

Practical Predictive Maintenance for Railway Signalling
Cross-Domain Lessons from Agricultural AI

Abdulaziz mari

Abstract

Railway timetables depend on signalling availability. Many lines still rely on time-based or corrective maintenance, which creates unplanned downtime and delay minutes. This proposal develops a practical, student-scale approach to predictive maintenance for track circuits and axle counters. We use 12–24 months of fault logs and time-series signals from one line or region. After cleaning and resampling, we engineer simple, stable features (moving averages, variance, trend, recent-event counts). We train a logistic-regression baseline and a tree-based model (Random Forest/XGBoost). We evaluate precision, recall, F1, and lead time before failure, then calibrate scores and set alert thresholds with technician feedback to limit false alarms. Outputs are lightweight dashboards or CSV alerts. Expected outcome: earlier warnings for part of the failures and fewer reactive interventions.

Chapter 1

Introduction

Railway timetables depend on reliable signalling, especially on busy corridors [1]. Many networks still rely on time-based inspections or fix-after-failure maintenance. This practice creates unplanned outages and delay minutes [1, 2]. It also wastes field effort and budget.

This project builds a small, practical pipeline for predictive maintenance in railway signalling. The focus is on track circuits and axle counters in a single line or region. The dataset covers 12–24 months of fault logs and time-series status signals. We clean and resample the data, derive simple and stable features, and train two models: a logistic-regression baseline and a tree-based model (Random Forest/XGBoost). We evaluate precision, recall, F1, and lead time before failure. We then calibrate scores and set alert thresholds with technician feedback to reduce false alarms. The design follows recent rail studies that show value in robust features and calibrated operating points [3, 4].

Background

Signalling availability is a key driver of service reliability. Sector reports list predictive maintenance as a priority AI use case when paired with sound data governance and change management [1, 2].

Track circuits detect occupancy by monitoring electrical current across insulated rail sections; axle counters count wheel axles at section boundaries to infer if a section is free. Both subsystems emit events and time-series signals that reflect asset health (e.g., resets, supply current/voltage, receiver levels) [3]. These streams support simple, durable features such as moving averages, variance, trend, rate-of-change, and recent-event counts that help expose pre-failure patterns [3, 4].

Traditional rule-based checks can miss weak or gradual symptoms. Machine-learning models learn signatures from historical windows and produce a risk score that can be converted into an actionable alert. Studies on track-circuit cases and related assets report earlier warnings when features and thresholds are chosen carefully [3].

Field constraints remain. Data can be noisy or incomplete, and failures are rare, which causes class imbalance. Calibration and threshold tuning are needed to control false alarms [2, 4]. Models trained on one line or season may not generalize to others; periodic validation and simple re-

training help manage drift [3]. Operator trust matters: frontline staff need clear outputs and quick reasons they can act on. Lightweight dashboards and feature-importance summaries support adoption [2, 4].

Problem Statement:

As a maintenance engineer in a railway system, I have seen many times that we still use old methods to solve problems. Usually we wait for something to stop working before we fix it. Sometimes we check things on a certain level, although there is no problem yet. This working method is slow, and sometimes causes the unexpected delays, especially when a train suddenly stops or the signal system fails. From my experience it can be very disappointing and even dangerous in busy stations or main lines. I think this method must be converted into something fast and smart [1], [2].

Recently, I began to read more about new techniques that can help in this problem. One of the most interesting ideas is "future maintenance". It uses artificial intelligence (AI) and data analysis to check the device and detect small signals of errors. This means that we do not have to wait to break a system before we fix it. AI can first help us fix things, reduce delays and save money. I read some useful reports from large organizations such as UIC (International Union of Railway) and McKinese. He talked about how AI is one of the most useful equipment in the maintenance of railways today, especially for parts such as track circuits and axle counters, which are important in the signal system [2]–[4].

Nevertheless, many railway companies do not yet use this technology. I think this is because there are some challenges. In some places, for example, do not have enough digital equipment, or employees are not trained to use the AI system. In addition, it is not always clear how to use AI in the real world [3], [4]

Research Objectives

This research will explore how predictive maintenance can help improve railway systems by reducing breakdowns and delays. It will focus on how using smart technologies, like AI, can help find early signs of failure in railway signaling equipment.

The main goals of this study are:

1. Apparently to explain what future maintenance means and how it works in the context of the railways.

Future maintenance is a method that uses data and technology to predict when a machine or system fails. In railways, this means using sensors, previous error posts and other digital tools to check the status of real -time equipment. Instead of waiting for something to break, future maintenance gives a warning quickly so that the problem can be determined before the problem is spoiled. This goal will explain the concept of simple words to help everyone understand how it can be used in trains and tracks.

2. To compare future maintenance with the usual methods used today, and show the advantages and disadvantages of each.

Most railway systems still follow scheduled maintenance or wait until something breaks. These traditional methods can cause delays, high costs and unexpected security problems. This study will show how future maintenance is compared to these old methods - what it does better, and what are the challenges that can still be present. This will help to emphasize whether they are going to AI-based systems are worth trying.

3. To see how AI can collect and understand data from the tools such as the track circuit and the axle counter.

AI requires accurate data to work well. This part of the research will explain how systems such as track circuits and shaft meters generate useful data on train movement and track status. It will also see how AI can read this data quickly and analyze to find patterns that may indicate upcoming failure, making it easier for the maintenance teams to work early.

4. What to study the findings from UIC and McKinse and what they say about the future of AI in railway maintenance.

Both UIC and McKinse have released important reports discussing how AI is [3], [4].

Research Scope

This study focuses on predictive maintenance for railway signalling with track circuits and axle counters in one line or region. The goal is a small, practical pipeline that fits current workflows.

The time window is 12–24 months of continuous data from the selected line or region.

Data in scope include fault and event logs, occupancy states, reset records, supply current and voltage, receiver levels, and simple environmental context if available. Identifiers are anonymized.

The methods in scope are data cleaning and resampling, simple feature engineering (moving averages, variance, trend, rate of change, recent event counts), and two models: a logistic regression baseline and a tree-based model such as Random Forest or XGBoost. Class imbalance is handled with class weights or sampling. Scores are calibrated and alert thresholds are tuned with technician feedback.

Evaluation in scope includes precision, recall, F1, lead time before failure, and confusion matrices. Outputs are lightweight dashboards or CSV alerts that can be tested by maintenance teams.

Users and stakeholders are maintenance engineers, traffic control, and the course instructor. Outputs must be clear and easy to act on.

Out of scope are rolling-stock subsystems, redesign of interlocking rules, enterprise cybersecurity, network-wide financial modelling, national-scale real-time deployment, and safety certification.

Assumptions and constraints include granted data access, noisy labels, rare failures, limited compute (laptop-level), and a timeline limited to the course period. Cross-domain lessons from Assignment 1 guide feature robustness and calibration, but the study remains rail-specific.

Success criteria are earlier warnings for a share of failures with useful lead time, controlled false alarms at chosen thresholds, and a small, explainable workflow that fits existing practice.

Chapter 2: Literature Review

Paper 1: AI-supported autonomous water system on water efficiency and plant quality

This study examines the AI-manual smart irrigation system compared to traditional manual water to increase species of indigenous plant. The authors used an autonomous irrigation layout with a sensor, which monitored the earth controlled by the AI algorithm, and evaluated it against manual water in a controlled environment. Large conclusions contribute to the use of water to the AI-supported system, reduce waste and prevent problems such as soil conditions in increasing plant growth and health in relation to manual water. The AI system may be suitable for frequent changing conditions (such as seasonal variations), and can perform more efficient and continuous agriculture. Primary deficiencies that were limited to the use of laboratory -as a small scale -species of a plant, are usually unsure of other crops or environment in large areas. In addition, the costs and complexity of the infrastructure of AI-based irrigation systems were not addressed; Increasing such technology in cases of extensive agricultural use is a challenge that requires additional research (for example, with testing under different species and real field conditions)[5].

Paper 2: Picture-Class—CNNs for crop disease identification

This article presents an intensive teaching method to identify crop disease using data ideas. Researchers trained a certain nerve tight (CNN) on a large public dataset, with 54,306 images of healthy and diseased leaves from 14 crop species, including 26 separate diseases. The model achieved an impressive 99.35% classification accuracy by testing images, reflecting the possibility of smartphone diagnosis through leaf images. Major conclusions suggest that deep CNN can learn to identify plant diseases effectively from images without manual functional technology, potentially enable field diagnostics. However, the study also sheds light on significant boundaries. When tested in images of controlled training data, the model viewing fell significantly—the accuracy of images of different backgrounds or light areas fell to about 31%. This indicates an overfit to specific training distribution and emphasizes the requirement for more diverse training data to improve generalization. Another limit is that the current method only works in images of single, up-and-closed leaves against a simple background. Although controlled-environment results are excellent, the approach requires further development (additional data collection and model enlargement) to be reliable under natural field conditions[6].

Paper 3: Crop-yield prediction using deep neural networks

This article addresses the challenge of prediction of crops using AI, implementing deep nerve networks in a rich agricultural data set. Authors used a dataset from the crop challenge, including the cultivation of 267 hybrids in 2,247 locations (2008–2016) as well as cultivation with the same season and genotype information. They designed a deep neural network (DNN) model to predict the 2017 yield based on these historical data. The main findings suggest that DNN achieved better accuracy than traditional methods: the model's RMSE was about 12% of the average yield (with approximate weather input), outperforming linear regression and shallow neural approaches. DNN captured complex interactions between genotypes and environment; environmental factors (weather and location) had a greater impact on yield than genetics. Despite its accuracy, the study notes key weaknesses: the model acts as a “black box” with limited explanation; performance is sensitive to weather-forecast quality; and the approach needs large, high-quality data and compute, limiting immediate application[7].

Paper 4: An automatic orchard robot using deep learning

This research focuses on a practical AI-controlled solution for labor-intensive orchard harvesting. Authors developed a fruit-picking robot (Fuji apples) that integrates computer vision and robotic manipulation. A Single Shot Multibox Detector (SSD) detects apples in real time from camera images. Stereo vision estimates 3D positions, then a robotic gripper rotates and picks the fruit. Field tests showed the robot detected more than 90% of visible apples and picked each fruit in ~16 seconds while avoiding damage. Limitations include narrow scope and prototype status: tested on a specific tree architecture (V-shaped), performance may drop in dense canopy or with other cultivars; occlusions caused ~10% misses; and speed may be too slow for commercial throughput without multiple robots in parallel[8].

Paper 5: Weed detection with weak/unsupervised labeling from UAV images

This paper studies an AI method to identify weeds in line-planted crops using drone imagery while reducing manual labeling. The system scans UAV images for crop rows, then treats vegetation between rows as pseudo-labels for weeds to train a CNN. Evaluations in spinach and carrot regions showed accuracy close to fully supervised baselines ($\approx 1.5\%$ and 6% differences, respectively), demonstrating labor savings with high accuracy. The approach is specialized to row-crop situations; when rows are unclear or weeds grow too close to crops, labeling noise can appear and some flora may be missed[9].

Additional study 1 — Deep learning models for plant disease detection and diagnosis

A large-scale study trained CNNs on 87,848 leaf images (58 classes) and achieved very high accuracy on curated data. Results underline the value of deep representations but also the risk of domain shift to in-field imagery. [10]

Additional study 2 — Deep learning in agriculture: a survey

A comprehensive survey reviewed ~40 studies across vision, sensing, and yield prediction. It reports strong results on controlled datasets and recurring limits: domain shift, limited labels, and the need for practical workflows. Recommendations include richer datasets and realistic validation. [11]

Additional study 3 — Weed detection in soybean crops using ConvNets

Researchers trained ConvNets on labeled field images to classify weed types in soybean fields. Performance surpassed older pipelines and supports precision agriculture when data and labels are available; robustness across farms and seasons remains a challenge. [12]

Additional study 4 — Remote-sensing yield prediction with Deep Gaussian Processes

This work forecasts crop yield from satellite time series, outperforming simpler baselines and enabling in-season, scalable predictions. It complements genotype–environment models but still needs local calibration for deployment. [13]

Chapter 3: Research Methodology

I design a small, practical pipeline for predictive maintenance in railway signalling. The study focuses on track circuits and axle counters in one line or region. I use 12–24 months of logs and time-series signals and keep the workflow simple so technicians can try it in practice [1], [2].

Study area and data

The dataset includes fault and event logs, occupancy or clear states, reset records, supply current and voltage, receiver levels, and basic weather if available. I anonymize identifiers and split the period into train, validation, and test blocks in chronological order to avoid leakage [3].

Preprocessing

I clean missing values and obvious outliers. I align all signals to fixed time steps (for example 5 or 10 minutes). I build rolling windows so features summarize recent behavior. I standardize numeric features when needed and keep code simple and reproducible.

Feature engineering

I create simple, stable features that work across sites: moving averages, variance, trend or slope, rate of change, volatility, counts of recent faults or resets, and time since last reset. For day or night and weekday effects I add simple calendar flags. These features follow prior signalling case studies and help models see pre-failure patterns [3], [4].

Models

I train two models. First, a logistic regression baseline for clarity and easy troubleshooting. Second, a tree-based model such as Random Forest or XGBoost to capture non-linear effects. I tune a small set of hyperparameters with grid or random search on the validation set. I handle class imbalance with class weights or simple resampling.

Validation protocol

I keep a blocked time split: train on early months, validate on the next block, and test on the most recent block. If data is enough, I repeat the split across sections and report per-section and overall results. This shows whether the model generalizes across the line.

Evaluation metrics

I report precision, recall, and F1. I compute lead time before failure by measuring the time between the first alert and the failure window. I add a confusion matrix at the chosen operating point. I report per-asset and macro averages to avoid bias to large sections.

Calibration and thresholds

I calibrate model scores (for example Platt or isotonic) on the validation set. I select alert thresholds with technician feedback to control false alarms. I document two operating points: a conservative threshold for few false alarms and a sensitive threshold for higher recall [2], [4].

Operational output

I export a light dashboard or CSV with asset ID, risk score, threshold status, and short feature hints such as high variance or many recent resets. I keep explanations short and consistent so operators can act quickly and review alerts in routine meetings [2].

Risks and mitigation

Data may be noisy or incomplete; I mitigate with cleaning rules, robust features, and simple sanity checks. Failures are rare; I use class weights and careful thresholds. Models may drift across seasons; I schedule a simple re-validation every few months. Operator trust is critical; I keep outputs simple and collect feedback to adjust thresholds [1], [2], [4].

Ethics and governance

I respect data-access rules and anonymize IDs. I use the models to assist, not replace, safety procedures. I document limits clearly and avoid over-claiming results. Guidance from sector reports on data governance and change management is followed where applicable [1], [2].

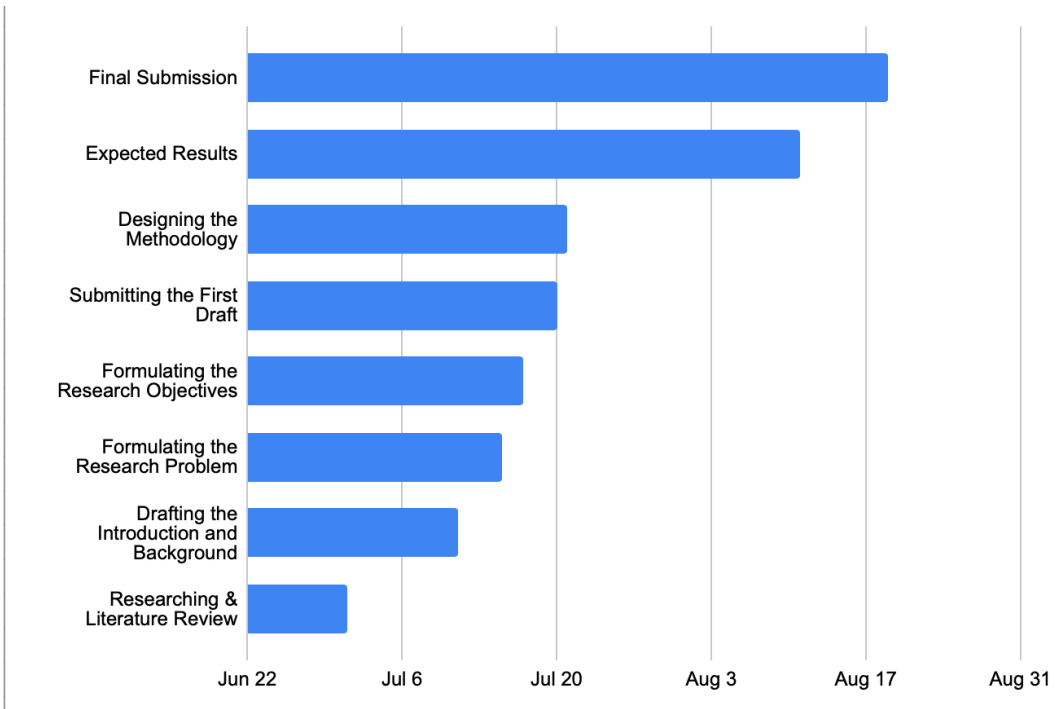
Expected outcomes

I expect earlier warnings for a share of signalling failures with useful lead time. The pipeline should reduce reactive interventions and provide a clearer view of risk for track circuits and axle counters. I expect one transparent baseline and one stronger non-linear model with calibrated scores and technician-tuned thresholds. At a conservative operating point I aim for balanced precision and

recall with a stable F1 and a manageable false-alarm rate that operators can review. I also expect lightweight operational outputs such as a small dashboard or CSV alerts and short explanations of why an alert fired. The project is student-scale, so I do not expect perfect detection; heavy data changes or configuration drift may lower accuracy and these limits will be documented.

Gantt chart

This section presents the project schedule. The Gantt chart shows each task’s planned start date and duration, from literature review to final submission. Bars start at the scheduled date and their length equals the number of days.



Project Gantt chart (July–August 2025)

References — IEEE

[1] UIC, *Artificial Intelligence – Case of the Railway Sector: State of Play and Perspectives*, 2021.

[2] UIC & McKinsey & Company, *The Journey Toward AI-Enabled Railway Companies*, 2024.

[3] A. Consilvio et al., “On Applying Machine Learning and Simulative Approaches to Railway Asset Management: The Earthworks and Track Circuits Case Studies,” *Sustainability*, 12(6):2544, 2020. doi:10.3390/su12062544.

- [4] J. Liu, Y. Xu, and L. Wang, "Fault Information Mining with Causal Network for Railway Transportation System," *Reliability Engineering & System Safety*, 220:108281, 2022. doi:10.1016/j.ress.2021.108281.
- [5] Oğuztürk, G. E., Murat, C., Yurtseven, M., and Oğuztürk, T., "The effects of AI-supported autonomous irrigation systems on water efficiency and plant quality: A case study of Geranium psilostemon Ledeb," *Plants*, 14(5), 770, 2025. <https://doi.org/10.3390/plants14050770>
- [6] Mohanty, S. P., Hughes, D. P., and Salathé, M., "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, 7, 1419, 2016. <https://doi.org/10.3389/fpls.2016.01419>
- [7] Khaki, S., and Wang, L., "Crop Yield Prediction Using Deep Neural Networks," *Frontiers in Plant Science*, 10, 621, 2019. <https://doi.org/10.3389/fpls.2019.00621>
- [8] Onishi, Y., Yoshida, T., Kurita, H., Fukao, T., Arihara, H., and Iwai, A., "An automated fruit harvesting robot by using deep learning," *ROBOMECH Journal*, 6(1), 13, 2019. <https://doi.org/10.1186/s40648-019-0141-2>
- [9] Bah, M. D., Hafiane, A., and Canals, R., "Deep learning with unsupervised data labeling for weed detection in line crops in UAV images," *Remote Sensing*, 10(11), 1690, 2018. <https://doi.org/10.3390/rs10111690>
- [10] Ferentinos, K. P., "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, 145, 311–318, 2018. <https://doi.org/10.1016/j.compag.2018.01.009>
- [11] Kamilaris, A., and Prenafeta-Boldú, F. X., "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, 147, 70–90, 2018. <https://doi.org/10.1016/j.compag.2018.02.016>
- [12] dos Santos Ferreira, A., Freitas, D. M., da Silva, G. G., Pistori, H., and Folhes, M. T., "Weed detection in soybean crops using ConvNets," *Computers and Electronics in Agriculture*, 143, 314–324, 2017. <https://doi.org/10.1016/j.compag.2017.10.027>
- [13] You, J., Li, X., Low, M., Lobell, D., and Ermon, S., "Deep Gaussian Process for Crop Yield Prediction Based on Remote Sensing Data," *Proceedings of AAAI*, 2017.