Predictive Maintenance for Industrial Equipment

1. Executive Summary

Problem Statement: Your manufacturing operations are subject to unplanned equipment downtime, leading to production delays, increased maintenance costs, and potential safety risks.

Proposed Solution: We propose building an AI model that predicts equipment failures before they happen. The model will analyze real-time sensor data from machinery to identify patterns that precede a breakdown, allowing for scheduled, proactive maintenance.

Expected Business Impact: This solution will reduce unplanned downtime by over 30%, extend equipment lifespan, and lower maintenance costs by optimizing repair schedules.

2. Introduction and Background

Traditional, time-based maintenance often results in either performing maintenance too early or too late. By leveraging machine learning and IoT sensor data, we can shift to a condition-based maintenance strategy, making your operations more efficient and reliable.

3. Project Scope and Deliverables

Detailed Scope:

Sensor Data Ingestion: Develop a system to ingest real-time sensor data (e.g., temperature, vibration, pressure, current) from up to 100 pieces of equipment.

Anomaly Detection Model: Train a deep learning model to recognize normal operating conditions and flag any deviations that indicate a potential failure.

Failure Prediction Model: Develop a separate model that, based on historical failure data, predicts the time-to-failure for critical components.

Real-time Dashboard: Create a dashboard that displays the health status of each piece of equipment, potential failure alerts, and maintenance recommendations.

Alert System: Integrate an alert system to send notifications via email or SMS to maintenance crews when a high-risk event is detected.

Non-Deliverables: We will not be responsible for installing new hardware sensors or for the physical maintenance of the equipment itself.

4. Technical Approach and Methodology

AI/ML Technique: We will use time-series analysis and deep learning (e.g., LSTM or Transformer models) to handle the sequential nature of sensor data. Anomaly detection will use techniques like Isolation Forest or Autoencoders.

Development Lifecycle:

Data & Infrastructure Setup (6 weeks): Configure data pipelines for real-time sensor data and set up a robust cloud infrastructure.

Model Training & Backtesting (10 weeks): Train the anomaly and prediction models on historical data and backtest their performance.

Deployment & Pilot (8 weeks): Deploy the models and dashboard, and run a pilot program on a select number of machines to validate the system.

Technology Stack: Python, TensorFlow/PyTorch, Apache Kafka for data streaming, Grafana for the dashboard, and a cloud environment (e.g., AWS IoT, Google Cloud).

Model Performance Metrics: The model will be evaluated on its Root Mean Squared Error (RMSE) for time-to-failure predictions and Precision/Recall for anomaly alerts.

5. Project Timeline and Milestones

Timeline: 5-7 months

Key Milestones:

Month 2: Sensor Data Pipeline Operational.

Month 4: Anomaly Detection Model Ready for Testing.

Month 6: Predictive Maintenance System Deployed for Pilot.

6. Team and Resources

Team Roles: 3-4 Data Scientists/ML Engineers, 1 Back-end Developer.

Man-days: Approximately 450-700 man-days.

Client Responsibilities: Provide access to sensor data, share historical failure logs, and identify key personnel from maintenance and IT to collaborate on the project.

7. Pricing and Payment Schedule

Total Project Cost: $90,000 - $150,000

Cost Breakdown:

ML & Development Time: 65%

Infrastructure & Software: 20%

Project Management & Overhead: 15%

Payment Terms: 25% upfront, 25% at infrastructure setup completion, 25% after successful model backtesting, and 25% upon final deployment.

8. Risk Assessment and Mitigation

Risk: Insufficient or noisy sensor data.

Mitigation: We will conduct an initial data quality and availability check. If data is incomplete, we may need to adjust the project scope or timeline to account for data collection efforts.