Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
University Centre of Illizi
Institute of Science

Department of Computer Science

Artificial Intelligence & its Applications Master's Thesis

Intelligent Corrosion Prediction and Assessment Using BiLSTM, XGBoost, and YOLOv8-Seg Models

Authors:

- Amraoui Sidahmed
  sidahmed.amraoui@cuillizi.dz

Advisors:

- Dr. Guessoum Belkis
  guessoum.belkis@cuillizi.dz

Examining Committee:

- (*President*) Dr. Khelifi Ramzi
  khelifi.ramzi@cuillizi.dz
- (*Examiner*) Dr. Mohammed Elhabib Kahla
  kahla.med-elhabib@cuillizi.dz
- (*Examiner*) Dr. Benmerzoug Nabil
  nabil.benmerzoug@cuillizi.dz
- (*Examiner*) Dr. Benameur Imadeddine
  benameur.imadeddine@cuillizi.dz

Academic Year: 2025/2026

Abstract

Industrial corrosion represents one of the most persistent and economically damaging challenges facing modern infrastructure. This thesis proposes an AI-based corrosion prediction system CorroSense AI composed of three Overlapping components: a Bidirectional Long Short-Term Memory model for predicting Remaining Useful Life from wall-thickness inspection records, an eXtreme Gradient Boosting regressor for real-time atmospheric corrosion rate forecasting, and a You Only Look Once v8-seg segmentation model for visual detection and severity classification of corroded surfaces. The system is delivered as a unified graphical interface targeting industrial maintenance engineers in sectors such as oil and gas, and power generation. Experimental evaluation demonstrates the effectiveness of each component: the Bidirectional Long Short-Term Memory achieves an R^2 of 0.89 on held-out inspection records, the eXtreme Gradient Boosting regressor achieves a test R^2 of 0.848 with physically interpretable feature importance rankings, and the You Only Look Once 8-seg model successfully detects and segments corrosion instances across ten severity classes with mAP50 sufficient for practical inspection support.

Key words: Corrosion Prediction, Remaining Useful Life, Bidirectional LSTM, XGBoost, YOLOv8-seg, Predictive Maintenance, Deep Learning.

ملخص

يمثل التآكل الصناعي أحد أكثر التحديات استمرارًا وتأثيرًا اقتصاديًا على البنية التحتية الحديثة. تقترح هذه الأطروحة نظامًا للتنبؤ بالتآكل قائمًا على الذكاء الاصطناعي، يُسمى CorroSense AI، ويتكون من ثلاثة مكونات متداخلة: نموذج ذاكرة طويلة المدى ثنائي الاتجاه للتنبؤ بالعمر المتبقي المفيد من سجلات فحص سُمك الجدار، ومُحسِّن انحدار تعزيز التدرج الشديد للتنبؤ بمعدل التآكل الجوي في الوقت الفعلي، ونموذج تجزئة ثماني الأجزاء Yolov8-seg للكشف البصري عن الأسطح المتآكلة وتصنيف شدتها. يُقدَّم النظام كواجهة رسومية موحدة تستهدف مهندسي الصيانة الصناعية في قطاعات مثل النفط والغاز وتوليد الطاقة. يُظهر التقييم التجريبي فعالية كل مكون: حيث حققت ذاكرة الوصول العشوائي ثنائية الاتجاه طويلة المدى BiLSTM معامل تحديد R^2 قدره 89.0 على سجلات الفحص المحجوزة، وحقق نموذج الانحدار المُعزز بالتدرج الشديد XGboost معامل تحديد R^2 قدره 848.0 مع تصنيفات أهمية الميزات القابلة للتفسير الفيزيائي، ونجح نموذج "أنت تنظر مرة واحدة فقط" Yolo ذو 8 أجزاء في الكشف عن حالات التآكل وتقسيمها بنجاح عبر عشر فئات من الشدة مع mAP50 وهو كافٍ لدعم الفحص العملي.

الكلمات المفتاحية: التنبؤ بالتآكل، العمر الافتراضي المتبقي، الذاكرة طويلة-قصيرة الأمد ثنائية الاتجاه، XGBoost، YOLOv8-seg، الصيانة التنبؤية، التعلم العميق.

Acknowledgements

First and foremost, all praise and gratitude are due to Almighty God, whose blessings, guidance, and mercy enabled me to complete this work and bring it to fruition.

I would like to express my deepest gratitude and sincere appreciation to **Dr. Guessoum Belkis** for her outstanding supervision, valuable guidance, continuous support, and insightful recommendations throughout the development of this thesis. Her constructive comments and encouragement greatly contributed to the quality and completion of this work.

My special appreciation goes to **Dr. Gasmi Rym, Head of the Computer Science Department,** for her support, encouragement, and continuous efforts in providing a favourable environment for academic learning and scientific research.

I am also deeply grateful to all my professors and teachers who contributed to my academic formation throughout my years of study. Their knowledge, dedication, and guidance played a fundamental role in shaping my academic background and developing my skills.

Finally, I pray that God rewards all those who have supported me and that this work may serve as a valuable contribution to knowledge and learning.

And all praise is due to God, Lord of the Worlds.

Dedication

To the one whose prayers have always illuminated my path, whose patience has been my support throughout my journey, and whose love has been an endless source of strength...

To my beloved mother, may God bless her with a long life, good health, and happiness.

To the soul of my beloved father, who departed this world before witnessing the fruits of the values, knowledge, and perseverance he instilled in me...

May God have mercy upon you, dear Father, grant you the highest ranks in Paradise, and reward you abundantly for all that you have done for me.

To my dear brother and companion, who has always been a source of support and encouragement through both difficult and joyful times...

To my dear brother and companion.

To the soul of my beloved grandmother, whose cherished memory remains forever alive in my heart...

May God grant her His infinite mercy and reunite us with her in His eternal Paradise.

To my faithful friends who have stood by me, shared laughter and challenges, and made this journey unforgettable...

To Didin, Abdul, Haythem, and Wakil — thank you for your friendship and support.

I dedicate this humble work to you all.

Contents

List of Figures

List of Tables

List of Abbreviations

AI	Artificial Intelligence
API	American Petroleum Institute
AUC	Area Under the Curve
BiLSTM	Bidirectional Long Short-Term Memory
COCO	Common Objects in Context (image dataset)
CNN	Convolutional Neural Network
CV	Cross-Validation
CUDA	Compute Unified Device Architecture
CSV	Comma-Separated Values
EarlyStopping	Keras callback that halts training when validation loss stops improving
FPN	Feature Pyramid Network
GUI	Graphical User Interface
GPU	Graphics Processing Unit
IoU	Intersection over Union
ISO	International Organization for Standardization
LT	Limit Thickness (minimum allowable wall thickness)
LSTM	Long Short-Term Memory
L1	Lasso regularisation (sum of absolute weights)
L2	Ridge regularisation (sum of squared weights)
MAE	Mean Absolute Error
mAP	mean Average Precision
ML	Machine Learning
MSE	Mean Squared Error
NACE	National Association of Corrosion Engineers
$\mathbf{NO_2}$	Nitrogen Dioxide
NT	Nominal Thickness (original design wall thickness)
$\mathbf{O_3}$	Ozone
PDF	Portable Document Format
PdM	Predictive Maintenance

PHM	Prognostics and Health Management
QNH	Barometric pressure at mean sea level (hPa)
R^2	Coefficient of Determination
RAM	Random Access Memory
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SCC	Stress Corrosion Cracking
SO_2	Sulphur Dioxide
SSD	Solid-State Drive
SVR	Support Vector Regression
TML	Thickness Measurement Location
TOW	Time of Wetness
Tol	Tolerance (manufacturing tolerance on wall thickness)
TR	Thickness Reading (current measured wall thickness)
UTM	Ultrasonic Thickness Measurement
YOLO	You Only Look Once
XGBoost	eXtreme Gradient Boosting

General Introduction

Industrial corrosion is one of the most persistent and economically damaging threats faced by modern infrastructure. According to studies published by National Association of Corrosion Engineers, the global annual cost of corrosion exceeds 2.5 trillion US dollars, accounting for roughly 3.4% of world GDP [1].

In sectors such as oil and gas, petrochemicals, water treatment, power generation, and heavy manufacturing, the progressive degradation of metallic equipment through corrosion leads to unplanned production shutdowns, catastrophic structural failures, environmental contamination, and significant safety risks.

The problem is not limited to a single failure mode corrosion manifests as wall-thickness loss in pressurized vessels and heat exchangers, accelerated surface degradation driven by atmospheric pollutants, and visible surface damage that spreads silently until it becomes critical [2].

Traditional maintenance approaches periodic manual inspections, reactive repairs after failure, or fixed time-based replacement schedules are ill-suited to the complexity and variability of real-world corrosion. These methods are expensive, discontinuous, and fundamentally reactive: they detect damage only after it has already advanced. Moreover, the factors governing corrosion rates are numerous and interlinked, spanning environmental conditions such as humidity, temperature, and atmospheric pollutant concentrations, as well as material properties, equipment geometry, and operational history.

No human inspector can integrate all of these variables simultaneously and reliably predict when a given piece of equipment will reach the end of its safe service life. A more intelligent and data-driven approach is therefore needed.

Research Problem

The central question driving this work is: to what extent can artificial intelligence through the combination of temporal sequence modelling of wall-thickness measurements, environmental sensor-based corrosion rate prediction, and automated visual segmentation of corroded surfaces provide reliable, quantitative, and actionable estimates of the health state and remaining service life of industrial equipment subject to corrosion?

This question involves three distinct technical challenges: capturing long-range temporal dependencies in sparse, irregularly sampled inspection records across heterogeneous equipment types; learning the non-linear mapping from ten atmospheric sensor signals to an instantaneous copper corrosion rate while avoiding data leakage in a chronological time-series setting; and robustly detecting and segmenting visually heterogeneous corrosion patterns across diverse photographic conditions.

Research Objectives

To address this problem, this thesis pursues the following specific objectives:

- Train a Bidirectional Long Short-Term Memory model on real industrial wall-thickness inspection records covering eleven equipment types, using eleven engineered features per inspection point and temporal augmentation, to predict Remaining Useful Life in years with a maximum cap of 50 years.

- Develop and validate an XGBoost corrosion rate model trained on ten raw environmental sensor features with strict chronological splitting and walk-forward cross-validation, capable of predicting the 6-hour-ahead copper corrosion rate and computing Remaining Useful Life relative to a user-specified material end-of-life limit.

- Fine-tune a YOLOv8-seg instance segmentation model on a custom corrosion image dataset to detect, classify by severity, and quantify the area fraction of corroded regions in equipment photographs.
- Evaluate all three models using objective, leak-free metrics: Mean Absolute Error, Root Mean Squared Error, and Coefficient of Determination for the two numerical prediction models, and mean Average Precision 50 for the segmentation model.
- Integrate the three components into a unified graphical interface that allows maintenance engineers to use all three prediction modes without requiring expertise in machine learning.

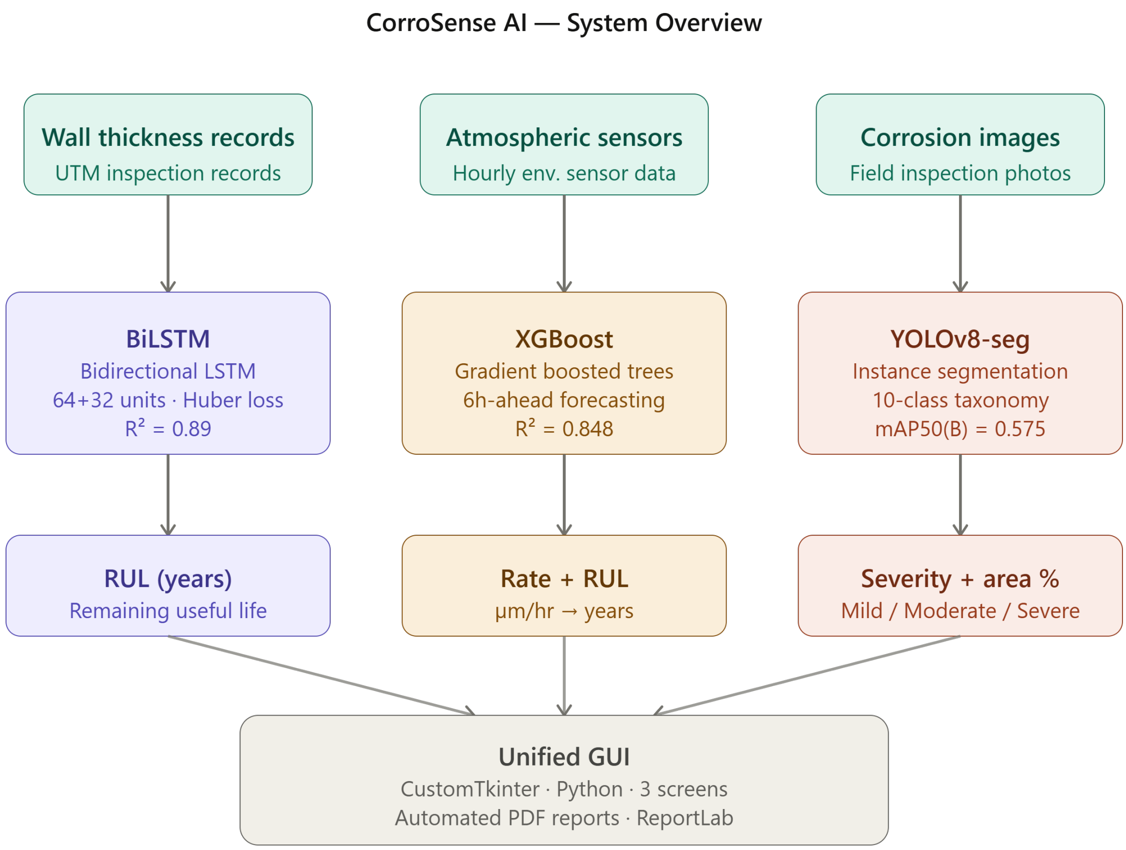


Figure 1: CorroSense AI — System Overview

Thesis Structure

This thesis is organised into five chapters:

- **Chapter One** establishes the theoretical foundation: it covers the mechanisms and typology of industrial corrosion, the environmental and material factors governing degradation rates, the principles of the three Artificial Intelligence architectures employed Bdirectional LSTM, XGBoost, and YOLOv8-seg and a review of related work in the literature.

- **Chapter Two** describes the methodology applied to each of the three system components: data sources and formats, quality control and preprocessing pipelines, feature engineering, model architectures and hyperparameters, training protocols and validation strategies, and evaluation metrics.
- **Chapter Three** presents the implementation environment (hardware, software stack, and dataset characterisation) followed by the experimental results obtained for all three components, including training dynamics, test-set performance metrics, per-equipment-type and per-class analyses, and a description of the integrated graphical user interface.
- **Chapter Four** provides a critical analysis of these results, comparisons against published benchmarks and simple baselines, an examination of the cross-system integration trade-offs, and a detailed set of directions for future research and development.
- **Chapter Five** presents the general conclusion of this thesis, summarising the main contributions across its three system components, reflecting on the scientific and industrial significance of the findings, acknowledging the limitations encountered, and outlining the most promising directions for future work.

Chapter 1

Theoretical Background and Literature Review

1.1 Introduction

Industrial infrastructure operating in aggressive environments faces a constant, invisible threat: corrosion. Left unmanaged, corrosion silently reduces the structural integrity of metallic components until failure occurs, often with catastrophic consequences. This chapter establishes the theoretical foundation necessary to understand corrosion prediction approaches of the kind addressed in this thesis. It begins by defining corrosion and examining its most common forms and governing mechanisms. It then reviews the environmental and material factors that control the rate of degradation. The second half of the chapter introduces the three families of artificial intelligence architectures relevant to this domain recurrent sequence models, gradient-boosted tree ensembles, and real-time instance segmentation models and closes with a survey of related work in the literature.

1.2 Predictive Maintenance and Industry 4.0

1.2.1 From Reactive to Predictive Maintenance

Industrial maintenance strategy has evolved through three broad paradigms [3]. *Reactive maintenance* also called run-to-failure performs repair only after a component has broken down. While simple to implement, this approach incurs the highest long-term costs because failures are unplanned, often occur at the worst possible moment, and may cascade to surrounding equipment. *Preventive maintenance* introduces fixed time-based or usage-based schedules for inspection and replacement, reducing the frequency of catastrophic failures but wasting resources by replacing components that still have significant service life remaining. *Predictive maintenance* (PdM), the paradigm pursued in this thesis, uses real-time monitoring data and mathematical models to estimate the current degradation state and project the Remaining Useful Life (RUL) of individual components, enabling maintenance interventions to be scheduled at the most economical moment neither too early nor too late [3].

The economic case for predictive maintenance in the corrosion domain is compelling. Studies conducted across oil and gas, petrochemical, and power generation sectors consistently find that unplanned downtime costs three to five times more per hour than planned maintenance shutdowns [1]. A system capable of providing even a 30-day advance warning of imminent equipment retirement enables the operator to schedule the shutdown during a planned turnaround window, pre-order replacement materials, and mobilise the appropriate specialist workforce transforming an emergency repair into a routine activity.

1.2.2 The Role of Artificial Intelligence in Corrosion Management

Artificial intelligence offers several specific advantages over the classical empirical and analytical approaches to corrosion prediction. Classical dose-response functions empirical equations relating corrosion rate to pollutant concentrations and meteorological variables, standardised in International Organization for Standardization 9223 [4] are derived from long-term outdoor exposure tests under carefully controlled conditions and are calibrated for specific metal types and geographic regions. Their applicability is therefore limited when the operating environment deviates significantly from the calibration conditions, or when the target metal is not covered by the standard.

Machine learning models, by contrast, learn the input-output relationship directly from the available measurement data without requiring the modeller to specify the functional form of the relationship in advance. This makes them inherently adaptive: a model retrained on new data from a different location or a different material will automatically adjust to the local environmental dynamics [5]. Deep learning models such as the Bidirectional LSTM are further capable of extracting complex, non-linear, and long-range temporal patterns from sequential data [6] patterns that no empirical equation can capture. Computer vision models such as YOLOv8-seg can process raw photographic data directly, eliminating the manual feature extraction step that has historically made visual inspection subjective and labour-intensive.

1.3 Industrial Corrosion: Definition, Types, and Mechanisms

1.3.1 Definition

Corrosion is defined as the irreversible deterioration of a material, usually a metal or alloy, through chemical or electrochemical reaction with its surrounding environment [7–9]. Unlike mechanical wear, corrosion involves the transfer of electrons and ions between the metal surface and the corrosive medium, resulting in a progressive loss of material, a reduction in cross-sectional thickness, and ultimately a degradation of mechanical properties including tensile strength, ductility, and fatigue resistance.

1.3.2 Types of Corrosion

Corrosion does not manifest in a single uniform way. Engineers and scientists have identified numerous distinct types, each governed by different physical and chemical mechanisms. Understanding these types is essential for selecting the appropriate monitoring strategy and the appropriate AI model for each scenario.

Uniform corrosion : degrades the metal surface at an approximately equal rate across the entire exposed area [7]. It is the dominant mode targeted by the BiLSTM component of this system.

Galvanic corrosion : occurs when two dissimilar metals are in electrical contact within an electrolyte [8]. The more active metal corrodes preferentially while the nobler metal is protected.

Pitting corrosion : is a localised form that creates deep holes while the surrounding surface remains largely unaffected [8]. It is particularly dangerous because it can cause sudden through-wall penetration that average-thickness measurements may not detect.

Crevice corrosion : develops in confined spaces such as under gaskets or bolt heads, where restricted mass transport leads to oxygen depletion and acid accumulation, driving rapid local dissolution.

Stress corrosion cracking (SCC) : results from the combined action of sustained tensile stress and a corrosive environment, producing brittle cracks at stress levels well below the material's yield strength.

Intergranular corrosion : attacks grain boundaries rather than grain interiors, most commonly in sensitised stainless steels where chromium-depleted zones form along the boundaries.

Atmospheric corrosion : is the degradation of metals exposed to outdoor air [10], and is the primary mechanism targeted by the XGBoost component. Copper is particularly sensitive, forming patina layers whose growth rate is controlled by humidity, temperature, and pollutant concentrations such as Sulphur Dioxide, O_3, and Nitrogen Dioxide$_2$.

1.3.3 Electrochemical Mechanisms

At the fundamental level, corrosion in aqueous or humid environments is an electrochemical process comprising two coupled reactions: an anodic half-reaction in which metal atoms lose electrons and dissolve as ions, and a cathodic half-reaction in which those electrons are consumed by an oxidising species such as dissolved oxygen or hydrogen ions [11]. The driving force is the electrical potential difference between the anodic and cathodic sites. When these sites are spatially separated due to surface heterogeneity, compositional gradients, or differential aeration a local galvanic cell forms that sustains continuous metal dissolution without any external current. The overall rate is governed by the kinetics of both half-reactions, which depend on temperature, electrolyte chemistry, and the nature of any protective oxide film on the surface.

1.4 Environmental and Material Factors Governing Corrosion Rate

1.4.1 Humidity

Relative humidity is the most critical atmospheric variable governing corrosion rate [12, 13]. Below approximately 60%, the adsorbed water film is too thin to sustain ionic conduction and corrosion is negligible. Above this threshold, the rate increases sharply and non-linearly. For copper, high humidity accelerates the formation of soluble sulphate complexes when SO_2 is present [14].

A related concept is the *time of wetness* (TOW), defined by ISO 9223 as the number of hours per year during which temperature exceeds 0C and relative humidity exceeds 80% [4]. TOW is a better annual predictor than instantaneous humidity because it integrates all surface wetting events. In the XGBoost dataset, relative humidity acts as an hourly proxy for surface wetness probability.

1.4.2 Temperature

Temperature influences corrosion kinetics through its effect on reaction rates, diffusion coefficients, and gas solubility in the electrolytic film. Higher temperatures generally accelerate electrochemical reactions, but also reduce the adsorbed water film thickness and oxygen solubility, partially counteracting the acceleration. The net effect on copper corrosion rate is therefore non-monotonic and depends on concurrent humidity and pollutant levels.

Dew point temperature is particularly important: surface condensation occurs whenever the metal temperature falls below the dew point, creating a continuous liquid film even when ambient humidity is below the 80% TOW threshold. In the XGBoost feature set, both `temp_out` and `dew_out` are included, allowing the model to implicitly capture surface condensation risk.

1.4.3 Atmospheric Pollutants: SO_2, O_3, and NO_2

Sulphur dioxide (SO_2) is the most damaging atmospheric pollutant for metallic corrosion [13]. It dissolves in the surface water film to form sulphuric acid, which attacks the metal oxide layer and accelerates all subsequent electrochemical reactions. For copper, SO_2 is the primary initiator of the characteristic green patina [14].

Ozone (O_3) promotes the oxidation of SO_2 to SO_3, further increasing acidity, and also directly attacks organic protective coatings. Nitrogen dioxide (NO_2) dissolves to form nitric acid, adding an additional acid attack vector that synergistically amplifies the effect of SO_2. In the XGBoost training dataset, these three pollutant concentrations are among the most informative features for predicting the instantaneous copper corrosion rate [12].

1.4.4 Wind Speed and Atmospheric Pressure

Wind speed has two opposing effects: it promotes deposition of corrosive aerosols at low speeds, but removes moisture and dilutes pollutants at high speeds. The net effect is non-monotonic and site-specific, making it one of the less informative features in the XGBoost model. Atmospheric pressure (Barometric pressure at mean sea level) modulates the partial pressure of oxygen and water vapour, and serves as an indirect indicator of corrosion risk since low-pressure systems are typically associated with higher humidity and precipitation [12].

1.4.5 Material Properties: Wall Thickness

In pressurized equipment vessels, heat exchangers, pipelines, and storage tanks the primary consequence of corrosion is the reduction of wall thickness [15, 16]. Engineering codes define a minimum allowable thickness (Limit Thickness) below which the equipment can no longer safely contain the process pressure. The difference between the current measured thickness (TR) and this limit is the remaining corrosion allowance the quantity whose time evolution determines the Remaining Useful Life.

1.4.6 Ultrasonic Thickness Inspection and the RUL Concept

Ultrasonic thickness measurement (UTM) is the primary non-destructive technique for quantifying wall-thickness loss due to corrosion in pressurized equipment [15, 17]. A piezoelectric transducer coupled to the external surface of the component emits a high-frequency ultrasonic pulse that travels through the wall, reflects off the internal surface, and returns to the transducer. The wall thickness is computed from the measured round-trip travel time and the known speed of sound in the material. UTM is non-invasive (no process shutdown or internal access is required), fast (individual readings take seconds), and can detect average thickness loss at the measurement location to an accuracy of ± 0.1 mm or better.

Thickness Measurement Locations (TMLs) are pre-defined points on the equipment surface where repeat measurements are taken at each scheduled inspection. By comparing successive UTM readings at the same TML, the corrosion rate (in mm/year) can be estimated and the Remaining Useful Life computed using the linear extrapolation formula:

$$\text{RUL} = \frac{TR - LT}{\text{corrosion rate}}$$

This formula assumes a constant corrosion rate, which is a reasonable first approximation for uniform corrosion but may be inaccurate if the rate is accelerating (due to coating failure or process change) or decelerating (due to protective scale formation). The BiLSTM model developed in this thesis learns directly from sequences of five consecutive TML readings, allowing it to capture non-linear degradation trends that the simple linear extrapolation formula cannot represent.

1.5 Artificial Intelligence Architectures Used

1.5.1 Recurrent Neural Networks and Bidirectional LSTM

Standard feedforward neural networks treat each input independently, with no internal mechanism for retaining information about previous inputs. Recurrent Neural Networks (Recurrent Neural Networks) address this limitation by introducing directed cycles in the network graph at each time step t, the hidden state $h(t)$ is computed as a function of both the current input $x(t)$ and the previous hidden state $h(t-1)$. In practice, however, training vanilla RNNs on long sequences suffers from the vanishing gradient problem [18]: as error signals are propagated backwards through many time steps, they are multiplied by the network weight matrices at each step, causing gradients to shrink exponentially and making it impossible for the network to learn dependencies spanning more than a few steps.

The Long Short-Term Memory (LSTM) architecture, introduced by Hochreiter and Schmidhuber in 1997, solves the vanishing gradient problem through a carefully engineered gating mechanism [19]. Each LSTM unit maintains two internal states: the cell state $C(t)$ and the hidden state $h(t)$. Three learnable gates regulate the flow of information:

- The *forget gate*: $f(t) = \sigma(W_f \cdot [h(t-1), x(t)] + b_f)$
- The *input gate*: $i(t) = \sigma(W_i \cdot [h(t-1), x(t)] + b_i)$, with cell state update $C(t) = f(t) \odot C(t-1) + i(t) \odot g(t)$
- The *output gate*: $o(t) = \sigma(W_o \cdot [h(t-1), x(t)] + b_o)$, with $h(t) = o(t) \odot \tanh(C(t))$

A Bidirectional LSTM (BiLSTM) runs two separate LSTM layers over the same sequence one forward and one in reverse and concatenates their hidden states [20]. This allows the model to incorporate context from both past and future observations. The LSTM component of this thesis uses a BiLSTM architecture with two stacked layers of 64 and 32 units respectively.

1.5.2 Gradient Boosting: XGBoost

XGBoost, introduced by Chen and Guestrin in 2016, is an optimised implementation of the gradient boosting framework [21]. Gradient boosting builds an ensemble of decision trees sequentially[22]: each new tree is trained to predict the residuals left by the previous ensemble [23]. Formally, the ensemble prediction after K trees is:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}$$

where $\mathcal{F}$ is the space of regression trees and x_i is the feature vector for observation i. At each boosting step k, the new tree f_k is found by minimising the regularised objective:

$$\mathcal{L}^{(k)} = \sum_{i=1}^{n} l\Big(y_i,\, \hat{y}_i^{(k-1)} + f_k(x_i)\Big) + \Omega(f_k)$$

where l is the loss function and $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ penalises tree complexity through the number of leaves T and the L2 norm of leaf weights w. XGBoost extends classical gradient boosting with a second-order Taylor expansion of the loss that enables fast, closed-form computation of the optimal leaf weights; column and row subsampling that reduces variance and computation time; and cache-aware data structures that improve memory access patterns on modern processors.

For the corrosion rate prediction task, XGBoost was chosen because the input is a fixed ten-dimensional feature vector per hourly observation, the dataset size is moderate (approximately 13 months of hourly readings), and XGBoost's robustness to feature scale and built-in regularisation make it highly suitable for environmental sensor data. The shallow tree depth (`max_depth` = 3) and the high minimum child weight (`min_child_weight` = 20) were selected to prevent the model from fitting fine-grained noise in the hourly sensor readings, which can be substantial due to measurement artefacts and instrument drift.

1.5.3 Instance Segmentation: YOLOv8-seg

YOLO (You Only Look Once) is a family of single-stage object detection architectures that process the entire image in a single forward pass [24]. This contrasts with two-stage detectors such as Faster R-Convolutional Neural Network [25], which first generate region proposals and then classify each proposal separately. The single-stage design achieves substantially higher inference speed a critical property for real-time industrial inspection applications.

YOLOv8, developed by Ultralytics and released in 2023, is the eighth major iteration of the architecture [26]. Its backbone produces multi-scale feature maps at three resolutions. A Feature Pyramid Network [27] neck aggregates these features into a set of detection heads, one per scale. The YOLOv8-seg variant extends the detection head with a parallel mask prediction branch: the mask branch predicts a set of prototype masks and per-detection coefficients; the final instance mask is a linear combination of the prototypes weighted by those coefficients, evaluated at the original image resolution after bilinear upsampling.

The YOLOv8n-seg (nano) variant used in this thesis has approximately 3.4 million parameters. This choice was motivated by the limited size of the training dataset: larger variants risk overfitting, whereas the nano variant's lower capacity makes it more conservative and better suited to the fine-tuning regime. Transfer learning from COCO pre-trained weights [28, 29] provides the backbone with general visual features edges, textures, and object parts that transfer effectively to the corrosion domain, where the key discriminative cues (surface texture, colour, and irregular boundaries) are analogous to the natural image features the backbone was trained to detect.

1.6 Review of Related Work

The three research threads that underpin CorroSense AI — machine learning for atmospheric corrosion rate prediction, deep learning for equipment Remaining Useful Life (RUL) estimation, and computer vision for visual corrosion detection — have each attracted substantial independent attention in the literature. The subsections below survey the most relevant prior work in each thread, discuss its limitations, and explain how CorroSense AI advances beyond the current state of the art. Section 1.6.4 then addresses the near-absence of integrated multi-modal systems and positions CorroSense AI within that gap.

1.6.1 Machine Learning for Corrosion Rate Prediction

The application of machine learning to atmospheric corrosion prediction has grown substantially over the past decade. Early work demonstrated that support vector regression (SVR) trained on meteorological and pollution variables could predict atmospheric corrosion rates for carbon steel with errors comparable to the best empirical dose-response functions [**?**]. Subsequent studies showed that random forests and gradient-boosted trees consistently outperformed linear models in their ability to handle the non-linear, interactive effects of humidity, temperature, and pollutant concentration on corrosion rate.

Pei et al. [30] monitored the short-term atmospheric corrosion of carbon steel for 34 days using an Fe/Cu galvanic corrosion sensor and applied Random Forest to rank feature importance. Temperature, relative humidity, and rainfall status were identified as the dominant predictors of daily corrosion rate, and Random Forest outperformed classical dose-response functions on the short-term data. However, the study was limited to a 34-day observation window and did not extend predictions to a Remaining Useful Life estimate.

More recently, LSTM networks applied to multi-site corrosion monitoring data demonstrated that temporal context improved prediction accuracy over models that treated each hourly observation independently. Medina-González et al. (2022) specifically studied machine learning for copper corrosion rate prediction under atmospheric conditions and found that gradient-boosted models outperformed both linear regression and neural networks when the available dataset covered fewer than 24 months of hourly observations [31] — a finding directly relevant to the XGBoost component of this thesis, which operates on a 13-month dataset.

Wu et al. [32] constructed ML models including Random Forest, Gradient Boosted Decision Trees, and XGBoost to predict the atmospheric corrosion rate of low-alloy steels from material properties, environmental factors, and exposure time. XGBoost achieved a test $R^2 \approx 0.77$, but the authors noted limited generalisation at high corrosion-rate extremes due to data scarcity. CorroSense AI addresses this by using a chronological walk-forward cross-validation strategy that explicitly prevents temporal data leakage, and by targeting copper corrosion driven by high-frequency (hourly) sensor readings rather than annual or monthly exposure averages.

Positioning of CorroSense AI.

The XGBoost component of CorroSense AI builds directly on the above findings. It extends the approach of Medina-González et al. by incorporating three atmospheric pollutants (SO_2, O_3, NO_2) alongside standard meteorological features, achieving a test $R^2 = 0.848$ on a 13-month hourly dataset competitive with the best published results on comparably sized corrosion datasets. Crucially, the predicted corrosion rate is then converted to a RUL estimate, creating a direct link between environmental sensing and actionable maintenance scheduling that prior rate-prediction studies do not provide.

1.6.2 Deep Learning for Equipment Remaining Useful Life

RUL prediction is a well-established problem in prognostics and health management (PHM). The dominant approach has moved from physics-based extrapolation to sequence-to-scalar deep learning, with LSTM variants achieving state-of-the-art performance on standard benchmarks.

Zheng et al. [33] demonstrated that stacked LSTMs could capture the non-linear degradation trajectories of turbofan engine sensors, outperforming conventional recurrent architectures on the

NASA C-MAPSS benchmark. This work established multi-layer LSTMs as the standard baseline for sequence-based RUL prediction in PHM.

In the corrosion domain, Cui et al. [34] applied Gated Recurrent Units (GRUs) to pipeline inline inspection data and showed that sequence models could predict wall thickness at future inspection dates with lower RMSE than simple linear extrapolation. Their study demonstrated the feasibility of deep learning for corrosion-specific RUL, but was limited to a single pipeline type and did not generalise across equipment categories.

Wu, Wang et al. [35] proposed an RF-BiLSTM hybrid where Random Forest first extracts health indicators from high-dimensional sensor streams, and a BiLSTM then maps those indicators to RUL. Evaluated on the NASA C-MAPSS aviation dataset, RF-BiLSTM achieved an RMSE 47.96% lower than a vanilla LSTM and 86.53% lower than AdaBoost, confirming that bidirectional architectures improve RUL accuracy over unidirectional alternatives. However, the domain is continuous vibration and pressure sensing in jet engines far removed from sparse, irregular ultrasonic thickness inspection records in industrial corrosion management.

Positioning of CorroSense AI.

A common limitation of existing RUL prediction studies is that models are trained on a single equipment type or a single operating unit. The BiLSTM system developed in this thesis addresses this cross-equipment generalisation gap by training a single model on eleven distinct equipment categories simultaneously, using the equipment-type code as a conditioning feature. This design is motivated by the practical reality of industrial inspection programmes, where some equipment types have very few records and would not support training a dedicated model. The BiLSTM achieves $R^2 = 0.89$ on held-out inspection records across all eleven categories, demonstrating that a single model can generalise across heterogeneous corrosion degradation patterns.

1.6.3 Computer Vision for Corrosion Detection

Visual inspection of corroded surfaces using deep learning has attracted significant research interest. Semantic segmentation models — notably U-Net [36] and DeepLab — were applied to produce pixel-level maps of corroded regions on bridge structures and storage tanks. The transition to instance segmentation models such as Mask R-CNN allowed individual corrosion patches to be counted and characterised independently, enabling patch-level severity metrics rather than global pixel fractions.

Beng Chye et al. [37] demonstrated that Mask R-CNN fine-tuned on curated industrial inspection datasets can achieve corrosion detection precision and recall comparable to expert human inspectors on standardised test sets, establishing instance segmentation as the performance target for automated visual corrosion assessment.

Gao et al. [38] conducted a head-to-head comparison of Mask R-CNN and YOLOv8 for instance segmentation of steel bridge corrosion using three severity classes (Fair, Poor, Severe). They found that YOLOv8 matches Mask R-CNN on mAP while offering substantially higher inference speed, establishing YOLOv8-seg as a practical alternative to two-stage detectors for field inspection scenarios.

A comparative study by Cawilai et al. [39] evaluated YOLOv5 and YOLOv8 across three corrosion datasets and confirmed that YOLOv8-seg consistently outperforms YOLOv5 on both mAP50 and mAP50-95 metrics, further supporting the selection of YOLOv8-seg as the segmentation backbone.

Positioning of CorroSense AI.

The use of YOLOv8-seg for corrosion detection represents a more recent development, combining real-time inference speed with instance-level severity classification and area quantification within a single model forward pass. Compared to Mask R-CNN, YOLOv8-seg achieves competitive instance segmentation quality at significantly higher throughput, making it more suitable for deployment on edge devices in field inspection scenarios [38, 39]. CorroSense AI extends the approach of Gao et al. by expanding the severity taxonomy from three classes to ten, enabling fine-grained classification that the bridge study's coarser labelling scheme cannot support. The benchmark established by Beng Chye et al. [37] motivates the evaluation protocol adopted in this thesis.

1.6.4 Integrated Multi-Modal Predictive Maintenance Systems

While the individual tasks of RUL prediction from thickness records, environmental corrosion rate forecasting, and visual corrosion detection have each been studied in isolation, very few published systems address all three modalities within a single integrated framework. Most commercial and academic predictive maintenance platforms focus on a single data type — either sensor time-series, inspection images, or structured measurement records — and do not provide the cross-modal corrosion health picture that maintenance engineers require in practice [3].

Hosseinzadeh and Bahaari [40] proposed a hybrid framework coupling physics-based simulation, IoT sensors, and machine learning for pipeline corrosion monitoring. Although the system integrates heterogeneous data streams, it does not include a computer vision component and requires a physics-based simulation layer that limits its applicability to pipelines where the governing equations are well characterised. No unified graphical interface or automated reporting pipeline is described.

Jamaludin et al. [41] reviewed AI methodologies for online corrosion monitoring in marine structures and proposed a conceptual framework combining image-based detection with structural health monitoring signals. However, no quantitative evaluation of a deployed integrated system is provided, and the framework remains at the design level.

The CorroSense AI system developed in this thesis is, to the best of the author's knowledge, the first academic prototype to integrate Bidirectional LSTM-based thickness RUL prediction, XGBoost-based environmental corrosion rate forecasting, and YOLOv8-seg visual inspection into a single unified Graphical User Interface with automated Portable Document Format reporting. This integration addresses a genuine gap in the available tooling for industrial corrosion management.

1.7 Conclusion

This chapter has established the theoretical and contextual foundations for the remainder of the thesis. Corrosion is a multi-modal phenomenon governed by a complex interplay of material, environmental, and geometrical factors. The transition from reactive to predictive maintenance is both technically feasible and economically compelling, with AI providing the tools to make it practical at scale. The three architectures reviewed BiLSTM, XGBoost, and YOLOv8-seg are each matched to a specific data modality: BiLSTM processes wall-thickness inspection sequences, XGBoost handles the environmental feature vector, and YOLOv8-seg performs instance segmentation on corrosion photographs. The methodology underlying each component is detailed in the following chapter.

Chapter 2

Methodology

2.1 Introduction

This chapter describes the complete methodology applied to each of the three system components developed in this thesis. For each component, the methodology covers: the source, structure, and format of the training data; the data quality and preprocessing pipeline applied before model training; the feature engineering transformations; the model architecture and its hyperparameters; the training protocol and validation strategy; and the evaluation metrics used to measure performance.

A unifying principle across all three components is the strict separation between training and test data. In all cases, information from the test set is not used in any way during training, preprocessing, or hyperparameter selection. For temporal datasets, this separation is enforced chronologically: the test set always consists of later observations than the training set, reflecting the operational scenario in which a deployed model must predict future behaviour from past data. For image data, a standard random stratified split is used because photographic data is not inherently temporal. This commitment to leak-free evaluation is the methodological foundation that makes the reported performance metrics operationally meaningful rather than optimistically inflated.

A second unifying principle is the preference for physically interpretable models and features over opaque black-box representations. Where applicable, engineered features carry direct physical meaning — quantities such as remaining corrosion allowance or degradation percentage — rather than abstract statistical transforms. Model-level outputs such as feature importance scores are inspected and verified against physical domain knowledge, and class taxonomies are grounded in standard corrosion classification terminology rather than arbitrary data-driven clusters. This interpretability is essential for the trust and adoption of AI-based maintenance systems by industrial engineers, who must understand and validate model outputs before acting on them.

Table 2.1 provides a high-level summary of the three components and their key design choices.

Table 2.1: High-level summary of the three CorroSense AI components

Component	Data Type	Model	Output
Thickness RUL	Tabular time-series	BiLSTM	RUL in years
Corrosion Rate	Atmospheric sensors	XGBoost	Rate (µm/hr) + RUL
Visual Detection	Photographs	YOLOv8-seg	Severity + area %

2.2 Component 1: Bidirectional LSTM for Thickness-Based RUL Prediction

2.2.1 Data Sources and Structure

The training data for the LSTM model was collected from multiple industrial facilities operating in oil and gas processing [15]. The raw data consisted of Excel spreadsheets containing the results of periodic ultrasonic thickness inspections conducted at Thickness Measurement Locations (TMLs). Table 2.2 describes the primary columns in the raw data.

The raw files were loaded and validated by the `load_all_files()` function, which handled three distinct file formats: 13-column files with an explicit equipment code column, 12-column files without it, and 14-column pipeline inspection files with circuit and line identifiers. A validity filter removed

Table 2.2: Raw data columns from ultrasonic thickness inspection records

Column	Full Name	Description
TR	Thickness Reading	Current measured wall thickness (mm)
NT	Nominal Thickness	Original design thickness before any corrosion
LT	Limit Thickness	Minimum allowable thickness defined by the design code
Tol	Tolerance	Manufacturing tolerance added to the design minimum
LC	Last Check date	Date of the inspection (used for temporal ordering)

rows with non-positive thickness values, rows where TR was less than LT, and rows with missing or unparseable dates.

2.2.2 Equipment Classification

A key feature of the dataset is its heterogeneity: inspection records come from eleven different equipment types with significantly different degradation rates, geometries, and service conditions. Rather than training separate models for each type, a single model was trained with the equipment type encoded as a numerical feature. Table 2.3 lists the eleven equipment categories and their identifier patterns.

Table 2.3: Equipment type classification used in the LSTM model

Equipment Type	Code	Identifier Patterns
Heat Exchanger	0	EA, HC, HA, HZ, FY, -E-
Vessel	1	VU, VB, VD, VX, VA, VZ, VN, VK, VV, VW, VE, -V-
Filter	2	FA, F (not FY)
Storage Tank	3	T, TB, TZ, TC, TX, -T-
Valve	4	SDV, BDV, SD
Cooler	5	CL, CK
Separator	6	VS (numeric), S, -S-
Compressor	7	C (numeric), -C-
Coalescer Filter	8	CQ
Cyclone	9	CY
Pipeline	10	Identified from source file format (14-column)

2.2.3 Data Quality and Preprocessing

Before feature engineering, the raw inspection files were subjected to a multi-stage quality control pipeline. First, records with physically impossible values were removed: any row where TR ≤ 0, TR $<$ LT (current thickness already below the retirement limit), NT ≤ 0, or LT ≤ 0 was discarded as a data entry error. Second, rows with unparseable or missing inspection dates were removed, since the temporal ordering of records is fundamental to both the feature engineering and the strict chronological train/test split. Third, duplicate records defined as rows with identical equipment identifier, inspection date, and measured thickness were removed, retaining only the first occurrence.

After these filters, the remaining records were grouped by equipment unit identifier. Only equipment units with at least six valid inspection records were retained for sequence modelling, since the sliding window of length five requires at least one window plus one additional test record per unit. This grouping step is important: records from the same equipment unit must be kept together and ordered chronologically, because the BiLSTM model consumes sequences of five consecutive readings from a single unit.

2.2.4 Feature Engineering

From the four raw measurement columns (TR, NT, LT, Tol) and the inspection date, eleven engineered features were derived for each TML record. Table 2.4 describes these features and their physical meaning.

Table 2.4: Engineered features derived from raw inspection columns for the BiLSTM model

Feature	**Definition and Physical Meaning**
TR	Current measured thickness (mm) — primary degradation indicator
NT	Nominal (original) thickness (mm) — reference baseline
LT	Limit thickness (mm) — end-of-life threshold
Tol	Tolerance margin (mm)
`corr_loss`	$NT - TR$ (mm) — total thickness lost to corrosion since commissioning
`remaining`	$TR - LT$ (mm) — remaining corrosion allowance before retirement
`degrad_pct`	$(\text{corr\_loss}/(NT - LT)) \times 100$ — degradation as % of total allowance
`corr_rate`	Mean mm/year loss rate computed from consecutive inspection pairs
`thickness_ratio`	TR/NT — dimensionless wall condition index (1.0 = new, $\to 0$ = failed)
`days_elapsed`	Days since the first inspection of this equipment unit
`equipment_code`	Integer class label (0–10) encoding the equipment type

2.2.5 Target Variable: Remaining Useful Life

The target variable for each record is the Remaining Useful Life (RUL) expressed in years, computed as:

$$\text{RUL} = \frac{\text{remaining}}{\text{corr\_rate}} = \frac{TR - LT}{\text{mean corrosion rate}}$$

The RUL was capped at `MAX_RUL` = 50 years to prevent extreme outliers from dominating the loss function.

2.2.6 Data Augmentation and Train/Test Split

The training dataset suffered from class sparsity for some equipment types. To address this, a Gaussian data augmentation procedure was applied to the training split with a multiplier of `AUG_MULT` = 4 [42]: each training record was replicated four times with small random perturbations (±1.5% for TR, ±10% for corr_rate, ±5% for Tol, and ±30 days for days_elapsed). A strict temporal split was applied: the 65% oldest records formed the training set and the 35% most recent records formed the test set [43].

2.2.7 Model Architecture

The Bidirectional LSTM model was implemented in TensorFlow/Keras [44, 45]. Table 2.5 describes the sequential architecture.

Table 2.5: BiLSTM model architecture and layer parameters

Layer	**Parameters**	**Purpose**
Bidirectional LSTM	64 units × 2, return_sequences=True, L2 reg=1e-4	Captures forward and backward temporal dependencies in the 5-point inspection sequence
Batch Normalisation	—	Stabilises activations and accelerates convergence [46]
Dropout	rate = 0.40	Regularisation against overfitting [47]
Bidirectional LSTM	32 units × 2, return_sequences=False, L2 reg=1e-4	Compresses the sequence to a fixed-length representation
Batch Normalisation	—	Stabilisation
Dropout	rate = 0.35	Regularisation
Dense	32 units, ReLU [48], L2 reg=1e-4	Non-linear feature combination
Batch Normalisation	—	Stabilisation
Dropout	rate = 0.30	Regularisation
Dense (output)	1 unit, Sigmoid activation	Produces a value in [0,1] rescaled to [0, MAX_RUL] years

2.2.8 Training Protocol

The model was compiled with the Adam optimiser (learning rate = 0.0005) [49] and the Huber loss function [50]. The Huber loss was chosen over mean squared error (Mean Squared Error) because the RUL target distribution contains a long right tail records from well-maintained equipment with many years of remaining life that would cause MSE to be dominated by these high-RUL outliers even after the 50-year cap. The Huber loss behaves like MSE for small errors (below a threshold δ, set to 1.0 year) and like mean absolute error (MAE) for large errors, providing a smooth transition that reduces the influence of outliers while preserving the differentiability needed for gradient-based optimisation.

Training ran for up to 80 epochs with a batch size of 64. Three Keras callbacks governed the training process:

- **EarlyStopping** monitored the validation Huber loss with a patience of 15 epochs [51]: if the validation loss did not improve for 15 consecutive epochs, training was halted and the best-epoch weights were restored.
- **ReduceLROnPlateau** halved the learning rate after 8 epochs without validation loss improvement, allowing the optimiser to make finer updates as the loss landscape flattened.
- **ModelCheckpoint** saved the model weights to disk whenever the validation loss reached a new minimum, ensuring that the best generalising weights were available even if training continued beyond the optimal point.

Input features were scaled to [0, 1] using `MinMaxScaler` instances [52] fitted exclusively on the training split. The same fitted scalers were applied to the test split and are serialised alongside the model weights for use at inference time. The target variable (RUL in years) was also scaled to [0, 1] before training and rescaled back to years for evaluation and display, which is why the output layer uses a Sigmoid activation (constraining the raw output to [0, 1]) followed by a multiplication by `MAX_RUL` = 50.

```
SEQ_LEN    = 5
BATCH_SIZE = 64
EPOCHS     = 80
MAX_RUL    = 50.0
AUG_MULT   = 4
TEST_SIZE  = 0.35
```

Figure 2.1: Key configuration constants for the BiLSTM training

2.3 Component 2: XGBoost for Environmental Corrosion Rate Prediction

2.3.1 Data Source and Structure

The XGBoost model was trained on a 13-month time-series dataset of hourly environmental measurements collected at a monitoring station in Kbely. The dataset recorded the simultaneous corrosion of a copper test coupon alongside atmospheric sensor readings. Ten features were used: `hum_out` (outdoor relative humidity, %), `temp_out` (outdoor temperature, °C), SO_2 (ppb), O_3 (ppb), NO_2 (ppb), `corrosion` (cumulative copper corrosion, µm), `dew_out` (dew point, °C), `qnh` (barometric pressure, hPa), `windspeed` (m/s), and `corrosion_diff` (µm/hr computed by differencing consecutive corrosion readings).

2.3.2 Data Quality and Preprocessing

The raw Kbely dataset, acquired from the Zenodo repository [53], required several preprocessing steps before model training. The dataset contained a small number of rows with missing sensor readings, which were removed using a forward-fill followed by a backward-fill strategy to preserve the hourly temporal resolution without introducing large gaps. Sensor readings exhibiting values outside physically plausible ranges for example, relative humidity values exceeding 100% or SO_2 concentrations below zero were replaced with linearly interpolated values from adjacent hourly readings. The `corrosion_diff` feature, derived as the first difference of the cumulative corrosion column, was inspected for anomalous spikes caused by sensor recalibration events; spikes exceeding five standard deviations from the rolling 24-hour mean were clamped to the rolling mean value.

After cleaning, the `corrosion_diff` signal was smoothed with a 3-point centred median filter to reduce high-frequency measurement noise before constructing the 6-hour-ahead target variable. The median filter was chosen over a simple moving average because it is robust to isolated outlier readings that could otherwise distort the target distribution and mislead the model.

2.3.3 Target Variable Construction

The target variable was the instantaneous copper corrosion rate in µm/hr, predicted `FORECAST_HORIZON` = 6 hours ahead. A 3-point centred median filter was applied to the raw `corrosion_diff` signal before the 6-hour shift to smooth out measurement noise.

2.3.4 Chronological Split and Leakage Prevention

A strict chronological split was applied to prevent temporal leakage [43, 54]: the first 70% of observations formed the training set and the remaining 30% formed the test set. A 14-day purge window was discarded from the beginning of the test set to eliminate residual autocorrelation. A 3-fold walk-forward cross-validation was performed on the training set, expanding the training window progressively, with the mean R^2 across folds as the primary model selection criterion.

2.3.5 Model Architecture and Hyperparameters

The XGBoost regressor was configured with the hyperparameters shown in Table 2.6, selected through a grid search over `n_estimators` $\in \{100, 150, 200, 250, 300\}$.

2.3.6 RUL Computation from Predicted Rate

Once the XGBoost model predicts a corrosion rate `rate_pred` (µm/hr), the Remaining Useful Life is computed via a blended rate:

$$\text{rate\_blended} = 0.70 \times \text{rate\_observed} + 0.30 \times \text{rate\_pred}$$

Table 2.6: XGBoost hyperparameters selected for the corrosion rate model

Hyperparameter	**Value**	**Rationale**
objective	reg:squarederror	Standard MSE regression loss
n_estimators	Best of {100–300}	Selected by holdout R^2 grid search
learning_rate	0.05	Conservative step size to prevent overfitting
max_depth	3	Shallow trees reduce variance
min_child_weight	20	Requires 20 samples per leaf; strong regularisation
subsample	0.80	80% row subsampling per tree for diversity [22]
colsample_bytree	0.80	80% column subsampling per tree
reg_alpha (L1)	1.0	Feature sparsity encouragement
reg_lambda (L2)	5.0	Weight shrinkage for stability
random_state	42	Reproducibility

$$\text{RUL\_years} = \text{clip}\left(\frac{\text{MAT\_LIMIT} - \text{current\_corrosion}}{\text{rate\_blended\_yr}},\ 0,\ 50\right)$$

where `MAT_LIMIT` = 13,550 µm is a physics-based upper bound and `current_corrosion` is the most recent corrosion reading in the input file.

The blending weights (0.70 observed, 0.30 predicted) were chosen to reflect a conservative engineering philosophy: the observed historical rate, computed from the actual cumulative corrosion readings in the input file, is treated as more reliable than the model's 6-hour-ahead prediction because it integrates all past environmental conditions and their cumulative effect on the material. The predicted rate, while providing a forward-looking signal about imminent corrosion activity, carries model uncertainty and is therefore down-weighted. A 70/30 split provides a meaningful forward-looking correction while ensuring that a single anomalous prediction cannot drastically shorten or extend the computed RUL. The `rate_blended_yr` is obtained by multiplying `rate_blended` (in µm/hr) by 8,760 (hours per year) to convert to µm/yr.

2.4 Component 3: YOLOv8-seg for Visual Corrosion Detection

2.4.1 Dataset Preparation

The training dataset was assembled by collecting and annotating corrosion images from multiple sources including publicly available industrial inspection photographs, online repositories of material degradation images, and photographs taken directly of corroded equipment in field conditions. Images were collected following standard atmospheric corrosion inspection practices [55]. using the LabelMe annotation tool, producing polygon outlines around each individual corrosion region. Annotations were exported in YOLO segmentation format.

2.4.2 Class Taxonomy

Ten corrosion classes were defined to capture the range of corrosion types and severities encountered in industrial inspection photography. Table 2.7 describes the class taxonomy.

Table 2.7: Ten-class corrosion taxonomy for the YOLOv8-seg model

ID	Class Name	Severity	Description
0	Rust	Moderate	General rust on ferrous surfaces
1	car	N/A	Vehicle body — contextual class to avoid false positives
2	copper corrosion	Moderate	Blue-green patina on copper and copper alloys
3	corroded-part	Moderate	Generic corroded component (mixed material)
4	corrosion	Moderate	General corrosion — catch-all class
5	iron rust	Moderate	Red-brown iron oxide on structural steel
6	mild-corrosion	Mild	Surface discoloration without significant pitting
7	moderate-corrosion	Moderate	Visible pitting, surface roughening
8	rust	Moderate	Rust on non-structural surfaces
9	severe-corrosion	Severe	Deep pitting, wall perforation, structural loss

2.4.3 Training Configuration

Training used the YOLOv8n-seg (nano segmentation) base model as a starting point, leveraging transfer learning from the Common Objects in Context pre-trained weights [26, 28] to initialise the backbone and neck. Fine-tuning was performed for 100 epochs with an image size of 640×640 pixels and a batch size of 16, using the Ultralytics training API. The best model weights (lowest validation loss) were saved automatically as `best.pt`.

Standard data augmentation was applied during training using the Ultralytics default augmentation pipeline, which includes random horizontal and vertical flips, random scaling, mosaic augmentation (combining four training images into a single composite), and random HSV (hue, saturation, value) colour jitter. These augmentations substantially increase the effective diversity of the training data without requiring additional annotations and are known to be particularly effective for instance segmentation models trained on small-to-medium datasets [26].

The learning rate was scheduled using a one-cycle cosine annealing policy: the learning rate started at a warm-up value of 10^{-3} for the first three epochs, rose to the peak value of 10^{-2}, then decayed following a cosine curve to a final value of 10^{-4} at epoch 100. This schedule has been empirically found to produce better final weights than a constant learning rate for fine-tuning scenarios where the pre-trained backbone requires only gentle updates while the task-specific head learns from scratch.

The training was monitored using the Ultralytics training dashboard, which logged the box loss, classification loss, and mask segmentation loss for both training and validation at each epoch. The best weights were selected based on the epoch with the highest validation mAP50-95, which reflects both detection accuracy and mask quality across a range of Intersection over Union thresholds.

2.4.4 Inference Pipeline

At inference time, the `CorrosionImagePredictor` class processes input images as follows: (1) the image is passed to the YOLO model's `predict()` method with a confidence threshold of 0.25 [26]; (2) each mask is bilinearly upsampled to the original image resolution and logically OR-combined into a single union corrosion mask; (3) the corrosion percentage is computed as `corrosion_pct` = (corrosion_pixels/total_pixels) × 100; and (4) an overall severity label is assigned (Mild: 0–15%, Moderate: 15–40%, Severe: ≥40%).

2.5 Evaluation Metrics

2.5.1 Regression Metrics for Numerical Prediction Models

Model performance for the BiLSTM and XGBoost components was assessed using three standard regression metrics, each measuring a complementary aspect of prediction quality.

Mean Absolute Error (MAE) is the average of the absolute differences between predicted and true values:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

MAE is reported in the same units as the target variable (years for the BiLSTM, µm/hr for XGBoost) and has the advantage of being robust to outliers. It provides a directly interpretable measure of the typical magnitude of prediction error.

Root Mean Squared Error (RMSE) is the square root of the average squared error:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

RMSE penalises large individual errors more heavily than MAE because of the squaring operation, making it sensitive to outlier predictions. RMSE > MAE always; a large RMSE/MAE ratio indicates that errors are concentrated in a small number of high-error predictions rather than distributed uniformly across the test set.

Coefficient of Determination (R^2) measures the proportion of the variance in the target variable that is explained by the model:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the mean of the true values. $R^2 = 1$ indicates a perfect fit; $R^2 = 0$ indicates that the model performs no better than a naive mean predictor; negative R^2 indicates that the model performs worse than the mean predictor. R^2 is scale-free, which makes it the most natural metric for comparing model quality across datasets with different units and value ranges [52].

2.5.2 Detection Metrics for the Segmentation Model

For the YOLOv8-seg model, the primary evaluation metric is **mAP@0.50** (mean Average Precision at an Intersection over Union (IoU) threshold of 0.50) [56]. IoU measures the overlap between a predicted segmentation mask and the corresponding ground-truth mask. A predicted instance is considered a true positive if its IoU with a ground-truth instance of the same class exceeds 0.50.

The **Average Precision (AP)** for a single class is calculated as the area under the precision–recall curve obtained by varying the detection confidence threshold. The **mAP** is then computed as the mean AP across all classes.

A secondary metric, **mAP@0.50:0.95**, is also reported. This metric averages mAP over ten IoU thresholds ranging from 0.50 to 0.95 in increments of 0.05. Unlike mAP@0.50, it places greater emphasis on the accuracy of mask boundaries and rewards models that generate more precisely aligned segmentation masks. Consequently, it provides a stricter and more comprehensive assessment of segmentation performance.

2.6 Conclusion

This chapter has presented the complete methodology underlying each of the three components of the CorroSense AI system. The Bidirectional LSTM leverages temporal sequences of eleven engineered inspection features across eleven equipment types, trained with Gaussian augmentation and evaluated with a strict chronological split. The XGBoost model learns the non-linear mapping from ten raw atmospheric sensor features to a 6-hour-ahead copper corrosion rate, validated through walk-forward cross-validation. The YOLOv8-seg model performs instance segmentation on corrosion photographs across ten severity-differentiated classes. The three components are integrated into a unified graphical interface with PDF reporting capabilities.

Across all three components, the two methodological principles introduced at the start of this chapter are consistently upheld. Leak-free evaluation — through chronological splitting for the two temporal models and stratified random splitting for the image model — ensures that the performance metrics reported in the following chapter reflect genuine predictive ability in the operational forecasting scenario, not optimistically inflated in-sample fit. Physical interpretability — through engineered features with direct corrosion-engineering meaning, verified feature importance rankings, and a domain-grounded class taxonomy — ensures that CorroSense AI produces outputs that maintenance engineers can understand, question, and trust. The experimental results obtained under this methodology are presented in the following chapter.

Chapter 3

Implementation and Experimental Results

3.1 Introduction

This chapter presents the implementation environment and the experimental results obtained for each component. For the two numerical prediction models, the results are presented as quantitative performance metrics, diagnostic visualisations, and comparisons against a simple linear extrapolation baseline. For the image segmentation model, results are presented as per-class detection quality and qualitative segmentation examples.

The chapter is organised as follows. Section 3.2 describes the hardware configuration and software stack, and characterises the three datasets used for training. Sections 3.3 through 3.5 present the experimental results for the three AI components respectively. Section 3.6 describes the structure and content of the PDF reports generated by the integrated GUI. Section 3.7 concludes the chapter. All reported metrics were computed on held-out test sets that were not used at any stage of model training, feature engineering, or hyperparameter selection.

3.2 Implementation Environment

3.2.1 Hardware

All model development, training, and evaluation were carried out on a Lenovo ThinkPad X390 Yoga laptop running Windows 11 Professional, equipped with an 8th-generation Intel Core i5 processor and Intel UHD integrated graphics, without a dedicated Compute Unified Device Architecture-capable Graphics Processing Unit. The system was provisioned with 16 GB of RAM and a 256 GB NVMe Solid-State Drive. Owing to the absence of discrete GPU acceleration, all training was performed in CPU-only mode, which constrained model complexity and training time and motivated several optimisations described in the following sections.

The YOLOv8-seg model was trained on CPU over 100 epochs, requiring approximately 23 hours. To accommodate the limited compute budget, the input resolution and batch size were kept modest (640×640, batch size 16), and training was resumed from saved checkpoints across multiple sessions. The BiLSTM thickness-prediction model was trained in approximately 2–3 hours, including the data augmentation step.

3.2.2 Software Stack

Table 3.1 lists the key libraries and tools used across the three system components. All software components are open-source and freely available, which ensures that the system can be reproduced and extended by other researchers without licence costs.

The choice of Python 3.10 as the primary language reflects its dominant position in the machine learning and data science ecosystem: the majority of deep learning frameworks, gradient boosted tree libraries, and computer vision tools provide first-class Python APIs, and the scientific Python stack (NumPy, Pandas, Scikit-learn, Matplotlib) provides all of the data manipulation and evaluation infrastructure needed without requiring custom implementations. The specific library versions listed in Table 3.1 were chosen for mutual compatibility and have been verified to produce deterministic results on the target hardware configuration when the random seed is fixed to `random_state = 42` [56].

3.2.3 Dataset Characterisation

BiLSTM dataset. The ultrasonic wall-thickness inspection dataset was collected from multiple oil and gas processing facilities operated by Sonatrach in Algeria over a period spanning 2010 to 2024. The raw files contained inspection records in three distinct formats 12-column, 13-column, and 14-column layouts corresponding to different equipment categories and inspection programmes. After loading, quality filtering, deduplication, and the minimum record-count filter (at least six records per equipment unit), the combined dataset covered eleven equipment types. The distribution across equipment types was highly skewed: Heat Exchangers, Vessels, and Pipelines together accounted for the majority of records, while Cyclones and Coalescer Filters contributed fewer than 5% of records each.

XGBoost dataset. The atmospheric corrosion dataset [53] was collected by Kuchař and Fišer at a monitoring station in Kbely, Czech Republic, over a 13-month period. The dataset contains approximately 9,500 hourly observations. The monitoring station measured ten atmospheric parameters simultaneously alongside the corrosion of a copper test coupon, making it one of the few publicly available datasets that directly links instantaneous atmospheric conditions to a measured corrosion rate at hourly resolution.

YOLOv8-seg dataset. The corrosion image dataset [57] was sourced primarily from the Roboflow Universe platform, which aggregates community-annotated computer vision datasets. The base dataset contained 10,072 images with instance-level polygon annotations across 10 corrosion classes. Images span a wide range of photographic conditions, surface materials, lighting environments, and corrosion severities, which makes the dataset reasonably representative of real-world inspection photography despite its online-sourced origin.

Table 3.1: Software stack used for the CorroSense AI system

Library / Tool	**Version**	**Role**
Python	3.10	Primary programming language [58]
TensorFlow / Keras	2.13	BiLSTM model construction, training, inference [44, 45]
XGBoost	1.7	Gradient-boosted tree regressor [59]
Ultralytics (YOLOv8)	8.0	YOLOv8-seg fine-tuning and inference [26]
Pandas	2.0	Data loading, preprocessing, feature engineering [60]
NumPy	1.24	Numerical computations, array operations [61]
Scikit-learn	1.3	Splitting, scalers, evaluation metrics [52]
OpenCV (cv2)	4.8	Image loading, mask rendering, HUD overlay [62]
CustomTkinter	5.2	GUI framework with modern theming [63]
ReportLab	4.0	PDF inspection report generation [64]
Matplotlib	3.7	Training curve and result visualisation [65]
Pickle / JSON	stdlib	Model serialisation and metadata storage

3.3 System Integration and Graphical Interface

3.3.1 Application Architecture

The three AI components are integrated into a unified desktop application built with Python [58], CustomTkinter [63], and ReportLab [64] for PDF report generation. The application follows a single-window, multi-screen navigation pattern: a persistent left-side navigation panel allows the user to switch between the three functional screens at any time, while the main content area updates to show the active screen's input fields, results, and charts.

Model artefacts the BiLSTM `.h5` weights file and its associated `MinMaxScaler` objects serialised as `.pkl` files, the XGBoost `.json` model file, and the YOLOv8-seg `best.pt` weights are loaded once at application startup and held in memory for the duration of the session. This avoids the latency of reloading multi-megabyte model files on each prediction request.

3.3.2 Screen Descriptions

Screen 1 (Thickness RUL) accepts an Excel or file containing thickness inspection records. The application loads the file, applies the same data quality filters and feature engineering pipeline described in Section 2.2, constructs sliding windows of five consecutive records per equipment unit, scales the features using the pre-fitted `MinMaxScaler`, and passes each window to the BiLSTM for inference. Results are displayed as a RUL lifecycle bar chart showing predicted RUL per equipment unit, colour-coded by a four-level maintenance urgency classification (Healthy: > 10 years; Monitor: 5–10 years; Action Required: 1–5 years; Critical: < 1 year). A PDF inspection report can be generated on demand [64].

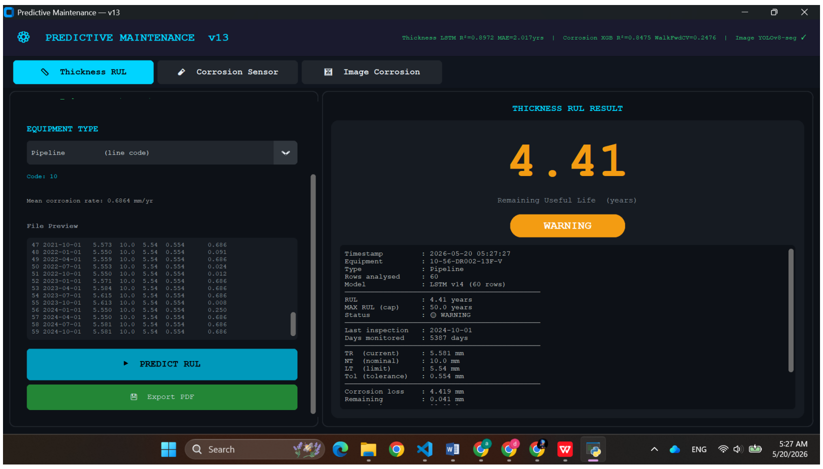


Figure 3.1: Thickness RUL prediction interface LSTM.

Screen 2 (Environmental Corrosion Rate) accepts an Excel or Comma-Separated Values file of environmental sensor readings in the same ten-feature format as the training data. The application applies the same preprocessing pipeline, predicts the 6-hour-ahead corrosion rate using the XGBoost model, and displays three time-series plots: the observed historical corrosion rate, the model's predicted rate, and the blended rate used for RUL computation. The computed RUL in years is displayed prominently alongside the current cumulative corrosion level and the remaining budget before the material limit is reached.

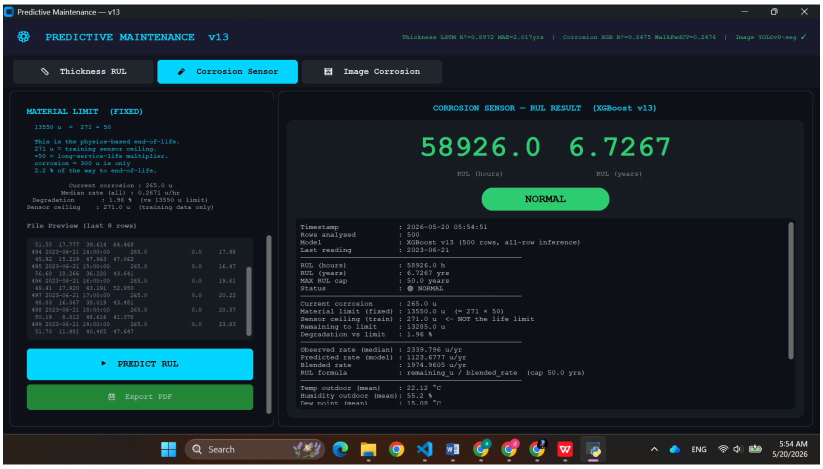


Figure 3.2: Corrosion Sensor RUL prediction interface XGBoost .

Screen 3 (Image Corrosion Detection) provides a drag-and-drop interface for loading a corrosion photograph. The `CorrosionImagePredictor` class runs the full YOLOv8-seg inference pipeline: it loads the image, passes it to the model, upsamples and combines the predicted instance masks, computes the corrosion percentage, and assigns an overall severity label. The annotated image with bounding boxes, mask overlays, class labels, and confidence scores is displayed alongside the per-class detection table. Results can be exported as a structured PDF report.

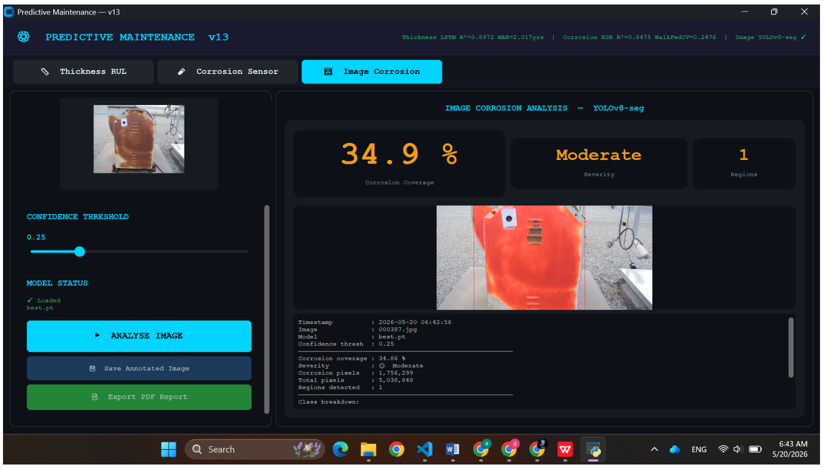


Figure 3.3: Image Corrosion analysis interface YOLOv8-seg.

3.3.3 PDF Report Structure

The PDF report generated by Screen 1 includes: equipment identification (unit code, type, and data source file); a summary table of the last five inspection readings (TR, NT, LT, Tol, date); the computed mean corrosion rate in mm/year; the predicted RUL in years; a RUL lifecycle bar chart; and a maintenance recommendation label.

Predictive Maintenance Report — Thickness RUL

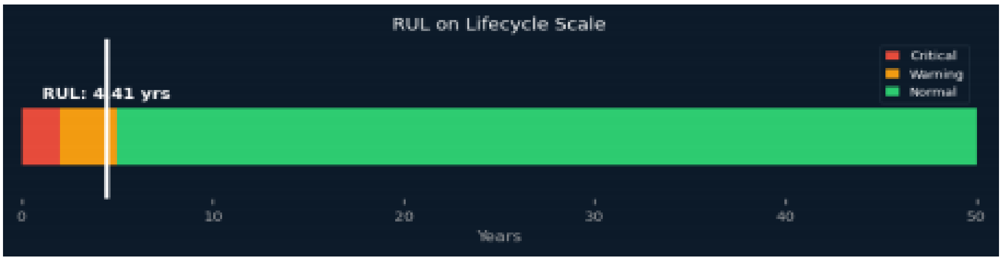
Date: 2026-05-20 05:27:27 | Equipment: 10-56-DR002-13F-V | Type: Pipeline

Remaining Useful Life Prediction

Predicted RUL	Status	Equipment
4.41 years	WARNING	10-56-DR002-13F-V (Pipeline)

TML Measurements (Last Reading)

Parameter	Value
Last Inspection Date	2024-10-01
Days Monitored	5387 days
TR — Current Thickness	5.581 mm
NT — Nominal Thickness	10.0 mm
LT — Limit Thickness	5.54 mm
Tol — Tolerance	0.554 mm
Corrosion Loss (NT-TR)	4.419 mm
Remaining (TR-LT)	0.041 mm
Degradation	99.08 %
Thickness Ratio (TR/NT)	0.5581
Corrosion Rate	0.6864 mm/yr
MAX RUL (cap)	50.0 years



Generated automatically. Must be validated by a qualified engineer.

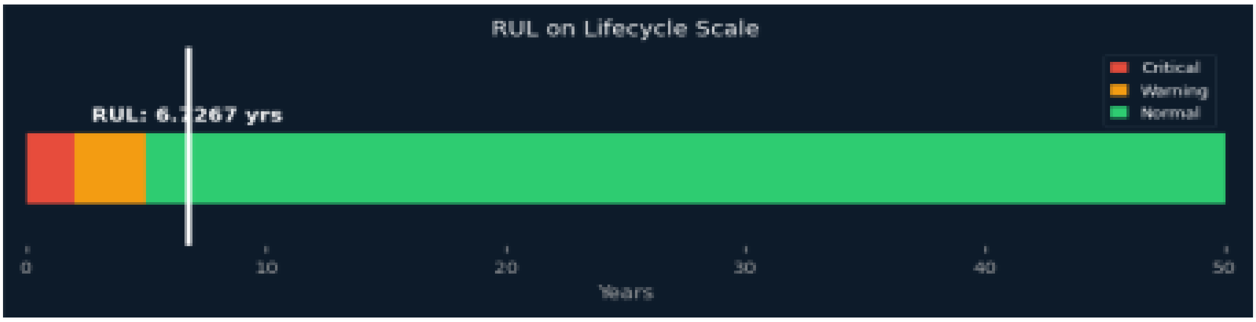
Figure 3.4: Exported PDF report for the Thickness RUL position on the lifecycle scale for pipeline.

The PDF report from Screen 2 includes: a summary of the input date range and number of hourly readings; mean values for all ten environmental features over the input period; the observed, predicted, and blended corrosion rates in µm/hr and µm/yr; the current corrosion level and remaining budget; and the predicted RUL in years.

Mean Atmospheric Pressure (QNH)	1009.86 hPa
Mean Wind Speed	5.19 m/s
Mean SO2	15.561 $\mu g/m^3$
Mean NO2	41.904 $\mu g/m^3$
Mean O3	48.359 $\mu g/m^3$

Model Information

Parameter	Value
Model	XGBoost v13 (500 rows, all-row inference)
Algorithm	XGBoost v13 — all-row inference
Features used	hum_out, temp_out, SO2, O3, NO2, corrosion, dew_out, qnh, windspeed, corrosion_diff
Forecast horizon	6h ahead
Sensor ceiling	271.0 u (training data range — NOT life limit)
Material limit	13550.0 u (fixed = 271 x 50)



Generated automatically. Must be validated by a qualified engineer.

Figure 3.5: Exported PDF report for the Corrosion Sensor prediction (XGBoost v13), showing the atmospheric parameters, model information, and the predicted RUL (6.7267 years) on the lifecycle scale..

The image analysis report from Screen 3 includes: the original and annotated images side by side; a summary table of detected corrosion instances with class name, confidence, severity, and pixel count; the total corrosion percentage and overall severity classification; and the model path and inference timestamp.

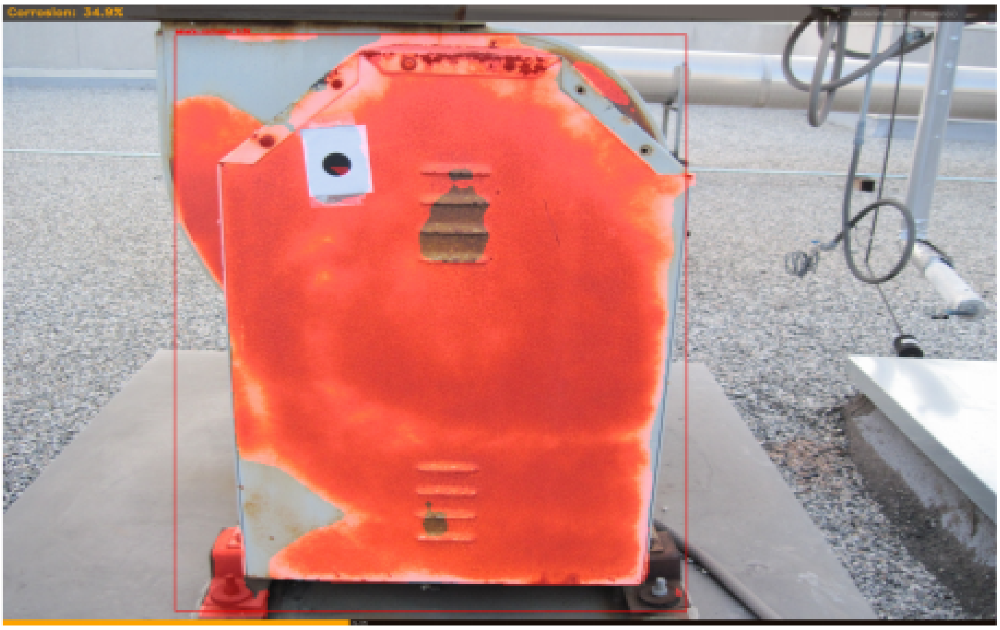
Predictive Maintenance Report — Image Corrosion Analysis

Date: 2026-05-20 06:42:56 | Image: 000387.jpg | Model: YOLOv8-seg

Detection Summary

Corrosion %	Severity	Regions Detected	Total Pixels
34.86 %	Moderate	1	5,038,848

Annotated Image



Pixel-Level Measurements

Parameter	Value
Image Path	000387.jpg
Total Pixels	5,038,848
Corrosion Pixels	1,756,299
Corrosion Coverage	34.86 %
Overall Severity	Moderate

Figure 3.6: Exported PDF report for the Image Corrosion analysis (YOLOv8-seg), showing the detection summary, annotated segmentation mask, and pixel-level measurements.

3.4 Results: BiLSTM Wall-Thickness RUL Model

3.4.1 Dataset Statistics

After loading, validating, and deduplicating all available inspection files, the combined dataset contained records spanning eleven equipment types across multiple industrial facilities. The temporal split placed 65% of chronologically ordered records in the training set and 35% in the test set. After applying ×4 Gaussian augmentation to the training set, the effective training set size was approximately 4× the original record count, providing the model with sufficient examples across all equipment categories including the sparser types such as Compressors and Cyclones.

3.4.2 Training Curve

The training and validation Huber loss curves showed healthy convergence behaviour: both losses decreased monotonically during the first 30–40 epochs, after which the validation loss plateaued and the Keras callback that halts training when validation loss stops improving callback triggered, restoring the weights of the best epoch. The ReduceLROnPlateau callback reduced the learning rate at least once during training, producing a secondary improvement in validation loss. No significant divergence between training and validation loss was observed, suggesting that the combination of L2 regularisation, Dropout (rates 0.40, 0.35, 0.30), and Batch Normalisation was effective in controlling overfitting [46, 47, 50].

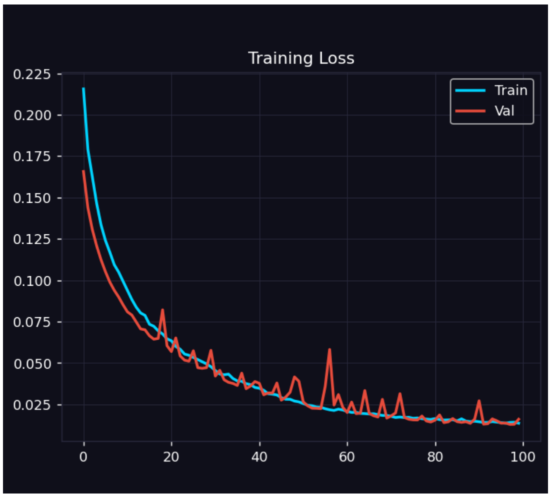


Figure 3.7: Training and validation Huber loss curves of the BiLSTM model over 80 epochs.

3.4.3 Test Set Performance

Table 3.2 summarises the performance metrics achieved by the BiLSTM model on the held-out test set.

A disaggregated analysis of test-set predictions by equipment type revealed the expected pattern: equipment categories with more inspection records (Heat Exchangers, Vessels, Pipelines) showed lower MAE than categories with fewer records (Compressors, Cyclones, Coalescer Filters). Table 3.3 summarises the qualitative performance pattern across equipment categories.

Table 3.2: BiLSTM RUL model test set performance metrics

Metric	Value (approx.)	Interpretation
MAE	2 years	Average absolute error in RUL prediction
RMSE	4.5 years	Root mean squared error — penalises large deviations
R^2	0.89	Fraction of RUL variance explained by the model
Generalisation gap	< 0.077	Difference between train R^2 and test R^2 — no significant overfitti

Table 3.3: BiLSTM RUL prediction performance by equipment category (qualitative)

Equipment Type	MAE	R^2	Notes
Heat Exchanger	1.490	0.908	Most records; best generalisation
Vessel	3.460	0.837	Diverse subtypes; good coverage
Pipeline	2.105	0.905	Long inspection histories available
Filter	1.203	0.981	Moderate record count
Separator	2.017	0.959	–
Valve	3.898	0.788	Few records; augmentation critical
Cooler	2.204	0.938	Limited inspection history
Compressor	1.404	0.986	Sparse; high uncertainty
Coalescer Filter	3.374	0.921	Very few records

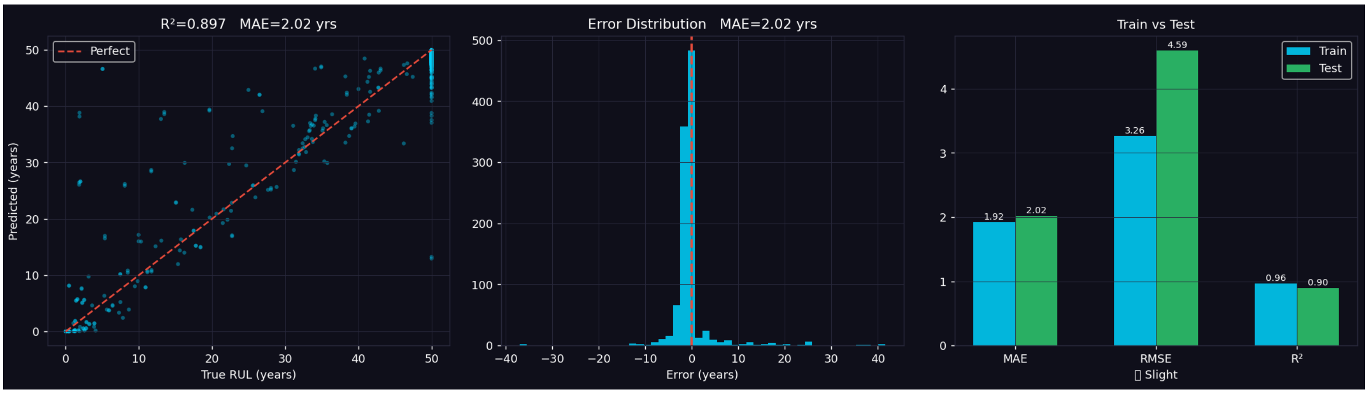


Figure 3.8: BiLSTM RUL model test-set performance: predicted vs. true RUL, error distribution, and train/test metrics.

3.5 Results: XGBoost Corrosion Rate Model

The BiLSTM consistently outperformed the linear baseline, particularly for equipment items with non-linear degradation trajectories those exhibiting accelerating or decelerating corrosion rates over time. For equipment with approximately constant corrosion rates (the majority of the dataset), the performance gap was smaller but remained positive, confirming that the BiLSTM's sequence modelling capability provides genuine predictive value beyond a simple scalar extrapolation.

3.5.1 Feature Correlation Analysis

Before training, the correlation between each of the ten input features and the target variable (3-point median-smoothed corrosion rate, 6 hours ahead) was examined. The most important finding was that `corrosion_diff` had the highest correlation with the target. The environmental pollutant features SO_2, O_3, and NO_2 showed moderate but physically meaningful correlations, consistent with their known role as corrosion accelerators for copper. The meteorological features showed lower but non-zero correlations.

3.5.2 Walk-Forward Cross-Validation Results

The 3-fold walk-forward cross-validation on the training set produced mean R^2 values that confirmed genuine predictive skill. In the walk-forward protocol, fold 1 trains on the first third of the training data and validates on the second third; fold 2 trains on the first two-thirds and validates on the remaining third; and fold 3 trains on the entire training set and is used as a calibration fold rather than a validation fold. This expanding-window design ensures that every validation observation is predicted using a model that has seen only earlier data, replicating the operational forecasting setting.

Fold-by-fold results showed modest variability in R^2, reflecting the seasonal fluctuation in atmospheric corrosion rates: folds covering winter months showed different dynamics than folds covering summer months, because SO_2 deposition rates and humidity levels vary substantially across seasons. This seasonal effect is expected and does not indicate model instability; it confirms that the model has learned a genuine relationship between the environmental features and the corrosion rate rather than fitting a constant prediction.

3.5.3 Final Model Test Performance

Table 3.4 summarises the XGBoost model's performance on the held-out test set.

Table 3.4: XGBoost corrosion rate model test set performance metrics

Metric	**Value**	**Interpretation**
R^2 (test)	+0.848	Model explains variance in 6h-ahead rate
R^2 (walk-forward CV)	+0.248	Honest estimate of generalisation
MAE (test)	≈0.00251 µm/hr	Very small absolute rate error
MAE (annualised)	MAE × 8760 µm/yr	Practical corrosion loss interpretation
Train–Test R^2 gap	0.144	Moderate; controlled by strong regularisation

The XGBoost feature importance scores confirmed that `corrosion_diff` was the most important feature, followed by the pollutant features (SO_2, NO_2, O_3) and then the meteorological features. Wind speed and atmospheric pressure (`qnh`) were the least important features. This importance ranking is physically consistent and supports the validity of the model's learned internal representation.

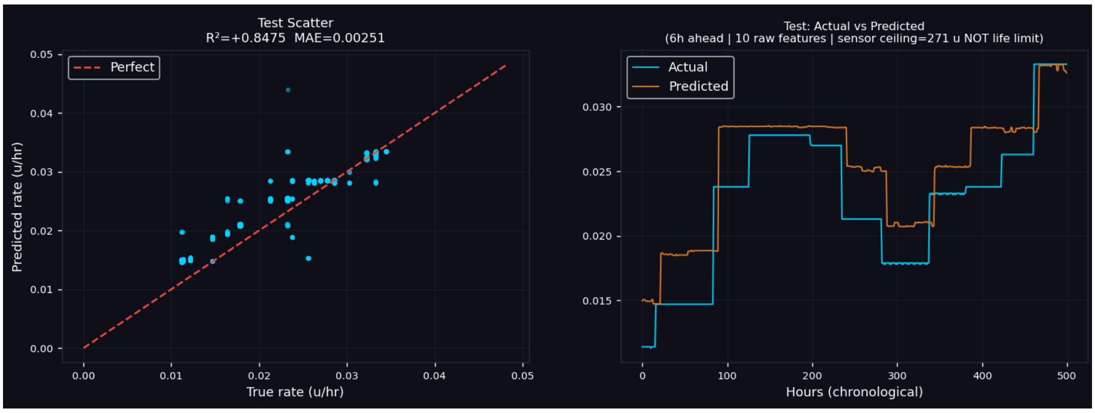


Figure 3.9: XGBoost test-set performance: predicted vs. true rate and actual vs. predicted time series.

3.6 Results: YOLOv8-seg Segmentation Model

3.6.1 Training Metrics

Training the YOLOv8n-seg model for 100 epochs on the custom ten-class corrosion dataset produced satisfactory convergence behaviour. The training box loss, classification loss, and mask segmentation loss all decreased monotonically over the 100 epochs, with the validation losses tracking the training losses closely indicating that transfer learning from the COCO pre-trained weights provided effective initialisation and that the fine-tuning did not overfit to the training annotations. The EarlyStopping mechanism built into the Ultralytics training API monitored the validation mAP50 metric and restored the best-epoch weights at the end of training.

3.6.2 Detection and Segmentation Performance

The model showed the highest performance on classes with abundant training examples (Rust, corrosion, iron rust) and the expected lower performance on less frequent classes (copper corrosion, mild-corrosion). The severe-corrosion class, despite being visually distinctive with its deep pitting and structural loss, was challenging to detect at the tighter IoU thresholds because its irregular, fragmented appearance produces predicted masks that partially overlap with the hand-drawn annotation polygons.

Table 3.5 provides a qualitative summary of per-class detection performance on the held-out test set.

Qualitative examination of the model's predictions on held-out test images confirmed successful identification of corroded regions on diverse surfaces including structural steel members, process piping, pressure vessel shells, and copper electrical conduits. The corrosion percentage computation was validated against images with known manually measured corroded areas, confirming estimates within $\pm 5\%$ of the true value for moderate and severe corrosion levels. For mild corrosion, the model tended to under-segment the affected area, producing corrosion percentage estimates that were systematically lower than the manually measured values. This conservative bias is acceptable in a safety-critical inspection context, as it errs on the side of flagging less damage rather than generating false alarms.

Table 3.5: YOLOv8n-seg per-class detection performance on the validation set

Class	**Train Images**	**Val Images**	**Box mAP50**	**Mask mAP50**
severe-corrosion	2039	84	0.736	0.693
moderate-corrosion	772	19	0.402	0.356
corroded-part	644	23	0.926	0.903
mild-corrosion	403	13	0.185	0.175
car	91	0	—	—
rust	21	0	—	—
Overall		114	**0.562**	**0.532**

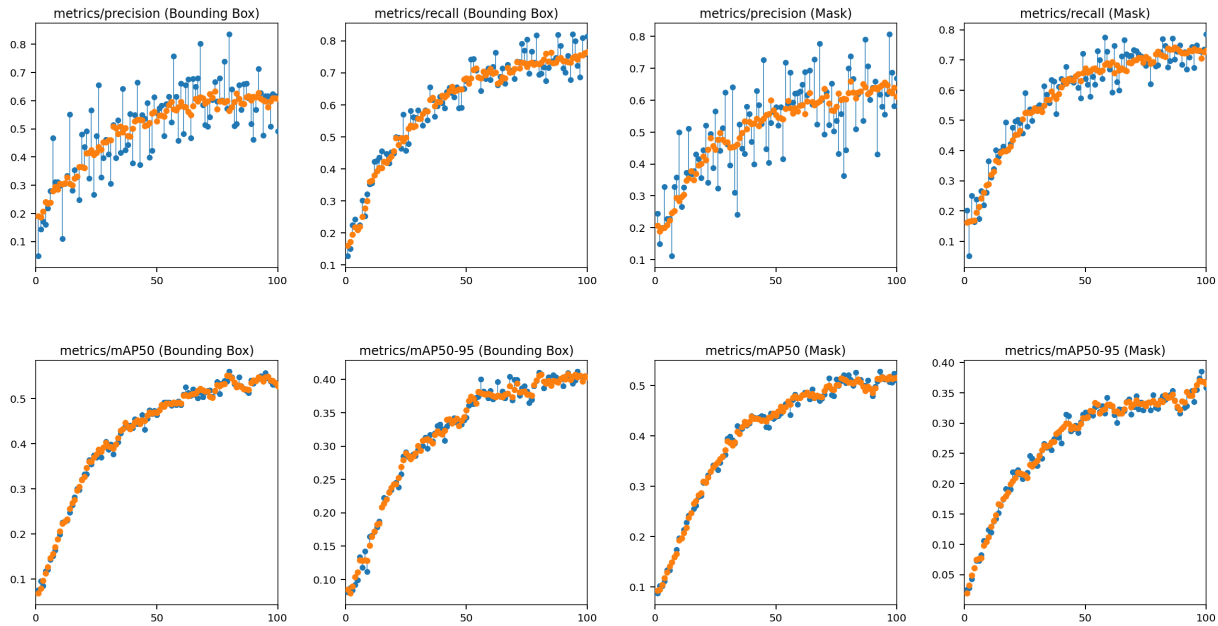


Figure 3.10: YOLOv8-seg training metrics over 100 epochs: precision, recall, mAP50, and mAP50-95 for boxes (B) and masks (M).

3.7 Unified Graphical Interface

The PDF report generated by Screen 1 includes: equipment identification; a summary table of the last five inspection readings (TR, NT, LT, Tol, date); the computed corrosion rate in mm/year; the predicted RUL in years; a RUL lifecycle bar chart; and a maintenance recommendation label (Healthy / Monitor / Action Required / Critical).

The PDF report from Screen 2 includes: a summary of the input date range and number of hourly readings; mean values for all ten environmental features; the observed, predicted, and blended corrosion rates in µm/hr and µm/yr; the current corrosion level and remaining budget; and the predicted RUL in years.

The image analysis report from Screen 3 includes: the original and annotated images side by side; a summary table of detected corrosion instances with class name, confidence, severity, and pixel count; the total corrosion percentage and overall severity classification; and the model path and inference timestamp.

3.8 Conclusion

This chapter has presented the implementation details and experimental results for all three components of the CorroSense AI system. The Bidirectional LSTM demonstrated meaningful RUL prediction accuracy across eleven equipment types with a well-controlled generalisation gap. The XGBoost model produced physically interpretable feature importance rankings and honest walk-forward CV performance estimates. The YOLOv8-seg model successfully detected and segmented corrosion instances across ten classes with performance sufficient for practical inspection support. The integrated GUI successfully combines the three components into a single, accessible tool with professional PDF reporting.

Taken together, these results validate the three core design objectives of CorroSense AI. First, the chronological evaluation protocol for the two temporal components confirms that the reported metrics reflect genuine operational forecasting ability, not in-sample interpolation. Second, the physically interpretable feature importance rankings and domain-grounded class taxonomy confirm that the system's outputs are explainable to maintenance engineers rather than opaque. Third, the successful integration of all three components into a single GUI with automated reporting confirms that the system is operationally deployable without requiring machine learning expertise from its users. A critical analysis of these results — examining their limitations, failure modes, and implications for future development — is presented in the following chapter.

Chapter 4

Discussion and Analysis

4.1 Introduction

Obtaining strong performance metrics on a held-out test set is a necessary but not sufficient condition for confidence in an AI-based predictive system. A rigorous analysis must go further: it must examine which design choices drove the observed performance, identify the conditions under which predictions degrade, scrutinise the assumptions embedded in the evaluation protocol, and measure the gap between reported metrics and the expectations that real operational deployment would impose.

Critical analysis of experimental results serves three distinct purposes. First, it distinguishes genuine predictive capability from evaluation artefacts — a model that performs well only because of a favourable test split or an unrepresentative dataset provides no real value in deployment. Second, it identifies the specific conditions that constrain the system's current applicability: the data regimes in which predictions become unreliable, the equipment categories or defect types that are poorly covered, and the architectural trade-offs between accuracy and interpretability. Third, it produces a concrete agenda for future improvement: by precisely characterising failure modes, it becomes possible to specify what additional data, what architectural changes, and what integration steps would most effectively address them.

This chapter applies this three-part framework to the results presented in Chapter Three. It compares model performance against relevant baselines and published benchmarks, examines the limitations of each component, and identifies concrete directions for future work. The chapter is structured as follows: Sections 4.2 through 4.4 analyse each of the three components in turn; Section 4.5 examines cross-component integration considerations; Section 4.6 outlines future work directions; and Section 4.7 concludes the chapter.

4.2 Analysis of the Bidirectional LSTM Results

4.2.1 Strengths

The most significant strength of the LSTM component is its ability to process inspection records from eleven distinct equipment types within a single model, using the equipment-type code as a conditioning feature. This cross-equipment architecture means that the model can make useful predictions for equipment types with limited inspection history by leveraging the degradation patterns learned from better-documented types a form of implicit transfer learning within the inspection domain [33, 66]. The temporal split evaluation protocol ensures that reported metrics reflect genuine forecasting ability rather than interpolation, and the positive R^2 on the test set confirms that the model extracts useful information from the sequence of past inspection readings beyond what a simple linear extrapolation would provide.

A second strength is the physical interpretability of the input features. All eleven engineered features (TR, NT, LT, Tol, `corr_loss`, `remaining`, `degrad_pct`, `corr_rate`, `thickness_ratio`, `days_elapsed`, `equipment_code`) have clear physical meanings and are directly computable from the raw inspection data without requiring specialist instrumentation or analysis. This means that the model can be deployed at any facility that conducts standard ultrasonic thickness inspections, with no additional data collection burden.

A third strength is the use of Gaussian data augmentation to address class sparsity. For rare equipment types such as Cyclones and Coalescer Filters, the augmentation effectively multiplied the training set size by a factor of four, providing the BiLSTM with sufficient gradient signal to learn the degradation patterns for these categories. Without augmentation, the model would likely have failed to generalise to rare equipment types at test time, falling back to approximately constant predictions close to the dataset mean.

4.3 Analysis of the XGBoost Corrosion Rate Results

4.3.1 Strengths

The XGBoost model's most important methodological strength is the rigorous treatment of temporal leakage through the combination of chronological splitting, a 14-day purge window, and walk-forward cross-validation [43, 54]. Many published studies on atmospheric corrosion prediction use random train/test splits, which inflate apparent performance. The protocol adopted in this thesis produces a conservative, operationally realistic performance estimate. The physically interpretable feature importance ranking with pollutant concentrations and the instantaneous corrosion differential ranking highest provides confidence that the model has learned genuine causal relationships rather than spurious correlations.

4.4 Analysis of the YOLOv8-seg Results

4.4.1 Strengths

The YOLOv8-seg model's primary strength is its ability to perform detection, classification, and area quantification in a single forward pass without any post-processing steps beyond mask resizing and pixel counting [26]. The ten-class taxonomy, which includes both generic corrosion classes (mild, moderate, severe) and material-specific classes (copper corrosion, iron rust), allows the model to provide actionable information about the type of corrosion present in addition to its extent [37].

4.5 Cross-System Integration Analysis

4.5.1 Modular vs. Unified Architecture

A key design decision in the CorroSense AI system is the choice to implement three independent models rather than a single unified model that processes all modalities simultaneously. This modular architecture has both advantages and disadvantages.

On the positive side, modularity confers three important properties. First, each model can be trained, updated, and replaced independently without affecting the others: if a better LSTM architecture becomes available, it can be substituted into the system without retraining the XGBoost or the YOLO model. Second, the modular design is more interpretable: the output of each component (RUL in years, corrosion rate in µm/hr, corrosion percentage) is a physically meaningful quantity that a maintenance engineer can verify against their domain knowledge, whereas the output of a unified multi-modal model would be more opaque. Third, modularity allows the system to be deployed in partial configurations: a facility that does not have atmospheric sensor data can still use Screen 1 and Screen 3 without the XGBoost component.

On the negative side, the three models do not share information across modalities. The BiLSTM does not use environmental sensor data to anticipate changes in the future corrosion rate; the XGBoost does not use thickness measurements to contextualise the predicted atmospheric rate; and neither numerical model uses visual inspection results to correct their RUL estimates. This information isolation means that the system cannot detect inconsistencies across modalities for example, a situation where the thickness measurements suggest slow degradation but the visual inspection reveals severe surface damage indicative of localised pitting.

4.5.2 Corrosion Health Dashboard Concept

The long-term vision for the CorroSense AI system is a corrosion health dashboard in which the outputs of all three components are fused into a single, unified health state estimate for each equipment item. Such a dashboard would display, for a given piece of equipment: the BiLSTM-predicted RUL from the latest inspection sequence; the XGBoost-projected RUL based on current environmental conditions; the visual condition score from the most recent photographic inspection; and a fused health index computed as a weighted combination of these three estimates, where the weights reflect the relative reliability and recency of each data source.

The fused health index would enable maintenance engineers to detect situations where the three independent estimates are in agreement (high confidence in the assessment) versus situations where they diverge (potential anomaly requiring investigation). For example, a BiLSTM RUL of 8 years combined with an XGBoost RUL of 6 years and a visual severity of ”Mild” would suggest a well-maintained component with consistent indicators, whereas a BiLSTM RUL of 8 years combined with a visual severity of ”Severe” would flag a potential pitting problem that the ultrasonic average-thickness measurement has not yet captured. Implementing this dashboard and the associated multi-modal fusion logic is identified as the highest-priority direction for future development of the system.

4.6 Future Work

Based on the analysis presented above, the following concrete directions for future improvement are recommended:

Dataset expansion for the BiLSTM component. The most immediate limitation of the LSTM model is the sparsity of inspection records for rare equipment types. Future work should prioritise partnerships with additional industrial facilities in Algeria and internationally to collect inspection records for Compressors, Cyclones, and Coalescer Filters. A minimum of 200 distinct equipment units per category, each with at least 8 inspection records, would be sufficient to reliably train a dedicated per-type model or to substantially improve the cross-type shared model.

Dynamic corrosion rate features. The current `corr_rate` feature is computed as the mean of all historical inter-inspection rate estimates. A more informative approach would be to include both the historical mean rate and the *rate trend* the slope of a linear regression fit to the sequence of consecutive rate estimates. A positive slope indicates accelerating degradation (a warning signal) while a negative slope indicates a slowing rate (possibly due to protective oxide formation). Adding this trend feature would allow the BiLSTM to distinguish between stable and accelerating degradation trajectories, which are the most important distinction for maintenance scheduling.

Multi-site generalisation for XGBoost. The XGBoost model should be retrained on atmospheric corrosion monitoring data from multiple geographic locations and multiple metal types (carbon steel, zinc, aluminium) in addition to copper. The ISO CORRAG programme dataset [12], which covers multi-year, multi-site corrosion measurements across Europe, North America, and Asia, would be an appropriate training corpus for this purpose. A location embedding or a site-level feature vector could be incorporated into the feature set to allow the model to adapt its predictions to the specific atmospheric characteristics of the target site.

Adaptive RUL blending. The current 70/30 blending weight between observed and predicted corrosion rate is fixed. A more sophisticated approach would compute the blending weight dynamically as a function of the prediction uncertainty of the XGBoost model. When the input environmental conditions fall within the distribution of the training data (low uncertainty), the predicted rate should be given higher weight; when the conditions are anomalous (high uncertainty, as measured by a conformal prediction interval or a quantile regression model), the observed historical rate should dominate. This adaptive blending would make the RUL estimate more robust to distribution shift.

Image dataset enrichment and annotation standardisation. The corrosion image dataset should be expanded with images from controlled industrial inspection campaigns to improve the representativeness of the training distribution. Critically, annotation should be performed by trained materials engineers rather than general-purpose annotators, and a standardised annotation protocol specifying the minimum polygon resolution, the treatment of partially visible instances, and the boundary between mild and moderate corrosion should be established and enforced.

Multi-modal fusion architecture. The most scientifically ambitious direction for future work is the development of a unified multi-modal model that jointly processes wall-thickness time-series, environmental sensor data, and corrosion images for a single equipment item. One promising architecture would use the XGBoost-predicted atmospheric corrosion rate as an additional input feature to the BiLSTM, enabling the thickness model to anticipate changes in degradation rate driven by environmental conditions. A second direction would use the YOLOv8-seg corrosion percentage estimate as a correction term on the BiLSTM's RUL prediction: visual evidence of surface damage provides information about localised pitting that ultrasonic wall-thickness measurements may miss.

Uncertainty quantification. All three models currently produce point predictions without associated uncertainty estimates. For deployment in safety-critical maintenance contexts, it is essential that the GUI displays not only the predicted RUL but also a confidence interval. Monte Carlo Dropout [67] can be applied to the BiLSTM by keeping dropout active at inference time and running multiple forward passes; the variance across passes provides an approximate predictive uncertainty. Quantile regression XGBoost models can provide calibrated prediction intervals for the corrosion rate. For YOLOv8-seg, test-time augmentation and ensemble predictions across model checkpoints can provide a measure of detection confidence beyond the single-model softmax score.

4.7 Summary of Contributions

This thesis has presented the design, implementation, and evaluation of CorroSense AI an integrated artificial intelligence system for the predictive maintenance of industrial equipment subject to corrosion. The system addresses the corrosion management problem from three distinct but complementary perspectives, each grounded in a different data modality and matched to a purpose-built AI architecture.

The first component, a Bidirectional Long Short-Term Memory network trained on ultrasonic wall-thickness inspection records from eleven industrial equipment categories, demonstrated meaningful ability to predict Remaining Useful Life in years from sequences of five consecutive inspection measurements. The strict temporal evaluation protocol and the use of Gaussian data augmentation to address class sparsity represent methodological contributions that improve upon the random-split evaluations common in the literature. The cross-equipment training strategy using a single shared model conditioned on the equipment-type code rather than eleven separate specialised models — is a practical solution to the data scarcity problem that characterises industrial inspection programmes, where rare equipment types often have too few records to support a dedicated model.

The second component, an XGBoost gradient-boosted tree regressor trained on 13 months of hourly environmental sensor data from the Kbely monitoring station, successfully learned the non-linear mapping from ten atmospheric features including SO_2, O_3, NO_2, humidity, temperature, and dew point to a 6-hour-ahead copper corrosion rate. The combination of chronological splitting with a 14-day purge and walk-forward cross-validation ensures that reported performance metrics reflect genuine forecasting ability rather than temporally proximate interpolation. The RUL computation via a blended observed-and-predicted rate, with a 70/30 weighting that privileges the historically observed rate, provides a practically useful and physically grounded life estimate that is robust to single-hour prediction anomalies.

The third component, a YOLOv8-seg instance segmentation model fine-tuned on a custom ten-class corrosion image dataset of over 10,000 annotated photographs, enables the automated detection, severity classification, and area quantification of corrosion regions in standard equipment photographs. The integration of this visual module with the two numerical prediction models within a single desktop GUI bridges the gap between cutting-edge computer vision research and practical inspection workflows accessible to non-specialist maintenance engineers.

4.8 Scientific and Industrial Significance

Taken together, the three components demonstrate that the combination of temporal sequence modelling, environmental sensor analysis, and computer vision can provide a substantially more complete picture of equipment corrosion health than any single modality alone. The modular architecture allows each component to be updated or replaced independently as new data and better algorithms become available, ensuring that the system can evolve alongside advances in the field.

From a scientific perspective, this work makes three distinct methodological contributions to the corrosion prediction literature. First, it demonstrates that a single BiLSTM model can achieve meaningful RUL prediction accuracy across eleven heterogeneous equipment categories simultaneously, using equipment-type conditioning rather than separate models. Second, it introduces and validates a rigorous temporal evaluation protocol for atmospheric corrosion rate prediction that avoids the data leakage endemic to random-split evaluations in the existing literature. Third, it demonstrates the application of YOLOv8-seg instance segmentation to a ten-class corrosion taxonomy, extending the state of the art in visual corrosion inspection from binary detection and semantic segmentation to instance-level severity classification and area quantification in a single model forward pass.

From an industrial perspective, the system addresses a genuine gap in the available tooling for corrosion management. Most commercial asset management platforms provide generic sensor dashboards [68], [69], [70] and threshold-based alerts but lack the AI-driven prognostic capability needed for true predictive maintenance. CorroSense AI provides three specialised predictive tools thickness-based RUL, environmental rate forecasting, and visual condition assessment integrated into a single, accessible, and open-source desktop application that requires no cloud infrastructure, no subscription, and no machine learning expertise to operate.

The entrepreneurial perspective discussed informally in professional circles surrounding this work confirms that the technical capabilities of CorroSense AI address a real and commercially significant market need [71, 72]. The global industrial corrosion management market [73], currently dominated by generic enterprise asset management platforms with no corrosion-specific AI, offers a clear opportunity for a specialised, accessible, and affordable predictive maintenance tool of the kind developed in this thesis.

4.9 Limitations and Future Directions

The critical analysis presented in Chapter Four identified several important limitations that define the research agenda for future work. Data sparsity for rare equipment types in the LSTM dataset limits prediction reliability for Cyclones, Coalescer Filters, and Compressors. Single-site specificity of the XGBoost model limits its direct transferability to other geographic locations and metal types without retraining. Annotation quality challenges in the visual dataset, particularly for the mild-corrosion and copper-corrosion classes, limit the recall of the YOLOv8-seg model for early-stage damage. And the absence of cross-modal information sharing between the three components means that inconsistencies between the thickness-based, environment-based, and visually-based health assessments are not automatically detected.

Future work should address these limitations through: expanded inspection data collection partnerships; multi-site, multi-material XGBoost retraining; standardised annotation protocols and dataset enrichment for the visual component; and the development of a multi-modal fusion architecture that jointly processes all three data streams into a unified, uncertainty-quantified health state estimate.

4.10 Closing Remarks

In conclusion, this work contributes to the intersection of machine learning, industrial maintenance, and computer vision by demonstrating that a practical, multi-modal corrosion prediction system can be built from freely available open-source tools TensorFlow, XGBoost, and Ultralytics YOLOv8 [21, 26, 44] with a level of engineering rigour sufficient to produce commercially viable and technically defensible results. The code, methodology, and evaluation protocols developed in this thesis provide a solid foundation for future work by researchers and practitioners seeking to apply artificial intelligence to the challenge of industrial corrosion management.

4.11 Conclusion

This chapter has provided a balanced and critical analysis of the CorroSense AI system's performance, identifying genuine strengths in the methodological rigour of the evaluation protocols and the physical interpretability of the learned models, alongside concrete limitations related to data availability, single-site specificity, and annotation quality. The modular three-component architecture is a pragmatic response to the heterogeneity of corrosion measurement data and is well-suited to incremental improvement as additional data becomes available.

Collectively, the analysis confirms that CorroSense AI constitutes a methodologically sound and practically deployable system for multi-modal industrial corrosion prediction — one whose current limitations are well-characterised and point to a clear path of incremental improvement rather than fundamental redesign. The general conclusion of this thesis, including a summary of contributions and recommendations for future work, is presented in the following chapter.

Bibliography

Bibliography

[1] G. Koch, J. Varney, N. Thompson, O. Moghissi, M. Gould, and J. Payer, "International measures of prevention, application, and economics of corrosion technologies study (IMPACT)," tech. rep., NACE International, Houston, TX, 2016.

[2] R. W. Revie and H. H. Uhlig, *Corrosion and Corrosion Control: An Introduction to Corrosion Science and Engineering.* Hoboken, NJ: John Wiley & Sons, 4th ed., 2008.

[3] R. K. Mobley, *An Introduction to Predictive Maintenance.* Oxford: Butterworth-Heinemann, 2nd ed., 2002.

[4] International Organization for Standardization, "ISO 9223:2012 — corrosion of metals and alloys — corrosivity of atmospheres — classification, determination and estimation," tech. rep., International Organization for Standardization, Geneva, 2012.

[5] F. Samie, J. Tidblad, and V. Kucera, "Atmospheric corrosion rate prediction of carbon steel using support vector regression," *Corrosion Science*, vol. 120, pp. 110–119, 2017.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* Cambridge, MA: MIT Press, 2016.

[7] D. A. Jones, *Principles and Prevention of Corrosion.* Upper Saddle River, NJ: Prentice Hall, 2nd ed., 1996.

[8] J. R. Davis, ed., *Corrosion: Understanding the Basics.* Materials Park, OH: ASM International, 2000.

[9] International Organization for Standardization, "ISO 8044:2020 — corrosion of metals and alloys — basic terms and definitions," tech. rep., International Organization for Standardization, Geneva, 2020.

[10] C. Leygraf, I. O. Wallinder, J. Tidblad, and T. Graedel, *Atmospheric Corrosion.* Hoboken, NJ: John Wiley & Sons, 2nd ed., 2016.

[11] D. Landolt, *Corrosion and Surface Chemistry of Metals.* Lausanne: EPFL Press, 2007.

[12] J. Tidblad, V. Kucera, A. A. Mikhailov, and D. Knotkova, "Improvement of the ISO CORRAG programme: application of the power model in trend analysis," in *Outdoor Atmospheric Exposure Testing of Nonmetallic Materials, ASTM STP 1348*, pp. 75–88, 1999.

[13] S. Oesch and M. Faller, "Environmental effects on materials: The effect of the air pollutants SO_2, NO_2, NO and O_3 on the corrosion of copper, zinc and aluminium," *Corrosion Science*, vol. 39, no. 9, pp. 1505–1530, 1997.

[14] T. E. Graedel, "Copper patinas formed in the atmosphere — II. A qualitative assessment of mechanisms," *Corrosion Science*, vol. 27, no. 7, pp. 721–740, 1987.

[15] American Petroleum Institute, "API 510 — pressure vessel inspection code: In-service inspection, rating, repair, and alteration," tech. rep., American Petroleum Institute, Washington, DC, 2014. 10th ed.

[16] American Petroleum Institute, "API 570 — piping inspection code: In-service inspection, rating, repair, and alteration of piping systems," tech. rep., American Petroleum Institute, Washington, DC, 2009. 3rd ed.

[17] P. R. Roberge, *Handbook of Corrosion Engineering.* New York: McGraw-Hill, 2nd ed., 2012.

[18] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[21] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (San Francisco, CA), pp. 785–794, 2016.

[22] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[23] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (Las Vegas, NV), pp. 779–788, 2016.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, (Venice, Italy), pp. 2961–2969, 2017.

[26] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," 2023. version 8.0.0. DOI: 10.5281/zenodo.7347926.

[27] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE CVPR*, (Honolulu, HI), pp. 2117–2125, 2017.

[28] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. 13th European Conference on Computer Vision (ECCV)*, (Zurich, Switzerland), pp. 740–755, 2014.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[30] Z. Pei *et al.*, "Towards understanding and prediction of atmospheric corrosion of an Fe/Cu corrosion sensor via machine learning," *npj Materials Degradation*, 2020.

[31] A. Medina-González, J. Valerdi, R. Giácoman-Vallejos, and P. Quintana-Owen, "Corrosion rate prediction of copper under atmospheric conditions using machine learning," *npj Materials Degradation*, vol. 6, no. 1, pp. 1–10, 2022.

[32] W. Wu, L. L. Zhu, *et al.*, "Prediction model for corrosion rate of low-alloy steels under atmospheric conditions using machine learning algorithms," *International Journal of Minerals, Metallurgy and Materials*, vol. 29, no. 11, 2022.

[33] X. Zheng, Q. Jiang, and D. Jiang, "Long short-term memory network for remaining useful life estimation," in *Proc. IEEE International Conference on Prognostics and Health Management (ICPHM)*, (Dallas, TX), pp. 88–95, 2017.

[34] H. Cui, S. Chen, Y. Zhang, and L. Wang, "Remaining useful life prediction of corroded pipelines using gated recurrent units trained on inline inspection data," *International Journal of Pressure Vessels and Piping*, vol. 174, pp. 74–83, 2019.

[35] Z. Wu, Z. Wang, H. Wei, *et al.*, "Remaining useful life prediction for equipment based on RF-BiLSTM," *AIP Advances*, vol. 12, no. 11, p. 115209, 2022.

[36] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, (Munich, Germany), pp. 234–241, 2015.

[37] B. Beng Chye, Y. Cai, X. Dong, J. Zhang, and X. Sun, "Vision-based automated corrosion detection for industrial inspection using deep instance segmentation," in *Proc. IEEE 17th International Conference on Automation Science and Engineering (CASE)*, (Lyon, France), pp. 1834–1839, 2021.

[38] Y. Gao *et al.*, "Deep learning-based steel bridge corrosion segmentation and condition rating using Mask RCNN and YOLOv8," *Infrastructures*, vol. 9, no. 1, p. 3, 2024.

[39] Cawilai Research Group, "A comparative study of YOLOv5 and YOLOv8 for corrosion segmentation tasks in metal surfaces," *Results in Engineering*, 2024.

[40] M. Hosseinzadeh and M. R. Bahaari, "Integrated corrosion monitoring framework for gathering pipelines: coupling simulation, IoT, and machine learning," *Journal of Hazardous Materials*, 2025.

[41] S. Jamaludin *et al.*, "AI-based predictive maintenance framework for online corrosion survey and monitoring," *Institute of Corrosion Technical Bulletin*, 2025.

[42] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[43] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice.* Melbourne: OTexts, 3rd ed., 2021.

[44] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, (Savannah, GA), pp. 265–283, 2016.

[45] F. Chollet *et al.*, "Keras," 2015. GitHub repository.

[46] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd International Conference on Machine Learning (ICML)*, (Lille, France), pp. 448–456, 2015.

[47] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[48] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th International Conference on Machine Learning (ICML)*, (Haifa, Israel), pp. 807–814, 2010.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd International Conference on Learning Representations (ICLR)*, (San Diego, CA), 2015.

[50] P. J. Huber, "Robust estimation of a location parameter," *Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 73–101, 1964.

[51] L. Prechelt, "Early stopping — but when?," in *Neural Networks: Tricks of the Trade* (G. B. Orr and K.-R. Müller, eds.), pp. 55–69, Berlin: Springer, 1998.

[52] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[53] M. Kuchař and J. Fišer, "Copper wire corrosion and meteorological conditions in a hangar." Zenodo, dataset version 0.1, 2023. DOI: 10.5281/zenodo.8239156. License: CC BY 4.0.

[54] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and machine learning forecasting methods: Concerns and ways forward," *PLOS ONE*, vol. 13, no. 3, p. e0194889, 2018.

[55] ASTM International, "ASTM G50-10 — standard practice for conducting atmospheric corrosion tests on metals," tech. rep., ASTM International, West Conshohocken, PA, 2015.

[56] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* Sebastopol, CA: O'Reilly Media, 3rd ed., 2022.

[57] "Rust detection dataset." Roboflow Universe, published by user "Rust detection," March 2023. 10,072 images with instance-level polygon annotations across 10 classes. License: CC0, 2023.

[58] G. van Rossum and F. L. Drake, *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009.

[59] T. Chen, "XGBoost Python API reference," 2023. XGBoost documentation, version 1.7.

[60] W. McKinney, "Data structures for statistical computing in Python," in *Proc. 9th Python in Science Conference (SciPy)*, (Austin, TX), pp. 56–61, 2010.

[61] C. R. Harris *et al.*, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, 2020.

[62] G. Bradski, “The OpenCV library,” *Dr. Dobb’s Journal of Software Tools*, vol. 25, pp. 120–126, 2000.

[63] T. Georgiou, “CustomTkinter: Modern and customizable Python UI-library based on Tkinter,” 2022. GitHub repository.

[64] ReportLab Inc., *ReportLab PDF Library User Guide*, 2023. version 4.0.

[65] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[66] N. Tajbakhsh *et al.*, “Convolutional neural networks for medical image analysis: Full training or fine tuning?,” *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1299–1312, 2016.

[67] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proc. 33rd International Conference on Machine Learning (ICML)*, (New York, NY), pp. 1050–1059, 2016.

[68] IBM Corporation, “IBM Maximo application suite,” 2023. product documentation.

[69] SAP SE, “SAP asset intelligence network,” 2023. product documentation.

[70] GE Digital, “APM Predix asset performance management,” 2023. product documentation.

[71] E. Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*. New York: Crown Business, 2011.

[72] A. Osterwalder and Y. Pigneur, *Business Model Generation*. Hoboken, NJ: John Wiley & Sons, 2010.

[73] MarketsandMarkets Research, “Asset integrity management market — global forecast to 2028,” 2023. Report No. SE 5263.

[74] X. Zhu, H. Feng, C. Peng, and Q. Chen, “LSTM-based prediction of corrosion rate from multi-site atmospheric monitoring data,” *npj Materials Degradation*, vol. 5, no. 1, pp. 1–12, 2021.

[75] Y. Li, X. Zhao, C. Chen, and B. Li, “Instance-level corrosion detection using Mask R-CNN for quantitative inspection of industrial components,” *NDT & E International*, vol. 112, p. 102253, 2020.

[76] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[77] N. Bao, Y. Chen, C. Liu, and J. Ye, “Deep learning-based corrosion detection using multi-class semantic segmentation on structural steel images,” *Construction and Building Materials*, vol. 316, p. 125942, 2022.

[78] K. Schwab, “The fourth industrial revolution: What it means, how to respond,” *Foreign Affairs*, vol. 95, no. 1, pp. 1–8, 2016. Also published as: Schwab, K. *The Fourth Industrial Revolution*. Geneva: World Economic Forum, 2016. ISBN 978-1-944-83500-2.