>XGBoost</b>  | : | eXtreme Gradient Boosting                                                          |

# ABSTRACT

Full Name : Eyad Abdullah K Alshaye

Thesis Title : Supervised Machine Learning Models For Predicting The Maximum  
Depth Of Corrosion Defects Based On Historical In-Line Inspection  
Data

Major Field : Material Science and Engineering

Date of Degree : December 2024

Oil and gas pipelines serve as the primary means of fluid transportation in the industry due to their efficiency, reliability, and cost-effectiveness. However, pipeline corrosion poses a significant risk, leading to loss of containment, operational interruptions, and potential loss of life. Recent trends in the introduction of novel and corrosive fluid services, such as hydrogen and carbon dioxide (CO<sub>2</sub>), are expected to exacerbate corrosion-related issues and failures. Consequently, pipeline inspection techniques, particularly In-Line Inspection (ILI), are playing an increasingly vital role in corrosion monitoring. This thesis establishes a framework for utilizing ILI data to develop machine learning models for corrosion prediction. To demonstrate this framework, various linear, non-linear, and tree-based machine learning techniques were employed with the objective of predicting the maximum depth of corrosion defects based exclusively on historical ILI data. In the first study, a

limited training and hyperparameter tuning strategy was employed. Despite this, the best model outperformed the Naïve forecasting benchmark by 28.30% on the test set, with a Root Mean Square Error (RMSE) of 0.406 mm. When more extensive training and hyperparameter tuning strategies were utilized, the best model achieved a 41.5% improvement over the same benchmark, predicting the maximum depth with an RMSE of 0.368 mm. Notably, the accuracy of these models exceeded that of the majority of service-based corrosion prediction models found in the literature. This success was achieved using a historical ILI dataset characterized by extreme variations in maximum depth distribution and changes in reporting criteria. These findings suggest that the proposed framework is a superior alternative to service-based models, particularly given the vast amount of ILI data available to oil and gas pipeline operators. Additionally, the service-agnostic nature of this framework supports the integration of process and service parameters, which can further enhance the efficacy of machine learning models for corrosion prediction.

## ملخص الرسالة

الاسم الكامل: إياد عبدالله خليفه الشايع

عنوان الرسالة: نماذج تعلم الآلة للتنبؤ بالعمق الأقصى لعبوب التآكل استنادًا إلى بيانات الفحص الداخلي التاريخية

التخصص: هندسة وعلم المواد

تاريخ الدرجة العلمية: ديسمبر 2024

تعتبر أنابيب النفط والغاز الوسيلة الأساسية لنقل الموائع في الصناعة، وذلك بفضل كفاءتها وموثوقيتها وفعاليتها من حيث التكلفة. ومع ذلك، فإن تآكل الأنابيب يشكل خطرًا كبيرًا، مما قد يؤدي إلى فقدان الاحتواء، وانقطاع العمليات، واحتمالية فقدان الأرواح. من المتوقع أن تؤدي الاتجاهات الحديثة في إدخال موائع جديدة و شديدة التآكلية، مثل الهيدروجين وثاني أكسيد الكربون، إلى تفاقم المشكلات والفشل المرتبطين بالتآكل. وبالتالي، تلعب تقنيات فحص الأنابيب، وخاصة الفحص الداخلي، دورًا متزايد الأهمية في مراقبة التآكل. تؤسس هذه الرسالة إطارًا لاستخدام بيانات الفحص الداخلي لتطوير نماذج تعلم الآلة للتنبؤ بالتآكل. لتوضيح هذا الإطار، تم استخدام تقنيات تعلم الآلة المختلفة، بما في ذلك النماذج الخطية وغير الخطية ونماذج الأشجار، بهدف التنبؤ بالعمق الأقصى لعبوب التآكل مع الاستناد حصريًا على بيانات الفحص الداخلي التاريخية. في الدراسة الأولى، تم استخدام استراتيجية تدريب محدودة الفعالية. على الرغم من ذلك، تفوق النموذج الأفضل في هذه الدراسة على معيار التنبؤ البسيط بنسبة 28.30% في مجموعة الاختبار، مع خطأ متوسط الجذر التربيعي قدره 0.406 مم. عندما تم استخدام استراتيجيات تدريب أكثر شمولاً، حقق النموذج الأفضل تحسناً بنسبة 41.5% مقارنة بنفس المعيار، متنبئاً بالعمق الأقصى قدره 0.368 مم. ومن الجدير بالذكر أن دقة هذه النماذج تجاوزت تلك الخاصة بمعظم نماذج التنبؤ بالتآكل المعتمدة على الموائع الموجودة في الأوراق العلمية المنشورة. تم تحقيق هذا النجاح باستخدام مجموعة بيانات تاريخية للفحص الداخلي تتميز بتيابينات شديدة في توزيع العمق الأقصى وتغيرات في معايير التقرير. تشير هذه النتائج إلى أن الإطار المقترح يعد بديلاً متفوقاً على النماذج المعتمدة على الخدمات، خاصةً بالنظر إلى الكمية الكبيرة من بيانات الفحص الداخلي المتاحة لمشغلي أنابيب النفط والغاز. بالإضافة إلى ذلك، يدعم الطابع غير المرتبط بالموائع لهذا الإطار دمج بيانات إضافية تتعلق بنوع الموائع وطرق التشغيل، مما يمكن أن يعزز بشكل أكبر فعالية نماذج تعلم الآلة في التنبؤ بالتآكل.

# **Chapter 1**

## **INTRODUCTION**

### **1.1 In-Line Inspection**

Pipelines are the primary means of fluid transportation utilized in the oil and gas industry, with an estimated length of 2.13 million kilometers worldwide in 2024 [1]. This length is expected to increase, as this method of transportation has proven to be reliable, safe, and economical compared to other means [2]. However, pipeline corrosion remains a chronic issue faced by operators due to the inherent corrosivity of fluids commonly encountered in the oil and gas sector [3]. Furthermore, as the industry expands to include corrosive fluids such as hydrogen and carbon dioxide, the challenges associated with corrosion become even more complex [4], [5]. Consequently, Non-Destructive Testing (NDT) techniques and asset integrity solutions play a crucial role in ensuring the safe and reliable operation of pipelines. Among these techniques, in-line inspection (ILI) is the primary NDT method employed for pipeline inspection. In this method, a tool equipped with multiple ultrasonic (UT) and Magnetic Flux Leakage (MFL) sensors is inserted into the pipeline to obtain thickness measurements as shown in Figure 1 [6], [7]. Notably, in some regions, ILI is

mandated by law as one of the permissible methods for assessing pipelines, alongside direct assessment and hydrostatic pressure testing [8], [9].

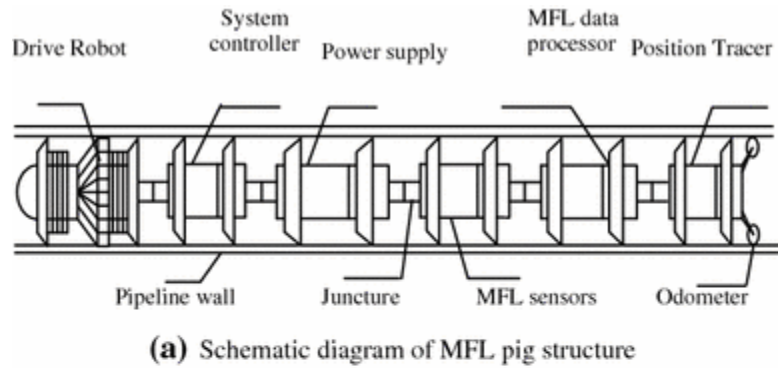


**Figure 1: (a) a schematic diagram showing the structure and components that make the In-Line Inspection tool. (b) ILI tool inserted into a pipeline [7]**

As illustrated in Figure 1, the tool is equipped with sensors, an odometer, an MFL data processor, and other components. The tool, along with its internal components, is isolated from the fluid used to propel it along the pipeline [6]. The odometer and position tracer are responsible for tracking the distance traveled relative to the pipeline's starting point. Sensors are strategically distributed across the internal diameter of the pipeline to measure the characteristics of defect features, as depicted in Figure 2 [10]. Additionally, data acquisition and storage devices are incorporated to retain the collected data for further

analysis and application. Finally, various signal processing techniques are employed to convert the MFL/UT signals into dimensional measurements of the variables under investigation.

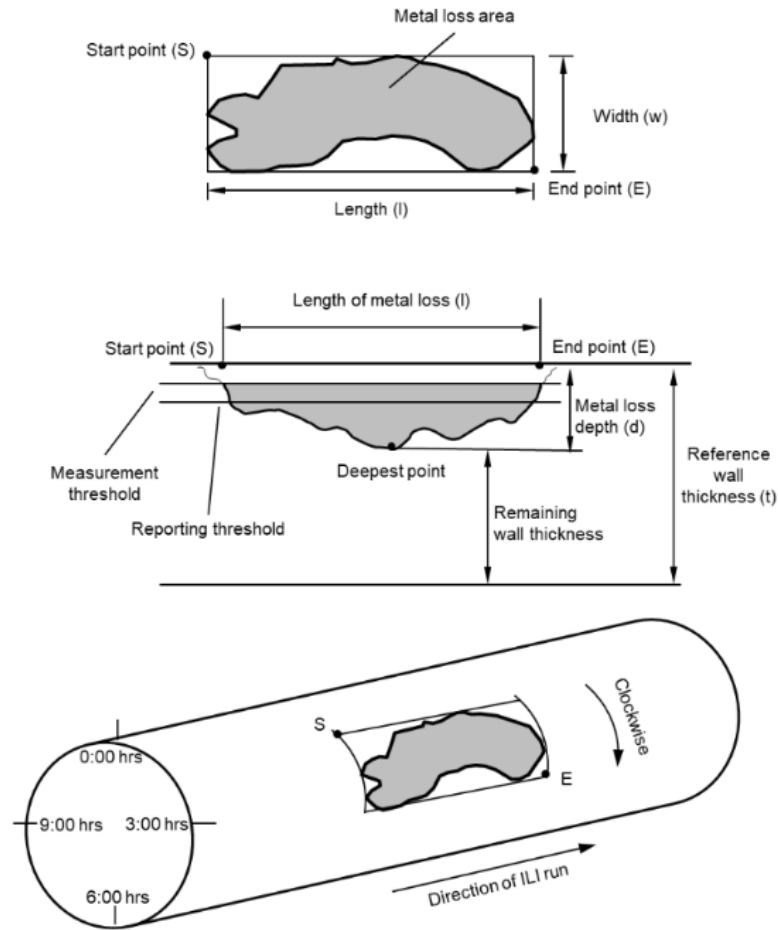


**Figure 2: Schematic illustrations of parameters associated with pipeline defects during ILI run [10]**

The ILI process is governed by various international standards, including those set by the Pipeline Operator Forum (POF) and the American Petroleum Institute (API). For instance, API STD 1163 specifies the qualification of ILI systems, while ASNT ILI-PQ outlines the qualification and certification requirements for ILI personnel [11], [12]. Additionally, POF 100 stipulates specifications and requirements for ILI of pipelines [10], and POF 110

defines the specifications for a universal data format for ILI [13]. To further enhance the reliability of ILI, POF 310 provides recommendations for field verification, helping to distinguish between systematic issues and statistical outliers [14]. This comprehensive framework ensures the availability of extensive historical data reflecting the actual conditions of pipelines, thereby facilitating the development of generalizable and accurate machine learning models.

## **1.2 Service-Based Corrosion Prediction Machine Learning Models**

Despite the advantages of utilizing historical ILI data for machine learning model development, no published research has yet attempted this approach. This gap in the literature may be attributed to the limited availability of ILI data, as the only public dataset of actual ILI data was released in late 2021 [15]. Conversely, numerous researchers have focused on predicting corrosion defect maximum depth and corrosion rate using data that reflect physical, mechanical, process, and operational parameters of oil and gas pipelines. Although this focus is not directly related to the analysis presented in this paper, examining such machine learning models is constructive for highlighting both the advantages and shortcomings of various approaches to corrosion prediction, as well as potential remedies for these issues.

Several machine learning models have been developed using a dataset comprising measurements of buried coated steel pipes in Southern Mexico [16]. This dataset contains 12 predictors measuring different attributes of the pipeline, such as chemical concentrations, resistivity, soil type, and coating condition. The target variable is the measured maximum pitting corrosion depth, with a total of 259 observations collected over

a three-year period. Various supervised machine learning techniques were employed to predict the maximum pitting depth, with the most accurate model achieving a root mean square error (RMSE) of 0.2909 mm on the test set. This model utilized a hybrid approach combining support vector regression and the Firefly Algorithm, as shown in Table 1 [17].

In addition to the aforementioned models, other studies have developed predictive models for corrosion rate and metal loss in pipelines by employing different parameters. For example, parameters representing the geometrical profile of the pipeline, fluid dynamic multiphase variables obtained through simulations, and two deterministic corrosion models were used as training variables for an Artificial Neural Network (ANN) model to predict corrosion rate and metal loss [18]. Although the exact number of data points was not reported, the author noted that it was limited. Another dataset, consisting of four predictors measuring temperature, CO<sub>2</sub> partial pressure, flow velocity, and pH, was utilized to predict CO<sub>2</sub> corrosion rate [19]. Unlike the previous datasets, this dataset was derived from experiments rather than actual measurements, comprising 721 observations. In this case, the best-performing machine learning technique was support vector machines, which achieved an RMSE of 0.4069 on the test set. Lastly, a dataset containing 60 measurements of a 5.5 km multiphase oil and gas subsea pipeline was used to build a corrosion prediction model [20]. Eight training variables describing physical and process parameters were employed, with the best model also being a support vector machine, achieving an RMSE of 2.7% on the test set. Table 1 below presents a summary of the machine learning models developed to predict defect maximum depth and corrosion rate.

**Table 1: a summary of the machine learning models developed to predict defect maximum depth and corrosion rate**

| <b>Training Variables</b>                                                                                                                                       | <b>Target</b>      | <b>Best Performing Model</b>                                    | <b>No. Data Points</b>                          | <b>Accuracy Measure</b> | <b>Accuracy Result</b>            |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------|-----------------------------------------------------------------|-------------------------------------------------|-------------------------|-----------------------------------|
| Elevation, Inclination, Concavity,<br><br>Flow regime, Hold-up, Pressure, Gas flow, Total flow, Liquid velocity, Gas velocity, De Waard model, and NORSOK model | CR//Metal loss     | ANN                                                             | No specified, but small according to the paper. | MAPE                    | 31 [18]                           |
| Coating type, Soil type, Redox potential, pH, Resistivity, Water content, Bulk density, Chloride, Bicarbonate, Pipe-to-soil potential and Sulphate content      | Maximum depth (mm) | CatBoost                                                        | 259                                             | RMSE                    | 0.8330 [21]                       |
|                                                                                                                                                                 |                    | Kriging                                                         |                                                 |                         | 0.91443 [22]                      |
|                                                                                                                                                                 |                    | Hybrid model of Support Vector Regression and Firefly Algorithm |                                                 |                         | 0.2909 [17]                       |
| Temperature, CO <sub>2</sub> partial pressure, Flow velocity, and pH                                                                                            | CR                 | Least Square Support Vector Machine                             | 721 (experimental dataset)                      | MSE, RMSE*              | MSE = 0.1656 [19] RMSE = (0.4069) |

|                                                                                                                                                                        |    |                         |    |      |           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------|----|------|-----------|
| Temperature, Liquid holdup, HOL, Pressure, Partial pressure of CO <sub>2</sub> , Liquid flow rate, Wall shear stress (liquid phase), wall shear stress (gas phase), pH | CR | Support Vector Machines | 60 | RMSE | 2.7% [20] |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|-------------------------|----|------|-----------|

\* RMSE was not reported in the original paper, it was calculated by taking the square root of MSE.

Several conclusions can be drawn from the analysis of the current corrosion prediction machine learning models presented in the previous table. First, despite the limited training data, these models have achieved commendable accuracies. In fact, some models have outperformed deterministic techniques, such as the de Waard and NORSOK models [18]. This demonstrates the validity of employing machine learning to predict both corrosion rate and defect maximum depth. The observed accuracy can be attributed to the inclusion of all variables that govern the corrosion phenomenon during model development. However, this approach also introduces limitations, as these models are system-specific, require substantial data collection and processing, and do not account for the temporal aspect during model development. For instance, it is evident from Table 1 that collecting the datasets necessary for developing these models is challenging. This is reflected in the small training datasets and the repeated use of the same datasets across different studies. Furthermore, due to the specific choice of training variables, all models are system-specific and can only predict corrosion rate and maximum depth for certain phenomena, such as CO<sub>2</sub> corrosion and pitting corrosion in buried pipelines. It remains unclear whether these

models can be applied to systems with different distributions of training variables. For example, can they accurately predict CO<sub>2</sub> corrosion when its concentration exceeds the range of the dataset? Similar considerations apply to other corrosives, such as H<sub>2</sub>S and water content. Lastly, none of the models directly incorporate the time of inspection as a training variable, which may significantly limit their predictive capabilities by neglecting the dynamic nature of corrosion, especially if corrosion datasets are to be treated as time series data.

### **1.3 Contribution Of This Work**

The approach adopted in this research exclusively utilizes ILI datasets to develop maximum depth prediction models. This choice minimizes data collection and processing activities, as ILI inspections are conducted regularly as part of asset integrity requirements, regardless of their role in machine learning. Moreover, ILI datasets do not impose assumptions regarding the specific service conditions or corrosive substances present in the pipeline, resulting in general models applicable to all pipelines in the oil and gas industry. These two factors combined ensure that extensive datasets are readily available for training the models, leading to accurate and generalizable models that can be developed without significant investment in data collection and processing efforts. Additionally, these models are flexible and can be further enhanced by incorporating system-specific datasets, which is expected to improve accuracy and facilitate data-driven examinations of system-specific factors contributing to pipeline corrosion. Figure 3 below summarizes the model development workflow for each approach, highlighting both the advantages and limitations.

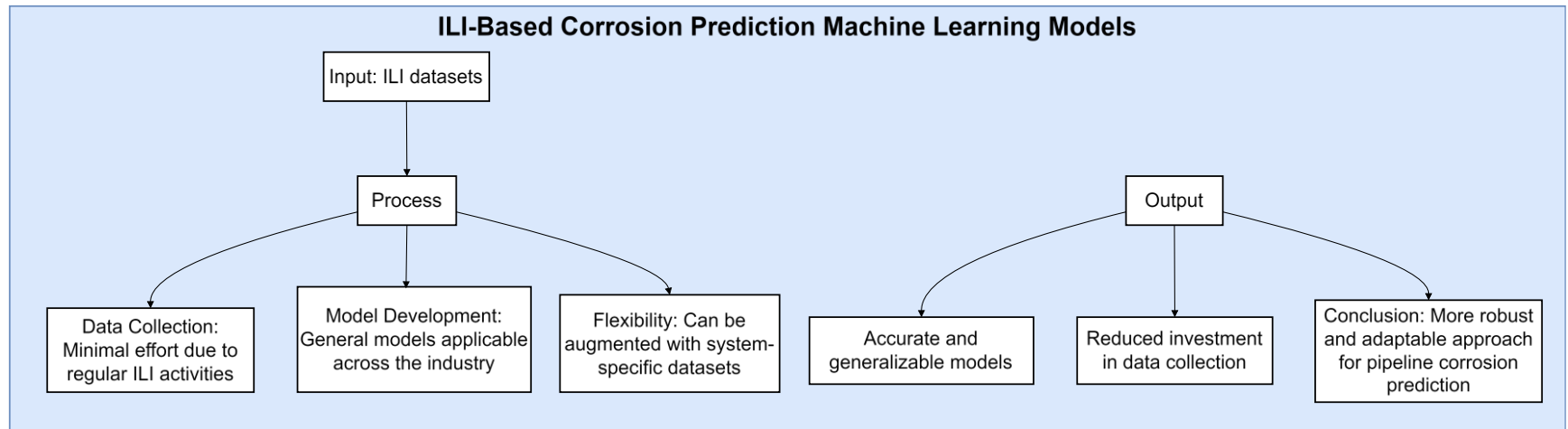
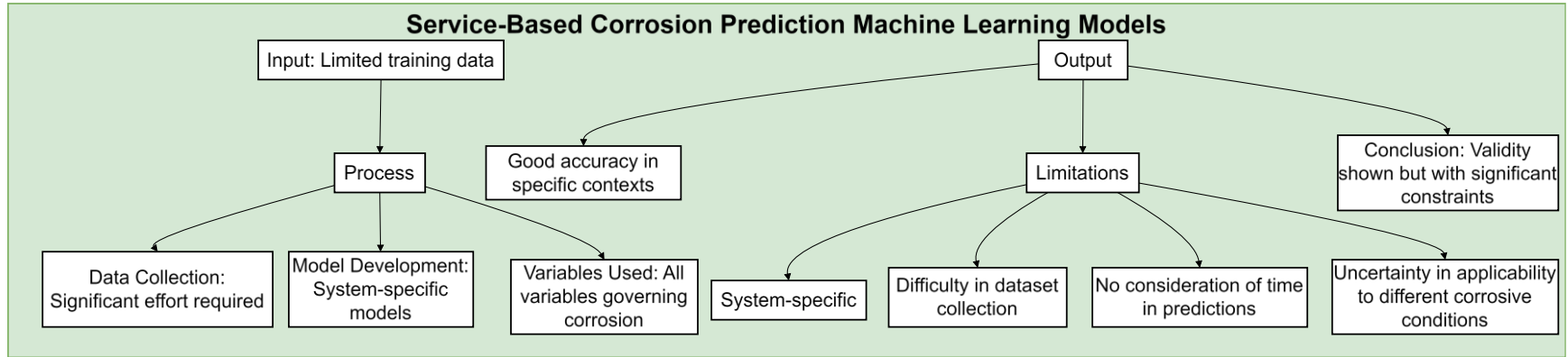


**Figure 3: Comparison of machine learning approaches for predicting pipeline corrosion**

This research explores the potential of supervised machine learning techniques to enhance the prediction of corrosion defects depth in pipelines, aiming to establish a reliable framework for integrating ILI with other asset integrity solutions. The following are the significant contributions of this work.

- Investigates the application of supervised machine learning techniques to predict the maximum depth of corrosion defects based on historical ILI dataset.
- Historical ILI data from a 250 km pipeline in the oil and gas industry is utilized, comprising four ILI runs conducted two years apart, totaling over 3 million defects.
- Analyzes 17 linear, non-linear, and tree-based machine learning techniques, focusing on two key training variables: inspection year and distance from the measurement location to the origin. The performance of these models will be compared to the Naïve forecasting benchmark to assess their capability in modeling ILI data.
- Analyzes four tree-based machine learning techniques: XGBoost, Random Forests, DART, and LightGBM with linear trees, employing six predictors representing the inspection year, defect location, circumferential orientation, and defect dimensions. The models will be trained using the Tree-structured Parzen Estimator (TPE) search algorithm and hyperband pruning with 500 trials for hyperparameter tuning. The performance of these models will be compared to a naïve forecasting benchmark to assess their capability in modeling ILI data.
- Establishes an alternative pipeline corrosion prediction framework that prioritizes the use of ILI datasets to develop service-agnostic models applicable to all pipelines in the oil and gas industry, as well as other industries utilizing similar techniques.

This thesis is organized as follows: Chapter 2 discusses the processes of data acquisition and description. Chapter 3 presents the first study, which surveys various machine learning techniques utilizing limited training data and hyperparameter tuning strategies. Chapter 4 introduces the second study, where the best models identified in Chapter 3 are further explored with additional training and hyperparameter tuning strategies. Subsequently,

Chapter 5 outlines the recommendations and limitations associated with the proposed framework, while Chapter 6 provides a conclusion to the thesis.

## **CHAPTER 2**

### **DATA ACQUISITION AND DESCRIPTION**

#### **2.1 Data Acquisition**

In this research, historical ILI data will be utilized to develop machine learning models. The dataset contains over 3 million observations collected across four inspection runs, as detailed in Table 2 [15]. It includes 15 features that characterize various attributes of each defect. These attributes can be categorized into four groups: pipeline/inspection run-specific features, the location of the defect relative to a reference point, circumferential orientation, and defect dimensions. For the purposes of this study, six attributes will be analyzed: the inspection year (Year), the distance in meters from the defect starting point to the pipeline origin (SPD.m), the distance of the most significant point of the defect relative to the pipe joint in meters (SIPRD.m), the most significant circumferential location of the defect measured in degrees (SIPO.deg), and the dimensions of the defect, including length, width, and maximum depth measured in millimeters (L.mm, W.mm, and MD.mm, respectively).

**Table 2: The number of data points (defects) obtained from ILI runs for each inspection year [15]**

| <b>Year</b> | <b>Data Points<br/>(Defects)</b> |
|-------------|----------------------------------|
| 1           | 735,755                          |
| 3           | 832,681                          |
| 5           | 759,683                          |
| 7           | 870,301                          |

## **2.2 Data Description**

Figure 4 illustrates the distribution of maximum depths recorded during ILI runs, with each bin measuring 0.07 mm in width. The majority of defects exhibit a maximum depth of 0.07 mm or less. This phenomenon can be attributed to the threshold of the ILI tool utilized for these measurements; consequently, any defect with a depth shallower than this threshold is reported as a defect with a depth of 0.07 mm [15]. The distributions of maximum depth exhibit significant variability over the years. Notably, years 5 and 7 demonstrate a pronounced exponential decay in the number of defects as depth increases. This variability is typical in real ILI datasets, where factors such as reporting criteria, ILI tool sensitivity, and the dynamic nature of corrosion contribute to differing distributions over time.

In addition to these observations, certain depth intervals show a marked absence of defects. Specifically, the first three intervals contain two bins with no reported defects, while the last interval displays four gaps within the first millimeter. Although it is anticipated that

the number of shallower defects will increase over time, this should not adversely impact the occurrence of deeper defects. For instance, despite the differing y-axis scales in the initial intervals, it is evident that numerous defects measuring 1 mm in depth are absent in the later intervals. Furthermore, the increase in the number of shallow defects should not come at the expense of the total number of defects; rather, the total number of defects should rise concurrently with the number of shallow defects as time progresses. However, a cross-reference of Figure 4 with Table 2 reveals no such trend. As a result, the average maximum depths for the four intervals, as well as the median, showed a downward trend except for Year 3 as shown in Table 3. In addition, the same table also shows a decreasing trend in the standard deviation that follows the same pattern as the average and median, where the lowest standard deviation is observed in Year 7, which is the testing set. This will affect the evaluation metrics that are dependent on the variance of the data, in particular,  $R^2$ , which increases if the variance of the data is high. As a result, it is expected that  $R^2$  of the training data will be much higher than that of the testing data even if other evaluation metrics do not justify such increase.

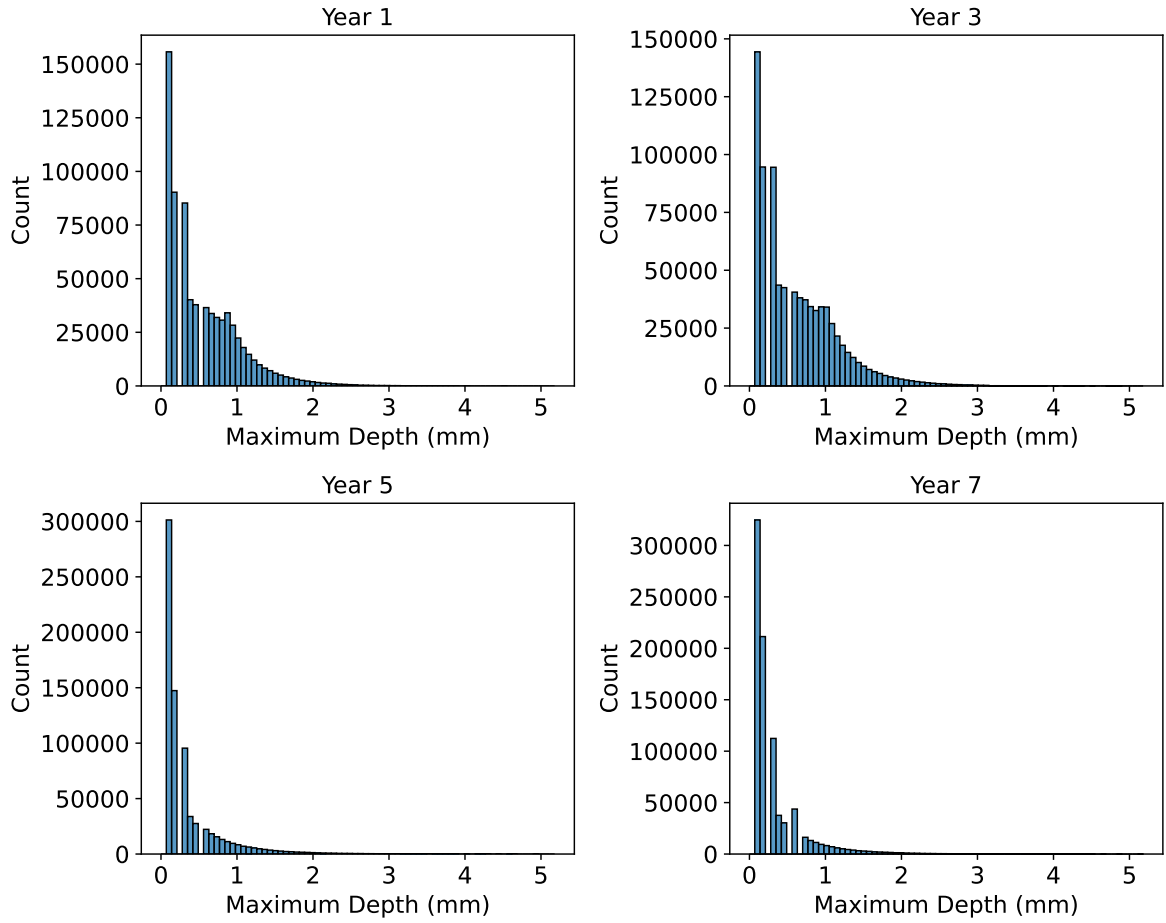


**Figure 4: Histograms of the number of defects for equally spaced maximum depth intervals for each ILI run.  
Each bin represents a 0.07 mm maximum depth interval**

**Table 3: Summary statistics for the target variable (Maximum depth) for all ILI runs as well as the training dataset. All quantities have a unit of millimeter**

| <b>Statistical Measure</b> | <b>Year 1</b> | <b>Year 3</b> | <b>Year 5</b> | <b>Year 7</b> | <b>Training Data</b> | <b>All Years</b> |
|----------------------------|---------------|---------------|---------------|---------------|----------------------|------------------|
| Min                        | 0.07          | 0.07          | 0.07          | 0.07          | 0.07                 | 0.07             |
| Q1                         | 0.14          | 0.21          | 0.07          | 0.07          | 0.07                 | 0.07             |
| Median                     | 0.42          | 0.49          | 0.14          | 0.14          | 0.35                 | 0.28             |
| Mean                       | 0.5599        | 0.6312        | 0.3446        | 0.3173        | 0.5151               | 0.4613           |
| Q3                         | 0.85          | 0.92          | 0.42          | 0.42          | 0.78                 | 0.71             |
| Max                        | 4.49          | 4.85          | 4.92          | 5.18          | 4.92                 | 5.18             |
| Standard Deviation         | 0.4834        | 0.5232        | 0.4248        | 0.3870        | 0.4955               | 0.4767           |

Figure 5 illustrates the distribution of various training variables across all inspection years. The distance from the defect starting point to the pipeline origin reveals distinct clusters with a higher number of defects. These clusters can be attributed to physical locations where corrosion is more pronounced, such as areas with steep slopes, regions experiencing increased or decreased fluid velocity, and locations associated with directional changes in flow (e.g., elbows, reducers). Additionally, areas characterized by low velocities may also contribute to these clusters. Although this parameter does not directly measure these factors, it encodes relevant information that can represent specific damage mechanisms, including erosion, under-deposit corrosion, and corrosion issues related to flow stagnation. Finally, the initial portion of the pipeline has significantly fewer defects compared to the rest of the pipeline.

There is a higher incidence of defects at both low and high values of the SIPRD.m variable, particularly in the vicinity of 12 meters. This phenomenon may be linked to field girth welding and coating issues commonly encountered during pipeline construction activities. Regarding the circumferential location of the maximum point of the defect, the sectors representing the top and bottom portions of the pipeline exhibit a greater number of defects compared to other segments. This observation may be attributed to specific corrosion damage mechanisms, such as top-of-the-line corrosion, under-deposit corrosion, and pitting corrosion. Finally, both defect length and width demonstrate a sharp decay in distribution as these values increase.

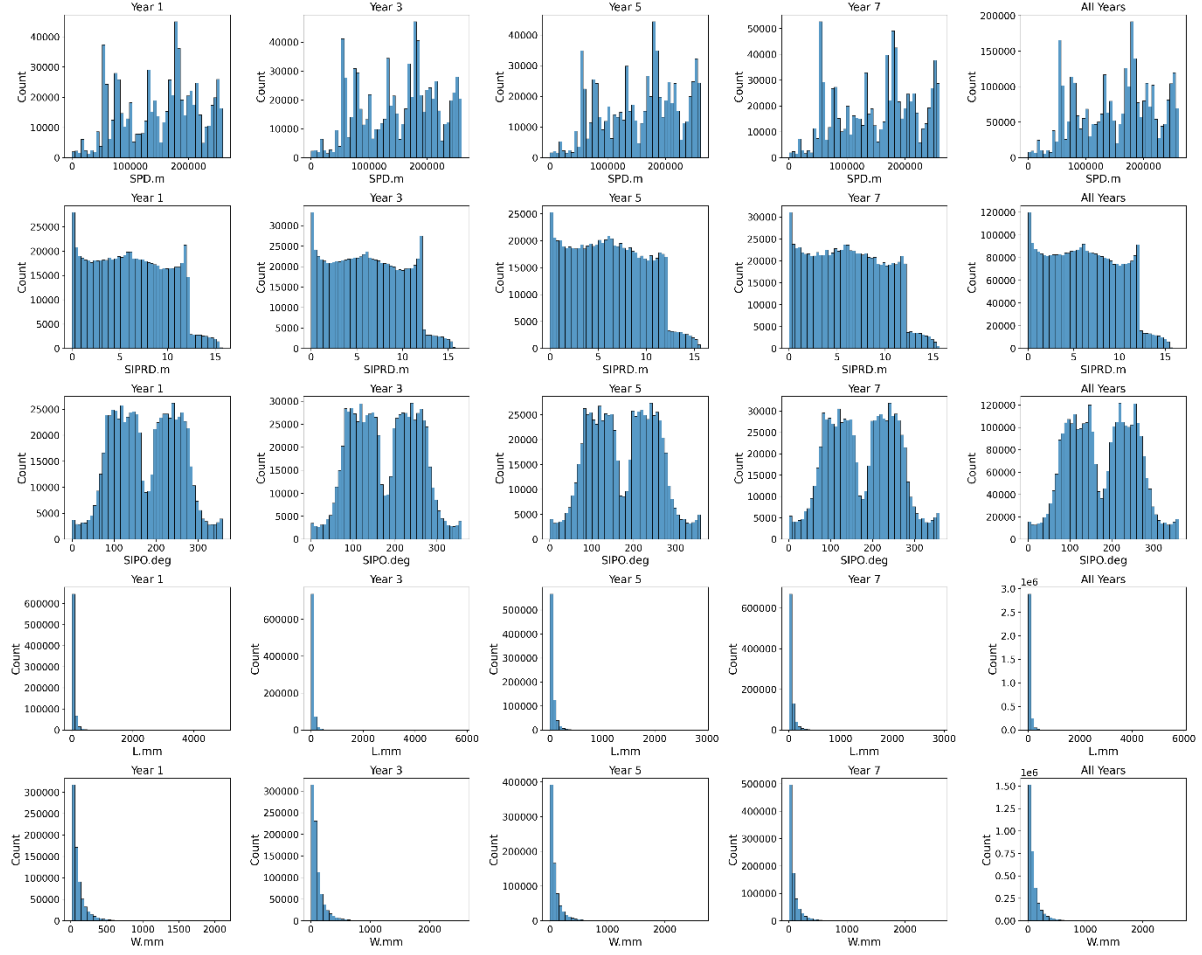


**Figure 5: Histograms showing (from top to bottom): defect location relative to the pipeline origin, defect location relative to the pipe joint, circumferential location of the defect, and the defect's length and width for all inspection runs**

Figure 6 illustrates the correlation between all training and target variables, revealing a negative correlation between inspection year and maximum depth, which is an artifact of the dataset as previously discussed. This trend may affect model accuracy, as a near-constant number of defects would typically suggest a positive correlation between maximum depth and inspection year. On the other hand, defect length and width are both positively correlated with maximum depth, and strongly correlated with each other. The latter observation may inflate their importance in feature importance calculations.



**Figure 6: Heat map illustrating the correlation between the variables utilized in this study**

# **CHAPTER 3**

## **SUPERVISED MACHINE LEARNING MODELS**

### **FOR PREDICTING THE MAXIMUM DEPTH OF**

### **CORROSION DEFECTS BASED ON HISTORICAL**

### **IN-LINE INSPECTION DATA**

#### **3.1 Methods**

##### **3.1.1 Data Cleaning and Pre-Processing**

Six types of transformations may be necessary before developing machine learning models. These transformations include creating dummy variables, removing predictors that exhibit only a single value, imputation of missing data, decorrelation, normalization, and general transformation [23]. In this analysis, the predictors utilized, specifically the inspection year and the distance from the defect to the pipeline origin, do not necessitate the first three transformations. Additionally, the correlation coefficient between these two predictors was computed as -0.0219, suggesting that decorrelation is unnecessary. However, Figure 4 and

Figure 5 illustrate the necessity for centering and scaling (normalization), which ensures that the predictors have a standard deviation of 1 and a mean of zero. Moreover, the requirement for data transformation is contingent upon the skewness of the data, which was calculated using Equation 3.1 presented below [23]:

$$skewness = \frac{\sum (x_i - \bar{x})^3}{(n - 1) v^{\frac{3}{2}}}$$

(Eq. 3.1)

$$where \quad v = \frac{\sum (x_i - \bar{x})^2}{(n - 1)}$$

Where,  $x$  is the predictor variable,  $n$  is the number of samples, and  $\bar{x}$  is the mean of the predictor. If the distribution of the predictor is almost symmetric, the value of the skewness metric will be close to zero. If the predictor exhibits a right or left skewness, however, the skewness metric will reflect that by diverging from zero in either the positive or negative direction based on the direction of skewness.

Several transformations, such as logarithmic, square root, or inverse-based, can be used. Instead, statistical methods can be employed to identify the most appropriate transformation by identifying a family of transformations. This approach was proposed by Box and Cox and is called Box-Cox transformation, which is given by [23]:

$$x^* = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(x) & \text{if } \lambda = 0 \end{cases}$$

(Eq. 3.2)

The skewness results confirmed that the predictors are skewed as the ratio of the largest to smallest value was 1530 and 5 for the distance from the pipeline origin and inspection year, respectively, and the sample skewness for the same variables are -0.201 and -0.0185, respectively. Based on this, the Box-Cox transformation in Eq. 3.2 was used to empirically identify an appropriate transformation, which yielded  $\lambda = 1$ . Consequently, the recommended pre-processing guidelines were used to identify if the machine learning technique requires pre-processing, as shown in Table 4 [23].  $\checkmark$  indicates that the method is required for the model, X indicates that it is not required, and  $\checkmark^*$  indicates that the pre-processing method was performed even though it is not necessarily required.

**Table 4: Recommended pre-processing methods for different models, the packages used for implementation, and the number of tuning parameters [23]**

| Category   | Model                  | Package       | Version | # Tuning Parameters | Normalize | Box-Cox |
|------------|------------------------|---------------|---------|---------------------|-----------|---------|
| Linear     | Linear regression      | brulee [24]   | 0.3.0   | 2                   | X         | ✓*      |
|            | PLS                    | mixOmics [25] | 6.26.0  | 2                   | ✓         | ✓       |
|            | Poisson regression     | glmnet [26]   | 4.1-8   | 2                   | X         | ✓*      |
| Non-Linear | MARS                   | earth [27]    | 5.3.3   | 1                   | X         | ✓*      |
|            | Bagged MARS            | earth         | 5.3.3   | 0                   | X         | ✓*      |
|            | MLP                    | brulee        | 0.3.0   | 5                   | ✓         | ✓       |
|            | KNN                    | kknn [28]     | 1.3.1   | 3                   | ✓         | ✓       |
|            | SVM (linear)           | kernlab [29]  | 0.9-32  | 2                   | ✓         | ✓       |
|            | SVM (polynomial)       | kernlab       | 0.9-32  | 4                   | ✓         | ✓       |
|            | SVM (RBF)              | kernlab       | 0.9-32  | 3                   | ✓         | ✓       |
| Tree-based | Bagged decision trees  | rpart [30]    | 4.1.23  | 0                   | X         | X       |
|            | BART                   | dbarts [31]   | 0.9-26  | 4                   | X         | X       |
|            | Boosted decision trees | xgboost [32]  | 1.7.7.1 | 7                   | X         | X       |
|            | Cubist rules           | Cubist [33]   | 0.4.2.1 | 3                   | X         | X       |
|            | Decision trees         | rpart         | 4.1.23  | 3                   | X         | X       |
|            | Random forest          | ranger [34]   | 0.16.0  | 2                   | X         | X       |
|            | Rule fit               | xrf [35]      | 0.2.2   | 8                   | ✓         | X       |

### 3.1.2 Machine Learning Models

To investigate the performance of maximum depth prediction using machine learning, this study will utilize 17 supervised machine learning techniques. These techniques are categorized into linear, non-linear, and tree-based machine learning methods. Specifically, we will employ linear regression, Poisson regression, and Partial Least Squares (PLS) for linear regression models. For non-linear regression models, we will utilize Multilayer Perceptron (MLP), Support Vector Machines (SVM) with linear, polynomial, and Radial Basis Function (RBF) kernels, Multivariate Adaptive Regression Splines (MARS), bagged MARS, and K-Nearest Neighbors (KNN). Finally, we will apply bagged decision trees, random forests, boosted decision trees, Bayesian Additive Regression Trees (BART), rule fit, cubist, and decision trees for tree-based regression models. Table 4 presents the models employed in this analysis, the selected packages for technique implementation and the number of tuning parameters utilized for each technique. All models used in this study were developed using R programming language (version 4.3.3) [36], along with the tidymodels machine learning framework [37]. All models were trained on an Intel i7-6820HK 2.70GHz CPU utilizing 4 cores in parallel.

The primary metric used to evaluate the effectiveness of each technique will be the Root RMSE. Additionally, performance metrics such as MAE and Mean Absolute Percentage Error (MAPE) will be reported for the final model on the test set. Since there are no available benchmarks for accuracy comparison and considering the time series nature of the data, the Naïve forecasting method has been selected for benchmarking. This method compares the previous and actual measurements to derive the evaluation metric, as illustrated in Eq. 3.3 [38], [39].

$$\text{Benchmark} = \text{RMSE}(MD_t - MD_{t-1}) \quad (\text{Eq. 3.3})$$

where MD represents the defect's maximum depth in mm, and the RMSE is computed for the difference between the current reading and the previous run's reading. Table 5 presents the number of matches with respect to consecutive ILI runs. It is evident that this method of matching possesses significant limitations, as it assumes that the distance measurements can be accurately utilized to track and match defects, and that the length of the defect remains unchanged between the two runs. Nevertheless, the values align with RMSEs obtained from other researchers, as illustrated in Table 1. The more conservative value of 0.5209 mm will be adopted as a benchmark in this study. Furthermore, other accuracy metrics such as MSE and MAPE can be calculated using the same approach.

**Table 5: Naive forecasting benchmark value for all consecutive years in the ILI dataset**

| <b>Year</b> | <b>Number matching defects</b> | <b>RMSE (mm)</b> |
|-------------|--------------------------------|------------------|
| Years 1&3   | 116                            | 0.5373           |
| Years 3&5   | 0                              | NA               |
| Years 5&7   | 32                             | 0.5209           |

### 3.1.3 Proposed Method

This research aims to build a maximum depth prediction regression model based on historical ILI data. The data is segregated by year (1, 3, 5, and 7), which does not reflect the pipeline's age but references the inspection year to the initial implementation of the ILI

program. The training dataset will include the first three inspection intervals, while the last year (Year 7) will be reserved for testing. The training variables are the distance from the defect to the pipeline origin measured in meters (SPD.m) and the inspection year (Year), while the target variable is the defect maximum depth measured in millimeters (MD.mm). The RMSE for each model will be compared to the Naïve forecasting benchmark to establish the model's effectiveness at predicting defect maximum depth. Additionally, the time required to train the model will also be reported.

The training strategy is split into two stages. The first is the initial screening of 17 models using randomly selected sample data from the training dataset to determine the best-performing technique. After that, the best model will be trained on the full training dataset and tested on the test set to determine the test accuracy of the model. The dataset used for the initial screening consists of 30,000 data points assembled by randomly selecting 10,000 data points from each year in the first three inspection intervals. Three Cross-Validation folds (CV) will be used to determine the model's RMSE using only one repetition. Furthermore, hyperparameter tuning will be done via a grid search of 10 combinations of tuning parameters. After that, the model with the lowest RMSE will be trained on the full training data using three cross-validation folds and 25 grid search combinations for hyperparameter tuning. Finally, the best-performing sub-model will be trained on the whole training data and tested on the test data to report the final training and testing RMSEs, respectively. The overall process is illustrated via a flowchart in **Error! Reference source not found..**

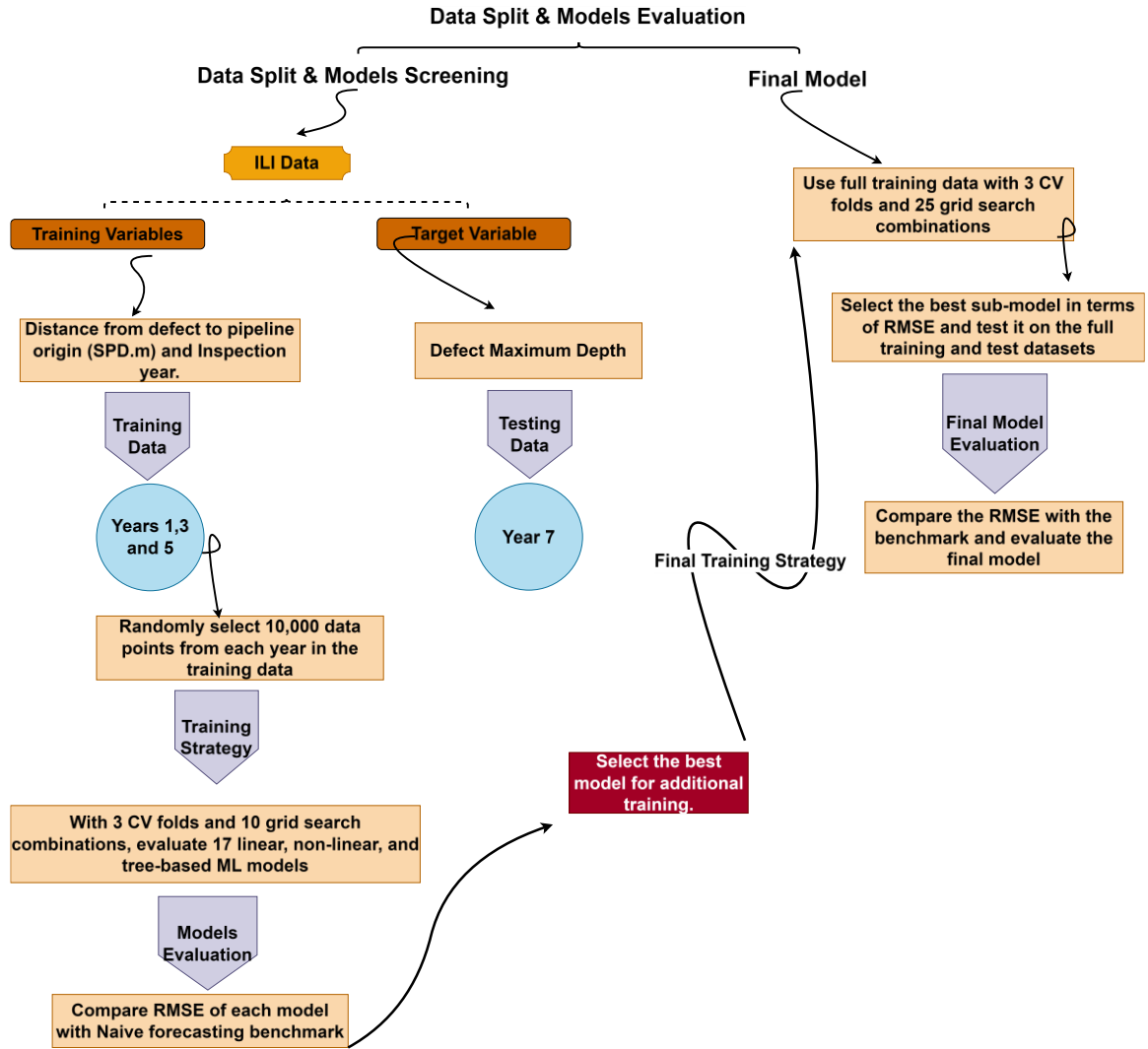


Figure 7: Overview of the research method

## 3.2 Results and Discussions

### 3.2.1 Models Screening

Table 6 ranks the 17 models developed for initial screening in terms of accuracy and performance relative to the Naïve forecasting benchmark. Out of all the models, only two performed worse than the benchmark: mainly bagged decision trees and linear regression.

The other models, which were the most accurate, outperformed the benchmark-boosted decision trees, achieving an 11.15% improvement over the benchmark in predicting the defect maximum depth. Furthermore, tree-based models consistently achieved higher scores than the other models, with the best five models in terms of accuracy, all belonging to this category. Conversely, linear models were generally the worst at this regression task, ranking 10<sup>th</sup>, 11<sup>th</sup>, and last in accuracy. Finally, non-linear models achieved acceptable results, except for SVM models, which had poor accuracy and training time performance. Based on these results, the boosted decision trees model was selected for additional training on the entire training dataset.

**Table 6: A summary table showing the best-performing sub-model from all the trained models ranked by RMSE. It also shows the training time in seconds required to train each model**

| <b>Rank</b> | <b>Model</b>           | <b>RMSE mean</b> | <b>RMSE STD</b> | <b>%Benchmark</b> | <b>Training Time (Sec)</b> |
|-------------|------------------------|------------------|-----------------|-------------------|----------------------------|
| 1           | Boosted decision trees | 0.4686337        | 0.002713014     | 11.15             | 256.14                     |
| 2           | BART                   | 0.4707769        | 0.002766971     | 10.65             | 945.06                     |
| 3           | Rule fit               | 0.4712407        | 0.002123441     | 10.54             | 1223.83                    |
| 4           | Random forest          | 0.4725822        | 0.002644525     | 10.22             | 155.01                     |
| 5           | Decision trees         | 0.4736607        | 0.002341969     | 9.97              | 5.66                       |
| 6           | KNN                    | 0.4750845        | 0.00202483      | 9.64              | 1410.31                    |
| 7           | Bagged MARS            | 0.4754932        | 0.002438454     | 9.55              | 13.36                      |
| 8           | MLP                    | 0.4757457        | 0.002689359     | 9.49              | 221.45                     |
| 9           | MARS                   | 0.4759113        | 0.002627042     | 9.45              | 2.48                       |
| 10          | PLS                    | 0.4868237        | 0.002548118     | 7.00              | 8.49                       |
| 11          | Poisson regression     | 0.4889379        | 0.002565004     | 6.54              | 6.97                       |
| 12          | Cubist rules           | 0.492583         | 0.003610034     | 5.75              | 161.41                     |
| 13          | SVM (RBF)              | 0.4926754        | 0.003157855     | 5.73              | 1357.64                    |
| 14          | SVM (polynomial)       | 0.4934075        | 0.003078422     | 5.57              | 466.61                     |

|    |                             |           |             |        |        |
|----|-----------------------------|-----------|-------------|--------|--------|
| 15 | SVM<br>(linear)             | 0.5058963 | 0.00284465  | 2.97   | 302.32 |
| 16 | Bagged<br>decision<br>trees | 0.5464708 | 0.001129019 | -4.68  | 49.56  |
| 17 | Linear<br>regression        | 0.5939556 | 0.0342727   | -12.30 | 11.46  |

To further evaluate the performance of all models, the RMSE results are illustrated in Figure 8, which provides insights into the accuracy of each grid search combination. Additionally, the Naïve forecasting benchmark is shown via the red horizontal line. Ignoring the two models that performed worse than the benchmark, only the KNN model performed worse than the benchmark in some grid search combinations, whereas the worst-performing combinations for the other models were all more accurate than the benchmark. In addition, the eight best-performing models all showed significant improvement as a result of hyperparameter tuning, except for BART, where its accuracy remained constant despite the different hyperparameters. This suggests that additional training, especially when coupled with a more extensive training strategy, might result in further improvement in defect maximum depth prediction. Finally, the number of tuning parameters greatly influenced each model's accuracy. Cross-referencing Table 6 with Table 4 shows that, out of the top five performing models, only one has two tuning parameters, while the others range from three to eight. These results show that, despite the unfavorable training strategy adopted in this research, most models performed better than the Naïve forecasting benchmark even during the least accurate grid search combination. Furthermore, most of the top-performing models showed improvements upon hyperparameter tuning. Adopting

a more exhaustive training strategy is expected to yield significant improvements in the accuracy of these models.

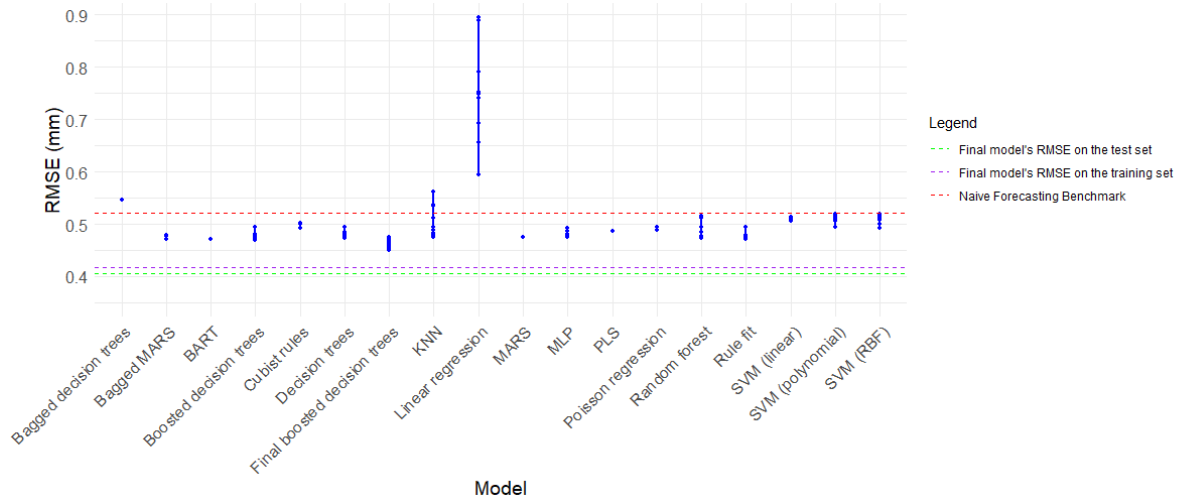


**Figure 8: Comparison of (RMSE) across all models developed in this research. The marker points indicate the RMSE for each grid search combination during hyperparameter tuning. Additionally, the RMSE of the Naïve forecasting benchmark, as well as the final model, are indicated by horizontal lines**

### 3.2.2 Final Model

The boosted decision trees technique was then trained on the whole training data utilizing three-fold cross-validation and 25 grid searches. Table 7 shows the tuning parameters of the most accurate sub-model, its RMSE on both the training and testing datasets and the MAE and MAPE of both the naive forecasting method and the final model on the test set. The accuracy of the model more than doubled compared to the training on the sample data, achieving a 24.64 % improvement over the benchmark. This improvement is attributed to the availability of more training data and the fine-tuning of the hyperparameters with a larger search space. This best sub-model was then tested on the testing data, and its accuracy exceeded the benchmark by 28.30 %, indicating that the model can predict defect depth with an error of  $\pm 0.406$  mm on average. Additional performance metrics such as

MAE and MAPE also surpassed the benchmark, showing 11.44% and 130.6% improvements, respectively. The test set RMSE is better than the training data, which can be attributed to two reasons. The data points in Year 7 are simpler than those in the training data, and the model was not sufficiently trained on the training data. The former is clearly illustrated in Figure 4, where the distribution is more right-skewed in the last two years, and the exponential decay of the number of defects as depth increases is sharper compared to the first two periods. This is due to the different reporting criteria adapted across the years and the defect threshold of 0.07 mm [15]. As a result, there are 324,777 data points in year seven with a reported maximum depth of 0.07 mm.

**Table 7: The results of Boosted decision trees (the best-performing model) on the full training data as well as on the testing data along with the model hyperparameter values**

|                                        |             |
|----------------------------------------|-------------|
| <b>Number of trees</b>                 | 1939        |
| <b>Minimal node size</b>               | 25          |
| <b>Tree depth</b>                      | 13          |
| <b>Learning rate</b>                   | 0.253550431 |
| <b>Minimum loss reduction</b>          | 2.12288E-06 |
| <b>Proportion observations sampled</b> | 0.8865065   |
| <b>Iterations before stopping</b>      | 18          |
| <b>Training RMSE</b>                   | 0.417935    |
| <b>Test RMSE</b>                       | 0.406       |
| <b>Naive forecasting MAE</b>           | 0.298125    |
| <b>Test MAE</b>                        | 0.267504    |
| <b>Naive forecasting MAPE</b>          | 2.097631 %  |
| <b>Test MAPE</b>                       | 1.60501 %   |
| <b>Training %Benchmark</b>             | 24.64       |
| <b>Test %Benchmark</b>                 | 28.30       |

Moreover, it is important to examine the residuals of the model to identify areas where the model performed poorly. This was accomplished by plotting the residuals (*Residuals* =

$y_{actual} - y_{prediction}$ ) for the final model in Figure 9, where a geometric parameter,  $\alpha$ , was used to represent the density of points. The diagonal pattern is common in datasets with many observations and limited prediction values. The ILI dataset used in this analysis satisfies this condition, as shown in Figure 4, where most points are in the first 40<sup>th</sup> percentile. Furthermore, the model underfits the defects with large maximum depth, where the predictions tended to regress to the mean of the data. On the other hand, the model overpredicts some shallow defects, resulting in large error residuals. This is common in tree-based models as they regress to the mean when extreme data are encountered [23]. Therefore, the points with the largest residuals are concentrated at or around the mean. Additionally, summary statistics were calculated for the best-performing model's predictions on the test set in Table 8, where Q1 and Q3 represent the 25<sup>th</sup> and 75<sup>th</sup> percentiles, respectively. The predictions of the model were overall more conservative than the actual values; for example, Q1, the median, and the mean of the model were all larger than those of the test set values. However, the model under-performed at the extreme values, where the maximum value predicted by the model was 1.6975 mm, while the maximum value of the test set was 5.18 mm. Finally, the model also predicted negative maximum depth for several points, as shown in the table. This is physically impossible, and this is another shortcoming of the model. Despite these shortcomings, the model's performance was acceptable overall, as seen in Figure 10. This shows the distribution of error residuals on the test set, where each bin represents approximately 0.1 mm. The figure shows a right-skewed distribution indicating conservative predictions on average and low error residuals for most points in the testing dataset.

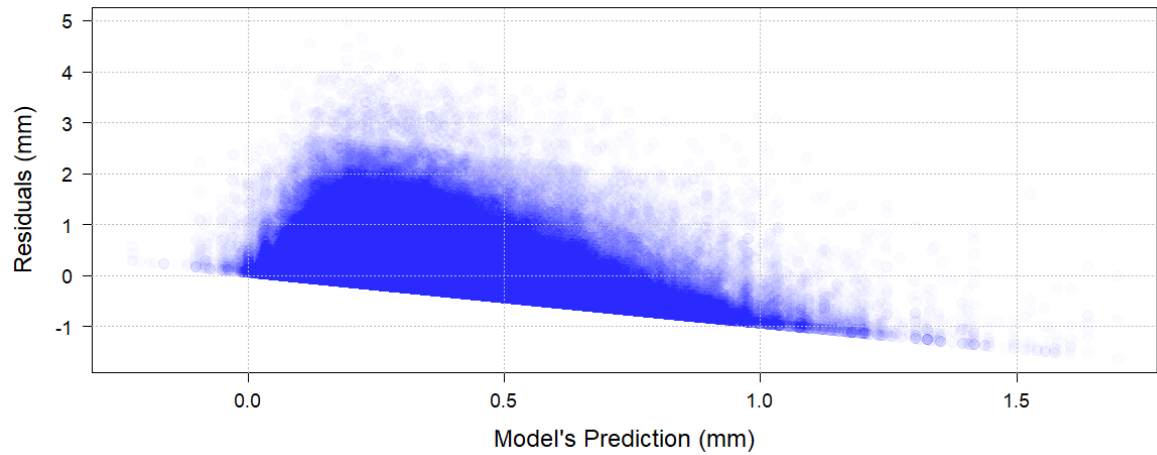


**Figure 9: A plot of the Error Residuals Vs the model predictions on the testing set. Darker colors indicate a higher density of points**

**Table 8: Summary statistics of the predictions of Boosted decision trees (the best-performing model) on the testing set**

| <b>Statistical measure</b> | <b>Max. Depth prediction (mm)</b> | <b>Actual max. depth (mm)</b> | <b>Residuals (mm)</b> |
|----------------------------|-----------------------------------|-------------------------------|-----------------------|
| Min.                       | -0.2269                           | 0.0700                        | -1.62752              |
| Q1                         | 0.1959                            | 0.0700                        | -0.19951              |
| Median                     | 0.2685                            | 0.1400                        | -0.08600              |
| Mean                       | 0.3069                            | 0.3173                        | 0.01035               |
| Q3                         | 0.3779                            | 0.4200                        | 0.11110               |
| Max.                       | 1.6975                            | 5.1800                        | 4.98673               |

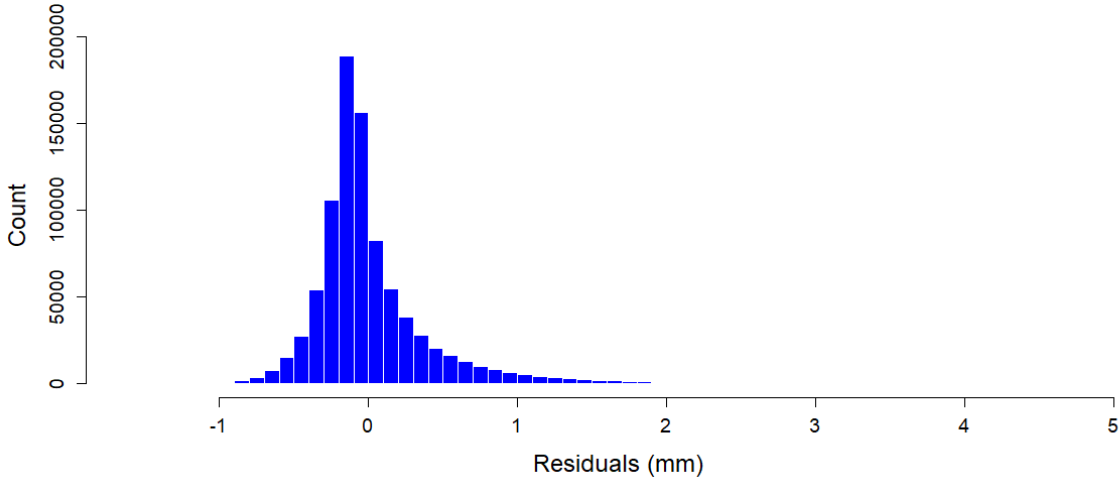


**Figure 10: A histogram showing the distribution of Error residuals for the model predictions on the testing set. There are 67 bins; each one is approximately 0.1 mm in width**

In conclusion, the results indicate that constructing machine learning models exclusively using ILI datasets to predict the maximum depth of corrosion defects is a valid approach. This assertion is supported by the observation that most models surpassed the benchmark, even when utilizing suboptimal hyperparameter combinations. Furthermore, surpassing the benchmark was achieved despite the unfavorable training strategy and the variations in maximum depth distribution throughout the years. This latter point is critically important for any proposed methodology aimed at addressing real-world ILI datasets. However, at this current stage, the models are not suitable for application due to the issues discussed in the preceding paragraph. It is anticipated that implementing a more comprehensive training strategy with the best-performing models, as demonstrated by Table 6, will mitigate the underfitting issues and the substantial error residuals observed in the final boosted decision trees model.

# **CHAPTER 4**

## **TREE-BASED MACHINE LEARNING MODELS**

### **FOR PREDICTING THE MAXIMUM DEPTH OF**

### **CORROSION DEFECTS BASED ON HISTORICAL**

### **IN-LINE INSPECTION DATA**

#### **4.1 Methods**

##### **4.1.1 Data Cleaning and Pre-Processing**

The tree-based methods utilized in this study do not require extensive pre-processing, with the exception of the LightGBM linear model, which recommends standard scaling of the predictors [40]. Furthermore, the predictors employed in this analysis do not contain any missing data, and none of the observations were removed from the original datasets.

### 4.1.2 Machine Learning Models

Tree-based machine learning techniques will be exclusively employed in this research due to their exceptional performance on tabular data. Specifically, this study will utilize Random Forests, XGBoost, DART, and LightGBM boosted trees with linear models. All hyperparameters relevant to regression analysis for these models will be explored. To achieve this, TPE algorithm will be employed to search for optimal hyperparameters, while Hyperband will be utilized for pruning. The following formulation outlines the proposed techniques.

Random Forests improve on decision trees by introducing two concepts that address the overfitting issues commonly encountered with decision trees: bootstrapping and feature subset selection during splitting [41]. These concepts introduce randomness, which in turn reduces the variance of the estimator.

Conversely, XGBoost employs Classification and Regression Trees (CART), differing from traditional decision trees in that each leaf contains a real score rather than decision values. Since this study focuses exclusively on regression machine learning with Mean Squared Error (MSE) as the loss function, the objective can be expressed as [32]:

$$\begin{aligned} obj^{(t)} = \sum_{i=1}^n [2(\hat{y}_i^{(t-1)} - y_i)f_t(x_i) + f_t(x_i)^2] + w(f_t) \\ + const \end{aligned} \tag{Eq. 4.1}$$

Where  $y_i$  is the target value for row  $i$ ,  $\hat{y}_i$  is the model prediction for the  $i^{\text{th}}$  row,  $w(f)$  is the regularization function,  $f$  represents the CART tree, and  $t$  denotes a tree such that  $t \in T$  where  $T$  is the number of boosted trees.

Boosted trees with linear models at the nodes build upon the same structure, with the distinction that the nodes contain linear models instead of constant values. DART, on the other hand, enhances these methods by incorporating dropout techniques adapted from deep neural networks to mitigate overfitting [42].

To calculate feature importance, Shapley values will be employed, specifically using TreeSHAP, which was developed for tree-based models [43]. This approach defines a value function using conditional expectations given by:

$$E_{X_j|X_{-j}}(\hat{f}(x)|x_j) \quad (\text{Eq. 4.2})$$

Where  $x$  is an instance,  $X_j$  represents the present predictors, and  $X_{-j}$  denotes the absent ones. Then, feature importance,  $I_j$ , is calculated by taking the average absolute Shapley values,  $\varphi_j$  as follows:

$$I_j = \frac{1}{n} \sum_{i=1}^n |\varphi_j^{(i)}| \quad (\text{Eq. 4.3})$$

For hyperparameter tuning, the TPE algorithm will be used for sampling, and the Hyperband algorithm will be employed for pruning [44], [45]. This combination has demonstrated superior performance compared to other methods for tabular data [46], thus it was exclusively utilized in this study. The study was conducted using Python 3.12.5, an

AMD Ryzen 9 7950X 16-core processor, and an RTX 4090 GPU. Table 9 lists the software packages and libraries used in this study along with their corresponding versions.

**Table 9: Software packages and versions used in machine learning model development and hyperparameter tuning**

| <b>Software</b> | <b>Version</b> | <b>Use</b>               |
|-----------------|----------------|--------------------------|
| scikit-learn    | 1.5.2          | Random Forests Model     |
| XGBoost         | 2.1.1          | XGBoost and final models |
| LightGBM        | 2.3.1          | LightGBM linear          |
| Optuna          | 4.0.0          | Hyperparameters tuning   |
| SHAP            | 0.46.0         | Feature importance       |
| CUDA            | 12.6           | GPU accelerated XGBoost  |
| Python          | 3.12.5         | Programming language     |

### 4.1.3 Proposed Method

This study utilizes the predictors Year, SPD.m, SIPRD.m, SIPO.deg, L.mm, and W.mm for training, with the target variable being MD.mm. The first three inspection intervals (Year 1, Year 3, and Year 5) will be used for training, while the final interval (Year 7) will serve as the testing set. Each model will be trained using 5 Cross-Validation folds (CV) and 500 trials of different hyperparameter combinations. Consequently, RMSE will be calculated for each out-of-sample fold, and the average RMSE will be reported for each trial. The best-performing model will then undergo an additional 100 trials to account for hyperparameter values that are near or at the specific range utilized during the search process. Following this, the hyperparameters associated with the lowest out-of-sample

RMSE will be applied to the final model. This model will then be trained on the entire dataset and utilized to predict the maximum depth of the test set.

All models will be compared to the Naïve forecasting benchmark to evaluate their accuracy. To achieve this, the distance from the starting and ending points of the defect relative to the pipeline origin will be used to match the defects across consecutive ILI runs. Subsequently, RMSE will be calculated using Eq 3.3.

## **4.2 Results and Discussions**

In this study, several models were developed to predict the maximum depth of corrosion defects. The hyperparameters utilized in these models, along with their optimal values and tuning ranges, are summarized in Table 10. Section 4.2.1 will detail the performance of these models, while the subsequent section will focus on the final model.

### **4.2.1 Initial Models**

Among the models evaluated, the random forests model exhibited the poorest performance, as illustrated in both Figure 11 and Table 11. Nonetheless, it surpassed the benchmark even when using the least favorable hyperparameters. With optimal hyperparameters, the random forests model exceeded the benchmark by 37.8%, achieving RMSE of 0.3779 mm. Interestingly, the best trial utilized a relatively low number of trees, despite the parameter range extending up to 1000 trees. Additionally, the maximum depth of the optimal tree was also on the lower side, as indicated in Table 10. This choice was likely to mitigate overfitting, a common concern associated with increased tree depth.

Figure 11 clearly demonstrates a significant accuracy gap between the random forests and boosting-based models. Furthermore, the random forests model required a greater number of trials to achieve the same RMSE as the other techniques. Both observations indicate that the underlying tree structure, decision tree, is not as effective in modelling ILI data. Furthermore, Table 12 shows the training time for all models where random forests required the longest time. This inefficiency can be attributed to the exact splitting method employed during tree construction, as well as scikit-learn's implementation of random forests, which lacks support for GPU acceleration. Consequently, this model experienced the longest training time among all evaluated models. The combination of low performance and extended training duration suggests that the random forests technique may not be the most effective choice for developing depth prediction machine learning models using ILI data. This training time limitation will pose a significant challenge in practical applications as the dataset expands.

**Table 10: Hyperparameters used for each model, along with the best value for each hyperparameter. Square brackets indicate the range considered during tuning. A dash (-) indicates that the corresponding parameter is not applicable for the model**

| Hyperparameter             | Random Forests     | XGBoost                                  | DART                                     | LightGBM              | Final XGBoost         |
|----------------------------|--------------------|------------------------------------------|------------------------------------------|-----------------------|-----------------------|
| number of trees/iterations | 334 [100, 1000]    | 311 [100, 500]                           | 380 [100, 500]                           | 445 [100, 500]        | 661 [300, 1000]       |
| tree maximum depth         | 13 [5, 50]         | 20 [1, 20]                               | 12 [1, 20]                               | 15 [1, 20]            | 30 [10, 50]           |
| min_samples_split          | 2 [2, 10]          | -                                        | -                                        | -                     | -                     |
| min_samples_leaf           | 7 [1, 10]          | -                                        | -                                        | -                     | -                     |
| max_features               | sqrt [sqrt, log2]  | -                                        | -                                        | -                     | -                     |
| max_leaf_nodes             | 100 [10, 100]      | -                                        | -                                        | -                     | -                     |
| min_weight_fraction_leaf   | 0.0003442 [0, 0.5] | -                                        | -                                        | -                     | -                     |
| booster                    | -                  | gbtree                                   | dart                                     | gbdt linear tree      | gbtree                |
| tree method                | -                  | hist                                     | hist                                     | hist                  | hist                  |
| learning rate              | -                  | 0.0759372 [0.01, 0.3]                    | 0.1667159 [0.01, 0.3]                    | 0.2984299 [0.01, 0.3] | 0.0741369 [0.01, 0.3] |
| gamma                      | -                  | 0.0048413 [0, 5]                         | 0.0014748 [0, 5]                         | 0.0057929 [0, 5]      | 0.0323618 [0, 0.1]    |
| min_child_weight           | -                  | 8 [1, 10]                                | 6.2813984 [1, 10]                        | 3.5827260 [1, 10]     | 5.4979688 [5, 15]     |
| subsample                  | -                  | 0.6534939 [0.5, 1]                       | 0.6575595 [0.5, 1]                       | 0.5113235 [0.5, 1]    | 0.7505404 [0.5, 1]    |
| sampling_method            | -                  | gradient_based [uniform, gradient_based] | gradient_based [uniform, gradient_based] | -                     | gradient_based        |
| colsample_bytree           | -                  | 0.9939477 [0.5, 1]                       | 0.9574214 [0.5, 1]                       | 0.9812707 [0.5, 1]    | 0.8874469 [0.5, 1]    |
| colsample_bylevel          | -                  | 0.8842332 [0.5, 1]                       | 0.9496533 [0.5, 1]                       | -                     | 0.9028283 [0.5, 1]    |

|                           |   |                                        |                                        |                        |                                        |
|---------------------------|---|----------------------------------------|----------------------------------------|------------------------|----------------------------------------|
| colsample_bynode          | - | 0.9701486<br>[0.5, 1]                  | 0.9808222 [0.5,<br>1]                  | 0.97960594<br>[0.5, 1] | 0.7834260<br>[0.5, 1]                  |
| lambda                    | - | 9.4656927 [0,<br>10]                   | 9.2688755 [0,<br>10]                   | 8.82043244 [0,<br>10]  | 14.2352818 [5,<br>15]                  |
| alpha                     | - | 6.4856226 [0,<br>10]                   | 7.1509553 [0,<br>10]                   | 8.68836387 [0,<br>10]  | 8.3635603 [5,<br>15]                   |
| grow policy               | - | depthwise<br>[depthwise,<br>lossguide] | depthwise<br>[depthwise,<br>lossguide] | -                      | lossguide<br>[depthwise,<br>lossguide] |
| maximum number of<br>bins | - | 891 [256,<br>1000]                     | 877 [256, 1000]                        | 930 [256, 1000]        | 1479 [512,<br>1500]                    |
| sample type               | - | -                                      | weighted<br>[uniform,<br>weighted]     | -                      | -                                      |
| rate_drop                 | - | -                                      | 0.0001793 [0, 1]                       | -                      | -                                      |

Another model utilized in this research is LightGBM with linear trees. The rationale for selecting this model was to address a primary issue encountered with boosting-based models: their lower performance at the extremes. However, this strategy did not yield the desired results, as evidenced by both Figure 11 and Table 11, which show that conventional XGBoost trees achieved superior accuracy. Despite this, the LightGBM model still significantly outperformed both the benchmark and the random forests model, a success attributed to its boosting architecture rather than its linear formulation.

Notably, although the LightGBM model was trained on a CPU, it completed training in the shortest time among all models as shown in Table 12. This reduced training time compared to the random forests model is expected, as LightGBM employs histogram-based binning for tree construction and internal data preprocessing. While XGBoost also utilizes this method, LightGBM demonstrates greater efficiency, as illustrated by its training times. This efficiency holds true regardless of whether linear trees are employed.

Consequently, while linear trees may not achieve the same level of performance as conventional boosted trees, the use of LightGBM is recommended for its rapid execution time. Although it remains uncertain whether LightGBM can match the accuracy of XGBoost, its faster execution can facilitate the narrowing of hyperparameter ranges during tuning. This smaller range of optimal values can then be trained using both packages, allowing for the identification of the best model in a significantly shorter timeframe than would be possible using only one package.

Both the XGBoost and DART models achieved very low RMSEs, exceeding the benchmark by 47.22% and 47.02%, respectively. However, the DART model required over 55% more training time. While these results make it challenging to recommend DART, the technique demonstrated continuous improvement in hyperparameter tuning, as indicated by the steeper slope in Figure 11. Given this trend and the negligible difference in RMSE, it is anticipated that DART could outperform the XGBoost model with additional trials. However, confirming this hypothesis is difficult, as the top 100 trials in terms of RMSE all exhibited a dropout rate below 0.03. Furthermore, the dropout rate for the best DART model was 0.0001793, as shown in Table 10. These suggest that the model achieves higher accuracies by closely mimicking the conventional XGBoost model. The wide range of this parameter used during hyperparameter training likely contributed to both the model's poor performance in lower trials and the slow rate of RMSE reduction as a function of trials. Nonetheless, these results indicate that lower dropout rates are associated with better performance.

In contrast, XGBoost outperformed all other models and exhibited a more rapid decrease in RMSE due to hyperparameter tuning, particularly in the earlier stages. The best model

was achieved with a tree depth of 20, raising questions about whether this technique has reached optimal accuracy or if it was constrained by the hyperparameter range. To address this uncertainty, an additional 100 trials were conducted for XGBoost, focusing on a targeted range of hyperparameters associated with the lowest RMSEs.

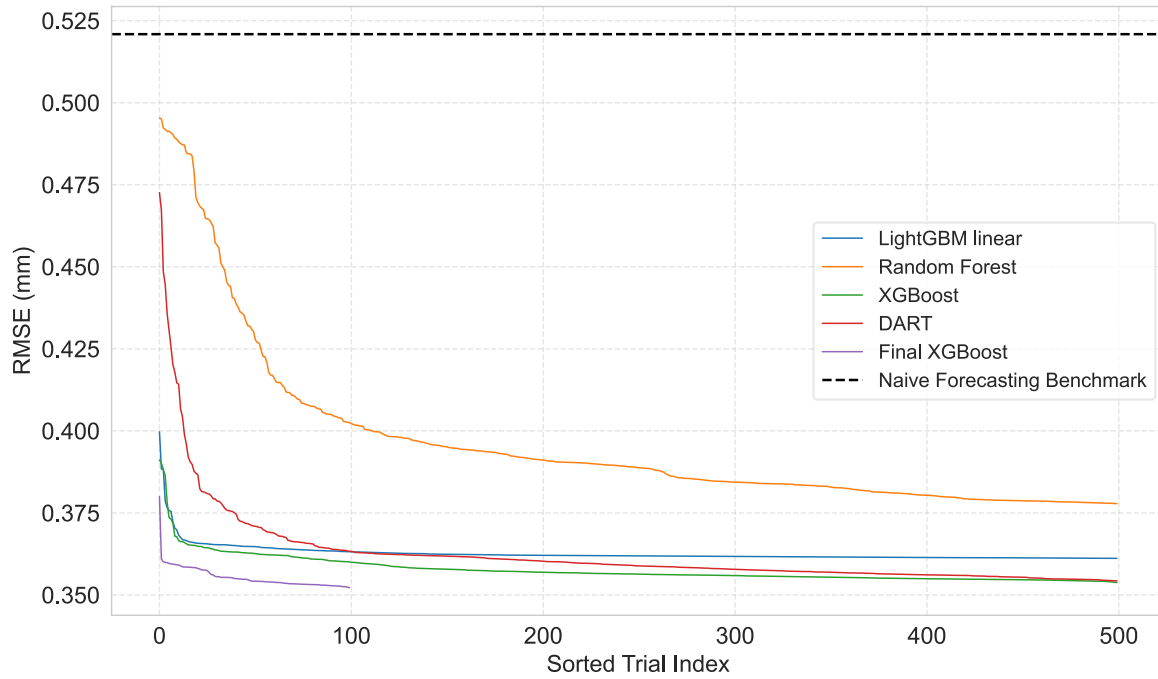


**Figure 11: RMSEs for all hyperparameter tuning trials conducted for each model developed in this study. The trials are sorted by RMSE values**

**Table 11: Evaluation metrics for each model, along with their relative performance compared to the Naive forecasting benchmark**

| <b>Model</b>             | <b>RMSE (mm)</b> | <b>R<sup>2</sup></b> | <b>Improvement over the benchmark (%)</b> |
|--------------------------|------------------|----------------------|-------------------------------------------|
| XGBoost                  | 0.35381          | 0.49021              | 47.2255                                   |
| Random Forests           | 0.37790          | 0.41862              | 37.8393                                   |
| DART                     | 0.35429          | 0.48882              | 47.0250                                   |
| LightGBM linear          | 0.36116          | 0.46882              | 44.2314                                   |
| Final XGBoost (CV)       | 0.35224          | 0.49471              | 47.8821                                   |
| Final XGBoost (training) | 0.29233          | 0.65199              | 78.1892                                   |
| Final XGBoost (testing)  | 0.36800          | 0.09561              | 41.5508                                   |

**Table 12: Training time, in seconds, for each model developed in this study**

| <b>Model</b>    | <b>Training time (Sec)</b> | <b>Library</b> | <b>Platform</b> |
|-----------------|----------------------------|----------------|-----------------|
| XGBoost         | 134466                     | XGBoost        | GPU*            |
| Random Forests  | 469829                     | Scikit-learn   | CPU             |
| DART            | 211071                     | XGBoost        | GPU*            |
| LightGBM linear | 5116                       | LightGBM       | CPU             |
| Final XGBoost   | 190673                     | XGBoost        | GPU*            |

\* The data was processed on the CPU while the model was executed on the GPU. This configuration may have increased memory consumption and reduced execution speeds due to the need for data transfer between the CPU and GPU.

## 4.2.2 Final Model

After conducting an additional 100 trials with the XGBoost model, the average out-of-sample RMSE was 0.35224 mm. Despite this extra training and a more targeted hyperparameter search, the model did not show substantial improvement. This outcome was anticipated, given the marginal decrease in accuracy observed in the original model during the later trials, as shown in Figure 11. Figure 12 presents a parallel coordinate plot of the tuning parameters. Combined with the information in Table 10, several observations can be made regarding the use of boosted trees, applicable to all XGBoost, DART, and linear tree models. First, high values of regularization parameters such as alpha and lambda significantly improve the out-of-sample accuracy. This is particularly important since the default values for these parameters are either zero or low in both XGBoost and LightGBM. Second, while the learning rate, number of iterations, and number of bins all contribute to increased accuracy and execution time, the number of bins emerged as the most significant factor.

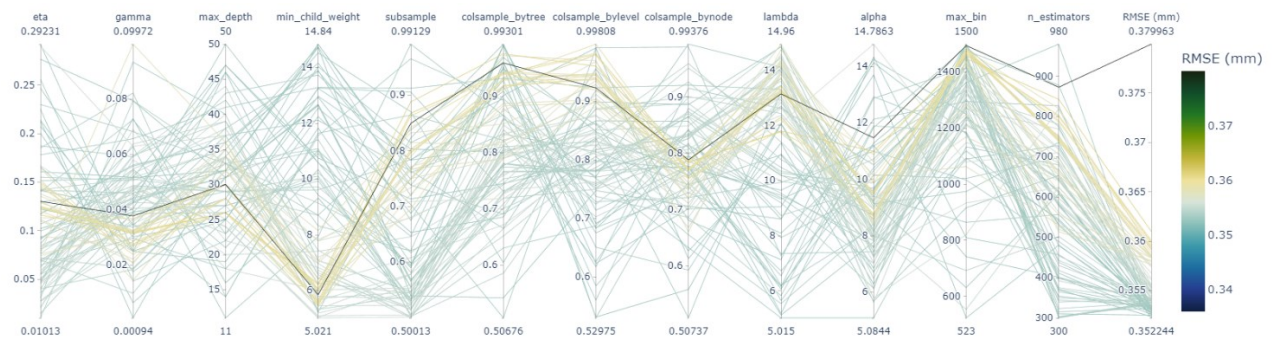


**Figure 12: Parallel coordinate plot of the final model hyperparameters. RMSE is represented in two ways: by the last column on the right and through the color gradient**

The best model was subsequently tested on the Year 7 test set, yielding a test RMSE of 0.36800 mm, which outperformed the benchmark by 41.5%. This RMSE value is close to that obtained during cross-validation (CV), but the  $R^2$  value was considerably lower. Since  $R^2$  is heavily influenced by the variance of the variable, the significantly reduced variance in Year 7, as shown in both Figure 4 and Table 3, contributed to its low value. This indicates that  $R^2$  may not be a reliable metric for assessing model performance, and RMSE is preferred for its consistency and direct interpretability. The error residuals ( $Residuals = y_{actual} - y_{prediction}$ ), are plotted against the model predictions on the test set in Figure 13. The largest errors are concentrated around the mean of the data, which is expected since XGBoost predictions tend to regress to the mean when encountering extreme data. Additionally, the diagonal shape observed is typical for datasets with multiple entries that share similar values, a characteristic noted in previous chapters.

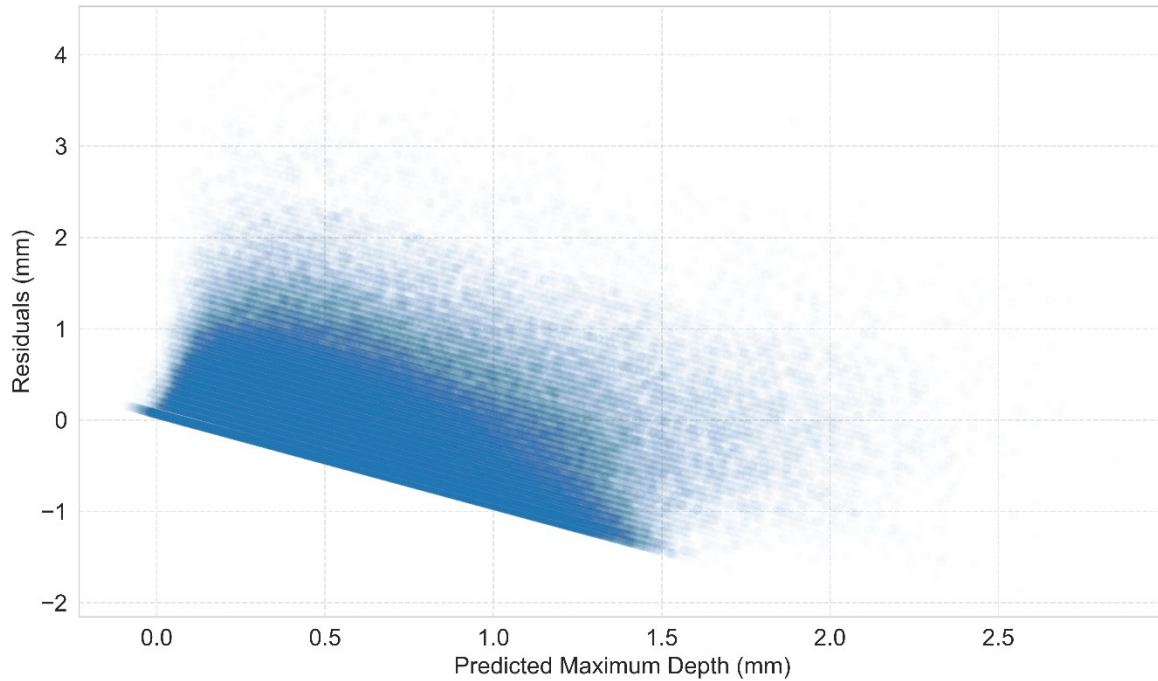


**Figure 13: Error residuals of the final model predictions on the test set plotted against the corresponding model predictions**

To further examine the model's predictions, a histogram of the residuals was plotted in Figure 14, with each bin measuring 0.01 mm in width. The right-skewed distribution indicates that the model's predictions were generally more conservative than the actual measurements. Furthermore, the high concentration of low residuals suggests that the model performed well for the majority of data points. This is supported by Table 13, which provides a statistical summary of actual and predicted depths, as well as residuals. Both the median and average of the residuals are slightly negative, while the 75th percentile (Q3) is slightly positive. However, the same table highlights the model's poor performance with deep defects, here, the maximum model prediction deviated by more than 2 mm from the actual measurement.

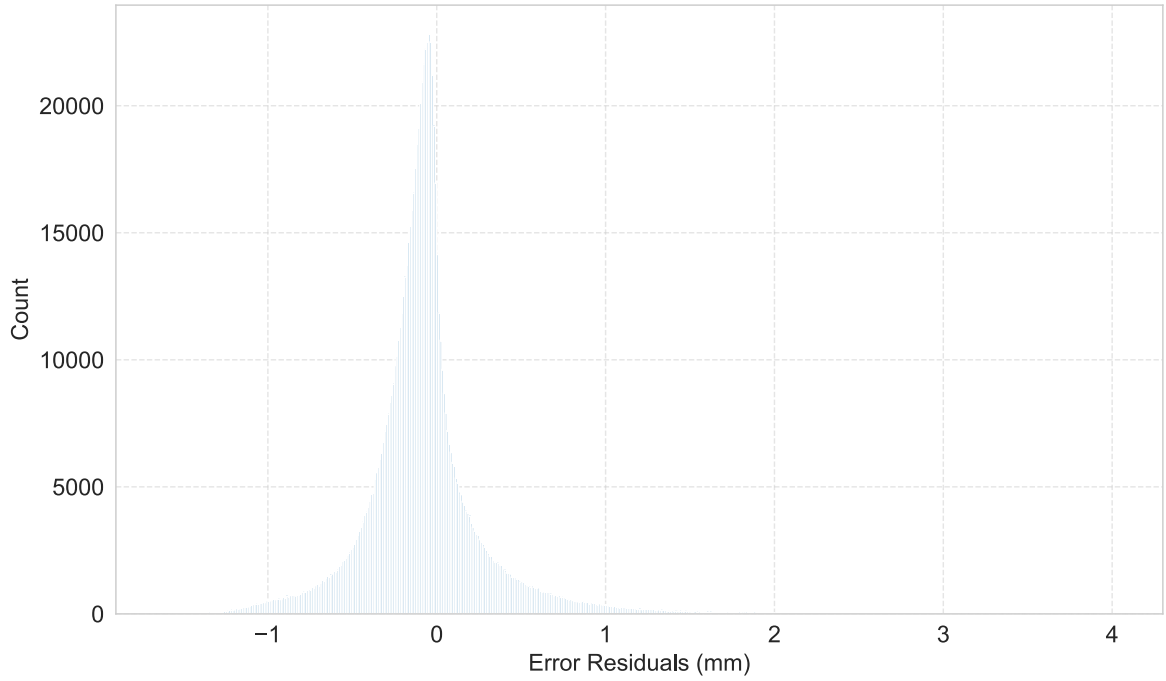


**Figure 14: Histogram of the error residuals from the final model predictions on the test set, with each bin representing an interval of 0.01 mm**

**Table 13: Statistical summary of the final model predictions on the test set. Q1 and Q3 denote the first and third quartiles, respectively**

| Statistical measure | Max. Depth prediction (mm) | Actual max. depth (mm) | Residuals (mm) |
|---------------------|----------------------------|------------------------|----------------|
| Min.                | -0.0832                    | 0.0700                 | -1.8507        |
| Q1                  | 0.1773                     | 0.0700                 | -0.2310        |
| Median              | 0.3049                     | 0.1400                 | -0.0873        |
| Mean                | 0.3921                     | 0.3173                 | -0.0748        |
| Q3                  | 0.5106                     | 0.4200                 | 0.0264         |
| Max.                | 2.8504                     | 5.1800                 | 4.2260         |

To better understand how the model makes predictions, feature importance was calculated using SHAP, as illustrated in Figure 15 and Figure 16 for the final model on the training data. Figure 15 displays the absolute contribution of each feature segregated by year, while Figure 16 shows the effect of feature values on the target variable. The analysis indicates that defect length and width had the most significant impact on predictions, followed by the inspection year, while other features had minimal influence. Notably, although all features exhibited fluctuations in importance across different inspection runs, the inspection year demonstrated a continuous increase, suggesting that the ILI data exhibit strong time-series behavior.

Due to changes in reporting criteria highlighted previously, Figure 16 indicates that later inspections are associated with lower depths. While this finding may seem counterintuitive, it aligns with the negative correlation between the inspection year and maximum depth shown in Figure 6. Additionally, the significant change in depth distribution illustrated in Figure 4 is evident in Figure 16, where Year 1 (denoted by dark green) contributes slightly to higher depths, while Year 3 (light green) plays a significant role in predicting deep defects. These mixed signals have greatly impacted model accuracy, as shown in Figure 17, which presents the first tree of the final model, specifically, the first four levels. The first split is based on the inspection year, which introduces potential errors due to the inconsistent relationship between the year and maximum depth.

Examining the other features, both defect length and width are more important than the inspection year. However, their contributions may be inflated due to the high positive correlation between these two features, as shown in Figure 6. In addition, the same figure shows that both variables are positively correlated with the maximum depth. This however

contradicts Figure 16, which consistently indicates that as defect length increases, so does depth, while width has the opposite effect—higher values contribute negatively to defect depth, whereas lower values are associated with greater depths.

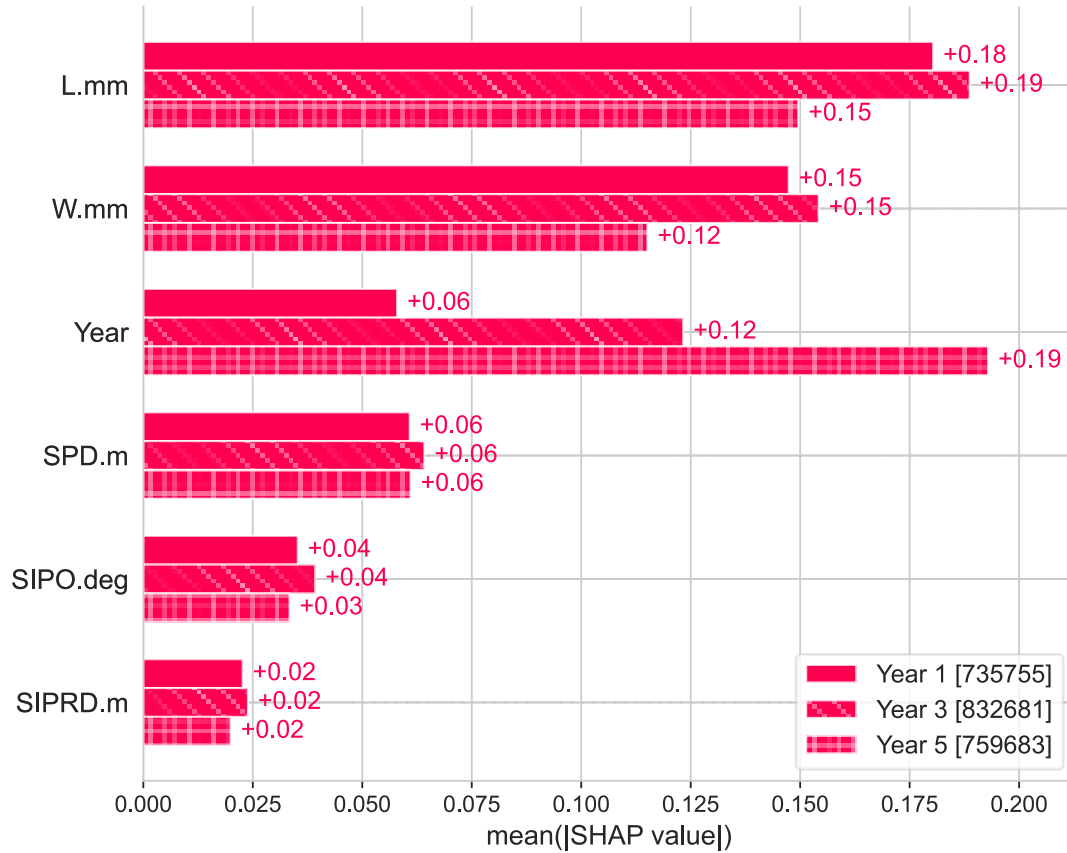


**Figure 15: Feature importance derived from SHAP analysis of the final model on the training data for each ILI run**

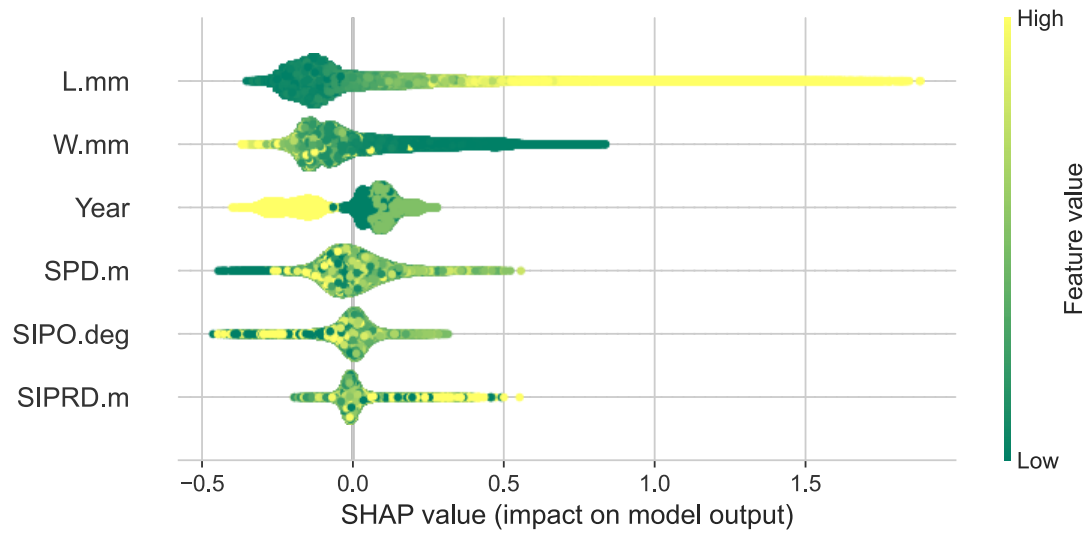


**Figure 16: Feature importance derived from SHAP analysis of the final model on the training data, with feature values indicated by color**

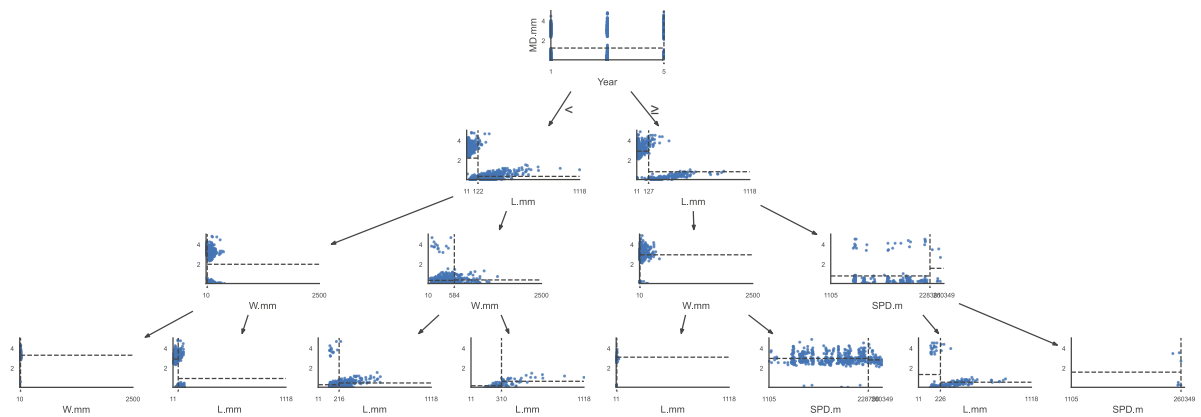


**Figure 17: The first four levels of the first tree in the final model developed using all training data. The plot displays observations with the highest and lowest residuals, with 1,000 observations for each category**

In conclusion, all models significantly exceeded the benchmark, with XGBoost and DART demonstrating superior performance compared to the others. The final model achieved consistently high performance on both cross-validation and testing datasets, providing more conservative predictions for the majority of observations. This success was accomplished despite the limitations of the datasets, where changes in defect distribution and reporting criteria are evident. However, the final model underperformed at extreme values, indicating a need for targeted training on different depth ranges, which could be addressed using semi-supervised machine learning. Furthermore, the strong time-series behavior of ILI data could be further leveraged through sequential models such as Recurrent Neural Networks (RNNs).

## **CHAPTER 5**

### **RECOMMENDATIONS AND LIMITATIONS**

This paper presents service-agnostic corrosion prediction models developed from historical ILI data, which outperformed both the benchmark and most service-specific corrosion prediction models in the research. These results were achieved despite significant changes in data distribution and defect reporting criteria, highlighting the potential of ILI-based machine learning models for corrosion prediction. Not only do these models demonstrate high accuracy, but they are also relatively easy to develop and implement, given the vast amounts of ILI data available to oil and gas companies. This represents added value, as these inspections will continue to be conducted regardless of their role in developing machine learning models.

Moreover, since the proposed framework is service-agnostic, the models can be further enhanced with service-specific data to improve predictive capabilities and generate data-driven insights and recommendations. Ultimately, these models can play a crucial role in asset integrity management frameworks, ensuring the reliable, safe, and efficient operation of pipelines in the oil and gas industry.

Despite these advantages, limitations persist regarding the prediction accuracy of deep defects and the traceability of defects across ILI runs. The current models, while generally accurate and more conservative than actual measurements for the majority of data points, encounter difficulties in predicting deep defects. As evidenced in Table 8 and Table 13, the reliability of these models for deep defect prediction is insufficient. This issue may be addressed through targeted training specifically focused on deep defects, which can be achieved using semi-supervised machine learning techniques. Furthermore, the study lacks a robust methodology for tracking defects over time, which impedes the establishment of accurate forecasting benchmarks and the monitoring of short-term corrosion rates. To overcome these challenges, future research should prioritize the development of robust defect-matching techniques and the enhancement of the overall predictive capabilities of the model

## **CHAPTER 6**

### **CONCLUSIONS**

This research establishes a framework for leveraging ILI data to develop machine learning models aimed at predicting the maximum depth of corrosion defects. Different linear, non-linear, and tree-based machine learning models were trained exclusively on historical ILI data. The results demonstrated significant improvements over the Naïve forecasting benchmark in both studies. When limited training and hyperparameter tuning strategies were employed, the best-performing model achieved a 28.30% improvement over the Naïve forecasting benchmark on the test set, with RMSE of 0.406 mm. In contrast, when enhanced predictors and training methodologies were utilized, the best models achieved a 41.5% improvement over the same benchmark on the test set, resulting in an RMSE of 0.368 mm. Furthermore, the accuracy of these models surpassed that of most service-based corrosion prediction models documented in the literature. These metrics indicate the success of the proposed framework in utilizing machine learning to construct pipeline corrosion prediction models based exclusively on ILI data. Notably, this success was achieved using an ILI dataset characterized by extreme variations in maximum depth distribution and changes in reporting criteria. These findings suggest that the proposed framework offers a more effective alternative to service-based models, particularly given

the extensive ILI data available to oil and gas pipeline operators. Additionally, the service-agnostic nature of this framework facilitates the integration of process and service parameters, which can further enhance the efficacy of machine learning models for corrosion prediction. Looking ahead, future research should explore the application of semi-supervised learning models to improve predictions for deeper defects. Moreover, the observed influence of the inspection year on prediction accuracy suggests that sequential models may yield even better performance

## References

- [1] “Oil and Gas Pipelines Industry Outlook by Capacity and Capital Expenditure Including Details of All Operating and Planned Pipelines to 2028,” GlobalData, London, UK, GDGE1381ICR-ST, Oct. 2024.
- [2] X. Wang *et al.*, “A review of development and application of out-of-pipe detection technology without removing cladding,” *Measurement*, vol. 219, p. 113249, Sep. 2023, doi: 10.1016/j.measurement.2023.113249.
- [3] A. Groysman, *Corrosion Problems and Solutions in Oil Refining and Petrochemical Industry*, vol. 32. in Topics in Safety, Risk, Reliability and Quality, vol. 32. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-45256-2.
- [4] *In-line inspection tool readiness for hydrogen pipelines*, White paper 520, May 2024.
- [5] A. H. El-Kady, M. T. Amin, F. Khan, and M. M. El-Halwagi, “Analysis of CO<sub>2</sub> pipeline regulations from a safety perspective for offshore carbon capture, utilization, and storage (CCUS),” *J. Clean. Prod.*, vol. 439, p. 140734, Feb. 2024, doi: 10.1016/j.jclepro.2024.140734.
- [6] *In-Line Inspection of Pipelines*, SP0102-2017.
- [7] B. Mao, Y. Lu, P. Wu, B. Mao, and P. Li, “Signal processing and defect analysis of pipeline inspection applying magnetic flux leakage methods,” *Intell. Serv. Robot.*, vol. 7, no. 4, pp. 203–209, Oct. 2014, doi: 10.1007/s11370-014-0158-6.
- [8] *49 CFR Part 192 -- Transportation of Natural and Other Gas by Pipeline: Minimum Federal Safety Standards*, vol. 49. Accessed: Nov. 25, 2024. [Online]. Available: <https://www.ecfr.gov/current/title-49/part-192>
- [9] *49 CFR Part 195 -- Transportation of Hazardous Liquids by Pipeline*, vol. 49. Accessed: Nov. 25, 2024. [Online]. Available: <https://www.ecfr.gov/current/title-49/part-195>
- [10] *Specifications and requirements for in-line inspection of pipelines*, Standard Practice POF 100.
- [11] *In-line Inspection Systems Qualification Standard*, STD API STD 1163, Washington D.C., 2021.
- [12] *ANSI/ASNT ILI-PQ: In-line Inspection Personnel Qualification and Certification (2023)*, ILI-PQ, Standards.
- [13] *Specifications and requirements for Universal POF Template (UPT) data format for in-line inspection of pipelines*, Standard Practice POF 110.
- [14] *Field verification for in-line inspection*, Recommended Practice POF 310.

- [15] F. Khan, R. Yarveisy, and R. Abbassi, "Cross-country pipeline inspection data analysis and testing of probabilistic degradation models," *J. Pipeline Sci. Eng.*, vol. 1, no. 3, pp. 308–320, Sep. 2021, doi: 10.1016/j.jpse.2021.09.004.
- [16] J. C. Velázquez, F. Caleyó, A. Valor, and J. M. Hallen, "Technical Note: Field Study—Pitting Corrosion of Underground Pipelines Related to Local Soil and Pipe Characteristics," *CORROSION*, vol. 66, no. 1, pp. 016001–016001–5, Jan. 2010, doi: 10.5006/1.3318290.
- [17] M. El Amine Ben Seghier, B. Keshtegar, K. F. Tee, T. Zayed, R. Abbassi, and N. T. Trung, "Prediction of maximum pitting corrosion depth in oil and gas pipelines," *Eng. Fail. Anal.*, vol. 112, p. 104505, May 2020, doi: 10.1016/j.engfailanal.2020.104505.
- [18] G. Masi, R. Vichi, M. Gentile, R. Bruschi, and G. Gabetta, "A Neural Network Predictive Model of Pipeline Internal Corrosion Profile," *Int. Conf. Syst.*, 2014, [Online]. Available: <https://api.semanticscholar.org/CorpusID:53479503>
- [19] S. Hatami, A. Ghaderi-Ardakani, M. Niknejad-Khomami, F. Karimi-Malekabadi, M. R. Rasaei, and A. H. Mohammadi, "On the prediction of CO<sub>2</sub> corrosion in petroleum industry," *J. Supercrit. Fluids*, vol. 117, pp. 108–112, Nov. 2016, doi: 10.1016/j.supflu.2016.05.047.
- [20] 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," *J. Nat. Gas Sci. Eng.*, vol. 85, p. 103716, Jan. 2021, doi: 10.1016/j.jngse.2020.103716.
- [21] H. Mesghali, B. Akhlaghi, N. Gozalpour, J. Mohammadpour, F. Salehi, and R. Abbassi, "Predicting maximum pitting corrosion depth in buried transmission pipelines: Insights from tree-based machine learning and identification of influential factors," *Process Saf. Environ. Prot.*, vol. 187, pp. 1269–1285, Jul. 2024, doi: 10.1016/j.psep.2024.05.014.
- [22] M. E. A. Ben Seghier, B. Keshtegar, M. Taleb-Berrouane, R. Abbassi, and N.-T. Trung, "Advanced intelligence frameworks for predicting maximum pitting corrosion depth in oil and gas pipelines," *Process Saf. Environ. Prot.*, vol. 147, pp. 818–833, Mar. 2021, doi: 10.1016/j.psep.2021.01.008.
- [23] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-6849-3.
- [24] M. Kuhn and D. Falbel, *brulee: High-Level Modeling Functions with "torch."* (2024). R. [Online]. Available: <https://github.com/tidymodels/brulee>
- [25] K.-A. Le Cao, F. Rohart, I. Gonzalez, and S. Dejean, *mixOmics: an R package for 'omics feature selection and multiple data integration.* (2023). R. [Online]. Available: <https://www.bioconductor.org/packages/release/bioc/html/mixOmics.html>
- [26] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization Paths for Generalized Linear Models via Coordinate Descent," *J. Stat. Softw.*, vol. 33, no. 1, 2010, doi: 10.18637/jss.v033.i01.
- [27] S. Milborrow, *earth: Multivariate Adaptive Regression Splines.* (2024). R. Accessed: Sep. 09, 2024. [Online]. Available: <https://CRAN.R-project.org/package=earth>
- [28] K. Schliep and K. Hechenbichler, *kkn: Weighted k-Nearest Neighbors.* (2024). R. [Online]. Available: <https://github.com/klausvigo/kkn>

- [29] A. Karatzoglou, A. Smola, and K. Hornik, *kernlab: Kernel-Based Machine Learning Lab*. (2024). R. [Online]. Available: <https://CRAN.R-project.org/package=kernlab>
- [30] T. Therneau, B. Atkinson, and B. R. port, *rpart: Recursive Partitioning and Regression Trees*. (Dec. 05, 2023). R. Accessed: Sep. 09, 2024. [Online]. Available: <https://cran.r-project.org/web/packages/rpart/index.html>
- [31] V. Dorie, *dbarts: Discrete Bayesian Additive Regression Trees Sampler*. (2024). R. [Online]. Available: <https://CRAN.R-project.org/package=dbarts>
- [32] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco California USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [33] M. Kuhn and R. Quinlan, *Cubist: Rule- And Instance-Based Regression Modeling*. (2024). R. [Online]. Available: <https://github.com/topepo/Cubist>
- [34] M. N. Wright and A. Ziegler, “ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R,” *J. Stat. Softw.*, vol. 77, no. 1, 2017, doi: 10.18637/jss.v077.i01.
- [35] K. Holub, *xrf: eXtreme RuleFit*. (Oct. 04, 2022). R. Accessed: Sep. 09, 2024. [Online]. Available: <https://cran.r-project.org/web/packages/xrf/index.html>
- [36] R Core Team (2024), *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [Windows 10 x64]. Available: <https://www.r-project.org/>
- [37] M. Kuhn and H. Wickham, *Tidymodels: a collection of packages for modeling and machine learning using tidyverse principles*. (2020). R. [Online]. Available: <https://www.tidymodels.org>
- [38] “Benchmark Methods & Forecast Accuracy · UC Business Analytics R Programming Guide.” Accessed: Sep. 08, 2024. [Online]. Available: [https://uc-r.github.io/ts\\_benchmarking](https://uc-r.github.io/ts_benchmarking)
- [39] C. Dhakal, “A Naïve Approach for Comparing a Forecast Model,” Aug. 2018.
- [40] G. Ke *et al.*, “LightGBM: a highly efficient gradient boosting decision tree,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, in NIPS’17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 3149–3157.
- [41] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, 2011.
- [42] K. V. Rashmi and R. Gilad-Bachrach, “DART: Dropouts meet Multiple Additive Regression Trees,” May 07, 2015, *arXiv*: arXiv:1505.01866. doi: 10.48550/arXiv.1505.01866.
- [43] S. M. Lundberg, G. G. Erion, and S.-I. Lee, “Consistent Individualized Feature Attribution for Tree Ensembles,” Mar. 07, 2019, *arXiv*: arXiv:1802.03888. doi: 10.48550/arXiv.1802.03888.
- [44] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for Hyperparameter Optimization”.
- [45] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, “Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization,” Jun. 18, 2018, *arXiv*: arXiv:1603.06560. doi: 10.48550/arXiv.1603.06560.

- [46] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” Jul. 25, 2019, *arXiv*: arXiv:1907.10902. doi: 10.48550/arXiv.1907.10902.

## Vitae

Name : Eyad Abdullah Alshaye |

Nationality : Saudi Arabian |

Date of Birth : 29 Marth, 1997 |

Email : iyadalshaye@gmail.com |

Address : 7661 Abdul Rahman Ibn Auf, 2848, Ash Shulah, 34266,  
Dammam, Saudi Arabia |

Academic Background : Master's Student, Materials Science and Engineering  
Department, KFUPM |

### **Papers:**

**Supervised Machine Learning Models for Predicting the Maximum Depth of Corrosion Defects Based on Historical In-line Inspection Data.** (Submitted to Engineering Applications of Artificial Intelligence Journal)

**Tree-based Machine Learning Models for Predicting the Maximum Depth of Corrosion Defects Based on Historical In-line Inspection Data.** (Submitted to the journal of Pipeline Science and Engineering)