POLITECNICO DI TORINO
Repository ISTITUZIONALE

Transformer-based Analysis of Vehicle CAN Bus Data for Predictive Maintenance: A Real Case Study

Original
Transformer-based Analysis of Vehicle CAN Bus Data for Predictive Maintenance: A Real Case Study / Yassine, A., Zorzan, F., Salvatori, L., Vassio, L., Cagliero, L.. - ELETTRONICO. - 4192:(2026). (10th International Workshop on Data Analytics solutions for Real-LIfe APplications (DARLI-AP) Tampere, Finland March 24, 2026).

Availability:
This version is available at: 11583/3009778 since: 2026-04-10T10:47:32Z

Publisher:
CEUR-WS.org

Published
DOI:

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

Transformer-based Analysis of Vehicle CAN Bus Data for Predictive Maintenance: A Real Case Study

Ali Yassine[1,2,*], Fabio Zorzan[2], Lucia Salvatori[2], Luca Vassio[1] and Luca Cagliero[1]

[1]Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129 Torino, Italia
[2]Tierra S.p.A., Corso Francesco Ferrucci, 112, 10138 Torino, Italia

Abstract

The use of machine learning techniques to analyze Controller Area Network (CAN) bus data transmitted by fleets of industrial vehicles has been increasingly explored. The main industrial applications include avoiding service disruptions and vehicle damage, improving operational efficiency, and reducing cybersecurity risks. However, the application of predictive maintenance techniques to industrial vehicle data is challenged by the high-dimensional and heterogeneous nature of the signals, their variable quality, and the limited availability of public benchmarks and human annotations. In this work, we describe a real-world industrial case study based on company data acquired from fleets of thousands of commercial vehicles over several years. We design a machine learning pipeline to early detect vehicle faults based on the analysis of CAN bus signals and evaluate the performance of several prediction models, including a newly proposed transformer-based architecture. Furthermore, we also demonstrate that existing public benchmarks fail to capture the complexity of real industrial scenarios, highlighting the need for more realistic and comprehensive analysis and benchmarks.

Keywords

Predictive maintenance, Vehicle systems, CAN bus data, Machine learning, Time series analysis

1. Introduction

Modern vehicles are complex cyber–physical systems composed of numerous Electronic Control Units (ECUs) and sensors that continuously monitor mechanical, electrical, and thermal subsystems. This increasing level of instrumentation enables detailed visibility into vehicle operation, introducing new opportunities to improve reliability, maintenance, and fault management [1].

Unexpected failures can result in costly downtime, safety risks, and operational issues, particularly in industrial fleet scenarios where vehicles operate continuously under diverse conditions [2]. Traditional maintenance strategies, such as reactive repairs or fixed-interval servicing, are often inadequate for such complex systems. For instance, reactive maintenance typically leads to unplanned downtime and expensive repairs, while scheduled preventive maintenance causes unnecessary interventions and inefficient use of resources.

Predictive Maintenance (PdM) aims to address these limitations by exploiting operational data to anticipate equipment failures and schedule maintenance proactively to reduce downtime, lower operational costs, and extend the life of industrial assets [3]. Classical PdM approaches rely on statistical modeling, condition monitoring, and reliability-centered metrics such as Remaining Useful Life (RUL) and Mean Time Between Failures (MTBF). These methods often use sensor-based measurements, including vibration, temperature, pressure, and oil analysis to detect signs of degradation. As a drawback, traditional approaches face challenges in modern vehicles due to the heterogeneous, high-dimensional nature of sensor signals and complex inter-dependencies between subsystems [1].

Published in the Proceedings of the Workshops of the EDBT/ICDT 2026 Joint Conference (March 24-27, 2026), Tampere, Finland
*Corresponding author.
ali_yassine@polito.it (A. Yassine); fzorzan@tierratelematics.com (F. Zorzan); lsalvatori@tierratelematics.com (L. Salvatori); luca.vassio@polito.it (L. Vassio); luca.cagliero@polito.it (L. Cagliero)
https://aliyassine26.github.io/ (A. Yassine)
0009-0004-3517-8754 (A. Yassine); 0009-0004-2919-3755 (F. Zorzan); 0000-0002-2920-1856 (L. Vassio); 0000-0002-7185-5247 (L. Cagliero)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-04-09

CEUR-WS.org/Vol-4192/DARLIAP-paper10.pdf

The Controller Area Network (CAN) bus represents a key data source for PdM in vehicles. It provides continuous streams of sensor measurements and control signals related to engine operation, cooling systems, and other critical subsystems. These multivariate time series capture both nominal operating conditions and subtle changes that may precede fault events. However, applying data-driven PdM methods to real-world CAN data remains challenging due to the high dimensionality and heterogeneity of the signals, their variable quality, and the limited availability of reliable fault annotations in industrial settings. Specifically, state-of-the-art approaches rely on transformer models [1], mainly addressing the detection of short-horizon anomalies or intrusion. Little attention has been paid to long-horizon fault anticipation and PdM using raw CAN data.

In this work, we address long-horizon PdM for industrial vehicles through a real-world case study involving large-scale CAN bus data collected from fleets of thousands of vehicles over multiple years.

Our **innovative contributions** are threefold:

- We develop a **machine learning pipeline for early fault detection in CAN bus data** and evaluate a variety of prediction models.
- We propose a **new transformer-based architecture** designed to effectively capture long-range temporal dependencies and cross-signal interactions in high-dimensional multivariate scenarios.
- We provide an **analytical comparison between the data considered in this industrial case study and the in-domain public benchmarks**, justifying the ad hoc pipeline and the experiments reported in the present study.

The remainder of the paper is organized as follows. Section 2 describes the CAN dataset and its comparison with public benchmarks. Section 3 presents the proposed transformer-based model. Section 4 details the experimental setup. Section 5 reports experimental results and discusses their implications. Finally, Sections 6 and 7 conclude the paper, outline directions for future work, and discuss the main limitations of the present work.

2. Vehicle Data & Problem Description

2.1. CAN Bus Data

Modern vehicles rely on the Controller Area Network (CAN) protocol to enable communication among electronic control units (ECUs) and sensors. The CAN bus provides a lightweight and reliable broadcast-based communication mechanism, allowing multiple ECUs to transmit messages containing sensor measurements, actuator states, and diagnostic information over a shared bus.

Each CAN message is identified by a unique identifier and carries a payload encoding one or more physical signals, such as engine speed, coolant temperature, pressures, or voltages. These raw messages are continuously generated during vehicle operation at signal-specific frequencies, ranging from a few milliseconds to several seconds, depending on the subsystem and sensor type.

In industrial fleet deployments, CAN messages are collected by on-board telematics units and transmitted to cloud-based infrastructures [4]. The resulting multivariate time series are stored at a fixed sampling resolution and made available for downstream analytics, including monitoring, diagnostics, and PdM.

2.2. Dataset Construction and Labeling

The dataset is constructed from raw CAN messages and fault logs collected continuously from a fleet of industrial vehicles. To obtain a structured learning dataset suitable for PdM, we transform the raw event streams into fixed-length temporal windows aligned on engine operating time.

Windowing and aggregation. For each vehicle, CAN messages are grouped into consecutive, non-overlapping windows of duration W, expressed in engine hours. Within each window, multivariate CAN signals are aggregated using summary statistics such as minimum, maximum, and average, resulting in a fixed-length representation that summarizes signal behavior over that interval. This process removes fine-grained, within-window temporal ordering, but preserves the temporal sequence at the level of windows.

Fault filtering and consolidation. Fault events are extracted from vehicle diagnostic logs and serve as operational ground-truth labels for fault occurrence. These events are mapped to predefined fault definitions, which include (i) system-reported diagnostic trouble codes generated by onboard vehicle controllers and (ii) data-driven fault indicators derived from CAN signals using predefined rules based on domain expertise. To avoid duplicated fault registrations caused by repeated triggering of the same condition, consecutive occurrences of the same fault are filtered out. This step ensures that each fault corresponds to a distinct underlying event rather than repeated notifications of an already active issue.

Label assignment and prediction horizon. Each time window is labeled based on the occurrence of fault events—defined from vehicle diagnostic logs and domain-informed fault definitions—within a future prediction horizon of length L, expressed in engine hours. Windows are assigned to one of three classes: *normal*, *pre-faulty*, or *faulty*. A window is labeled as faulty if it coincides with a fault occurrence, pre-faulty if it is not faulty and a fault event occurs within the next L engine hours following the window, and normal otherwise. This labeling scheme explicitly targets early fault prediction, which is the primary objective in predictive maintenance, while preventing information leakage from future observations.

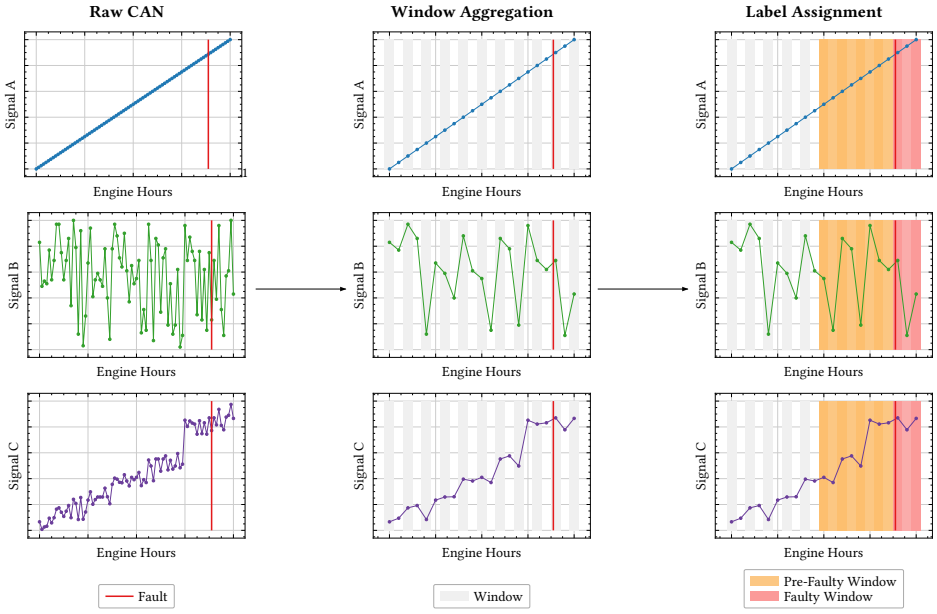


Figure 1: Illustration of the data processing pipeline for predictive maintenance. Multivariate time series derived from decoded CAN signals (left) are aggregated into fixed-length temporal windows (middle), which are then labeled as *pre-faulty*, or *faulty* based on fault events obtained from vehicle diagnostic logs (right). This visualization highlights the transformation from high-frequency sensor signals to structured window-level samples used for predictive maintenance model training.

Figure 1 provides an overview of the complete data processing pipeline, illustrating how decoded CAN signals represented as multivariate time series are transformed into window-level samples with aggregated features and future fault labels. Overall, this process results in a window-level dataset designed

to reflect realistic industrial constraints, including sparse fault events, noisy sensor measurements, and long operational horizons.

2.3. Our Case Study: Engine-Related Fault

This study considers a single fault prediction task related to engine operation. The fault is defined by an automotive domain expert and is directly related to two CAN signals, while being indirectly reflected in other engine-related signals.

The fault labels are treated as ground truth and are used for supervised training and evaluation of all models.

2.4. Comparison with Public Benchmarks

We compare the characteristics of our industrial fleet-scale dataset with other publicly available predictive maintenance benchmarks, i.e., SCANIA APS, SCANIA Component X, and MetroPT, to highlight the unique characteristics and challenges of our scenario (Table 1).

Unlike SCANIA APS, which provides tabular snapshots of operational states with a binary failure label, our dataset captures high-frequency operational signals over multiple years, enabling longitudinal modeling of component degradation. Compared to SCANIA Component X, which reports component failures using a time-to-event (TTE) format and contains multivariate features represented as histograms and numerical counters with irregular readout intervals, our dataset contains multiple diagnostic trouble codes (DTCs) per vehicle, reflecting a broader and more complex set of failure modes. While MetroPT offers high-frequency raw signals at 1 Hz, it is limited to a single vehicle with only a few documented failures, making it unsuitable for fleet-scale predictive modeling.

Our industrial dataset is derived from a real operational environment with heterogeneous vehicles, variable usage patterns, and diverse operating conditions. This introduces challenges not present in public benchmarks, such as imbalanced failure occurrences, uneven monitoring frequencies across vehicles, and high-dimensional feature representations that include both accumulated counters and multi-dimensional signals. Consequently, existing solutions tested on public datasets cannot be directly applied, as they often rely on simplified assumptions about failure distributions, unit homogeneity, or event sparsity.

Overall, the combination of fleet-scale coverage, heterogeneous high-frequency signals, and a complex failure landscape makes our dataset a challenging benchmark for predictive maintenance, bridging the gap between publicly available datasets and real-world industrial scenarios.

Table 1
Comparison of automotive and fleet-scale predictive maintenance benchmarks. TTE: Time-to-Event; DTC: Diagnostic Trouble Code. A dash (–) indicates information that is unavailable or not reported.

Dataset	Units	Signals	Resolution	Failures	Timespan	Data Format	Anomaly Type
SCANIA APS [5]	76,000	107	Snapshot	1,000	–	Snapshots	Event Label
SCANIA Comp. X [6]	33,121	14	Irregular / Low	2,000+	–	Histograms / Counters	TTE
MetroPT [7]	1	15	1 Hz	3	6 Months	Raw Time-Series	Event Label
Ours	50,000+	10+	High-Freq	1000+	3+ Years	Raw Time-Series	DTC Data-Driven Failures Pre-fault TTE

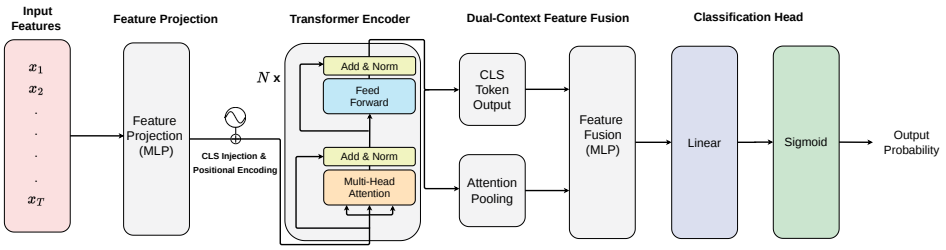


Figure 2: Overview of the CAN-Transformer framework. Multivariate CAN signal sequences are projected into a latent space, enriched with positional encodings, and processed by a transformer encoder stack. Global (CLS-based) and local (attention-pooled) representations are fused to produce fault predictions.

3. Methodology

This section presents the proposed framework for fault prediction from multivariate CAN time series.

3.1. Problem Formulation

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ denote a multivariate CAN time series segment of length T, where $\mathbf{x}_t \in \mathbb{R}^d$ represents d sensor measurements observed at time step t. The objective is to predict whether a fault will occur within a predefined future horizon based on the observed sequence.

We formulate this task as a binary classification problem:

$$f : \mathbb{R}^{T \times d} \to [0, 1], \quad p = f(\mathbf{X}), \tag{1}$$

where p is the probability of a fault within a predefined horizon. The problem features high-dimensional, noisy signals and strong class imbalance.

3.2. Overview of the CAN-Transformer Framework

The proposed framework, namely CAN-Transformer, is a transformer-based model designed to capture long-range temporal dependencies and cross-signal interactions in multivariate CAN data. Figure 2 illustrates the overall pipeline.

The architecture is inspired by transformer models in NLP, particularly BERT [8], and prior applications of BERT-style architectures to CAN data, such as CAN-BERT [9]. Unlike these works, which target intrusion detection on CAN ID sequences using unsupervised pretraining, our model addresses supervised fault prediction on rich multivariate sensor sequences.

Input Representation Each sensor channel is standardized offline via z-score normalization, then projected into a latent space through a lightweight MLP:

$$\tilde{\mathbf{x}}_t = \frac{\mathbf{x}_t - \mu}{\sigma}, \quad \mathbf{h}_t^{(0)} = \mathrm{MLP}(\tilde{\mathbf{x}}_t), \tag{2}$$

aligning heterogeneous signals for attention-based processing.

Sequence Encoding Following the BERT paradigm, a learnable classification token (CLS) is prepended to the input sequence, and positional embeddings are added to preserve temporal order:

$$\mathbf{H}^{(0)} = [\mathbf{h}_{CLS}; \mathbf{h}_1^{(0)}; \ldots; \mathbf{h}_T^{(0)}] + \mathbf{P}, \tag{3}$$

where $\mathbf{P}$ denotes the positional encoding matrix. This step ensures that the model can both summarize the sequence via the CLS token and capture the relative position of each timestep in the input.

Transformer Encoder The encoded sequence is processed by a stack of transformer encoder layers. Each layer consists of a multi-head self-attention module followed by a position-wise feedforward network, with residual connections and layer normalization:

$$\tilde{\mathbf{H}}^{(\ell)} = \text{LayerNorm}(\mathbf{H}^{(\ell-1)} + \text{MHA}(\mathbf{H}^{(\ell-1)})), \tag{4}$$

$$\mathbf{H}^{(\ell)} = \text{LayerNorm}(\tilde{\mathbf{H}}^{(\ell)} + \text{FFN}(\tilde{\mathbf{H}}^{(\ell)})). \tag{5}$$

This architecture enables the model to capture long-range temporal dependencies and complex interactions between sensor channels.

Dual-Representation Fusion Global (CLS) and local (attention-pooled) representations are concatenated:

$$\mathbf{h}_{fused} = \phi([\mathbf{h}_{CLS}; \mathbf{h}_{pool}]), \tag{6}$$

where ϕ is a linear transformation with normalization and nonlinearity.

Prediction Head The fused sequence representation $\mathbf{h}_{\text{fused}} \in \mathbb{R}^{d_{\text{model}}}$ is mapped to a scalar logit and passed through a sigmoid to produce a fault probability:

$$p = \sigma(\mathbf{w}^\top \mathbf{h}_{\text{fused}} + b), \tag{7}$$

where $\mathbf{w} \in \mathbb{R}^{d_{\text{model}}}$ and $b \in \mathbb{R}$ are trainable parameters optimized end-to-end.

Loss choice The fault prediction task exhibits severe class imbalance, as faulty periods are much rarer than normal periods. To address this, we employ an *Adaptive Focal Loss*, which extends standard Focal Loss [10] by using class-specific weighting to emphasize positive (faulty) examples while focusing training on hard-to-classify samples.

The loss is defined as:

$$\mathscr{L} = -\alpha_t (1 - p_t)^\gamma \log(p_t), \tag{8}$$

where p_t is the predicted probability for the true class, and α_t is a per-sample weight that depends on the class:

$$\alpha_t = \alpha_{\text{pos}} \cdot \text{target} + \alpha_{\text{neg}} \cdot (1 - \text{target}), \tag{9}$$

with the positive class corresponding to faulty periods and the negative class to normal periods. The $(1 - p_t)^\gamma$ term ensures that misclassified or uncertain examples contribute more to the loss, allowing the model to better learn rare and difficult fault events.

The model is trained using gradient-based optimization with regularization techniques such as early stopping and learning rate scheduling to ensure stable convergence and prevent overfitting.

Inference At inference, CAN data is segmented into fixed-length sequences and processed by the trained model to produce fault probabilities, enabling continuous monitoring and early warnings.

4. Experimental Setup

This section describes how the proposed model and baselines are trained and evaluated. It also defines the evaluation protocol and metrics used to assess PdM performance in industrial vehicle scenarios.

4.1. Training and Validation Procedure

The dataset is divided to ensure temporal and vehicle-level separation. First, 80% of the time windows (earlier operational periods) form the training set, and the remaining 20% (later periods) form the test set.

Second, the training set is split by vehicle: 80% of them are used for training and the remaining 20% for validation. This ensures validation evaluates the model's ability to generalize to unseen vehicles while preventing temporal leakage.

Preprocessing includes normalization of sensor signals, handling missing values, and feature aggregation over temporal windows.

The validation set is used to optimize model hyperparameters, select decision thresholds for fault detection, and implement early stopping during training. This procedure ensures that the model performance reported on the test set reflects generalization to unseen periods.

4.2. Baselines and Comparisons

To evaluate the performance of the proposed model, we compare it against a diverse set of classical and modern baseline models. These include: Neural Networks (Multi-Layer Perceptron, MLP); Tree-Based Models (Decision Tree, Random Forest); Support Vector Machines (Linear and RBF SVM); Probabilistic and Generative Models (Gaussian Process Classifier, Quadratic Discriminant Analysis, QDA, and Naive Bayes); and a heuristic baseline called *CAN Over Threshold*, which flags a fault whenever selected CAN signals exceed thresholds. The threshold hyperparameter was tuned using the validation set to maximize the F1 score.

All classical machine learning models are implemented using default scikit-learn[1] hyperparameters, except where otherwise noted. All baselines are trained and evaluated using the same dataset and preprocessing pipeline as the proposed CAN-Transformer to ensure a fair and reproducible comparison.

4.3. Evaluation Metrics

In line with the predictive maintenance objective, all models are trained and evaluated to detect *pre-faulty* periods, i.e., periods preceding an actual fault occurrence, rather than the identification of faults at the moment they occur, which aligns with the practical goals of predictive maintenance.

Performance is evaluated using standard classification metrics: Precision (P), Recall (R), and F1-score (F1). Since we consider a single fault type, these metrics are computed with respect to the positive class, corresponding to *pre-faulty* predictions.

Evaluation is conducted at the period level to reflect the predictive maintenance objective of early fault detection. Consecutive temporal windows are grouped into fixed-length periods of duration L, where L is an integer multiple of the window length W (i.e., $L = kW$ for some integer k). Model predictions are aggregated at the period level. A period is labeled as pre-faulty if it occurs within the prediction horizon preceding a fault event, and as faulty if it coincides with a fault occurrence; all remaining periods are labeled as normal. By construction, each faulty period is preceded by a pre-faulty period, while normal periods may occur before a pre-faulty period or after a faulty one.

4.4. Model Hyperparameters

The CAN-Transformer is trained using the following hyperparameter settings, selected based on validation performance to maximize period-level F1-score:

- **Architecture:** number of encoder layers = 2, hidden size (d_{model}) = 128, number of attention heads = 4, feedforward size (d_{ff}) = 256, dropout = 0.2, maximum sequence length = 1024, input sequence length = 32.

[1] https://scikit-learn.org

Table 2
Period-level test set performance comparison of all evaluated models for Engine Cooling Efficiency fault detection.

Model	Precision	Recall	F1-score
CAN-Transformer	0.351	0.986	**0.518**
MLP	**0.450**	0.380	0.412
Naive Bayes	0.230	0.938	0.369
QDA	0.205	0.751	0.322
CAN Over Threshold	0.134	0.993	0.236
Random Forest	0.093	0.999	0.170
Gaussian Process	0.080	**1.000**	0.149
RBF SVM	0.080	**1.000**	0.148
Linear SVM	0.071	**1.000**	0.133
Decision Tree	0.062	**1.000**	0.116

- **Optimization:** Adam optimizer with learning rate = 2e-4, weight decay = 1e-4, batch size = 128, early stopping patience = 5 epochs, total epochs = 25.
- **Loss:** Adaptive Focal Loss with class-specific weighting: $\alpha_{\text{pos}} = 3.0$, $\alpha_{\text{neg}} = 0.25$, $\gamma = 3.0$.
- **Dataset configuration:** lead time $L = 60$, window size $W = 6$, and fault duplicate filtering horizon $F = 72$, all expressed in engine operating hours.

Model Size and Hyperparameter Considerations The CAN-Transformer is intentionally designed to be lightweight, balancing model capacity with the risk of overfitting given the limited number of labeled fault events. Hyperparameters were tuned empirically by observing performance on the validation set, ensuring the model is both compact enough for efficient training and inference, and sufficiently expressive to capture essential temporal dependencies and cross-signal interactions in multivariate CAN data.

5. Results and Discussion

This section presents the period-level performance of the proposed CAN-Transformer model compared to a comprehensive set of classical and heuristic baselines on the dataset described in Section 2, restricted to failure-positive vehicles.

Table 2 shows that the CAN-Transformer achieves the highest F1-score, reflecting a strong balance between precision and recall. Classical models, such as Random Forests, SVMs, and Gaussian Processes, often attain near-perfect recall but extremely low precision, resulting in numerous false positives. Heuristic approaches (CAN Over Threshold) improve precision but are limited in overall F1-score, while simple models like MLP achieve slightly higher precision at the cost of much lower recall, meaning many faults would be missed in practice.

From an industrial perspective, both types of errors carry significant costs: false positives trigger unnecessary inspections and maintenance, increasing operational downtime and labor expenses, while false negatives risk missing real faults, potentially causing equipment damage and unplanned repairs. The CAN-Transformer mitigates these issues by leveraging long-range temporal dependencies and cross-signal interactions in high-dimensional, noisy CAN bus data. Its combination of high recall and moderate precision provides a practical trade-off, ensuring faults are rarely missed while keeping false alarms manageable.

These results demonstrate the effectiveness of attention-based sequence modeling for industrial vehicle fault detection, where both operational cost and fault risk must be carefully balanced.

6. Limitations

While this work provides insights into predictive maintenance on industrial fleets, several limitations should be acknowledged:

- **Data limitations:** The dataset, although large and high-frequency, contains noise due to sensor errors, device misconfigurations, and irregular sampling. Pre-fault periods are estimated from operational data rather than directly observed, so actual fault progression may differ. Some variables are aggregated or anonymized, limiting the interpretability of individual signals. Additionally, the dataset reflects a specific industrial environment, which may reduce the generalizability of findings to other vehicle types or operational contexts.
- **Evaluation constraints:** Ground-truth labels for pre-fault periods are inherently uncertain, and DTCs or maintenance records may not fully capture the onset or severity of faults. This introduces ambiguity into model evaluation, and reported metrics should be interpreted with this context in mind.
- **Generality and external validity:** While comparisons with public benchmarks highlight the challenges of fleet-scale industrial scenarios, the findings may not generalize to datasets with different types of vehicles, operational patterns, or sensor configurations. Additionally, some public datasets provide only simplified or snapshot data, which limits direct comparison of modeling performance.
- **Technical and operational constraints:** Variability in vehicle usage, maintenance schedules, and ECU software versions introduces heterogeneity in the data that may affect model performance. Models developed under these conditions may require adaptation before deployment in other fleets.

Acknowledging these limitations emphasizes the need for careful interpretation of the results and the continued development of robust predictive maintenance methods capable of handling noisy, heterogeneous, and high-dimensional industrial data.

7. Conclusion & Future Work

In this work, we presented CAN-Transformer, a transformer-based architecture for predictive maintenance on multivariate vehicle CAN bus data. Evaluated on real-world fleet data from thousands of commercial vehicles over several years, the model captures complex temporal dependencies and cross-signal interactions, enabling accurate fault prediction under challenging industrial conditions. CAN-Transformer outperforms classical machine learning baselines and heuristic approaches, achieving a balanced trade-off between precision and recall that is critical for practical deployment, where both false alarms and missed faults carry operational costs.

Future work includes extending the framework to multi-fault and multi-horizon prediction to improve early warning across diverse vehicle systems, integrating temporal attention explainability to aid operator interpretation, investigating transfer learning across different vehicle types or fleets to enhance generalization, and exploring online and continual learning strategies to adapt to evolving vehicle behavior and operational conditions in real time.

Overall, CAN-Transformer demonstrates that transformer-based sequence modeling is a powerful tool for industrial predictive maintenance, providing robust, interpretable, and scalable fault detection in complex, large-scale vehicular environments.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-5.2 in order to: proofreading and formatting assistance. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] Y. Lei, N. Li, L. Guo, N. Li, T. Yan, J. Lin, Machinery health prognostics: A systematic review from data acquisition to rul prediction, Mechanical Systems and Signal Processing 104 (2018) 799–834.

[2] S. Buccafusco, A. Megaro, L. Cagliero, F. Vaccarino, L. Salvatori, R. Loti, Profiling industrial vehicle duties using CAN bus signal segmentation and clustering, in: C. Costa, E. Pitoura (Eds.), Proceedings of the Workshops of the EDBT/ICDT 2021 Joint Conference, Nicosia, Cyprus, March 23, 2021, volume 2841 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2021. URL: https://ceur-ws.org/Vol-2841/DARLI-AP_7.pdf.

[3] R. K. Mobley, An Introduction to Predictive Maintenance, Plant Engineering, second edition ed., Butterworth-Heinemann, Burlington, 2002. URL: https://www.sciencedirect.com/book/9780750675314/an-introduction-to-predictive-maintenance.

[4] S. Buccafusco, L. Cagliero, A. Megaro, F. Vaccarino, R. Loti, L. Salvatori, Learning industrial vehicles' duty patterns: A real case, Comput. Ind. 145 (2023) 103826. URL: https://doi.org/10.1016/j.compind.2022.103826. doi:10.1016/J.COMPIND.2022.103826.

[5] APS Failure at Scania Trucks, UCI Machine Learning Repository, 2016. DOI: https://doi.org/10.24432/C51S51.

[6] Z. Kharazian, T. Lindgren, S. Magnússon, O. Steinert, O. A. Reyna, SCANIA Component X dataset: a real-world multivariate time series dataset for predictive maintenance, Scientific Data 12 (2025) 493. doi:10.1038/s41597-025-04802-6.

[7] B. Veloso, R. P. Ribeiro, J. Gama, P. M. Pereira, The metropt dataset for predictive maintenance, Scientific Data 9 (2022) 764.

[8] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[9] N. Alkhatib, M. Mushtaq, H. Ghauch, J.-L. Danger, Can-bert do it? controller area network intrusion detection system based on bert, in: Proc. IEEE/ACS AICCSA, 2022, pp. 1–8. doi:10.1109/AICCSA56895.2022.10017800.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 42 (2020) 318–327. doi:10.1109/TPAMI.2018.2858826.