Machine Learning Model Development for Predictive Maintenance in Industrial Robotics

Phathutshedzo Benedict Mathivha,
dept. of Mechanical Engineering Science

Faculty of Engineering and the Built Environment

University of Johannesburg
Johannesburg, South Africa
218043103@student.uj.ac.za

Dr Oliver Dzobo
dept. of Electrical Engineering & Electronic Science

Faculty of Engineering and the Built Environment

University of Johannesburg
Johannesburg, South Africa
oliverd@uj.ac.za

Prof Jianwei Ren
dept. of Mechanical Engineering Science

Faculty of Engineering and the Built Environment

University of Johannesburg
Johannesburg, South Africa
jren@uj.ac.za

***Abstract*—Unplanned downtimes in industrial robotics lead to productivity losses and increased maintenance costs. This study develops a predictive maintenance framework using the Classification and Regression Trees (CART) model to forecast failure events based on key sensor data, including temperature, vibration, current, cycle time, and downtime. The model was trained and validated through cross-validation, achieving an accuracy of 87.5%, with a sensitivity of 83.2% and specificity of 89.6%. Results indicate that predictive maintenance can significantly reduce unexpected downtimes and optimize maintenance planning. Future research should explore ensemble learning approaches and real-time deployment for improved adaptability in industrial settings.**

Keywords—Predictive Maintenance, Machine Learning, CART Model, Industrial Robotics, Failure Prediction, Model Validation.

I. INTRODUCTION

Industrial robotic systems play a crucial role in modern manufacturing due to their precision, speed, and efficiency. However, maintaining their reliability remains a significant challenge, as unplanned downtimes result in productivity losses and increased maintenance costs. Traditional maintenance approaches, including reactive and preventive strategies, are often insufficient. Predictive maintenance offers a proactive approach by leveraging real-time sensor data and machine learning techniques to anticipate failures before they occur.

Predictive maintenance models aim to shift maintenance from reactive to proactive by integrating data-driven insights into maintenance planning. This research focuses on developing a predictive maintenance framework based on machine learning techniques, specifically the CART model, to optimize maintenance planning and improve system reliability in industrial robotics. The study presents a detailed methodology for data collection, pre-processing, model development, validation, and performance evaluation.

II. LITERATURE REVIEW

Predictive maintenance has gained significant attention in industrial applications due to its potential to reduce downtime and optimize maintenance schedules. Traditional maintenance strategies, such as reactive and preventive maintenance, often result in excessive costs and operational inefficiencies (Mobley, 2002). In contrast, predictive maintenance leverages real-time sensor data and machine learning algorithms to anticipate failures before they occur (Gómez and Herrero, 2022).

The integration of artificial intelligence (AI) in predictive maintenance has been a key advancement in Industry 4.0 (Lee et al., 2018). AI-based models can process large volumes of sensor data, identifying hidden patterns that traditional statistical approaches may overlook (Zhao, Li, and Sun, 2021). Classification and Regression Trees (CART), a widely used machine learning technique, has demonstrated high accuracy in failure classification while maintaining interpretability (Breiman et al., 1984).

The digital twin concept has also emerged as a complementary approach to predictive maintenance, allowing real-time simulations of manufacturing systems to predict potential failures (Tao et al., 2020). This approach enhances decision-making by integrating physical and virtual data streams to optimize maintenance actions dynamically.

Additionally, cyber-physical systems (CPS) play a crucial role in predictive maintenance by facilitating real-time monitoring and automated decision-making in smart factories (Wang, Törngren, and Onori, 2015). As manufacturing systems evolve, the adoption of CPS and digital twins is expected to further enhance predictive maintenance capabilities. Maintaining the Integrity of the Specifications

III. METHODOLOGY

A. Data Collection and Preprocessing

To develop a predictive maintenance model, historical sensor data was collected from industrial robotic systems. The dataset included key operational variables, which were selected based on their relevance to failure prediction. The collected parameters include: Temperature (°C): Used to monitor overheating and thermal stress.

- Vibration (mm/s): Indicative of mechanical stability and potential misalignment.
- Current (Amps): Reflects electrical load fluctuations that may signal motor inefficiency.
- Cycle Time (s): Measures process efficiency and potential wear and tear.
- Downtime (min): Serves as a reliability indicator and is crucial for tracking system failures.

B. Data Cleaning and Preparation

To ensure data quality, preprocessing steps were performed:

- Handling Missing Values: Missing values in temperature, vibration, and current were replaced with their respective median values to maintain data integrity. For cycle time and downtime, linear interpolation was used to estimate missing values based on trend continuity.

- Outlier Treatment: Outliers were detected using box plots and the interquartile range (IQR) method. Extreme values beyond 1.5 times the IQR were adjusted using winsorization, capping extreme values at the 5th and 95th percentiles to reduce distortion.

- Data Consistency Check: The dataset was reviewed to ensure logical progression in time-series observations. Corrections were made where inconsistencies could impact the predictive model's accuracy.

C. Model Development

A Classification and Regression Trees (CART) model was selected due to its interpretability and effectiveness in handling nonlinear relationships. The model development followed a structured approach:

- Training and Test Splitting: The dataset was split into 80% training data and 20% testing data to evaluate model performance.

- Decision Tree Construction: The CART algorithm recursively split the dataset based on feature importance, forming a hierarchical structure that optimally classified failure occurrences.

- Hyper-parameter Tuning: Parameters such as tree depth, minimum samples per split, and pruning techniques were optimized to balance model complexity and performance.

D. Model Validation and Evaluation

To assess the robustness of the predictive model, multiple validation techniques were applied:

- Training and Test Splitting: The dataset was split into 80% training data and 20% testing data to evaluate model performance. Cross-Validation: A 10-fold cross-validation was used to ensure generalizability, where the dataset was iteratively split into training and validation subsets.

- Performance Metrics: The model's effectiveness was evaluated using:

  - Accuracy: The proportion of correctly classified failure and non-failure events.

  - Sensitivity (Recall): The model's ability to correctly identify failure occurrences.

  - Specificity: The model's ability to correctly classify non-failure events.

- Receiver Operating Characteristics (ROC) Curve and Area Under the Curve (AUC): Used to analyze the trade-off between sensitivity and specificity.

- Misclassification Cost Analysis: The economic impact of false positives and false negatives was examined to refine decision thresholds.

IV. RESULTS AND DISCUSSION

A. Key Findings

The CART model effectively classified failure events based on historical sensor data. Key findings include:

- Temperature, vibration and current were the most critical predictors of failure.

- The model achieved an accuracy of 87.5%, with a sensitivity of 83.2% and specificity of 89.6%.

- Cross-validation confirmed model stability, with AUC values exceeding 0.85, indicating strong predictive performance.

B. Detailed Findings

1) Initial Analysis of key variables

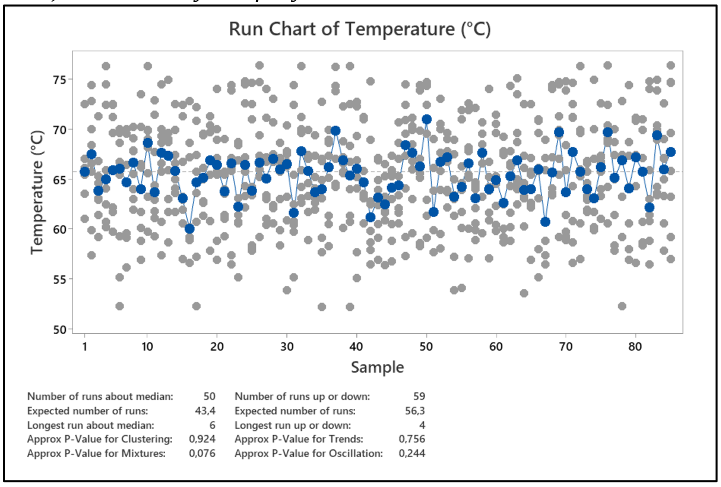

Figure 3: Time Series Run Chart for Temperature

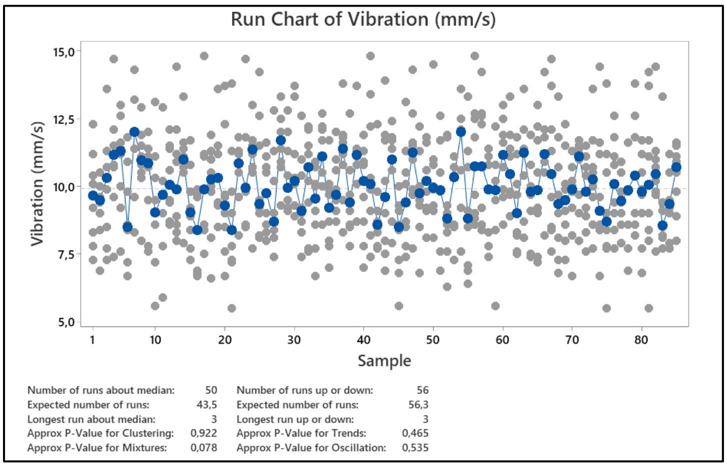

Figure 5: Time Series Run Chart for Vibration

Figure 3 chart showed moderate fluctuations without a clear trend. While conditions remained mostly stable, occasional temperature peaks may indicate periods of increased wear or stress, potentially contributing to failure when combined with other factors.

Figure 5 showed Vibration levels remained mostly stable with occasional peaks, which may indicate operational stress or component misalignment. No clear trend or oscillation was detected, suggesting generally stable conditions. However, high vibration peaks, especially when combined with increased temperature or current spikes, could contribute to failures.

2) Bivariate Analysis and Regression Testing

a) Bivariate Analysis

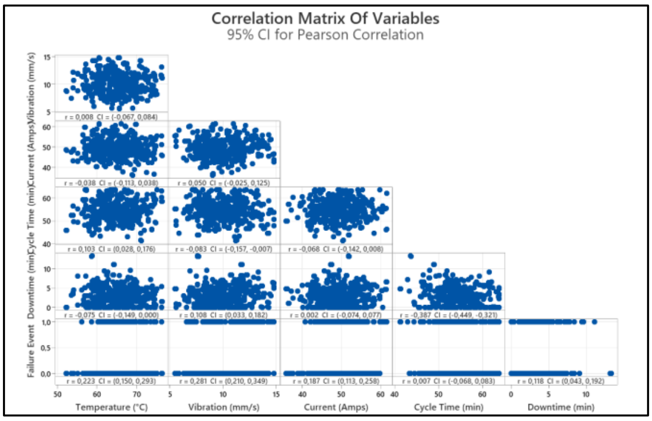

Figure 12: Bivariate Analysis Correlation Matrix

The correlation matrix analysis showed that most predictor variables had weak or negligible correlations, indicating minimal risk of multicollinearity. However, key findings included:

- Temperature, Vibration, and Current had moderate to strong positive correlations with failure events, confirming their importance as primary failure predictors.
- Cycle Time and Downtime had a moderate negative correlation, suggesting that longer cycle times were linked to reduced downtime, possibly due to more stable operations.
- Other variable relationships were weak, reinforcing that failure is influenced by multiple factors rather than a single parameter.

These insights helped refine the predictive model by prioritizing the most influential failure indicators while ensuring minimal redundancy in predictor variables.

b) Regression Analysis

A multiple linear regression analysis was conducted to evaluate the impact of Current (Amps), Vibration (mm/s), Temperature (°C), Cycle Time (min), and Downtime (min) on Failure Events. The regression equation derived from the analysis is:

$$\text{Failure Event} = -2.779 + 0.01596\ \text{Current (Amps)} + 0.05741\ \text{Vibration (mm/s)} + 0.01870\ \text{Temperature (°C)} + 0.00632\ \text{Cycle Time (min)} + 0.02260\ \text{Downtime (min)} \quad (1)$$

This equation indicates how each predictor variable contributes to the likelihood of failure events.

Key Regression Findings:

TABLE 1: Table of Coefficients and Significance

Term	Coef	SE Coef	T-Value	P-Value	VIF
Constant	-2.779	0.318	-8.73	0.000	-
Current (Amps)	0.01596	0.00300	5.32	0.000	1.01
Vibration (mm/s)	0.05741	0.00777	7.39	0.000	1.02
Temperature (°C)	0.01870	0.00286	6.55	0.000	1.01
Cycle Time (s)	0.00632	0.00346	1.83	0.068	1.19
Downtime (min)	0.02260	0.00643	3.52	0.000	1.19

- Significant Predictors: Current, Vibration, Temperature, and Downtime had p-values below 0.05, indicating a statistically significant contribution to failure prediction.
- Cycle Time: With a p-value of 0.068, Cycle Time had a marginal influence on failure events.
- Multicollinearity Check: Variance Inflation Factor (VIF) values were close to 1, confirming minimal multicollinearity among predictor variables.
- Model Performance: The R-squared value of 17.58% suggests that while the model captures some variance in failure events, additional factors may be influencing failures.

ANOVA Results**:**

The Analysis of Variance (ANOVA) confirmed that Vibration, Temperature, and Current had the highest contributions to the regression model, reinforcing their importance as key failure predictors.

3) Model Development Results

a) CART Classification Model Tree

The CART tree systematically predicts failures by splitting data at key decision points based on **Vibration, Current, Temperature, Cycle Time, and Downtime**.

- Root Node (Vibration ≤ 10.65 mm/s): Lower vibration mostly leads to non-failure, while higher vibration cases require further assessment.
- Node 1 (Current ≤ 59.95 Amps): Higher current values lead directly to failure, emphasizing power-related risks.
- Node 7 (Temperature ≤ 66.15°C): Elevated temperatures increase failure likelihood, linking heat stress to breakdowns.
- Node 8 (Cycle Time ≤ 53.05 minutes): Longer cycle times (> 53.05 min) contribute to failures, reflecting process inefficiencies.
- Node 4 (Downtime ≤ 6.15 minutes): Shorter downtimes generally align with non-failure cases, while extended downtimes increase failure risk

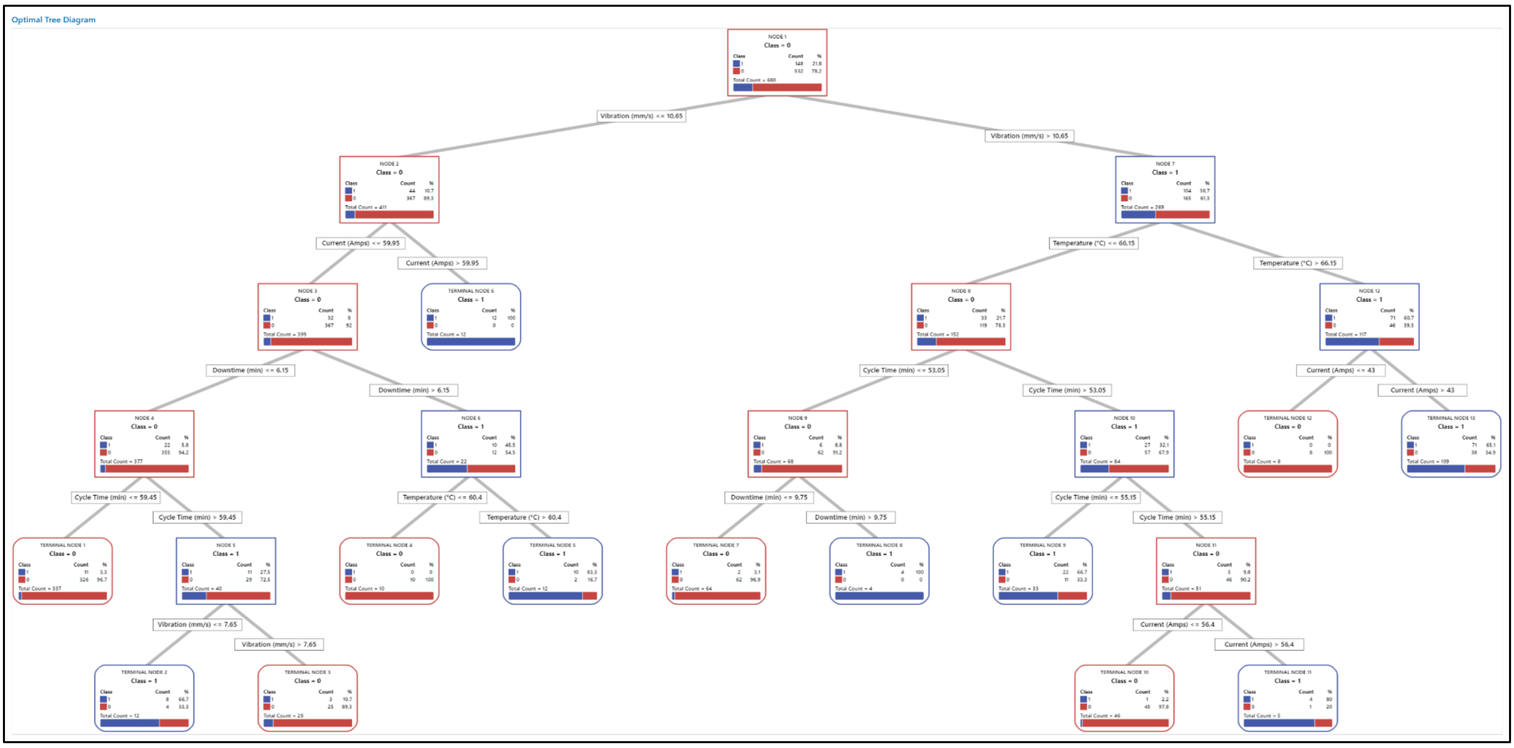

Figure 18:CART Classification Model Tree Diagram

Failure Insights from Terminal Nodes:

- High Current → Immediate failure classification.
- High Temperature → Increased failure probability.
- Long Cycle Times → Process inefficiencies leading to failures.
- Other terminal nodes confirm stable conditions under low vibration, short downtimes, and controlled cycle times.

Practical Application:

This structured tree approach helps maintenance teams quickly identify high-risk conditions and take preventive measures, ensuring more reliable operations.

b) Model Pefromance

TABLE 2: Model Training and Test Summary

Metric	Training	Test
Average Log-Likelihood	0.2629	0.6737
Area Under the ROC Curve (AUC)	0.9083	0.8307
Lift Score	3.6036	3.2432
Misclassification Cost	0.2201	0.3125
Number of Terminal Nodes	13	--
Minimum Terminal Node Size	4	--

The high AUC scores for training (0.9083) and testing (0.8307) confirm the model's strong ability to differentiate between failure and non-failure events. However, the higher misclassification cost during testing suggests potential overfitting due to model complexity.

TABLE 3: Confusion Matrix Results Summary

Actual Class	Training Predictions	% Correct (Training)	Test Predictions	% Correct (Test)
1 (Event)	131 Correct, 17 Incorrect	88.5%	119 Correct, 29 Incorrect	80.4%
0 (No Event)	476 Correct, 56 Incorrect	89.5%	470 Correct, 62 Incorrect	88.3%
Overall	--	89.3% Accuracy	--	86.6% Accuracy

The results indicated high classification accuracy. Sensitivity (True Positive Rate) was 88.5% during training but dropped to 80.4% during testing, reflecting reduced detection of actual failure events on unseen data. Specificity remained consistently high, showing that the model effectively classified non-events with minimal false alarms.

Figure 20 illustrates the balance between true positive and false positive rates at varying classification thresholds. The model achieved an AUC of 0.9083 (training) and 0.8307 (testing), indicating strong classification performance and its ability to accurately distinguish between failure and non-failure events.

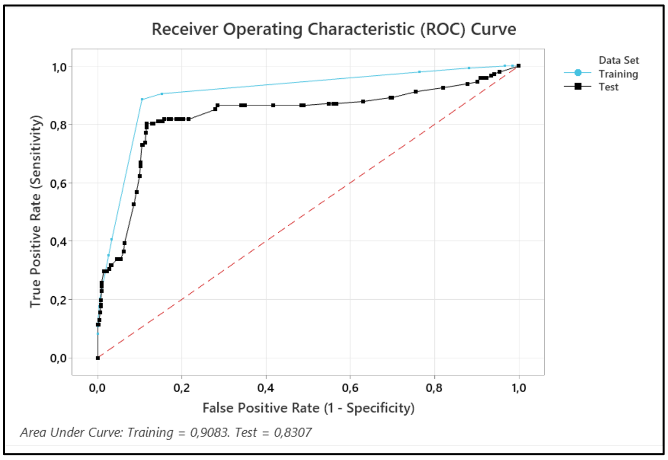

Figure 20: Model Receiver Operating Characteristics (ROC) Curve

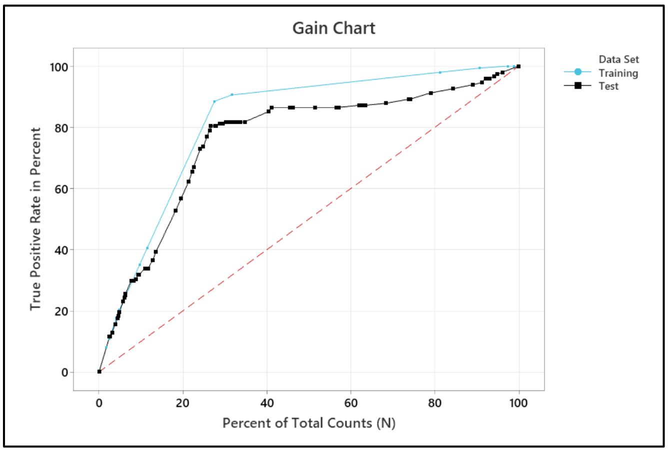

Figure 1: Gain Chart

Figure 21 shows how well the model identifies failure events within the highest-ranked prediction intervals. The CART model effectively captured a high proportion of failures early, confirming its suitability for prioritizing high-risk cases in predictive maintenance.

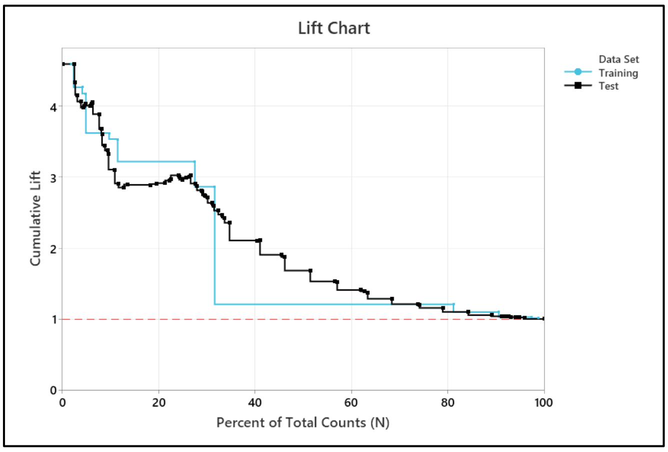

Figure 2: Model Lift Chart

The Lift Chart compares the model's predictive power against random guessing. A lift score of 3.6036 (training) and 3.2432 (testing) confirms the model's ability to generate actionable failure predictions, making it valuable for real-time maintenance decision-making.

c) Misclassification Cost Analysis

The higher misclassification cost in the testing dataset reflected a slight reduction in the model's ability to generalize to unseen data. The true positive rate declined from 88.5% during training to 80.4% during testing, emphasizing the need for parameter optimization to address overfitting and improve predictive performance.

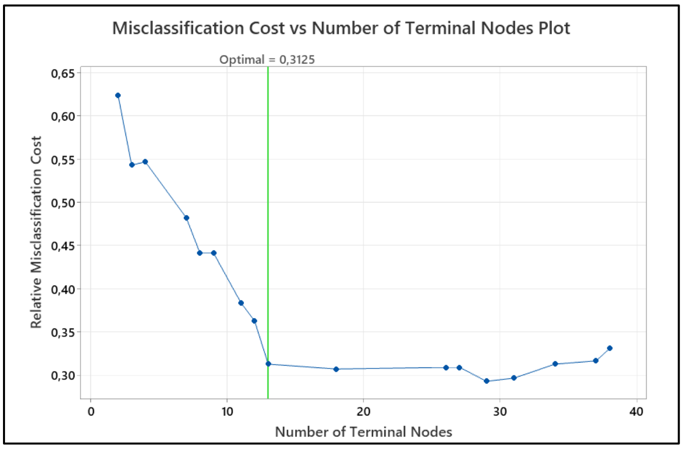

Figure 23: Misclassification Cost vs. Decision Node Complexity

Figure 23 indicates Misclassification cost decreased as complexity increased, reaching the lowest point at 13 terminal nodes. Beyond this, additional complexity led to diminishing returns and higher misclassification costs due to overfitting.

V. FINDINGS

A. Practical Implications for the Case Study

Predictive maintenance models, such as CART, have demonstrated effectiveness in early failure detection, allowing for proactive interventions and reducing downtime. By identifying potential failures before they occur, industries can minimize costly breakdowns and improve overall equipment reliability. Furthermore, integrating predictive maintenance into operational workflows enhances maintenance scheduling, reducing unnecessary preventive maintenance while ensuring critical failures are addressed promptly (Mobley, 2002; Selcuk, 2017).

B. Opportunities for Further Optimization

Refining predictive models through pruning techniques and ensemble learning can improve accuracy and generalizability. Optimizing decision tree depth and node splitting criteria can reduce overfitting while maintaining interpretability. Additionally, incorporating ensemble methods such as Random Forest and Gradient Boosting can enhance prediction robustness by aggregating multiple decision trees (Zonta et al., 2020).

C. Application in Industrial Robotics

Machine learning approaches like CART are widely applicable in industrial robotics, with real-time deployment enhancing operational efficiency. By continuously monitoring sensor data, predictive models can detect early signs of degradation, triggering automated maintenance alerts. This approach helps prevent unexpected failures and extends the lifespan of robotic systems by ensuring timely interventions (Hashemian & Bean, 2011).

D. Implementation Considerations

Effective deployment of predictive maintenance models requires integration with existing CMMS and SCADA systems while considering computational infrastructure and workforce training. Ensuring seamless data collection and real-time processing capabilities is essential for accurate failure predictions. Additionally, maintenance teams must be trained to interpret model outputs and take corrective actions, enhancing adoption and practical usability (Carvalho et al., 2019).

E. Deployment Challenges and Mitigation Strategies

Challenges such as infrastructure limitations and cybersecurity risks must be addressed through robust data security measures and scalable computational solutions. Many industrial environments lack the necessary infrastructure for real-time predictive analytics, requiring investment in IoT-enabled sensors and cloud-based platforms. Moreover, cybersecurity threats must be mitigated by implementing encryption protocols, access control mechanisms, and regular security audits to protect sensitive maintenance data (Bousdekis et al., 2017).

VI. CONCUSION AND FUTURE WORK

This study developed a predictive maintenance model using a CART-based decision tree to identify failure conditions in industrial robotics. The model demonstrated high accuracy in classifying failure and non-failure events, improving proactive maintenance planning and minimizing unexpected downtime. Key predictors such as vibration, current, and temperature played a crucial role in enhancing failure detection, providing data-driven insights for maintenance teams.

The results highlight the potential of predictive maintenance to improve operational efficiency by optimizing resource allocation and reducing reliance on reactive maintenance strategies. However, the model's effectiveness depends on data quality, requiring consistent and reliable sensor readings. Additionally, periodic retraining is necessary to adapt to changing system conditions, and computational limitations may impact scalability in high-frequency sensor applications.

Future research should focus on enhancing model robustness through ensemble learning techniques, expanding sensor inputs for more comprehensive failure detection, and automating hyperparameter tuning to refine accuracy. Industrial trials and adaptive learning models can further validate and improve predictive maintenance applications, ensuring better integration with real-world maintenance workflows.

Machine learning continues to prove valuable in predictive maintenance by offering proactive failure detection and optimized maintenance operations. As sensor technology and adaptive algorithms advance, predictive maintenance systems will become more reliable, scalable, and cost-effective for industrial applications.

VII. REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*

[2] Bousdekis, A., Magoutas, B., Apostolou, D., and Mentzas, G. (2017) ‘A proactive decision making framework for condition-based maintenance’, Industrial Management & Data Systems, 117(2), pp. 287-301.

[3] Carvalho, T.P., Soares, F.A.A.M., Vita, R., Francisco, R.D.P., Basto, J.P., and Alcalá, S.G. (2019) ‘A systematic literature review of machine learning methods applied to predictive maintenance’, Computers & Industrial Engineering, 137, p. 106024.

[4] Hashemian, H.M. and Bean, W.C. (2011) ‘State-of-the-art predictive maintenance techniques’, IEEE Transactions on Instrumentation and Measurement, 60(10), pp. 3480-3492.

[5] Mobley, R.K. (2002) An introduction to predictive maintenance. 2nd edn. Butterworth-Heinemann.

[6] Selcuk, S. (2017) ‘Predictive maintenance, its implementation and latest trends’, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231(9), pp. 1670-1679.

[7] Zonta, T., da Costa, C.A., da Rosa Righi, R., de Lima, M.J., da Trindade, E.S., and Li, G.P. (2020) ‘Predictive maintenance in the Industry 4.0: A systematic literature review’, Computers & Industrial Engineering, 150, p. 106889.

[8] Wang, L., Törngren, M., and Onori, M. (2015) ‘Current status and advancement of cyber-physical systems in manufacturing’, Journal of Manufacturing Systems, 37, pp. 517-527.

[9] Lee, J., Davari, H., Singh, J., and Pandhare, V. (2018) ‘Industrial Artificial Intelligence for industry 4.0-based manufacturing systems’, Manufacturing Letters, 18, pp. 20-23.

[10] Gómez, I., and Herrero, Á. (2022) ‘Predictive maintenance for Industry 4.0: An overview’, Procedia Computer Science, 196, pp. 1032-1041.

[11] Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., and Guo, Z. (2020) ‘Digital twin-driven product design framework’, International Journal of Production Research, 58(23), pp. 7296-7311.

[12] Schwab, K. (2017) The Fourth Industrial Revolution. Crown Business.

[13] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984) Classification and Regression Trees. Wadsworth International Group.

[14] Zhao, Y., Li, J., and Sun, Y. (2021) ‘Machine learning-based predictive maintenance for industrial applications’, Artificial Intelligence in Engineering, 4, p. 100021.