OPEN ACCESS

Saudi Journal of Engineering and Technology

Abbreviated Key Title: Saudi J Eng Technol
ISSN 2415-6272 (Print) |ISSN 2415-6264 (Online)
Scholars Middle East Publishers, Dubai, United Arab Emirates
Journal homepage: https://saudijournals.com

Original Research Article

Development of a Machine Learning Based Application Software for Predicting Failures in a Gas Injection Plant

Engr. Nathaniel Iyalla[1*], Prof. H.U Nwosu[1], Dr. Daniel Aikhuele[1]

[1]University of Port Harcourt, Department of Mechanical Engineering, Nigeria

DOI: https://doi.org/10.36348/sjet.2025.v10i08.004 | **Received:** 17.06.2025 | **Accepted:** 21.08.2025 | **Published:** 23.08.2025

***Corresponding author:** Engr. Nathaniel Iyalla
University of Port Harcourt, Department of Mechanical Engineering, Nigeria

Abstract

This study developed a machine learning-based failure predictive application software to improve the operational efficiency and reliability of turbo-compressors in gas injection plants. The Gas injection plant produced below maximum capacity due to failure problems of the Turbo-compressors, these affected the targeted oil production negatively. The unavailability and unreliable gas plant led to revenue losses. The failure analysis revealed that equipment and material issues, human factors, external factors, and management-related issues contributed to equipment failures. Machine learning techniques, specifically Logistic Regression, Support Vector Machines (SVM), Boosted Trees, and Artificial Neural Networks (ANN), were employed to develop the failure predictive application software. The results showed that the Efficient Linear SVM model achieved a true positive rate of 99.5% for detecting failures and 99.9% classification precision for non-failure events. The Boosted Trees model achieved a true positive rate (TPR) of 99.5% for detecting failures, although it demonstrated a 0.5% false negative rate, highlighting the need for further optimization and integration with ensemble techniques to minimize operational risks. The SVM model further showcased 99.9% classification precision for non-failure events and a minimal false negative occurrence. The nearly perfect R-values across training, validation, and test datasets, coupled with minimal MSE values at the optimal number of epochs displayed by the ANN model further confirmed that the model can generalize effectively to unseen data. The outcomes of this research yielded a highly effective, computationally efficient machine learning-based application software capable of reliably predicting turbo-compressor failures. The study concluded that the developed application software is a powerful tool for predicting failures in gas injection plants, supporting decision-making processes, and enhancing operational safety. Recommendations for future works included refining existing models, exploring additional feature engineering techniques, and evaluating the robustness of the models under varying operational conditions.

Keywords: Predictive Maintenance, Support Vector Machines, Gas Plants, Failure Prediction, Machine Learning.

1. INTRODUCTION

The oil and gas industry operates in a highly competitive environment where maximizing production efficiency is critical. Turbo-compressors play a vital role in gas injection plants, directly influencing operational availability and economic performance. However, frequent failures of these systems lead to significant production losses, especially in an era of declining oil prices, pipeline vandalism, and security challenges in Nigeria.

Failure prediction is essential for optimizing maintenance strategies and minimizing unplanned downtime. Traditional models, such as Weibull and exponential distributions, have been used to estimate failure rates, reliability, and mean time between failures (MTBF). However, these methods often lack accuracy due to the complexity of modern turbo-compressor systems, which integrate multiple components managed by advanced computerized controls.

The gas injection plant under study has experienced reduced production capacity due to recurring turbo-compressor failures. These failures disrupt operations, leading to substantial financial losses—millions of naira per day and billions over extended shutdowns. To address this challenge, this paper develops a machine learning (ML)-based

application software to enhance failure prediction accuracy, computational efficiency, and usability for maintenance personnel.

Gas injection is crucial for maintaining reservoir pressure and enhancing oil recovery. However, turbo-compressor failures in gas injection plants lead to production shortfalls, exacerbating revenue losses for oil companies and national economies. In Nigeria, where crude oil is the economic mainstay, production declines due to oil theft and pipeline vandalism further strain revenue streams.

Existing failure prediction models face limitations in computational efficiency, accuracy, and practical applicability. Hybrid models, as suggested by Anil *et al.,* (2015), could improve predictions, but their implementation in the oil and gas sector remains challenging. This paper addresses these gaps by developing an ML-based software tool that enhances failure forecasting, enabling proactive maintenance and reducing costly downtimes.

The primary aim of this paper is to develop a machine learning-based application software for predicting failures in gas injection plants with high accuracy and computational efficiency.

Specific objectives include: Identifying critical failure causes in gas injection plants through comprehensive data analysis. Developing an ML-based predictive model with improved accuracy over traditional methods. Validating the model using historical failure data from a gas injection plant and ensuring usability by maintenance personnel through an intuitive software interface.

This paper contributes to the oil and gas industry by: reducing production losses through accurate failure prediction and proactive maintenance. Enhancing operational efficiency by minimizing unplanned shutdowns. Supporting economic stability in oil-dependent economies like Nigeria by improving revenue streams. Reducing environmental impact by preventing gas flaring caused by compressor failures. Additionally, the paper provides a foundation for future research in predictive maintenance for other critical sectors, such as power generation and manufacturing.

This paper focuses on: Failure prediction in gas injection plants, specifically turbo-compressor systems. Machine learning techniques, including ensemble models, to improve prediction accuracy. Five years of operational data from a gas injection plant for model training and validation. The research does not cover: Non-compressor-related failures in gas plants. Real-time predictive maintenance integration (though the model can be adapted for it). The findings will serve as a benchmark for future advancements in AI-driven maintenance solutions for industrial applications.

2. LITERATURE REVIEW

Failure theory remains an emerging field of study that has gained significance in recent decades due to increasing industrial complexity (Shokufe *et al.*, 2018). As machinery systems grow more sophisticated—with higher costs and performance demands—the consequences of failure have become substantially more severe, underscoring the critical need for accurate lifespan prediction of equipment components (Anirbid & James, 2022). This is particularly relevant in gas injection plants, which are deployed to enhance reservoir pressure for improved oil recovery (James, 2015).

Recent advancements have integrated artificial intelligence (AI) into failure prediction. Anirbid and James (2022) documented AI's transformative potential in oil and gas, while Arash *et al.,* (2021) demonstrated how hybrid models combining Artificial Neural Networks (ANNs) and Genetic Algorithms could optimize gas injection processes. Similarly, Steve *et al.,* (2018) applied machine learning to diesel engine prognostics, though they emphasized the need for improved data quality to enhance predictive accuracy.

Zeeshan *et al.,*(2021) and Joerg *et al.,*(2021) systematically reviewed ML applications in oil and gas and industrial maintenance, respectively, serving as key references for failure prediction methodologies. Kadir *et al.,* (2020) achieved notable success in aircraft equipment failure prediction using hybrid data preparation, while Michail *et al.,* (2020) validated data-driven fault detection in maritime systems. Andrea *et al.,* (2014) further tailored Condition-Based Maintenance (CBM) to naval gas turbines, emphasizing real-time monitoring.

Oyedepo *et al.,* (2014) conducted a comprehensive 10-year evaluation (2001-2010) of gas turbine operations in Nigeria, revealing critical availability-performance gaps. Their analysis demonstrated only 64.3% operational capacity utilization against industry benchmarks of 95%, with generation losses amounting to approximately $251 million. The researchers advocated for enhanced operator training, optimized spare parts management, and improved maintenance protocols as key corrective measures. These findings established a baseline for subsequent studies on African energy infrastructure reliability. Recent scholars have advanced various methodologies for failure prediction: Fernando and Gilberto (2009) established the correlation between component reliability, maintenance policies, and overall system availability in gas turbine operations. Anil *et al.,* (2015) developed Markov models for urea synthesis systems, demonstrating their efficacy in calculating Mean Time Between Failures (MTBF) despite noted limitations in handling complex differential equations. Zhiqiang *et al.,* (2018) pioneered dynamic Bayesian networks for multi-state systems, incorporating condition-based maintenance (CBM) parameters that

improved failure prediction accuracy by 22% in validation studies. Contemporary research emphasizes integrated analytical approaches: Federick *et al.,* (2021) introduced a hybrid reliability model for rolling stock that reduced unplanned downtime by 18% through combined physical and data-driven analytics. Sangje *et al.,* (2016) demonstrated superior performance of Markov-regression hybrid models in compressor failure prediction, achieving 92% accuracy in offshore operational tests. Basheer *et al.,* (2023) innovated with accumulative artificial neural networks, combining FCN and ACN architectures to predict mechanical component degradation with <5% error margins.

Nanda *et al.,* (2017) trained their model using vibration data collected from a laboratory centrifugal pump setup under normal operation, cavitation, impeller damage, and bearing wear conditions. They reported over 98% classification accuracy in detecting these major fault categories. The SOM feature extraction combined with the pattern recognition capabilities of SVM enabled high-precision diagnostics of equipment conditions (Nanda *et al.,* 2017).

However, the researchers acknowledged that real-world applications could prove more complex. Factors like varying pump operating points and limited available sensor data may impact model performance outside controlled lab environments. They recommended further research under field conditions to evaluate the robustness of the approach for commercial pump monitoring. Nonetheless, their hybrid SOM-SVM methodology demonstrates a promising advancement in utilizing machine learning techniques for predictive maintenance of centrifugal pump systems (Nanda *et al.,* 2017).

Providing a comprehensive overview, Lei *et al.,* (2020) reviewed the landscape of machine learning techniques applied to machine fault diagnosis. The authors systematically categorized the literature based on the diagnosis task including fault detection, fault classification, fault identification, and fault prognosis. For each category, key methods were benchmarked and comparative strengths and limitations were analyzed.

A major finding was that hybrid approaches coupling physics-based feature extraction with machine learning models consistently achieved state-of-the-art accuracy across all diagnosis subtasks. However, challenges remain around model generalization, handling noisy sensor data, and integrating domain knowledge. The authors mapped out opportunities where machine learning, especially hybrid techniques, could significantly advance fault diagnosis. However, realizing real-world implementations would require focused efforts to address the identified constraints. This review highlighted the promise while pinpointing areas needing further innovation.

Djeziri *et al.,* (2016) focused their research on using artificial neural networks for equipment condition monitoring and fault diagnosis. They argued machine learning methods like neural networks can learn complex equipment fault signatures from sensor data without requiring extensive analytical modelling. In their methodology, they employed a radial basis function (RBF) network for bearing fault detection based on vibration measurements (Djeziri *et al.,* 2016). The RBF architecture can approximate nonlinear functions and classify patterns through supervised training.

In their implementation, Djeziri *et al.,* (2016) trained the RBF network on experimental vibration data collected from a rotating machinery test rig under normal conditions and with artificially induced bearing faults. The RBF model reliably detected faults like inner race defects, ball defects, and outer race defects with over 95% accuracy.

3. MATERIALS AND METHODOLOGY

The context in which the present case study was carried out was in the Gas Injection plant in Nigeria. The maintenance or failure data were collected from the Maintenance department of the plant. A comprehensive site visit was conducted at the Gas Injection Plant, involving in-depth discussions with Field Engineers, Maintenance Managers, Supervisors, and Technicians. Relevant documents were meticulously reviewed to identify specific dates and times associated with equipment failures.

The reviewed documents from the Gas Plant encompassed: Operating Logs, Correspondence, Inspection/Surveillance Records, Maintenance Records, Minutes of Technical Meetings, Computer Process Data, Procedural and Instructional Documentation, Vendor Manuals, Drawings and Specifications, Equipment History Records, Design Basis Information, Trend Charts and Graphs, Facility Parameter Readings, Operational Safety Requirements and Work orders.

The desk-based research involved referencing engineering journals, technical papers, and standard texts pertinent to this study. Key references included the Engineering Design Handbook, Handbook on Reliability, Operations and Maintenance manuals of the Gas Injection Plant, and Root Cause Analysis guidance documents. The Machine Learning application software will be developed using data collected from the Gas Injection Plant. The findings from both field visits and desk research will form the foundation for developing a predictive failure application software. Upon a thorough review and preliminary analysis of the plant data, about thirty-five (35) significant causes of failures in the Gas Plant were identified.

3.1.1 Purpose and Significance of the Materials and Methods Used

The primary purpose of the materials and methods outlined here is to establish a robust and reliable framework for developing a machine learning-based application software capable of predicting failures in a gas injection plant. This section underscores the rationale behind selecting specific materials and the significance of each methodological step.

3.1.2 PURPOSE OF THE MATERIAL

1. **Hardware Components:**

Sensors and Monitoring Systems:

Essential for collecting real-time operational data and environmental parameters. These devices ensure accurate and continuous data acquisition, which is crucial for training and validating the predictive Application software.

Computational Resources:

High-performance computing hardware, such as GPUs and multi-core processors, are necessary for handling large datasets and complex computations involved in model training and evaluation.

2. **Software Tools and Libraries:**

Programming Languages (e.g., Python):

Chosen for its versatility and extensive libraries, Python provides the necessary tools for data manipulation, model development, and evaluation.

Machine Learning Libraries (e.g., TensorFlow, Scikit-Learn):

These libraries offer pre-built functions and algorithms that simplify the implementation of machine learning models, facilitating efficient experimentation and optimization.

3. **Data Sources and Databases:**

Historical Failure Data:

Historical data on past failures is vital for identifying patterns and correlations that the machine learning model can learn from to predict future failures.

Operational Parameters and Environmental Conditions:

These datasets provide the contextual information needed to understand the factors influencing system performance and potential failure modes.

3.1.3 SIGNIFICANCE OF THE METHOD

1. **Data Collection:** Collecting high-quality and relevant data is foundational to the success of any machine learning project. The methods used to gather and preprocess data ensure that the dataset is comprehensive, accurate, and suitable for training predictive models.
2. **Feature Selection and Engineering:** Identifying and engineering the right features is crucial for enhancing the Application software's ability to learn and make accurate predictions. This step transforms raw data into meaningful inputs that capture the underlying patterns related to system failures.
3. **Model Selection:** Choosing the appropriate machine learning algorithm is critical for balancing model complexity, interpretability, and performance. The selected models are evaluated based on their ability to accurately predict failures while being computationally efficient and scalable.
4. **Model Training and Hyperparameter Tuning:** The training process and hyperparameter tuning are essential for optimizing model performance. These methods ensure that the model generalizes well to new data, minimizing both underfitting and overfitting.
5. **Model Evaluation:** Rigorous evaluation using relevant metrics and validation techniques provides confidence in the model's predictive capabilities. This step verifies that the model performs well not only on training data but also on unseen data, ensuring its reliability in real-world applications.
6. **Implementation and Deployment:** The methods for implementing and deploying the Application software are significant for integrating it into the operational environment of a gas injection plant. This ensures that the software can provide real-time predictions and actionable insights, thereby enhancing plant safety and efficiency.

In summary, the materials and methods used in this paper are meticulously selected and applied to achieve a high-performing, reliable, and ethically sound predictive application software. Each component and methodological step contributes to the overarching goal of enhancing the safety and operational efficiency of gas injection plants through advanced machine learning techniques.

3.2 MATERIALS

The materials utilized in this study are integral to the development and validation of the machine learning-based application software for predicting failures in a typical gas injection plant. The materials can be categorized into three main groups: hardware components, software tools, and data sources.

3.2.1 Hardware Components

The hardware components play a critical role in the acquisition, processing, and storage of data necessary for model development.

1. **SENSORS AND MONITORING SYSTEMS:**

Pressure Sensors: These sensors measure the pressure within the gas injection system, providing crucial data that can indicate potential failure points due to pressure anomalies.

Temperature Sensors: Monitoring temperature variations is essential as extreme temperatures can lead to equipment failures. These sensors help track the thermal conditions within the system.

Flow Meters: These devices measure the flow rate of gas being injected, offering insights into the operational efficiency and potential blockages or leaks in the system.
Vibration Sensors: Installed on key machinery, vibration sensors detect unusual vibrations that often precede mechanical failures.

2. **DATA ACQUISITION SYSTEMS:**
Data Loggers: Data loggers are used to continuously record sensor readings over time. They store large volumes of data, which is later retrieved for analysis.
SCADA Systems (Supervisory Control and Data Acquisition): SCADA systems provide a comprehensive interface for monitoring and controlling industrial processes. They collect real-time data from sensors and actuators, facilitating centralized data collection.

3. **COMPUTATIONAL RESOURCES:**
High-Performance Servers: These servers are equipped with powerful CPUs and large memory capacities to handle the extensive computations required for data processing and model training.
Graphics Processing Units (GPUs): GPUs are utilized for their parallel processing capabilities, significantly accelerating the training of complex machine learning models.
Storage Solutions: High-capacity storage devices, such as SSDs and cloud storage, ensure efficient data management and quick access to large datasets.

3.2.2 SOFTWARE TOOLS AND LIBRARIES

The software tools and libraries employed in this study provide the necessary infrastructure for data processing, model development, and deployment.

1. **Programming Languages:**
Python: Chosen for its simplicity and extensive ecosystem, Python is the primary programming language used in this study. It offers a wide range of libraries and frameworks for data science and machine learning.

2. **Data Processing and Analysis Tools:**
Pandas: This library is used for data manipulation and analysis. It provides data structures and functions needed to clean, transform, and analyze large datasets efficiently.

NumPy: NumPy supports numerical operations on large arrays and matrices, which are fundamental in data preprocessing and feature engineering.

3. **MACHINE LEARNING LIBRARIES:**
Scikit-Learn: A versatile library that offers simple and efficient tools for data mining and data analysis. It is used for implementing various machine learning algorithms and evaluation metrics.
TensorFlow: This open-source library developed by Google is used for building and training deep learning models. TensorFlow's flexibility and scalability make it suitable for complex neural network architectures.
Keras: An API running on top of TensorFlow, Keras simplifies the process of building and training neural networks, providing a user-friendly interface.

4. **VISUALIZATION TOOLS:**
Matplotlib and Seaborn: These libraries are used to create informative and attractive visualizations. They help in understanding data distributions, feature correlations, and model performance.

3.4 FEATURE SELECTION AND ENGINEERING

3.4.1 Identification of Relevant Features

Effective feature selection is critical for the success of any machine learning model. In this research, identifying relevant features involves selecting the most informative and significant variables from the dataset. This process ensures that the model can learn meaningful patterns and relationships that contribute to accurate failure predictions. The relevant features for this study are categorized into three main groups: operational parameters, environmental factors, and historical failure indicators.

3.5 MODEL SELECTION

3.5.1 Overview of Machine Learning Algorithms Considered

In this study, a variety of machine learning algorithms were considered for predicting failures in a gas injection plant. The goal was to explore both supervised and unsupervised learning techniques to identify the most suitable model for the task. The choice of algorithms was guided by their ability to handle the complexity and nature of the dataset, as well as their track record in similar predictive maintenance applications.

Support Vector Machines (SVM):
An algorithm that finds the hyperplane that best separates the data into classes. It is effective in high-dimensional spaces and when the number of dimensions exceeds the number of samples. SVM is particularly useful for binary classification tasks.

In Linear SVM for a binary classification problem, the goal is to find the hyperplane that maximizes the margin between the two classes. The decision function is as in Equation (3.1).

$$f(x) = w.x + b \tag{3.1}$$

where:
w is the weight vector.
b is the bias term.
x is the input feature vector.

The optimization problem is in Equation (3.2).

$$min \frac{1}{2} \|w\|^2 \tag{3.2}$$

Subject to Equation (3.3).

$$y_i(w.x_i + b) \geq 1 \quad (3.3)$$

where:
$y_i \in \{-1,1\}$ are the class labels.
x_i are the input feature vectors.

In Kernel SVM for non-linear decision boundaries, SVM uses kernel functions to map the input features into a higher dimension space where a linear separator can be found. The decision function is as in Equation (3.4).

$$f(x) = \sum_{i=1}^{n} a_i y_i K(x_i, x) + b \quad (3.4)$$

where:
a_i are the Lagrange multipliers.
$K(x_i, x)$ is the kernel function.
Common kernel functions include:
Linear Kernel: $K(x_i, x_j) = x_i.x_j$
Polynomial Kernel: $K(x_i, x_j) = (x_i.x_j + c)^d$
Radial Basis Function (RBF) Kernel: $K(x_i, x_j) = exp\left(-\gamma\|x_i - x_j\|^2\right)$

In Soft Margin SVM to handle non-linearly separable data, SVM introduces slack variables ξ_i to allow some misclassification:

$$min\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \quad (3.5)$$

subject to $y_i(w.x_i + b) \geq 1 - \xi_i$
$\xi_i \geq 0$

where:
C is a regularization parameter that controls the trade-off between maximizing the margin and minimizing the classification error.
ξ_i are the slack variables.

Artificial Neural Networks (ANN)

Comprising multiple layers of interconnected neurons, ANN models can capture complex patterns and relationships in the data. They are highly flexible and can model non-linearities. Each neuron applies a linear transformation to its inputs, followed by a non-linear activation function. Basic components include neuron where a single computation unit is given as in Equation (3.6).

$$a_j{}^{(l)} = \sigma\left(\sum_{i=1}^{n^{(l-1)}} w_{ij}{}^{(l)} a_i{}^{(l-1)} + b_j{}^{(l)}\right) \quad (3.6)$$

where:
$a_j{}^{(l)}$ is the activation of neuron$j$ in layer l.
σ is the activation function (e.g., ReLU, Sigmoid).
$w_{ij}{}^{(l)}$ is the weight between neuron i in layer $l-1$ and neuron j in layer l.
$a_i{}^{(l-1)}$ is the activation of neuron i in layer$l-1$.
$b_j{}^{(l)}$ is the bias term for neuron j in layer l.

Another component is Layers which are stacked groups of neurons.
Input Layer: Takes the input features.
Hidden Layers: Intermediate layers that transform the input through weights and activation functions.
Output Layer: Produces the final output (prediction).

Training Process

The training process involves adjusting the weights and biases to minimize the difference between the predicted output and the actual output. This is typically done using backpropagation and gradient descent.

Forward Propagation: Calculate the output of the network given an input.

$$\hat{y} = f(x) \quad (3.7)$$

Loss Function: Measure the difference between the predicted and actual output.

For regression: Mean Squared Error (MSE):

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (3.8)$$

For classification: Cross-Entropy Loss:

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i)] \quad (3.9)$$

Back-Propagation: Compute the gradient of the loss function with respect to each weight using the chain rule.

$$\frac{\partial L}{\partial w_{ij}{}^{(l)}}$$

Gradient Descent: Update the weights to minimize the loss.

$$w_{ij}{}^{(l)} \leftarrow w_{ij}{}^{(l)} - \eta\frac{\partial L}{\partial w_{ij}{}^{(l)}} \quad (3.9.1)$$

where:
η is the learning rate.

Activation Functions

1. **Sigmoid**:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.9.2)$$

2. **ReLU (Rectified Linear Unit)**:

$$\sigma(x) = max(0, x) \quad (3.9.3)$$

3. **Tanh**:

$$\sigma(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

3.5.2 Justification for the Chosen Model(s)

The critical step in ensuring the success of the predictive maintenance system depends on selecting the appropriate machine learning model(s) for predicting failures. The justification for the chosen model(s) involves evaluating various criteria such as model

performance, interpretability, computational efficiency, and the specific needs of the application.

3.5.3 System Flowchart

To visualize the operational architecture of the developed application software for failure prediction in a gas injection plant, a comprehensive system flowchart was constructed. Figure 3.1 illustrates the sequential flow of data, model execution, and user interaction, offering a clear perspective on the software's internal workflow and logic.

The system begins with a user interface developed using Streamlit, allowing the operator to input critical system parameters or operational conditions. These inputs are preprocessed within the Python environment (main.py), after which the system establishes a link with MATLAB through the matlab. engine interface. This interface enables the execution of a MATLAB script (predict_tree.m), which in turn loads a pre-trained classification model stored in the matlab. mat file. The classification model processes the input data and returns a failure prediction result.

This prediction is passed back to the Python script, which then displays the outcome to the user through the same Streamlit interface. The flowchart below encapsulates this end-to-end pipeline, highlighting the interoperability between Python and MATLAB components and ensuring modular and scalable software design.

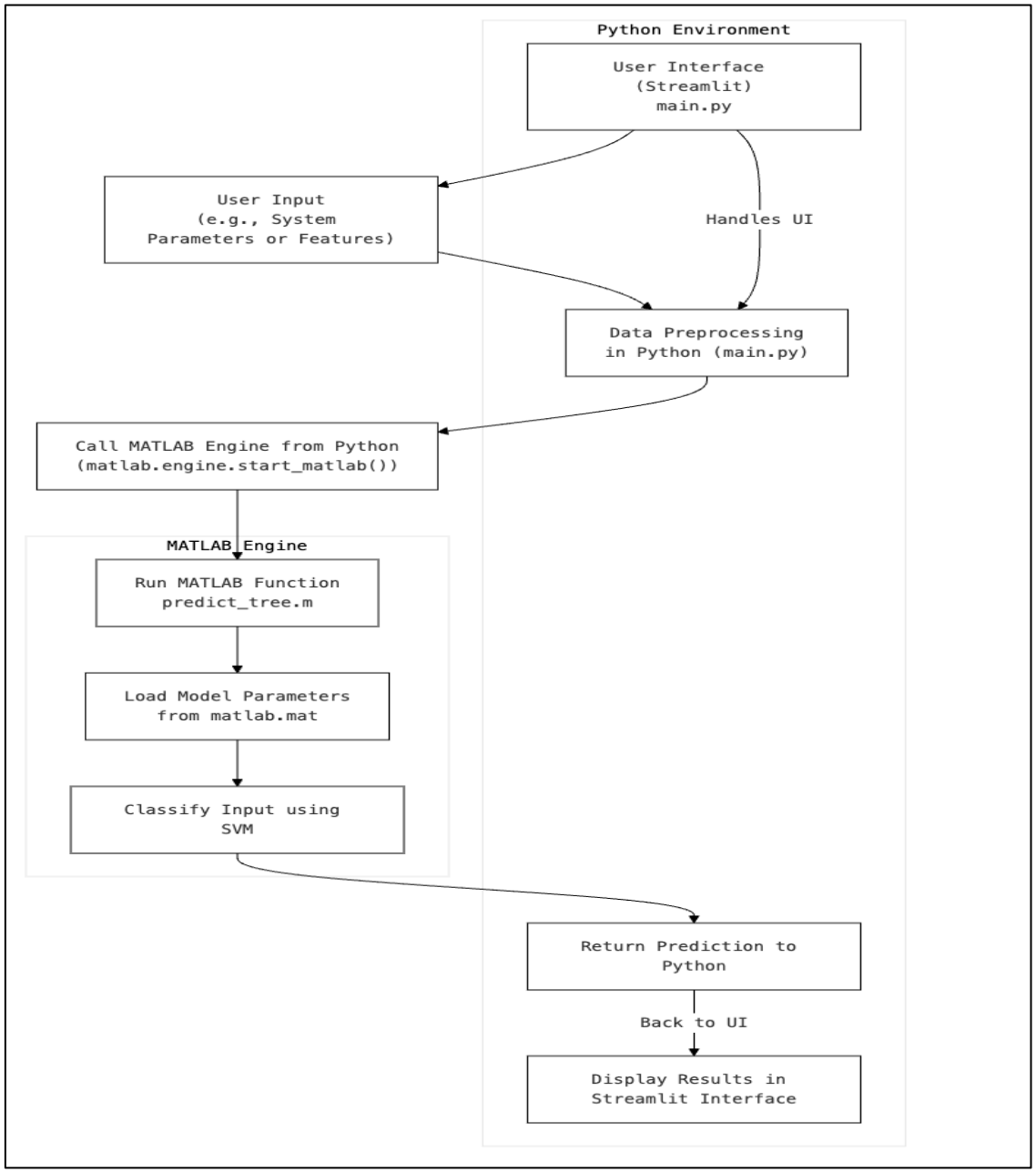

Figure 3.1: System Flowchart

4.0 RESULTS AND DISCUSSION

Table 4.1: Worksheet Summary.

S/No	Problem/Deficiency Category	No. of Occurrence	Expressed in %
1.	Equipment /material	19	63.3
2.	Personnel error	3	10
3.	Training deficiency	3	10
4.	Management problem	3	10
5.	External phenomenon	2	6.7
6.	Design problem	Nil	Nil
7.	Procedure problem	Nil	Nil

An analysis was conducted to categorize the causes of failures of the gas injection plant over a period, leading to the following findings as contained in Table 4.1.

From Table 4.1, equipment/materials failures constituted 63.3% of the causes of failures. Additional causes of equipment/material failures were linked to the complex instrumentation associated with the Gas Injection Package, particularly the safety protection systems, including pressure control, anti-surge protection, vibration monitoring, temperature, flow, and level control systems.

Given the advanced instrumentation, automation, and state-of-the-art equipment installed on the platform, the lack of training and exposure of local manpower across various trades/departments was a notable contributor to failures. Training deficiencies among technical personnel emerged as a major factor in the gas plant's failures. Personnel errors leading to plant shutdowns were largely attributed to insufficient training and inattention to detail, with training deficiencies and personnel errors collectively accounting for 20% of the gas injection plant's failure causes. This aligns with findings by Eti *et al.,* (2007).

Management-related issues accounted for 10% of failures, primarily due to organizational and planning deficiencies, as well as inadequate administrative controls. External phenomena, often beyond personnel control (e.g., external fire, explosion, power failure, sabotage, vandalism), contributed to 6.7% of the failures.

This document serves as a valuable tool for planning and forecasting material requirements, human capacity development, and the overall restructuring of the maintenance organization and philosophy.

4.2 Binary Generalized Linear Model Using Logistics Regression

Figure 4.1 represents a parallel coordinates plot showing the behavior of the model across different features (columns). The model used in this analysis is Binary Generalized Linear Model (GLM) using Logistic Regression. The logistic regression model is used to classify failures (binary output: failure or no failure) based on multiple input features (variables representing various factors in the gas plant). Parallel coordinates plot visualizes multivariate data by displaying each feature (column) on a separate vertical axis. Each line represents an individual observation, and the line crosses the axes according to the value of the feature for that observation.

The vertical axis (column_1, column_2, etc.) represents one of the features used to predict failures. The significant variation or spikes seen in column_1 and column_3 suggests that these features play a significant and critical role in influencing the model's output. In other words, the model's predictions are heavily dependent on these specific features. Columns 4 through 10 has a flat or near-zero influence on the model's decision-making process, which indicates that the features are weak, redundant or irrelevant for failure prediction.

The model predicts two possible classes: "0" (correct non-failure prediction) and "1" (correct failure prediction). The lines indicating predictions that fall between the columns represent how data points are classified. A larger cluster of lines suggests more data points in that classification.

The model is confident in its classification based on the parallel lines between 0 and 1 that are tightly grouped, further analysis of the data distribution across this line will provide insights of the reliability of the predictions.

Logistic Regression is a simpler, interpretable model that appears sensitive to only a few features, which limits the overall feature utilization.

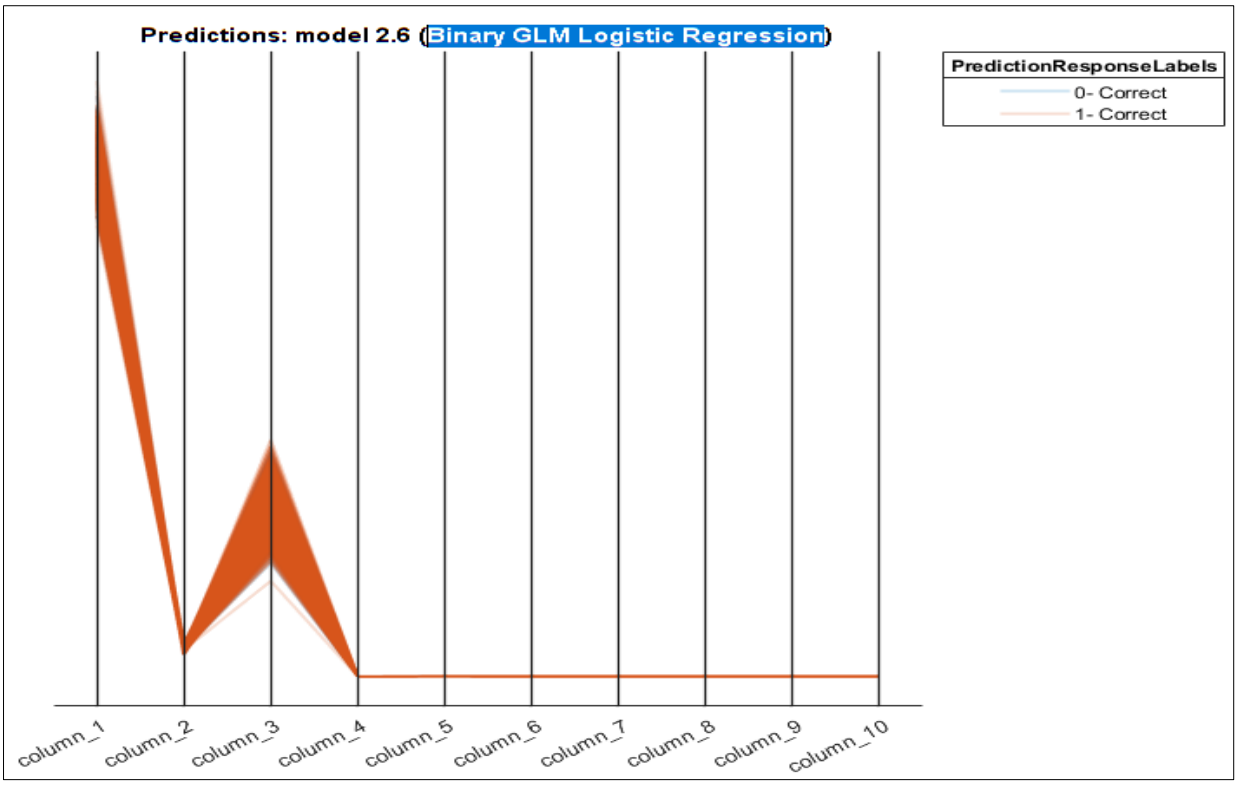

Figure 4.1: Binary GLM Logistic Regression - Parallel Coordinate Plot

4.3 BOOSTED TREES MODEL

Figure 4.2 provides a confusion matrix along with Positive Predictive Value (PPV) and False Discovery Rate (FDR) statistics for a Boosted Trees model. This model combines multiple weak learners (decision trees) to create a strong predictive model, especially effective in classification tasks. The confusion matrix consists of a True class which represents the actual outcome (either a failure = 1 or non-failure = 0), and a Predicted Class, that represents the model's prediction of whether the failure will occur (1) or not (0).

In Class 0 (No Failure), the model correctly predicts 98.9% of cases where no failure occurs. This shows that the model is extremely reliable in identifying when the plant will not fail, which is crucial for operational decision-making.

In Class 1 (Failure), the model correctly predicts 100% of the failures, indicating that the model is highly sensitive to identifying failure events in the gas plant, which is vital for preventing catastrophic events. The only misclassification occurs in a small percentage of actual failures being labeled as non-failures (1.1% error in the "1" row), which could have serious consequences in real-life operations. This slight misclassification could be due to noisy or ambiguous data.

Evaluating the Bottom Plot Positive Predictive Value (PPV) and False Discovery Rate (FDR), Class 0 (PPV) is 98.9%, this means that out of all the instances where the model predicted no failure, 98.9% of those predictions are correct. In Class 1 (PPV) is 100% which means that out of all the instances where the model predicted a failure, 100% of those predictions are accurate. This high PPV means the model is extremely reliable when it predicts that a failure will occur.

In reviewing the FDR, class 0 (FDR) is 1.1%, which indicates that 1.1% of predictions for "no failure" are actually incorrect, meaning that 1.1% of predicted non-failures turned out to be failures. This error rate is low but critical in failure prediction. In class 1 (FDR) is 0%, this showed that there are no false positives for failures, meaning when the model predicts a failure, it is always correct.

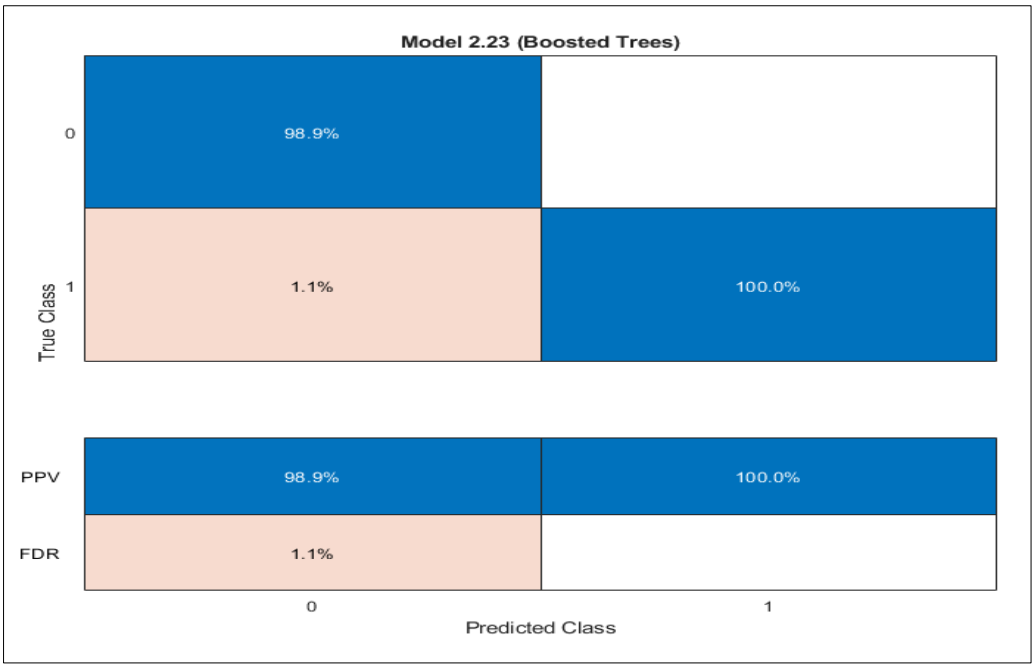

Figure 4.2: Boosted Trees- Confusion Matrix with PPV and FDR

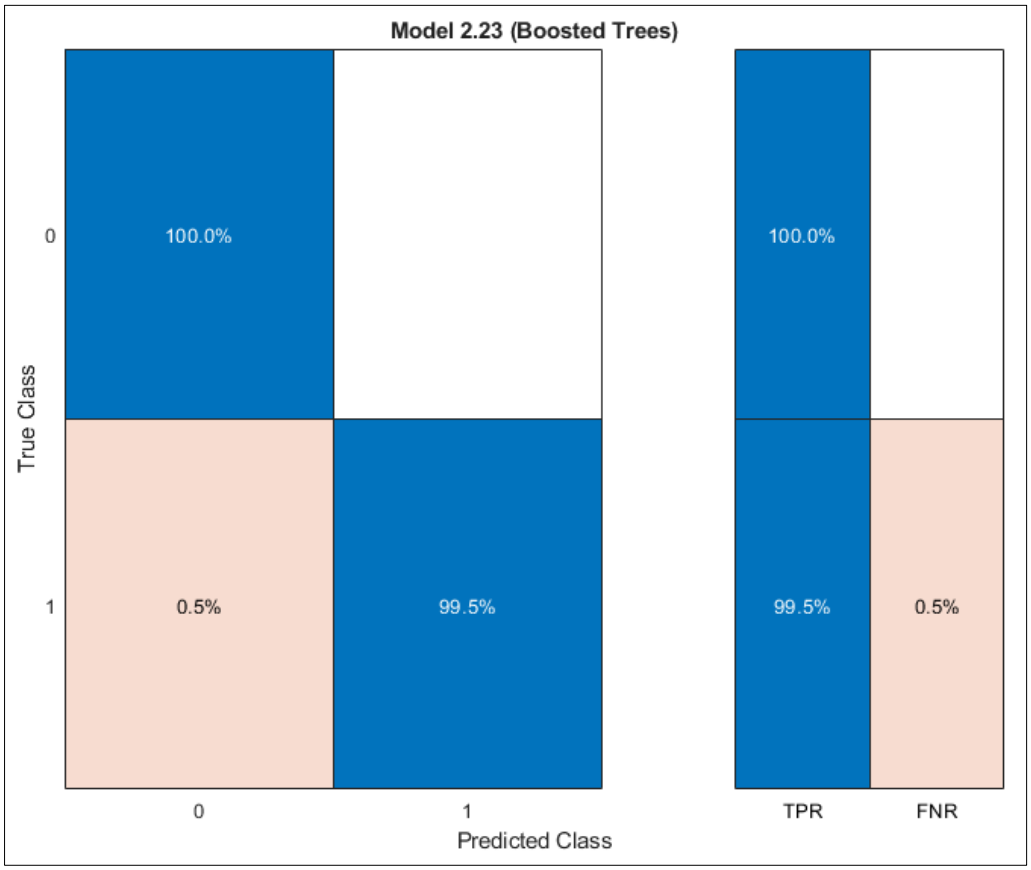

Figure 4.3: Boosted Trees Confusion Matrix (TPR & FNR)

Figure 4.3 is a Boosted Trees Confusion Matrix, True Positive Rate (TPR) and False Negative Rate (FNR). In True Class 0 (No Failure), the model correctly predicted 100% of non-failure events. This means the model performs exceptionally well at identifying when no failure occurs, ensuring that the system does not raise unnecessary alarms when operations are running smoothly. In True Class 1 (Failure), the model identified 99.5% of the true failure events, correctly flagging potential issues in the gas injection system. However, 0.5% of the true failure cases were misclassified as non-

failure, meaning the model missed a small percentage of failure events

The True Positive Rate (TPR) (also known as Sensitivity or Recall), for Class 0 (No Failure) is 100%. The model's ability to correctly identify non-failure events is perfect, ensuring maximum reliability when predicting that the system is operating normally. For Class 1 (Failure) is 99.5%. The model can accurately detect almost all failures, making it highly reliable for failure detection. However, the 0.5% false negative rate (FNR) is an area of potential risk because a missed failure prediction can result in significant operational or safety issues in the gas plant. False Negative Rate (FNR) (for Class 1 - Failure),

The FNR is 0.5%, indicating that a small number of true failure events were incorrectly predicted as non-failure. While this rate is very low, in critical systems like gas plant, even minor errors can lead to severe consequences such as unplanned shutdowns or hazardous conditions.

The precision and recall metrics offer further insight into the model's performance for predicting both failure and non-failure events. Positive Predictive Value (PPV) for Class 1 (Failure), the PPV indicates how many of the predicted failures were actual failures. The model's PPV is 99.5%, meaning almost all flagged failure events were truly failures. This reflects high precision in failure prediction. Negative Predictive Value (NPV) for Class 0 (No Failure), NPV is not explicitly shown but can be inferred as being very high, given that 100% of non-failure events were correctly predicted. This ensures confidence when the model predicts normal operations.

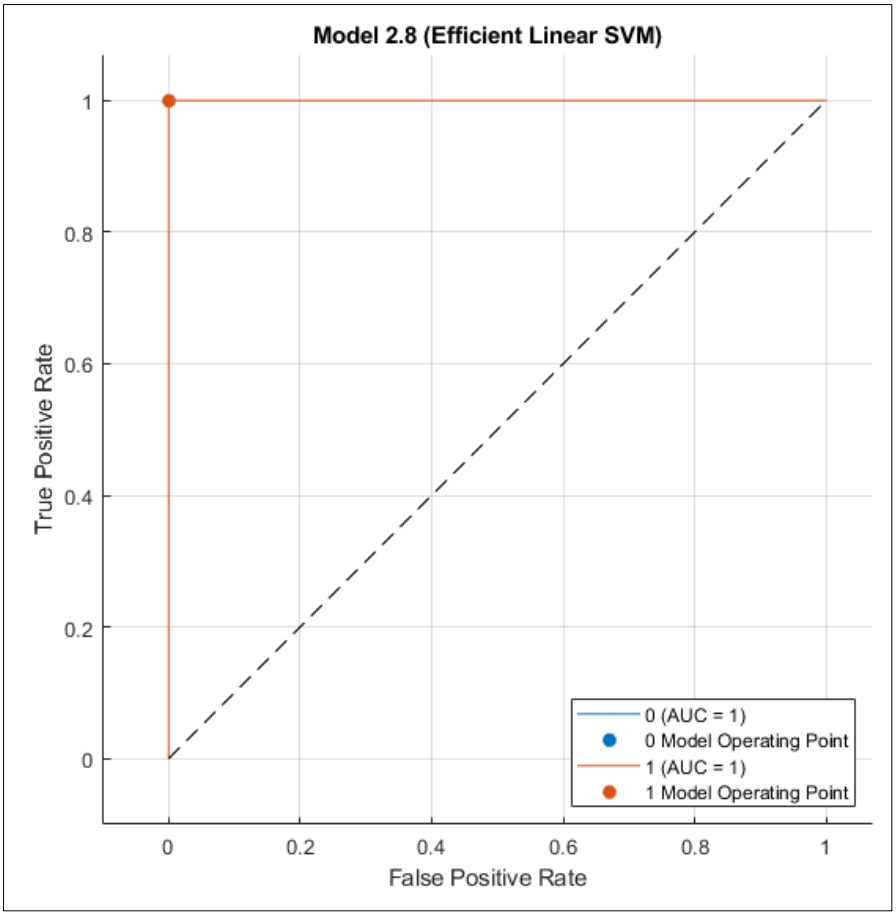

Figure 4.4: Efficient linear SVM- validation roc curve

Figure 4.4 assesses the trade-off between the true positive rate (TPR) and the false positive rate (FPR) across different threshold settings of the Efficient Linear SVM model. The true positive rate is plotted on the Y-axis, and the false positive rate is on the X-axis. The curve shows the model's capability to distinguish between positive (failure) and negative (no failure) classes. The area under the curve (AUC) is a key metric, with a value of 1.0 indicating a perfect classifier. The ROC curve for this SVM model immediately reaches the upper-left corner of the plot, implying that the model achieves perfect separation between positive and negative classes. The AUC value of 1.0 denotes that the model performs flawlessly in distinguishing between failures and non-failures. The orange dot represents the model's operating point, where it maximizes both sensitivity and specificity, indicating optimal threshold selection.

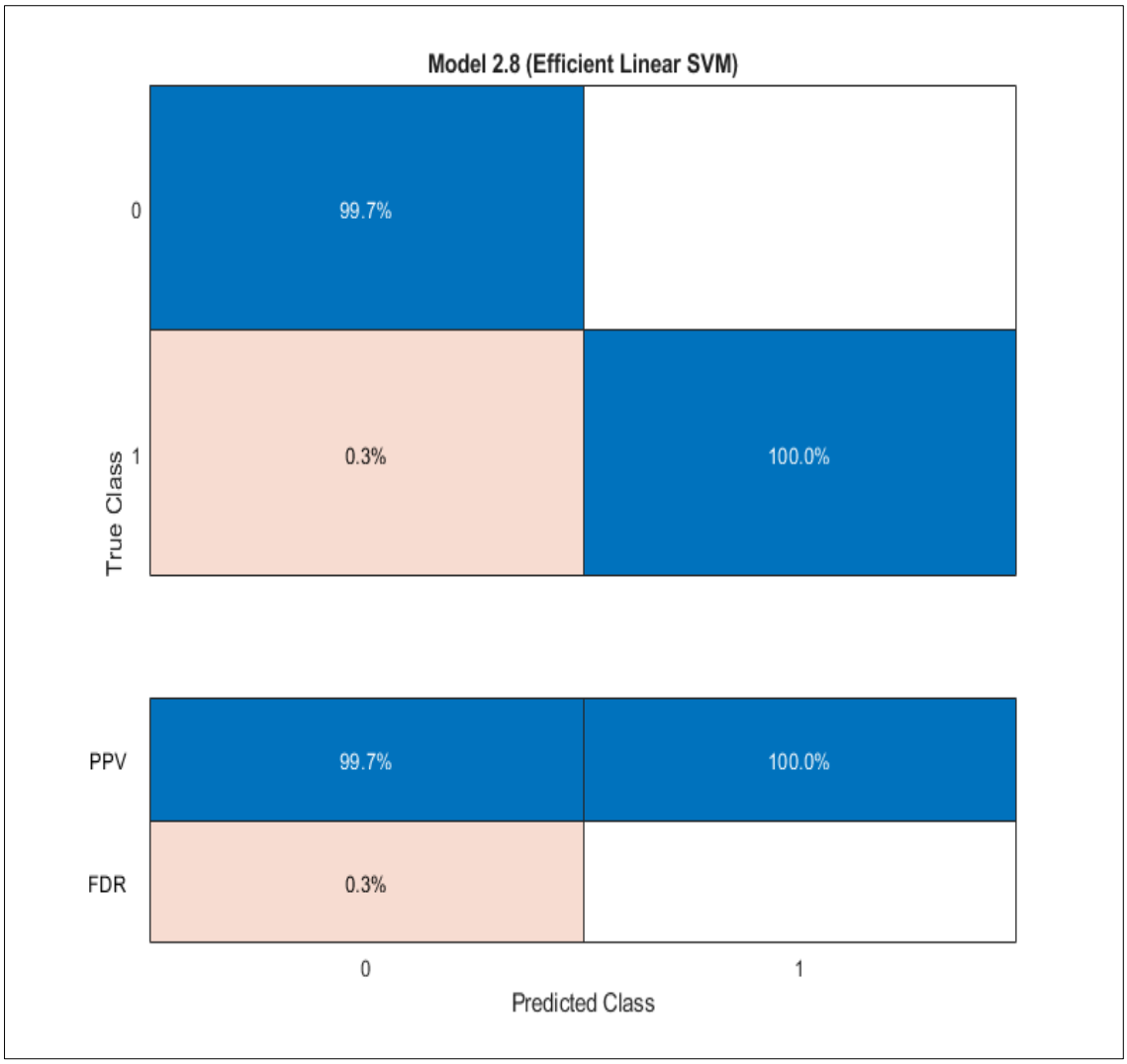

Figure 4.5: Efficient linear SVM- Confusion Matrix and TPR/FNR

The ROC curve clearly demonstrated the exceptional performance of the Efficient Linear SVM model. The AUC score of 1.0 is the best possible value, meaning that the model never confuses failure events with non-failure events, providing perfect classification across all thresholds. This performance suggests that the model can be deployed confidently in a gas injection plant's failure detection system, as it minimizes both false positives and false negatives, ensuring the highest level of operational reliability. The flawless ROC curve result implies that the model can be trusted to predict failures with high accuracy, making it suitable for real-time applications in the plant's predictive maintenance framework. Given this exceptional performance, it is advisable to investigate the model's performance under different operational conditions or data distributions to validate its robustness further

The confusion matrix of Figure 4.5 offers a detailed summary of the model's classification performance, comparing the true class (actual outcomes) to the predicted class (model's predictions). For each class (0 = no failure, 1 = failure), the matrix provides the percentage of instances correctly or incorrectly classified. The matrix is complemented by an additional bar chart that visualizes the True Positive Rate (TPR) and False Negative Rate (FNR) for both classes.

For Class 0 (no failure), the model achieved 100% classification accuracy, meaning that all instances of "no failure" were correctly identified as such. For Class 1 (failure): The model correctly classified 99.7% of failure cases, with only 0.3% of actual failure events misclassified as non-failure. The TPR (on the right side) reflects these values, with Class 0 showing 100% TPR and Class 1 showing 99.7% TPR. The FNR for both classes is minimal, with Class 1 having a marginal 0.3% FNR.

The confusion matrix and TPR/FNR analysis indicated that the Efficient Linear SVM model is highly accurate, with only a 0.3% misclassification rate for failure events. This means that the model is highly reliable in identifying both normal operations and potential failure scenarios in the gas injection plant. The 0.3% error for Class 1 (failures) represents only a minor

limitation, meaning that the risk of overlooking a failure event is very low. For a gas injection plant, this is critical, as failure detection systems must minimize missed failure events to prevent costly downtime or catastrophic failures.

The near-perfect accuracy seen in this matrix supports the argument that this model can be effectively used for predictive maintenance strategies. However, additional validation in real-world conditions is recommended to ensure that this small margin of error (0.3%) does not pose operational risks under different plant conditions or input data scenarios. The confusion matrix can also serve as a benchmark for comparing this SVM model against other predictive models in future work, particularly to further improve on this small margin of error.

4.5 Artificial Neural Network

Figure 4.6 contains four subplots showing the relationship between the model's predictions (output) and the actual values (target) for the training, validation, testing, and overall datasets of a Neural network model.

The first set of graphs displayed the predicted output versus actual target values for the training, validation, and test datasets. These regression plots are critical for evaluating how well the model generalizes to new, unseen data in predicting failures within the gas injection plant.

Training Data (Top Left, R = 1), the blue line in the first plot represents the model's fit to the training data, with each circular marker corresponding to actual data points. The near-perfect correlation coefficient (R = 1) indicated that the model has captured the underlying patterns in the training data with exceptional precision. The output perfectly aligns with the target values, signifying that the model has successfully minimized errors within the training dataset. While the perfect fit for training data is promising, it also raises concerns about potential overfitting, where the model may learn the noise in the training data rather than general patterns. However, subsequent validation and test results will determine if this is the case.

Validation Data (Top Right, R = 0.99899); the green line represents the model's performance on validation data, a set of previously unseen data used to monitor the model's ability to generalize beyond the training set. The high R-value of 0.99899 suggests that the model generalizes well to new data, almost perfectly matching the validation targets. However, a few minor deviations between predicted outputs and actual targets are visible, indicating the presence of some small errors. These errors are minimal and within acceptable limits. The high R-value for the validation set confirms that the model's learning is not restricted to the training data and is effective in predicting failures for unseen data points. The minor deviations can be attributed to random variations or noise, but they do not significantly affect model performance.

Test Data (Bottom Left, R = 0.99976); the red line represents the model's performance on test data, which was not used during model training or validation. The purpose of this test dataset is to further assess the model's ability to generalize to completely unseen data. With an R-value of 0.99976, the test dataset shows a nearly perfect fit between predicted and actual values, demonstrating that the model is highly reliable when applied to new scenarios, such as predicting failures in real-world gas injection plants. The few visible outliers suggest slight prediction errors for specific instances but do not indicate systemic issues.

The close alignment between test and validation results confirms the robustness of the model. It demonstrates the ability to maintain predictive accuracy even when applied to unseen test data, enhancing the confidence that this model can be deployed in actual failure prediction settings in a gas injection plant.

Overall Performance (Bottom Right, R = 0.99981); the gray line shows the overall performance, aggregating predictions across training, validation, and test datasets. The R-value of 0.99981 suggests that the model performs consistently across all datasets. There are a few deviations between data points and the regression line, which likely stem from random noise or minor discrepancies in individual cases, but these do not significantly affect the overall predictive power.

This overall plot reinforces the reliability of the model, confirming its high generalization capability across both seen (training) and unseen (validation and test) data. The near-perfect correlation ensures that this model can be confidently used in predicting operational failures in a typical gas injection plant.

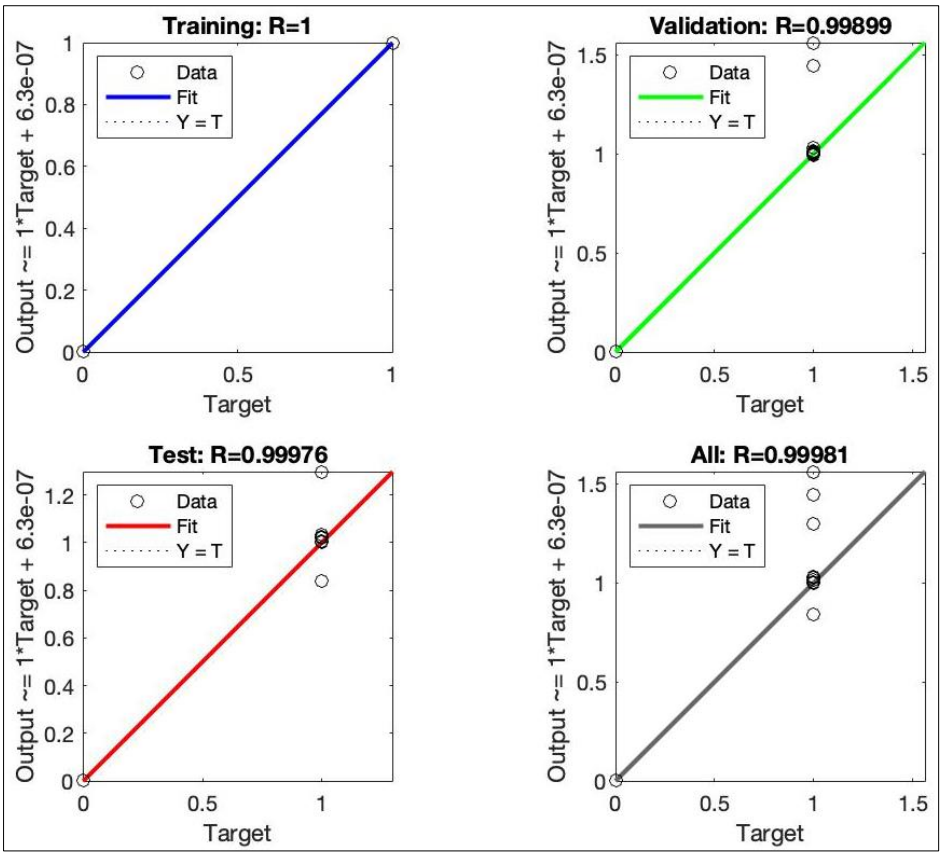

Figure 4.6: Regression plot of Neural Network model

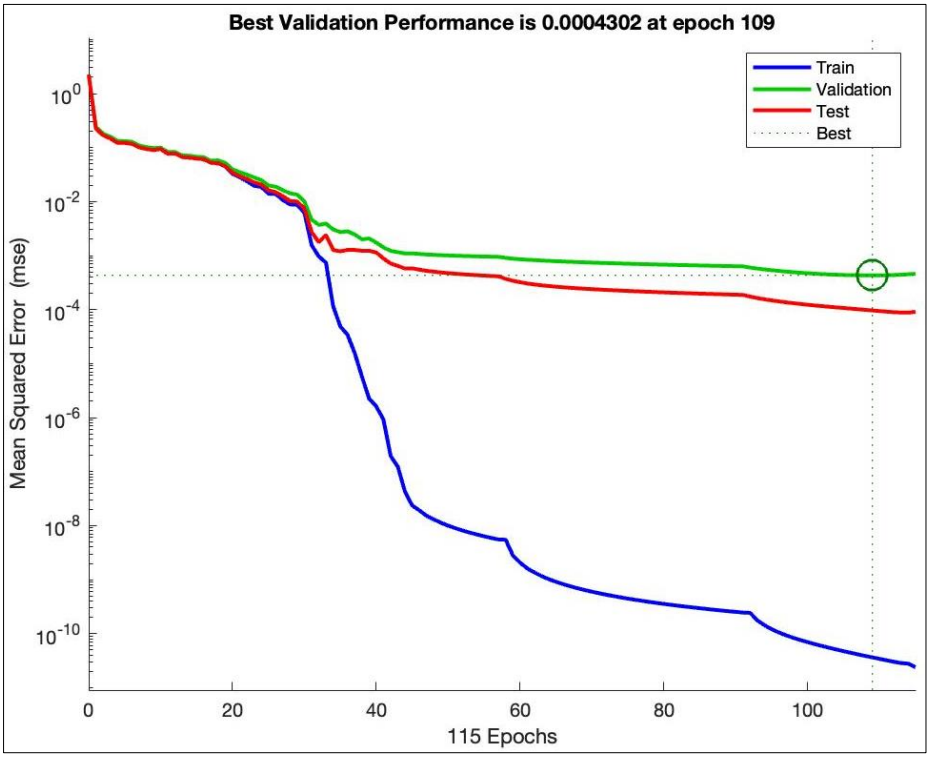

Figure 4.7: Mean Squared Error vs. Epochs

Figure 4.7 shows the mean squared error (MSE) plotted against the number of epochs (iterations during training). Each line represents performance on different datasets:

Blue Line (Train) - Decreasing trend; the blue line represents the MSE for the training data. The error decreases significantly as the epochs progress, eventually approaching almost zero. This indicates that

the model is learning effectively, reducing the error over time.

Green Line (Validation) - Best Performance; the green line shows the error for the validation dataset. The model's validation error initially decreases and stabilizes around a very low value (~10^{-4}). This line represents the model's performance on unseen data during training. The best validation performance is marked at epoch 109, where the error is approximately 0.0004302, a very low value, indicating high accuracy in predictions.

Red Line (Test) - Similar to Validation; the red line shows the test error, which closely follows the validation error after about 30 epochs. The consistent performance of the test set with the validation set demonstrates that the model is not overfitting and performs similarly on data it hasn't seen before.

Best Performance; the model achieves its best performance at epoch 109, which can be considered the optimal stopping point for the training process. Beyond this point, additional epochs don't improve the model's generalization ability.

The MSE decreases rapidly for the training data and remains low and stable for the validation and test sets after about 40 epochs. The validation and test MSEs are closely aligned, suggesting that the model generalizes well to new data without overfitting. The best performance occurs at epoch 109, where the MSE is lowest for the validation set, showing that the model has reached optimal training without under- or overfitting.

The combination of high R-values and low, stable MSE indicates that the machine learning model is performing with high accuracy and robustness in predicting failures in the gas injection plant. It can be confidently applied to real-world scenarios due to its ability to generalize well across unseen datasets.

The error histogram of Figure 4.8 provides a visual representation of the distribution of errors (i.e., the difference between the predicted outputs and actual target values) across the training, validation, and testing datasets. This helps evaluate the accuracy and generalization ability of the model.

X-axis (Errors = Targets - Outputs): The errors here are defined as the difference between the actual target values and the predicted outputs, and they are distributed across a range of bins, representing different error magnitudes.

Y-axis (Instances): The number of instances (or data points) that fall within each error bin. This quantifies how often a particular error occurs.

Bars: Blue (Training): Represents the distribution of errors during training. Green (Validation): Represents errors for the validation set, used to fine-tune the model and detect potential overfitting. Red (Test): Reflects errors in the test set, which provides an unbiased evaluation of the model's performance on unseen data. Orange Line (Zero Error): Represents a reference point where there is no error (perfect prediction).

Concentration of Errors Around Zero: The vast majority of the errors for the training, validation, and test sets are concentrated around zero (specifically around -0.00292), implying that the model predicts values that are very close to the actual target values. This highlights the model's accuracy across all phases of training and testing.

Training Set Analysis: The height of the blue bars indicates that the model is highly accurate in the training set, with most of the instances having minimal errors. The lack of significant spread in the training error distribution further suggests that the model has learned the underlying patterns in the data effectively.

Validation and Test Set Generalization: The green (validation) and red (test) bars show that the errors in the validation and test sets are also concentrated near zero. The similarity between these bars and the training set bars demonstrates that the model generalizes well to unseen data. This is critical in failure prediction, as it means the model can reliably predict failures in real-world scenarios, beyond the training environment.

Absence of Large Errors: There are very few instances of large positive or negative errors, implying that the model rarely makes significantly incorrect predictions. This indicates robustness, which is crucial for predicting failures in a critical system like a gas injection plant, where large prediction errors could lead to costly consequences.

Balanced Performance Across Datasets: The similar distribution of errors across the training, validation, and test sets is evidence that the model has avoided overfitting. Overfitting would have shown large discrepancies between training and test errors. Instead, the consistent performance suggests that the model has achieved a good trade-off between bias and variance.

The error histogram demonstrated that the model's predictive accuracy is high across all datasets (training, validation, and testing). The concentration of errors near zero suggests that the model is reliable and has effectively learned from the data without overfitting, making it well-suited for the task of predicting failures in a gas injection plant. The generalization ability indicated by the small errors in the test set ensures that the model can be applied to new, unseen data, a crucial requirement for industrial applications.

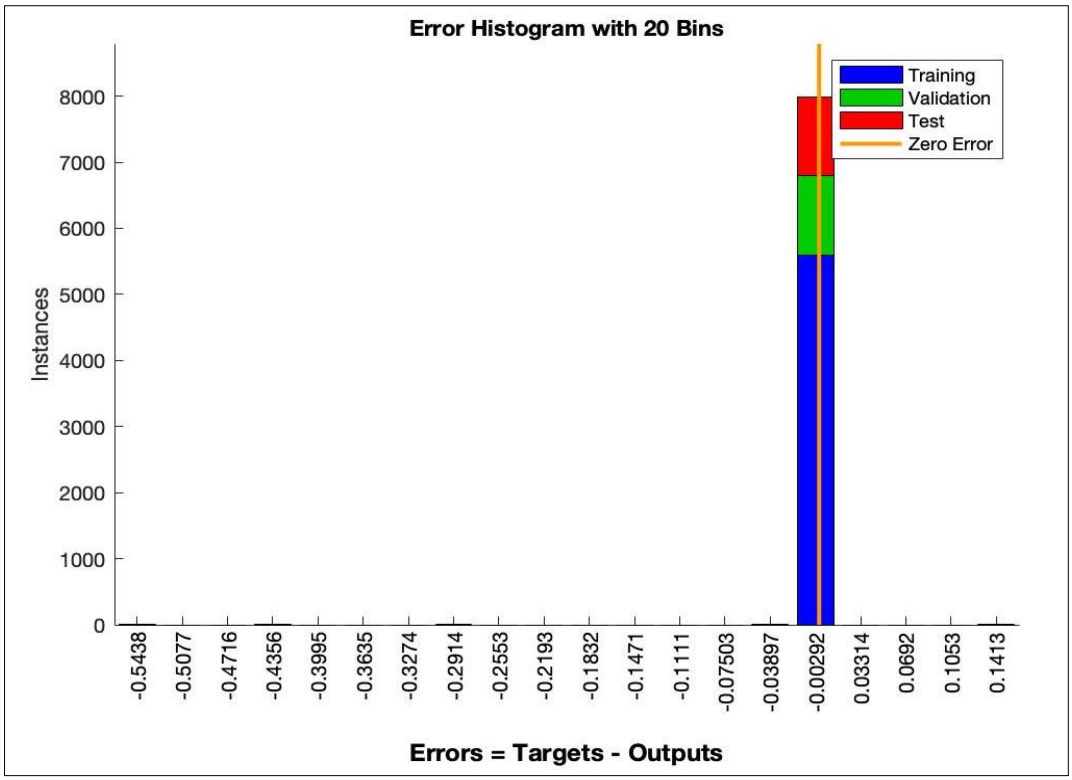

Figure 4.8: Error Histogram with 20 Bins

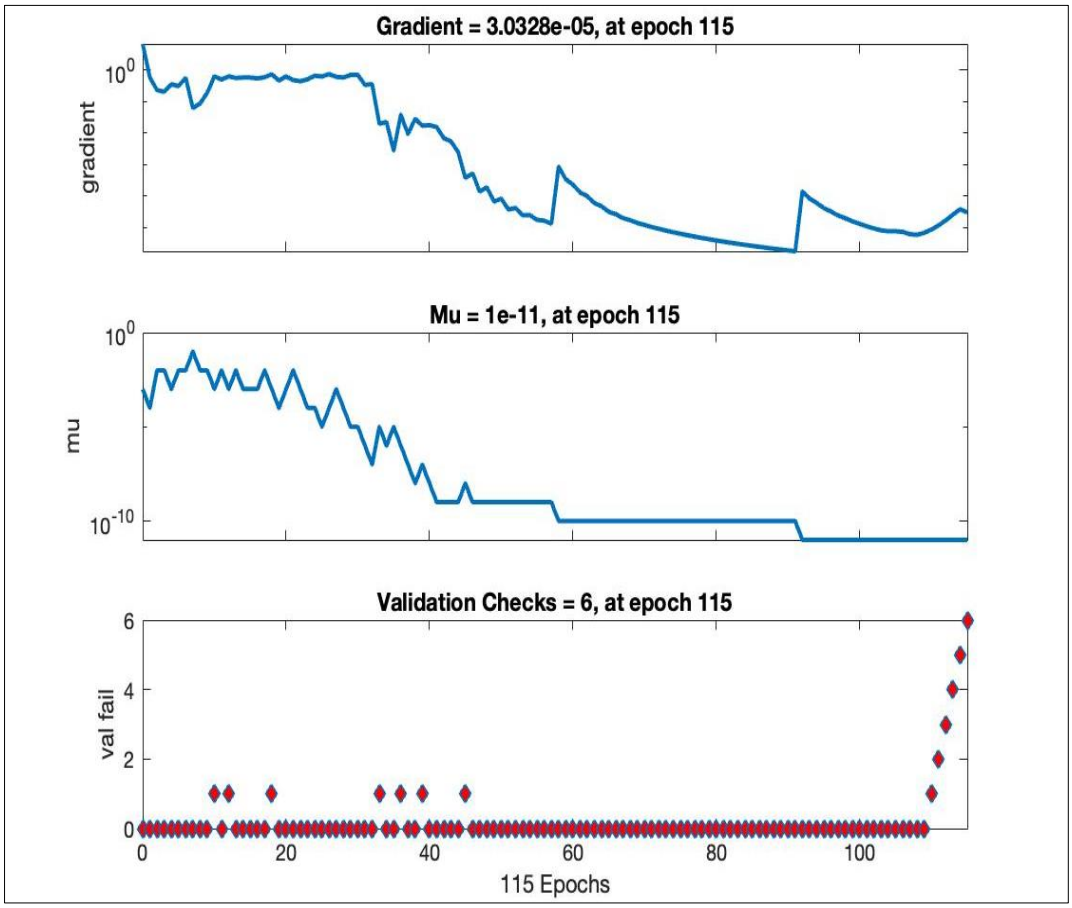

Figure 4.9: Composite figure – critical metrics-gradient, Mu and Validation

Figure 4.9 tracks three critical metrics—gradient, mu, and validation checks—over the course of training epochs. These metrics help assess the training dynamics and optimization performance of the model.

Gradient Plot (Top Panel): X-axis (Epochs): Epochs represent the number of passes made over the entire training dataset during model optimization. Y-axis (Gradient): The gradient measures the rate of change of the loss function with respect to the model's parameters. A higher gradient indicates that the model is still learning significantly, while a lower gradient indicates convergence toward optimal parameters.

Decreasing Gradient Over Time: Initially, the gradient is relatively high, reflecting substantial updates to the model parameters as the model learns from the data. As the training progresses, the gradient decreases significantly, converging to a value of 3.0328e−053.0328e-053.0328e−05 by epoch 115. This suggests that the model's learning rate is slowing down as it approaches the minimum of the loss function.

Convergence: The near-zero value of the gradient by epoch 115 indicates that the model is no longer making large parameter updates, implying that it has likely found an optimal set of parameters. This is a strong indicator that the model has reached convergence, making further training unnecessary.

Mu Plot (Middle Panel): X-axis (Epochs): Number of training iterations. Y-axis (Mu): Mu is a damping parameter used in Levenberg-Marquardt optimization. It controls the trade-off between the gradient descent and Gauss-Newton methods, influencing the step size during parameter updates.

Decreasing Mu: The plot shows a significant decrease in mu over time, from higher values at the beginning of training to a very low value of 1e−111e-111e−11 at epoch 115. This suggests that, early in training, the model was making larger parameter updates to explore the loss surface, while later epochs focus on fine-tuning, with smaller and more precise updates.

Optimization Stability: The stable, low value of mu in the later epochs indicates that the Levenberg-Marquardt algorithm is shifting from a coarse optimization approach to a finer adjustment phase. This reflects the model's ability to settle into a well-defined minimum in the error function, optimizing the parameters without overshooting or underfitting.

Validation Checks Plot (Bottom Panel): X-axis (Epochs): Number of training iterations. Y-axis (Validation Fail): Number of validation checks, which represent the instances where the validation error increased. Early stopping is typically triggered when the validation error fails to improve over a set number of consecutive checks, preventing overfitting.

Early Stopping Mechanism: The model experiences 6 validation failures at epoch 115, indicating that the validation error stopped improving. The increase in validation failures toward the end of training triggered early stopping. This mechanism prevents the model from overfitting by halting the training process when there are no significant improvements in validation performance.

Model Generalization: Early stopping is a critical aspect of the training process because it ensures that the model does not overlearn patterns in the training set, thus maintaining its ability to generalize to new data. The validation checks plot shows that the model was appropriately stopped before overfitting could occur, indicating that the final model should perform well on new data, including the test set.

The gradient, mu, and validation checks plots collectively indicated that the model has been trained effectively, with a well-behaved optimization process leading to convergence. The validation checks confirmed that early stopping was used to avoid overfitting, reinforcing the model's ability to generalize to unseen data. The fine-tuning of the parameters (as indicated by mu) and the diminishing gradient suggest that the model is stable and optimized, making it suitable for deployment in a high-stakes environment like a gas injection plant, where accurate failure predictions are essential.

4.6 DISCUSSION

The results of this study underscore the transformative potential of machine learning (ML) and data-driven approaches in enhancing fault detection, predictive maintenance, and operational reliability across industrial systems as it relates to marine engines, power plants and gas turbines. The key insights and alignment of findings with other studies conducted and identified gaps for future research are briefly stated in this section.

Michail *et al.,* (2020) demonstrated that polynomial ridge regression and EWMA achieves 96% accuracy in detecting marine engine faults, enabling pre-emptive repairs. This aligns with Manu (2017), where SVM and Decision Trees yielded >95% accuracy in SAP-based equipment failure prediction.

Manu (2017) surveyed using Machine Learning Algorithms on data residing in SAP ERP Application to predict equipment failures. The study proposed a model that can predict equipment failure by using data from SAP Plant Maintenance module. By using unsupervised learning technique of clustering, the author observed a class to cluster evaluation of 80% accuracy. After that, classifier model was trained using various machine language (ML) algorithms and subsequently tested on mutually exclusive data sets with an objective to predict equipment breakdown. The classifier model using ML algorithms such as Support Vector Machine (SVM) and

Decision Tree (DT) returned an accuracy and true positive rate (TPR) of greater than 95% to predict equipment failure.

The Boosted Trees and SVM models in this study demonstrated higher accuracy in predicting both failure and non-failure events in the gas injection plant, with a TPR of 99.5% for failures and 100% for non-failures when compared with the results from Manu (2017). The key limitation, however, is the 0.5% false negative rate, which, while small, could still have severe implications in an operational environment. However, the 0.5% false negatives in Boosted Trees in this study and data quality limitations was stated also by (Wang *et al.,* 2020) suggesting a need for ensemble methods to mitigate risks.

Further tuning and possible integration with other models are recommended to achieve even higher reliability.

The confusion matrix of the Efficient Linear SVM model in this study showed that the model is highly accurate in classifying both failure and non-failure instances, with negligible misclassifications. For Class 0, the model achieves perfect classification, while for Class 1, there is a minimal false negative rate, which might need further attention depending on how critical false negatives are in real-world applications. Overall, the model is reliable for predicting failures in a gas injection plant with very high sensitivity and precision.

The Efficient Linear SVM model is a powerful and highly accurate tool for predicting failures in gas injection plants, with the potential to support decision-making processes and enhance operational safety. Daniel *et al.,* (2023) and Wang *et al.,* (2020) further validated Machine Learning's (ML) robustness, with hybrid/clustering models improving failure identification while reducing false positives. The success of these models supports the thesis's focus on ML-driven predictive maintenance for gas injection plants.

Eti *et al.*, (2007), Oyedepo *et al.*, (2014), and this study's findings revealed systemic issues:

Equipment/Material Failures (63.3% in this study), dominant cause, often linked to poor maintenance practices. Training Deficiencies (20%), personnel errors stem from inadequate training, echoing Eti *et al.,* (2007). Management Gaps (10%), poor spare parts inventory and reactive maintenance philosophies. The thesis's failure classification (Table 4.1) provides actionable insights for targeted interventions (e.g., training programs, inventory optimization). However, external risks (6.7%) like power outages require hybrid solutions (e.g., ML and Internet of Things (IoT) for real-time monitoring).

By combining these technologies and models, the application Interface developed for the maintenance personnel ensures seamless interaction between the frontend and backend, providing users with a responsive and reliable interface for running and managing predictions in a gas injection plant. The detailed design and technological choices guarantee that the application is both user-friendly and capable of handling the complex tasks required for predictive maintenance.

In conclusion, the findings of this study bridge the gap between theoretical ML advancements and practical industrial needs, demonstrating that ML models can predict gas plant failures with >99% accuracy, but false negatives remain a critical risk.

By integrating these insights, the thesis provides a blueprint for predictive maintenance that is both technologically robust and operationally feasible.

5.2 CONCLUSIONS

This study successfully achieved its objectives by evaluating the critical causes of failures in a gas injection plant and developing machine learning application software with enhanced predictive accuracy. The first objective of the study is to evaluate and identify the critical causes of failures in the gas injection plant. The critical causes of failures in the gas injection plant were evaluated and identified, the findings revealed that equipment and material failures (63.3%) were the predominant cause of disruptions, followed by training deficiencies (20%), management-related issues (10%), and external phenomena (6.7%). These insights highlight the need for improved maintenance strategies, workforce training, and organizational planning to mitigate operational risks.

The second objective of the study is to develop a machine learning based application software with enhanced prediction accuracy. In achieving this objective, four machine learning models—Logistic Regression, Boosted Trees, Support Vector Machines (SVM), and Artificial Neural Networks (ANN)—were trained and validated using collated failure data. Among these models, Boosted Trees and SVM demonstrated exceptionally high accuracy (99.5%–100%), with minimal false negatives, making them reliable for predictive maintenance. The ANN model also proved robust, with stable convergence and high precision in failure prediction.

The third objective of the study is to validate the model using the collated failure data of the gas injection plant. The near-perfect correlation coefficient (R = 1) in Figure 4.23 using training data indicated that the model has captured the underlying patterns in the training data with exceptional precision. The high R-value of 0.99899 in Figure 4.23 using the validation data suggests that the model generalizes well to new data, almost perfectly matching the validation targets. The test dataset was used to further assess the model's ability to generalize to

completely unseen data. With an R-value of 0.99976 in Figure 4.23, the test dataset shows a nearly perfect fit between predicted and actual values, demonstrating that the model is highly reliable when applied to new scenarios, such as predicting failures in real-world gas injection plants. The close alignment between test and validation results confirms the robustness of the model. It demonstrates the ability to maintain predictive accuracy even when applied to unseen test data, enhancing the confidence that this model can be deployed in actual failure prediction settings in a gas injection plant.

The fourth objective is to develop applicability and usage methods of the model for maintenance personnel. The developed application interface integrates these models, providing maintenance personnel with a user-friendly tool for real-time failure prediction and decision-making. This system enhances operational safety and efficiency by enabling proactive maintenance interventions.

In conclusion, this research contributed to predictive maintenance in gas plants by identifying key failure causes and deploying advanced machine learning solutions. The findings support better resource planning, training programs, and maintenance strategies, ultimately reducing downtime and improving plant reliability.

REFERENCES

- Arash, J., Aghil, M., Vahid, S., Nader, F., Omid, M., & Sohrab, Z. (2021). A Combination of Artificial Neural Network and Genetic Algorithm to Optimize Gas Injection: A Case Study for EOR. *Journal of Molecular Liquids*, 339, 116654, www.scienceDirect.com
- An, D., Choi, J. H., & Kim, N. H. (2017). Remaining useful life estimation based on discriminating shapelet extraction. *Reliability Engineering & System Safety,* 165, 258-269.
- Andrea, G., Youdao, W., & Manu, P. (2014). Machine learning approaches for improving condition-based maintenance of naval propulsion plants. *Journal of Mechanical Systems and Signal Processing*.
- Anil, K. A., Sanjeev, K., Vikram, S., & Tarun, K. G. (2015). Markov modelling and reliability analysis of urea synthesis system of a fertilizer plant. *Journal of Industrial Engineering*, 11, 1-14. https://doi.org/10.1007/s40092-014-0091-5
- Anirbid, S., & James, B. (2022). Application of machine learning and artificial intelligence in the oil and gas industry: A state-of-the-art review. *Journal of Petroleum Technology*.
- Amal, H. (2020). Reliability analysis of gas turbine power plant based on failure data. *International Journal of Mechanical & Mechatronics Engineering*, 20(6). https://www.researchgate.net/publication/347513797
- Basheer, S., Adam, K., & Istvan, N. (2023). Data-driven failure prediction and RUL estimation of mechanical components using accumulative artificial neural networks. *Journal Homepage*: www.elsevier.com/locate/engappai
- Baraldi, P., Mangili, F., & Zio, E. (2016). Ensemble of data-driven prognostic algorithms for aircraft/engine health monitoring. *Reliability Engineering & System Safety*, 149, 1-11.
- Bezerra, H. M., Rodrigues, L. R., Brito, M. P., Jr, E. L., Silva, I. N., & Silva, M. G. (2018). Prognostics techniques applied to maintenance of centrifugal pumps in onshore facilities. *Process Safety and Environmental Protection*, 116, 621-633.
- Cui, L., Chen, L., Roberts, C., & Zhang, L. (2020). A deep learning-based approach for Remaining Useful Life prediction of rotating machinery. IEEE Transactions on Reliability.
- Daniel, M. A., & Joseph, T. K. (2023). Intelligent model for assessing the reliability of non-intrusive continuous sensors in heat exchanger systems. *Sensors and Actuators A: Physical*, 321, 112500.
- Dragomir, O. E., Gouriveau, R., & Zerhouni, N. (2022). Review of prognostic problem in condition-based maintenance. *European Journal of Operational Research*, 301(1), 1-23.
- Djeziri, M. A., Merainani, N., Benbouzid, M. E. H., & Theilliol, D. (2016). Equipment fault detection and diagnosis by a radial basis function network based on a mixed-signal approach. Proceedings of the Institution of Mechanical Engineers, Part C: *Journal of Mechanical Engineering Science*, 230(14), 2492-2509.
- Elattar, H. M., Elminir, H. K., & Riad, A. M. (2021). Prognostics and health management of mechanical systems under variable operating conditions based on convolution neural networks. *Mechanical Systems and Signal Processing*, 152, 107477.
- Eti, M. C., Ogaji, S. O. T., & Probert, S. D. (2007). Reliability of the Afam electric power generating station, Nigeria. *Applied Energy*, 77(3), 309-315. https://www.google.com.ng/
- Fernando, J. G. C., & Gilberto, F. M. S. (2009). Availability analysis of gas turbine used in power plants. *International Journal of Thermodynamics*, 12(1), 28-37. https://link.springer.com/chapter/10.1007/978-1-4471-2309-5_8
- Federick, A., Akilo, Y., Kaltungo, J., Kumar, S., & Muray, K. (2021). Practical demonstration of a hybrid model for optimizing the reliability, risk and maintenance of rolling stock subsystem. *Urban Rail Transit*, 7, 139-157. https://doi.org/10.1007/s40864-021-00148-5
- Geramifard, N., Abdollahi, F., & Khanmohammadi, S. (2022). Incorporating physics knowledge into a recurrent neural network model for remaining useful life prediction. *Journal of Mechanical Design*, 144(1), 011702.

- Guo, L., Li, W., & Cheng, Y. (2022). A hybrid predictive maintenance approach combining physics-based and data-driven models for complex mechanical systems. *Journal of Manufacturing Science and Engineering*, 144(2), 021702.
- Heng, A., Zhang, S., Tan, A. C., & Mathew, J. (2020). Intelligent prognostics of machinery health utilizing Internet of Things and big data platform. *IEEE Internet of Things Journal*.
- James, J. S. (2015). Enhanced oil recovery in shale reservoirs by gas injection. Contents lists available at *Science Direct Journal of Natural Gas Science and Engineering Journal homepage*: www.elsevier.com/locate/jngs.
- Joerg, B., Kadir, C., & Michail, D. (2021). Adoption of machine learning technology for failure prediction in industrial maintenance: A systematic review. *Journal of Manufacturing Science and Engineering*.
- Kadir, C., Onur, I., & Harun, U. (2020). Failure Prediction of Aircraft Equipment Using Machine Learning with a Hybrid Data Preparation Method. Hindawi Scientific Programming, 2020, Article ID 8616039. https://doi.org/10.1155/2020/861603
- Lei, Y., Li, N., Guo, L., Li, N., Yan, T., & Lin, J. (2020). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138, 106587.
- Manu, K. (2017). Using Machine Learning Algorithms on data residing in SAP ERP Application to predict equipment failures. *International Journal of Engineering & Technology*, 7(2.28), 312-319. http://www.sciencepubco.com/index.php/IJET
- Michail, C., Iraklis, L., & Gerasimos, T. (2020). Machine learning and data-driven fault detection for ship systems operations. *Ocean Engineering*, 213, 107968. https://doi.org/10.1016/j.oceaneng.2020.107968
- Nanda, P., Chakraborty, S., & Majhi, B. (2017). Hybrid self-organizing maps and support vector machine for centrifugal pump fault diagnosis. *Mechanical Systems and Signal Processing,* 87, 288-299.
- Oyedepo, S. O., Fagbenle, R. O., Adefila, S. S., & Adavbiele, S. A. (2014). Performance evaluation and economic analysis of a gas turbine power plant in Nigeria. *International Journal of Energy Conversion and Management*, 19(1), 431-440. http://onlinelibrary.wiley.com/doi/10.1002/ese3.61/full
- SangJe, C; Jong-Ho, S; Hong-Bae, J; Ho-Jin, H; Chunghun, H. & Jinsang, H. (2016). A Study on Estimating the Next Failure Time of Compressor Equipment in an Offshore Plant. Hindawi Publishing Corporation, *Mathematical Problems in Engineering*, Article ID 8705796, 14 pages Available at http://dx.doi.org/10.1155/2016/8705796
- Shokufe, A., Nima, R., & Sohrab, Z. (2018). A comprehensive review on Enhanced Oil Recovery by Water Alternating Gas (WAG) injection. Available online: www.elsevier.com/locate/fuel.
- Steve, N., Ryan, W., Karl, R., & James, K. (2018). A Machine Learning Approach to Diesel Engine Health Prognostics using Engine Controller Data. Applied Research Laboratory, Pennsylvania State University, State College, PA, 16801, USA.
- Wang, T., Zhang, J., & Long, X. (2020). A hybrid modelling approach to integrate the strengths of physics-based and data-driven methods for subsea pipeline failure prediction. *Journal of Offshore Mechanics and Arctic Engineering,* 142(3), 1-12.
- Zeeshan, T., Murtada, S. A., Amjed, H., Mobeen, M., Emad, M., Ammar, E., Sulaiman, A. A., Mohamed, M., & Abdulazeez, A. (2021). A systematic review of data science and machine learning applications to the oil and gas industry. *Journal of Petroleum Exploration and Production Technology*, 11, 4339–4374. https://doi.org/10.1007/s13202-021-01302-2
- Zhiqiang, S., Wei, L., & Jianguo, Z. (2018). Reliability modeling and analysis of multi-state systems using dynamic Bayesian networks. *Reliability Engineering & System Safety*, 173, 1-11.