2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

TOWARDS DATA-DRIVEN METRO RAIL MAINTENANCE FOLLOWING THE MLOPS PARADIGM

Oihana Garcia, Kerman López de Calle, Francisco Javier Díez, and Egoitz Konde
TEKNIKER, Iñaki Goenaga 5, Eibar 20600, Spain

Abstract

Railway maintenance, particularly in curved sections, is a complex and costly operation requiring effective solutions to minimise wear. Machine Learning (ML) models were developed to monitor curved tracks using accelerometer data for lubrication level prediction, sensor fault detection, and outlier analysis. To support this, a monitoring system designed within a Machine Learning Operations (MLOps) framework was implemented. While vibration alone proved inconclusive in predicting lubrication levels, a significant increase in outliers during lubrication system inactivity indicated potential rail issues. As a result, the solution offers near real-time insights, helping end users make data-driven maintenance decisions by presenting model outcomes.

Introduction

The maintenance of railways is a critical and costly operation. The Rail Market Monitoring (RMMS) report provided by the European Commission (2023) indicates that the EU-27 spent approximately €41.8 billion on rail infrastructure in 2020, with 25% of this sum allocated specifically to maintenance.

Among the critical aspects of subway system maintenance, rail lubrication is a key factor. Proper lubrication reduces friction between train wheels and rails, minimising wear, energy consumption, and noise. Studies have shown that effective rail lubrication significantly prolongs rail life and improves energy efficiency, particularly in urban transit systems (Wilson, 2006; Belikov et al., 2022).

In curved sections, the importance of lubrication becomes even more pronounced. The presence of curves in a rail system can result in increased lateral forces, leading to accelerated wear and higher risk of derailment if not properly managed (Wilson, 2006; Belikov et al., 2022). This increased pressure can also accelerate the wear of the track and joints and cause deformations that, if not detected in time, could compromise the safety of the train. In addition, when trains brake, the contact between wheels and rails becomes much stronger, which causes faster wear. This is particularly problematic in tight curves, where the extra pressure from the train increases rail damage. Railside lubrication in these scenarios has been shown to effectively reduce wheel flange wear and minimize these forces, contributing to the overall safety and longevity of the rail infrastructure (Song et al., 2024).

Experts and scholars have explored this issue extensively, focusing on material wear models, profile smoothing algorithms, and local contact theories. These efforts underscore the importance of targeted lubrication strategies to manage the challenges posed by wheel-rail interactions (Song et al., 2024).

This study focuses on supporting maintenance decisions by analysing rail and lubrication systems using Machine Learning (ML) models. The use case involves a lubrication system installed on a subway rail just before a curve, a critical area where wheel-rail wear is typically more severe.

Specifically, the study investigates how lubrication impact in curved sections can be assessed using accelerometer data and advanced analytics. Sensors integrated into the rails collect vibration data. Through the use of an end-to-end Machine Learning Operations (MLOps) pipeline, the study enables efficient model development, allowing for continuous updates and future additions as required.

The objective of this integration is to bridge the gap between data collection and actionable information to improve maintenance strategies in the metro system.

Setup

The implemented setup is designed to monitor and analyse the lubrication and behaviour of subway rails. This chapter describes the physical system installed on the subway, the data acquisition system, and the model deployment process.

Physical system

The physical hardware components are installed in a subway system situated within an urban city centre tunnel. These sensors and acquisition systems were integrated during nighttime hours when trains were not in operation, ensuring safety and avoiding service interruptions.

Accelerometers are the sensors installed in the subway system to monitor vibrations. Specifically, six Brüel & Kjær type 4508 accelerometers were placed along the curve to measure the interaction between the train wheels and the rails. They are widely used in railway applications for vibration monitoring (Zeng et al., 2024), detect-

ing cracks, and identifying other irregularities that could compromise rail integrity (Shafique et al., 2021).

The monitoring system complements the hardware with a dedicated software solution developed within a monitoring box to manage data capture as trains approach the measurement point. This software is structured into several key modules:

- Trigger Module: Monitors accelerometer AC1 (see Figure 1) every second. If the measured value exceeds 20 m/s^2, it initiates a 30 second data acquisition. This duration is sufficient to capture the train's passage across all installed sensors.
- Acquisition Module: Collects signals, merges the data into a single MATLAB file (.mat), and ensures acquisition at the correct frequency using National Instruments APIs. These APIs provide programmatic access to interact with hardware and software, enabling reliable data acquisition from connected sensors.
- External Connection Module: This module first stores the files locally on the acquisition system PC before uploading them to a cloud database. Transferring the data to the cloud ensures accessibility and places it in an easily reachable location for further analysis. Additionally, the module verifies successful data transfer and deletes local files upon confirmation to prevent storage issues on the acquisition system.

Lubrication system At the same location, a third-party lubrication system was installed, featuring four lubrication points. This system applies a controlled amount of lubricant directly to the rail, which is then spread by passing trains to ensure even coverage along the curve.

The lubrication dosage was modified throughout the duration of the study. This was done following the schedules presented in Table 1 which includes the the range of dates and the percentages of lubrication applied by the lubricators during these periods. The dosage percentage refers to the total volume (in cm^3) dispensed to each train passing by.

Table 1: Lubrication dosage details by date

Dosage (%)	Flow rate (cm^3/train)	Dosage Change Date
100	0.2	2023/06/13 07:30
200	0.4	2023/07/04 14:44
300	0.6	2023/09/19 00:00
0	0	2023/10/04 17:04
200	0.4	2023/11/03 11:16
0	0	2023/12/15 06:40
200	0.4	2023/12/20 08:42

Sensor placement and distances Figure 1 illustrates the placement of the installed components before the curved rail section. Trains on this track operate in a single direction, allowing the lubricant applied by the lubricators to spread consistently along the rail in the same direction.

The accelerometers were installed to monitor rail vibrations in relation to lubrication application. They were placed at 3 or 1.5 meter intervals along the track, while the lubricators were positioned 0.5 meters apart.

End-to-End Machine Learning Pipeline

This section describes the workflow designed to support the maintenance system, developed based on the MLOps paradigm. As shown in Figure 2, the workflow includes the three core phases of MLOps (John et al., 2021), which are detailed below: data acquisition, modelling and release.

Data Acquisition The data acquisition system implemented in this study uses an event-driven approach to ensure timely data collection and processing, minimising latency.

The data retrieval process begins with a trigger linked to the AC1 accelerometer, as described in the previous section. Once activated, the system captures a 30 second data window from all installed accelerometers. The collected data is then uploaded from the monitoring box to a secure cloud database, specifically to Azure Blob Storage, an object storage service. This cloud-based architecture provides flexibility, scalability, and ease of access for downstream tasks, accommodating the varying demands of real-world operations (Wang et al., 2019).

Upon detecting the arrival of a new file in the storage,

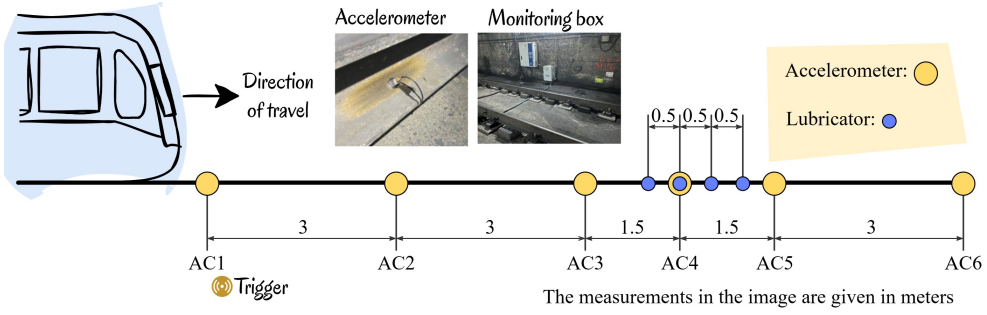


Figure 1: Sensor placement in a curved section of the subway rail system

preprocessing begins. A serverless computing function, implemented using Azure Functions, is triggered to process the data. This event-driven mechanism eliminates the need for dedicated hardware or complex orchestration systems, simplifying the overall implementation. The serverless architecture also ensures scalability and fault tolerance, making it well-suited for handling varying workloads (Shojaee Rad and Ghobaei-Arani, 2024).
Key Performance Indicators (KPIs), such as peak-to-peak amplitude, root mean square (RMS), and mean values, are computed during the preprocessing stage. These KPIs simplify the raw signal data, converting it into meaningful indicators that can be effectively analysed. For example, RMS quantifies the signal's power, providing a measure of overall activity, while peak-to-peak amplitude highlights extreme variations, which can indicate anomalies such as excessive vibration or noise. The computed KPIs are stored as CSV files in a separate container within Azure Blob Storage. These processed files are accessible for further analysis, visualization, or integration into ML pipelines.

Modelling and Release An end-to-end MLOps platform was used for all ML and MLOps-related processes. In the MLOps field, a wide range of tools support various aspects of the machine learning lifecycle. Moreschi et al. (2023) identified 84 MLOps tools, including both end-to-end platforms and specialised tools focused on specific subcategories. Each of these tools offers unique advantages, depending on the needs and context of a project. In this study, Azure Machine Learning (AML) was selected as the primary platform, primarily due to its compatibility with the existing Azure ecosystem, which is already in use for data storage and serverless capabilities. Additionally, using an end-to-end platform like AML provided significant benefits by streamlining the implementation process with built-in tools and features for experimentation, training, evaluation, versioning, and deployment, thus simplifying the workflow.
The MLOps principles serve as a guide to best practices in model lifecycle management (Kreuzberger et al., 2023). This implementation closely adhered to the following principles:

- **CI/CT/CD Automation:** Continuous Integration, Continuous Training, and Continuous Deployment were implemented using pipelines in Azure DevOps and AML. Each time new code is merged into the main branch, the code is built and tested and then uploaded to AML for model training. Once the training is complete, the model is deployed to an AML endpoint. AML's built-in tools simplify this process by facilitating database connection, storing model metrics, and supporting advanced deployment strategies. For instance, the blue-green deployment strategy was used to ensure zero downtime, keeping the models operational and preventing data loss.

- **Continuous ML Training:** The model is automatically retrained on a monthly basis with updated data. This process follows the same CI/CT/CD pipelines established for previous development and deployment cycles.

- **Continuous Monitoring:** To monitor input data, serverless functions were developed for each model, following a modular design where each function handles a specific task. Once KPIs are computed, these functions trigger the deployed models via an API to generate predictions. The results, along with the KPIs, are stored for later visualisation in dashboards like Power BI, providing end users with decision-making insights.

- **Workflow Orchestration:** An event-driven approach orchestrates each stage of the workflow. Data acquisition, preprocessing, and model inference are performed to generate near-real-time results every time a train passes. Additionally, pipelines are used to orchestrate the retraining and deployment phases, triggered automatically on a monthly basis or by the developer when new models or model updates are required. Finally, Power BI dashboards display the KPIs and model results to the end user.

- **Reproducibility and Versioning:** Reproducibility is ensured through the versioning of data, models, and code. Metadata for every training job and endpoint call is saved, providing traceability for compliance, auditing, and debugging purposes. This guarantees that experiments can be replicated with identical results.

- **ML Metadata Tracking and Logging:** Throughout the model lifecycle, metadata tracking and logging are essential. Information such as training date, duration, parameters, performance metrics, execution messages, and the data and code used in each model iteration is logged, enabling thorough tracking for every cycle.

As shown in Figure 2, the concept of MLOps incorporates multiple domains, integrating elements of data engineering (blue section), ML and DevOps (yellow section). By connecting these areas, MLOps enables a continuous, end-to-end workflow that is consistently updated.
Moreover, this approach is cost-effective due to its pay-per-use model. Serverless functions incur costs only when data is processed, and in the Azure Machine Learning (AML) environment, training and endpoints are also billed based on usage. Training is charged only when active, and AML's *batch* endpoints are only activated when processing requests, minimizing idle costs.
This setup minimizes expenses but introduces slight delays in triggers, training and API responses, typically a few minutes. However, predictions remain reliably available

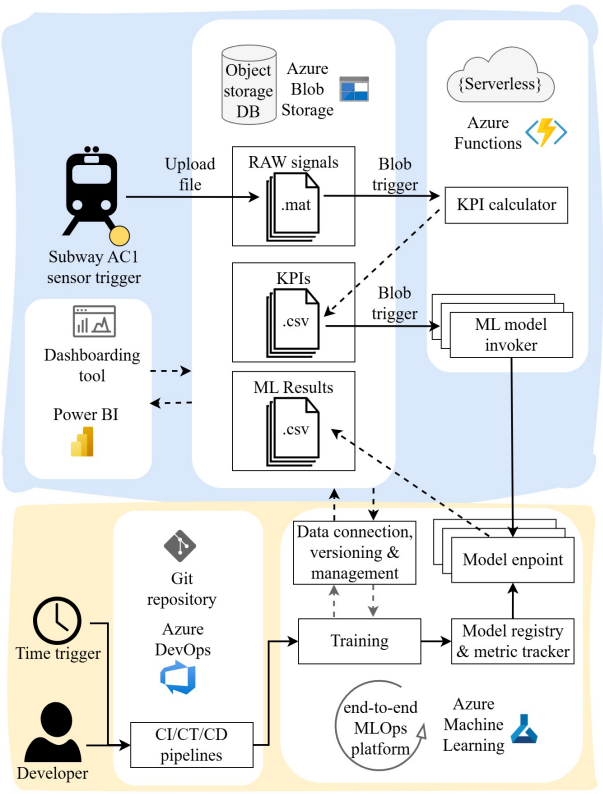


Figure 2: Logical Architecture

within 15 minutes of a train passing the curve. This trade-off between response time and cost efficiency aligns with the project's requirements, where a quarter of an hour delay is acceptable.

This integration brings the metro system closer to becoming a smarter, data-driven operation that supports maintenance and optimises resources.

Results and Discussion

This section presents the findings from the collected data, insights derived from the developed ML models, and their implications for improving rail maintenance strategies. The results are organized into key areas, including sensor reliability, anomaly detection and lubrication evaluation.

Sensor Health and Reliability

The analysis began by evaluating the functionality and reliability of the accelerometers. It was observed that Accelerometer 6 (AC6) had begun to record only noise, making its data unreliable. Similarly, Accelerometer 5 (AC5) displayed increasing signs of malfunction, with noisy signals during data collection.

Figure 3 presents a box plot showing the mean values for each sensor. Given the nature of the accelerometers, the expected mean value should be close to zero. AC5 exhibited a notably wide interquartile range (IQR), showing significant variability and inconsistency in its measurements. As a result, AC5 was considered unsuitable for further use. AC6 demonstrated pronounced skewness, confirming its malfunction and necessitating the exclusion of its data. In contrast, AC2, AC3, and AC4 showed minor deviations from zero, which, although not indicative of a malfunction, were likely due to offsets.

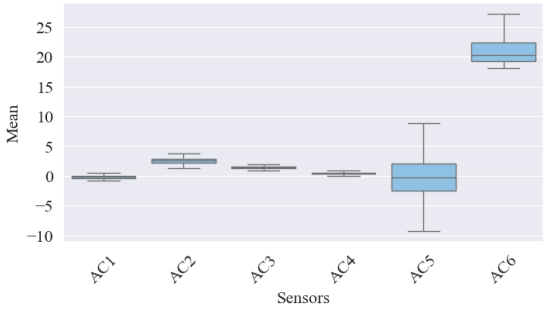


Figure 3: Boxplot of mean values of each accelerometer

Except for AC5 and AC6, the accelerometers exhibited strong correlations between their RMS values, as shown in Figure 4. While the correlation matrix reflects the overall similarity of the signals, additional checks confirmed that these correlations were not limited to static behaviour, but were also consistent over time. These correlations were used to develop a sensor health assessment model. An agglomerative clustering algorithm was used to group sensors based on their correlations. Using this hierarchical clustering method, the model distinguished between healthy and malfunctioning sensors, allowing unreliable data to be excluded from further analysis.

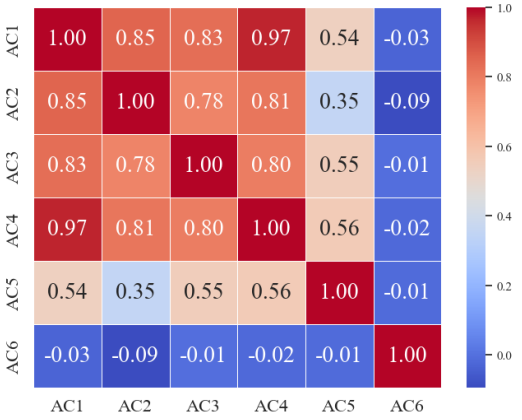


Figure 4: Accelerometers RMS Correlation Matrix

This approach not only enabled the identification of faulty sensors but also established a framework for detecting future sensor errors.

Outliers and Anomaly Detection

Analysis of the raw vibration signals from the accelerometers revealed irregular patterns, indicating the presence of anomalies in the data. These anomalies were categorised as follows and illustrated in Figure 5:

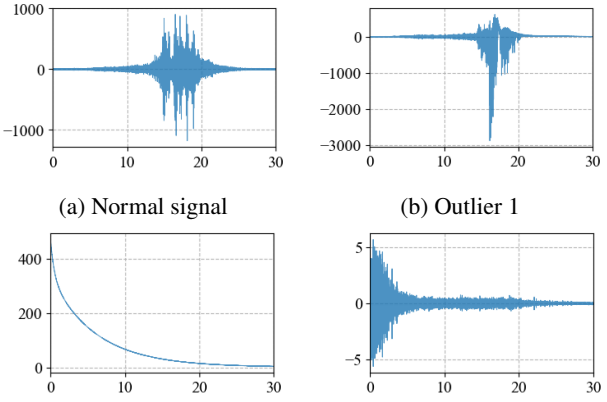
a) Normal signals: The typical accelerometer signal, which represents the expected behaviour of the system, is shown in Figure 5a. These signals have a mean

value close to zero and exhibit symmetry in the positive and negative Y directions, reflecting balanced vibrations under normal operating conditions. This symmetry is due to the rotational nature of the monitored components.

b) Outlier 1: This signal shows significant deviations, particularly with unusually low negative Y values, breaking the expected symmetry. This anomaly suggests a rotational imbalance, potentially caused by factors such as braking (wheel stoppage), rail defects, or unidentified mechanical issues.

c) Outlier 2: This irregularity occurs sporadically and lacks a consistent pattern. Potential causes include sensor malfunction, interference, or faults within the monitoring system box. Though its origin remains unclear, this anomaly signals possible underlying problems requiring further investigation.

d) Outlier 3: Characterised by a significantly lower amplitude compared to normal signals, this anomaly appeared after the initial 30 second data acquisition triggered by a passing train. Further analysis identified that residual vibrations from the same train had reactivated the trigger, leading to a second activation. To maintain the accuracy of the results, these signals were excluded from further analysis.



(a) Normal signal (b) Outlier 1

(c) Outlier 2 (d) Outlier 3

Figure 5: Raw vibration signals from the AC3 outliers

To address these anomalies, a classification model was developed using MLOps principles to identify and categorise irregular signals. A random forest classifier was selected to label the data patterns. The model was trained and validated on approximately one hundred manually labelled signals using the F1 score[1], which balances precision and recall in a single metric. For this multi-label classification, a macro-average[2] was applied to ensure that each class was given equal importance regardless of class imbalance. This approach avoided biasing the scoring metric towards more frequent classes. The classifier showed good performance for this use case, achieving an F1 score of 97.3%, indicating high precision and recall in distinguishing anomalous signals. Figure 6 presents the confusion matrix of the outlier classifier, obtained through cross-validation testing.

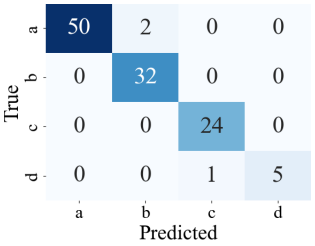


Figure 6: Outlier classifier Confusion Matrix

Before achieving this result, several iterations were carried out within the MLOps framework. Initially, about thirty signals were labelled. As more manually labelled data became available, the pipelines were re-executed and the models were then automatically trained, tested and deployed. During each iteration, F1 scores were recorded to track performance improvements. Figure 7 shows a screenshot of AML illustrating the evolution of the classification models through the iterations.

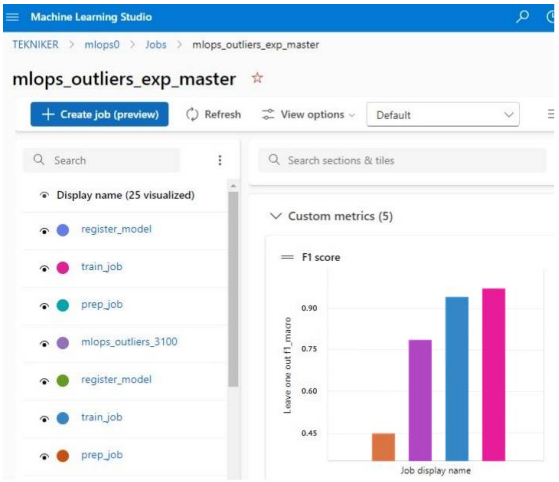


Figure 7: Screenshot of the Evolution of F1 scores in AML

Lubrication Analysis

To evaluate the efficacy of the rail lubrication process, the lubrication percentages presented in Table 1 were analyzed and compared against the vibration signals captured by the accelerometers.

Regression model The objective of the regression model was to evaluate whether the data from the installed sensors could reliably reflect the lubrication percentages, thereby providing a deeper understanding of rail conditions. To achieve this, an XGBoost regression model was

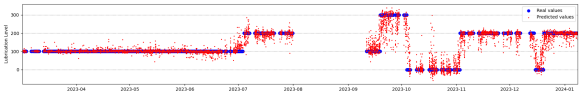
[1] $F1 = \frac{TP}{TP + \frac{1}{2}(FP+FN)}$

[2] $F1_{\text{macro}} = \frac{\sum_{i=1}^{n} F1_i}{n}$

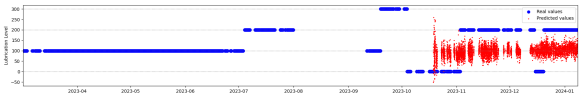
trained and tested using accelerometer KPIs.

Regarding the validation schema, given that some temporal correlation was expected, two testing scenarios were employed as shown in Figure 8: (a) using randomly sampled test points and (b) using the last 25% of the data as unseen test data. This way the behaviour of the temporal correlation could be better understood. The results demonstrate a significant discrepancy. In the random testing, the model performs well, achieving an R-squared value of 0.81, indicating a strong fit. However, in the forecasting scenario, the model exhibited poor performance, with an R-squared value of -0.25. Given that the doses were already seen during the forecasting testing, this might imply that other exogenous factors affect sensor readings.

The accelerometers revealed no clear patterns that would allow effective differentiation between lubrication percentages. While the model generalises well to randomly sampled data, it fails to extrapolate in a meaningful way over time. Several factors may contribute to this: changes in environmental conditions, train speed variability or rail surface condition over time may influence sensor readings independently of lubrication. This suggests that accelerometer signals alone are insufficient to accurately predict lubrication conditions.



(a) Random test data



(b) Final time-series data

Figure 8: Performance of the XGBoost Regression Models.

Anomalies at 0% lubrication At first, it was thought that the anomalies shown in Figure 5 were detected sporadically throughout the year. However, a significant increase in anomalies was observed when the lubrication dosage was set to 0%. As illustrated in Figure 9, nearly all recorded signals from the second week of October exhibited anomalies, classified as either outlier 1 or outlier 2. These abnormalities likely indicate issues such as increased friction or rail wear due to the absence of lubrication. This suggests that the model can effectively identify such outliers to detect rail-related problems.

After 16 October, the signal patterns returned to normal even though the lubricant dosage remained at 0%. Discus-

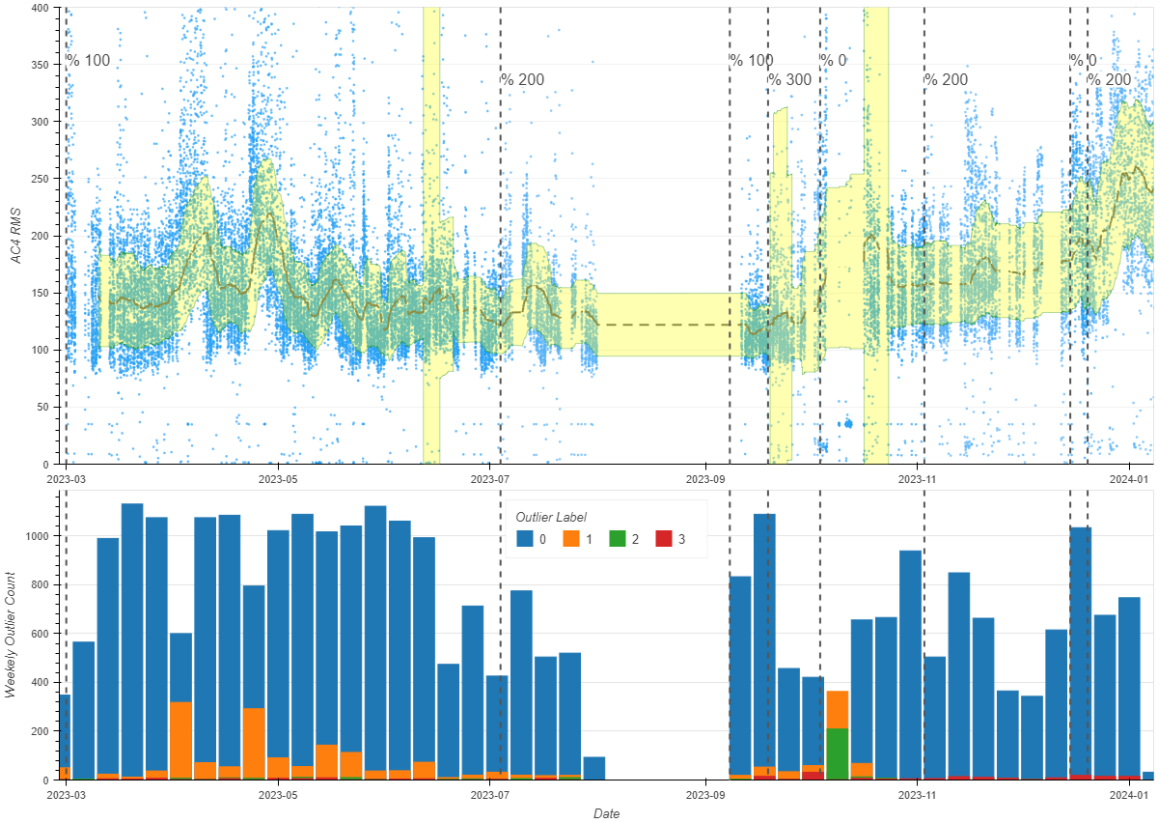


Figure 9: RMS analysis of AC4 accelerometer and weekly outlier detection

The top plot displays the RMS values recorded by the AC4 accelerometer. Blue points represent the raw RMS data, the black line shows the mean RMS within a window (with outliers excluded), and the yellow band indicates the interquartile range (IQR 1–3). The bottom plot presents the weekly count of detected outliers, categorized by type.

sions with the maintenance team revealed that this was due to manual lubrication applied by a separate maintenance team to remedy the earlier lack of lubrication. This case highlights the importance of integrating sensor data with maintenance activities to accurately diagnose rail conditions. The anomalies in the sensor data show that sensor readings can reliably detect lubrication-related problems - for example, the model detected numerous outliers on 5 October, while the metro team identified the problem 11 days later, on 16 October, and took action the same day. This illustrates how data-driven models can speed up problem detection, enable timely intervention and reduce reliance on subjective judgements.

By leveraging this model, metro maintenance personnel gain actionable insights through near real-time analysis. The system processes and presents data on a dashboard within minutes of a train passing, allowing personnel to quickly identify and address anomalies or potential issues. In the Figure 10, there is an example of a dashboard created in Power BI, which is sent to the end users.

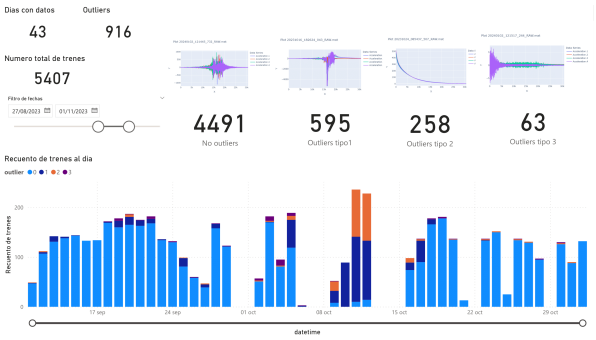


Figure 10: Example of a Power BI Dashboard

Conclusions

This study focused on analysing rail maintenance using accelerometer data and machine learning models. To support this, a monitoring and data analysis system integrating MLOps principles was developed, enabling seamless deployment and updates of models in production.

Implementing three machine learning models constituted a pivotal element of the system, which were the outcomes of a data analysis process. These outcomes included the identification of malfunctioning accelerometers and the development of an effective outlier detection model. While the performance of the regression model was inconclusive, indicating that vibration data alone is insufficient for accurately predicting lubrication levels, the outlier detection model proved effective in identifying irregularities in rail conditions. This model plays a crucial role in detecting rail issues, especially in the absence of lubrication, thus contributing valuable insights for rail maintenance and minimising wear.

The dashboards will enable metro experts and maintenance teams to quickly access actionable insights every time a train passes. This project is a step towards enabling data-driven decision-making for rail maintenance.

Future Work

Future work can be categorized into two areas: system enhancements and analytical improvements.

Currently, the system provides insight through visualisations such as graphs and dashboards. A valuable improvement would be the integration of a notification system to provide real-time alerts to maintenance teams when anomalies are detected. In addition, the system could be directly linked to lubrication management processes, allowing data-driven adjustments to lubrication percentages in real time based on rail conditions. This integration would increase the efficiency of the lubrication application. The current installation could also be extended to other locations along the rail network. Given the modular and scalable nature of the MLOps framework, the models could be relatively easily adapted and deployed in new areas, providing greater coverage and insight. The current solution is scalable thanks to the underlying cloud services. If required, model retraining, version control and remote endpoint management can be handled using built-in lifecycle management and deployment orchestration tools. However, there are practical challenges to scaling the system. Deploying models across multiple sites increases the need for coordinated maintenance, consistent monitoring and reliable data handling across sites.

In terms of analysis, the inclusion of additional sensors could significantly enhance the analytical capabilities of the solution. For example, the integration of microphones to capture acoustic data could provide additional information about rail conditions, such as detecting cracks or other irregularities based on sound patterns. This expanded data collection would refine anomaly detection and improve overall reliability and effectiveness.

Acknowledgments

This research has been supported by the project INARTRANS 4.0 “TRANSICION DIGITAL HACIA UNA INDUSTRIA AVANZADA EN SOLUCIONES DE INTELIGENCIA ARTIFICIAL PARA EL SECTOR DE LAS INFRAESTRUCTURAS DE TRANSPORTE” (“Digital transition towards an advanced industry in artificial intelligence solutions for the transport infrastructure sector”) PLEC2023-010343/MIG-20232067 funded by MICIU/AEI /10.13039/501100011033, and funded by the Elkartek programme, TCRINI2 project “NUEVAS TECNOLOGÍAS PARA LA INSPECCIÓN DE INFRAESTRUCTURAS CRÍTICAS EN EL SECTOR DEL TRANSPORTE” (“New technologies for critical infrastructure inspection in the transport sector”) ref: KK-2023/00029 sponsored by the Basque Government.

References

Belikov, A., Kreknin, K., Matsuk, Z., and Protsiv, V. (2022). Lubricants for rail transport liquid (plastic) for friction pair “wheel – rail”. (1):63–68.

European Commission (2023). REPORT FROM THE COMMISSION TO THE EUROPEAN PARLIAMENT AND THE COUNCIL eighth monitoring report on the development of the rail market under article 15(4) of directive 2012/34/EU of the european parliament and of the council.

John, M. M., Olsson, H. H., and Bosch, J. (2021). Towards mlops: A framework and maturity model. In 2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pages 1–8.

Kreuzberger, D., Kühl, N., and Hirschl, S. (2023). Machine learning operations (mlops): Overview, definition, and architecture. IEEE access, 11:31866–31879.

Moreschi, S., Recupito, G., Lenarduzzi, V., Palomba, F., Hastbacka, D., and Taibi, D. (2023). Toward end-to-end mlops tools map: A preliminary study based on a multivocal literature review.

Shafique, R., Siddiqui, H.-U.-R., Rustam, F., Ullah, S., Siddique, M. A., Lee, E., Ashraf, I., and Dudley, S. (2021). A novel approach to railway track faults detection using acoustic analysis. Sensors, 21(18).

Shojaee Rad, Z. and Ghobaei-Arani, M. (2024). Data pipeline approaches in serverless computing: a taxonomy, review, and research trends. Journal of Big Data, 11(1):1–42.

Song, R., Lu, C., Sun, L., Zhang, Z., Chen, D., and Shen, G. (2024). Analysis of wheel wear and wheel-rail dynamic characteristics of high-speed trains under braking conditions. 2024(1):9618500.

Wang, G., Nixon, M., and Boudreaux, M. (2019). Toward cloud-assisted industrial iot platform for large-scale continuous condition monitoring. Proceedings of the IEEE, 107(6):1193–1205.

Wilson, L. J. (2006). Performance measurements of rail curve lubricants.

Zeng, Y., Núñez, A., Dollevoet, R., Zoeteman, A., and Li, Z. (2024). A train-borne laser vibrometer solution based on multisignal fusion for self-contained railway track monitoring. IEEE Transactions on Industrial Informatics, pages 1–10.