AI-Driven Anomaly Detection for Hydraulic Systems: Enhancing PLC Monitoring and Industrial Reliability

Manuel Caipo Maximilian Dommermuth, and Jörn Kretschmer

Predictive maintenance is a cornerstone of Industry 4.0, yet its adoption in industrial environments is often hindered by fragmented data architectures, heterogeneous acquisition layers, and unclear scalability pathways. This work presents an open-source, modular framework for data architecture that bridges the gap between theoretical AI models and their operational deployment in manufacturing, with a focus on hydraulic systems. The framework enables cost-effective evaluation of AI-driven strategies for anomaly detection, machine state classification, and Remaining Useful Life (RUL) prediction—while remaining fully compatible with existing Programmable Logic Controller (PLC) monitoring that provides low-latency safety thresholds.

A real-world case study is conducted on a hydraulic system monitored by a Bosch Rexroth `ctrlX CORE` PLC, connected to an OPC UA (Open Platform Communications Unified Architecture) server. Machine states are identified via K-Means clustering and Hidden Markov Models (HMMs), whereas RUL is estimated using a Long Short-Term Memory (LSTM) network augmented with an attention mechanism and an XGBoost regressor. All models are orchestrated in a containerized Extract–Transform–Load (ETL) pipeline implemented with Dagster and Docker, acquiring sensor data, storing it in a PostgreSQL database, and delivering results to operators through Power BI dashboards for near-real-time decision support.

The architecture is designed for edge execution, reducing dependency on cloud infrastructure in the early adoption phase and enabling future migration once a positive return on investment (ROI) is demonstrated. By complementing PLC-based threshold alarms with statistical learning and historical insights, the framework enhances reliability, supports knowledge transfer, and reduces reliance on expert operators. The results illustrate how scalable AI–PLC integration can transform conventional monitoring into an adaptive, data-driven maintenance ecosystem aligned with Industry 4.0 principles.

Index Terms—**Anomaly detection, hydraulic systems, PLC monitoring, AI-driven, Docker containers, data pipelines, adaptive learning.**

NOMENCLATURE

AI	Artificial Intelligence
ACR	Asset Criticality Ranking
CPS	Cyber-Physical System
ctrlX CORE	Programmable Logic Controller from Bosch Rexroth
DL	Deep Learning
Docker	Platform for developing, shipping, and running containerized applications
Dagster	Data orchestration platform for building and managing data pipelines
ERP	Enterprise Resource Planning
ETL	Extract, Transform, Load
FMECA	Failure Mode, Effects, and Criticality Analysis
GUI	Graphical User Interface
HMM	Hidden Markov Model
Industry 4.0	Fourth Industrial Revolution
LSTM	Long Short-Term Memory
MES	Manufacturing Execution System
ML	Machine Learning
OPC UA	Open Platform Communications Unified Architecture
PdM	Predictive Maintenance
PLC	Programmable Logic Controller
PostgreSQL	Object-relational database management system
Power BI	Business Intelligence tool from Microsoft
ROI	Return On Investment
RUL	Remaining Useful Life
SHAP	SHapley Additive exPlanations
XGBoost	Extreme Gradient Boosting algorithm

I. INTRODUCTION

THIS work proposes a practical solution to one of the core challenges identified in [1] concerning machine learning and reasoning for predictive maintenance—specifically the issue of Big Data in industrial environments. The authors emphasize the importance of ensuring latency, scalability, and bandwidth. While edge computing is proposed as a solution, it is crucial to highlight that predictive maintenance does not necessarily require real-time decision-making. Instead, it demands periodic assessments (e.g., every 5 minutes, 1 hour, 7 hours, or 24 hours) . Predictive maintenance aims to reduce unexpected downtime by detecting anomalies and degradation before actual failure occurs.[2]

From this perspective, real-time Big Data challenges requiring complex and expensive infrastructure can be mitigated by adopting a periodic data-processing approach, tailored to the operational characteristics of the machine. Critical behaviors, such as immediate failures or rapid deviations from setpoints, are typically addressed by real-time monitoring and threshold-based PLC alarms, which inherently provide millisecond-level response times [3]. These alarms are specifically designed for components that fail within minutes or seconds, offering immediate warnings for process integrity or safety. In contrast, components exhibiting gradual degradation, which might not trigger immediate critical alarms, can be effectively assessed and predicted using machine learning (ML) and deep learning (DL) models. This distinction allows for a significant reduction in the burden on data infrastructure, enabling the entire data pipeline to be executed locally in a scalable architecture. Once economic benefits are evident, the solution can be strategically migrated to the cloud to ensure system availability and standardization.

From this perspective, real-time Big Data challenges requiring complex and expensive infrastructure can be mitigated by adopting a periodic data-processing approach, tailored to the operational characteristics of the machine. variables que indican un comprtamiento criticos definidas por un treshhols en plx, por que that fail within minutes or seconds are typically already monitored by PLC alarms [3], while components exhibiting gradual degradation can be effectively assessed using machine learning (ML) and deep learning (DL) models. This significantly reduces the burden on data infrastructure, allowing the entire data pipeline to be executed locally in a scalable architecture. Once economic benefits are evident, the solution can be strategically migrated to the cloud to ensure system availability and standardization.

This study introduces a localized analysis architecture for a hydraulic system monitored by a ctrlX PLC controller (Bosch Rexroth), which functions as both an edge device and OPC UA server. The proposed data pipeline processes signals extracted from the PLC through a message streaming service (Solace), followed by structured ingestion into a PostgreSQL database. The data is then pivoted and organized for efficient use, feeding three final tables at frequencies of 100 ms, 1 second, and 1 minute. The ETL pipeline is executed every 5 minutes to avoid overloading the local system.

As discussed in [1], data acquisition is a recognized challenge for deploying ML-based predictive maintenance solutions within Industry 4.0. However, this challenge is effectively addressed by structuring the dataflow to match the variability and sampling speed of sensor data. The system's scalability is ensured through an integrated streaming layer that enables migration and standardization of services, paving the way for modular and expandable architectures.

Another widely acknowledged challenge is data imbalance, especially when target failure events are rare. Nevertheless, this improves over time as more training data becomes available and models are adapted. The most persistent and unresolved challenge remains the lack of universal ML models suitable for every machine. However, recurring industrial scenarios—particularly in hydraulic systems—allow the development of localized model templates suited to specific machine distributions.

For condition monitoring and anomaly detection, the unsupervised clustering algorithm K-Means was used to identify abnormal patterns [4]. To reinforce these results, Hidden Markov Models (HMM) were applied to detect operational states. These models were supported by preprocessing techniques that calculated signal variation, oscillation, and envelope characteristics over time to distinguish critical operating states. Given the correlation between critical states and system runtime, a Remaining Useful Life (RUL) regression model was implemented using a Long Short-Term Memory (LSTM) neural network and XGboost, which captures temporal dependencies in the data.

The LSTM network outputs are periodically injected back into the PostgreSQL database via a pipeline orchestrated by Dagster. This enables prediction storage and continuous model retraining based on past prediction errors. The prediction pipeline is triggered every 6 minutes, starting one minute after the data ETL finishes. All microservices are containerized using Docker and executed locally on a single industrial computer.

This work demonstrates how a traditionally PLC-monitored hydraulic system can be enhanced with condition-based monitoring and predictive RUL estimation through edge-computing, machine learning, and a modular orchestration layer.

mds
August 01, 2025

II. Predictive Maintenance (PdM)

As presented by Lasi et al. [5], Industry 4.0 primarily describes IT-driven changes in manufacturing systems, encompassing not only technological but also organizational transformations. In the context of Cyber-Physical Systems (CPS), the implementation of PdM requires fully digitalized environments, where the real condition of the system is captured through sensor data and analyzed in real time.

PdM leverages historical data, domain knowledge, and computational models to forecast system behavior. By applying statistical or machine learning methods, it enables the identification of trends, patterns, and correlations, allowing for early detection of anomalies or potential failures. This approach supports informed decision-making and contributes to minimizing unplanned downtime [6].

As indicated in [7], it is crucial to perform an ACR – Asset Criticality Ranking to select assets that warrant deeper analysis (FMECA) and predictive resources. Subsequently, with an FMECA, one can make decisions regarding which Failure Mode, Effects, and Criticality Analysis should be conducted.

III. Related Work

sources,[2], [8], [9], [9], [10], [11], [12]

IV. System Architecture for Predictive Maintenance

This section presents the two main architectural components required to support a predictive maintenance solution aligned with Industry 4.0 principles:

- A database architecture for structured recording of machine faults and downtime events.
- A data flow architecture (ETL pipeline) to collect, process, and store real-time sensor data from PLCs into time-series tables.

Together, these architectures provide a unified data backbone for implementing anomaly detection and Remaining Useful Life (RUL) prediction models.

A. Database Architecture for Fault Collection

This architecture defines the structure and relationships between normalized tables used to log machine faults, operational states, and their root causes. It enables accurate tracking of downtime categories and failure contexts, forming the historical basis for supervised learning and operational analysis.

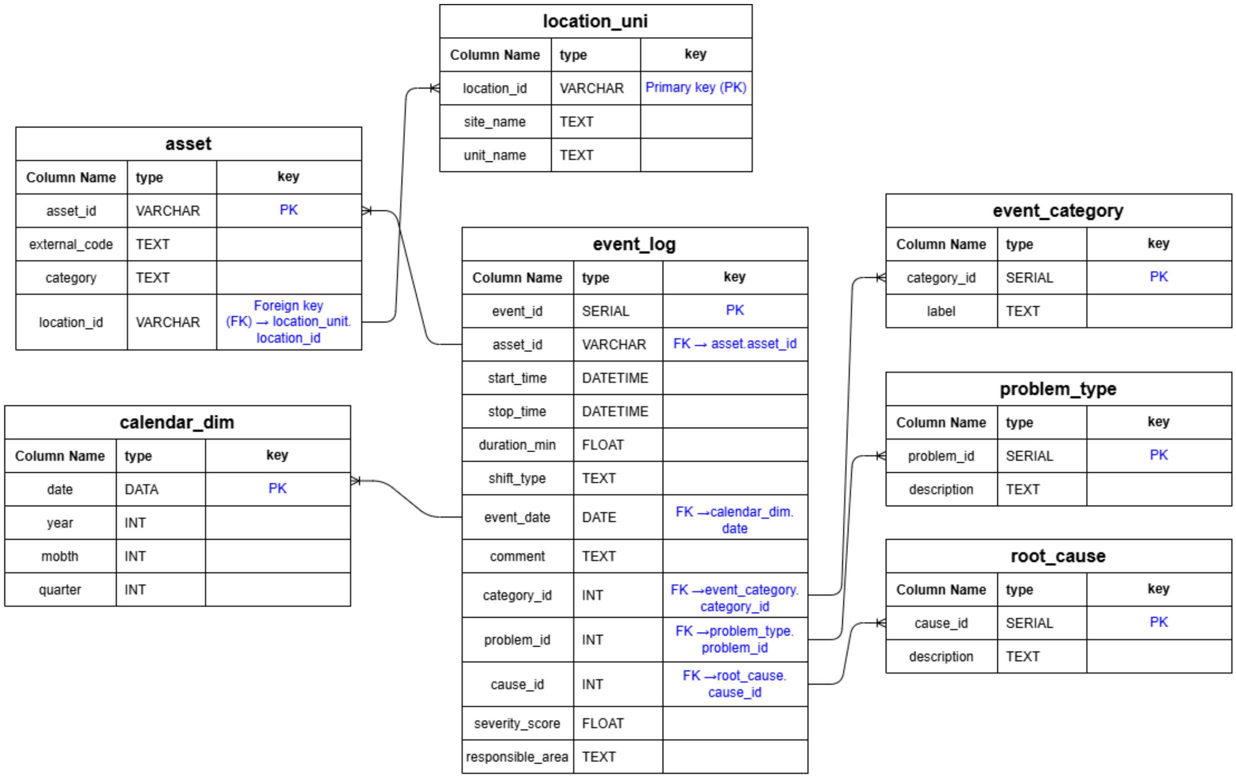

Fig. 1. Relational database schema for capturing machine faults and operational events as the foundation for predictive maintenance systems.

TABLE I
DOWNTIME CATEGORY DEFINITIONS

Category	Definition
Operational Standby	Equipment is available but idle due to external factors such as no upstream input.
Planned Maintenance	Scheduled inspections, calibration, or part replacement activities.
Unplanned Failure (Emergency)	Sudden stop due to unexpected system failure. In hydraulic systems, this could include pressure drops or actuator blockage.
Resource Unavailability	Downtime caused by lack of materials, such as low oil levels or blocked supply lines in hydraulic systems.

As shown in Figure 1, the system must include the following elements:

- A problem table that classifies typical hydraulic issues (e.g., “cylinder fails to extend/retract”, “slow or abnormal movement”).
- An asset table that registers each hydraulic machine and its corresponding identifier (e.g., cylinder, pump, cooler).
- An event log table that records equipment downtimes with precise timestamps, asset identifiers, and associated problem categories.
- A cause table that maps each problem to a potential root cause (e.g., internal leakage, non-functional cooler, trapped air).

Downtime events must also be categorized using a downtime category table. In the context of hydraulic systems, the main categories include:

- *Unplanned Failure (Emergency)*: unscheduled and critical stops due to unexpected system failure. In hydraulic applications, this includes pressure drops or actuator malfunctions.
- *Planned Maintenance*: scheduled inspections, calibration, or replacement of components.
- *Operational Standby*: periods when the machine is operable but inactive due to external conditions, such as missing production orders or upstream material delays.
- *Resource Unavailability*: downtime caused by lack of essential inputs, for example, insufficient oil supply or blocked delivery lines in hydraulic systems.

These structured classifications are essential. Without them, building an effective predictive maintenance model or production monitoring framework is nearly impossible. While sensor data (e.g., pressure, temperature, stroke length) can be analyzed, their real value emerges only when they are tied to the event log and root cause records. This linkage is what enables tracking of improvements in reducing unplanned failures and shifting them toward scheduled interventions.

Solutions based on machine learning or deep learning serve as practical tools to support operational decisions. They enable real-time visibility of equipment conditions and help maintenance planners and reliability engineers determine optimal intervention strategies. These strategies can then be integrated into enterprise maintenance systems (e.g., SAP PM), leading to better resource allocation, reduced downtime, and increased operational safety.

A key requirement for implementing this strategy is the

automation of downtime detection through edge computing at the OPC UA server level. The system should automatically generate downtime records in the database based on the `run_status` signal—whether derived from sensors or logic confirming actual machine activity. A predefined minimum threshold for acceptable inactivity must be established. Once exceeded, the system logs the event automatically.

Afterward, a graphical user interface (GUI) must prompt the operator to specify the cause of the stop and complete the event log with accurate contextual information.

This capability is fundamental to the realization of Industry 4.0. A reliable and structured collection of faults and downtimes is the foundation upon which predictive maintenance strategies are built. Only with this foundation can the impact of each solution be measured objectively and expanded or replicated across different machines and production environments.

B. ETL Architecture for Sensor Data Acquisition

In parallel, an ETL (Extract, Transform, Load) architecture handles the continuous acquisition of time-series sensor data from the PLC layer. This pipeline ingests raw data (e.g., pressure, flow rate), transforms it to a standardized format, and stores it in dedicated tables for analysis and visualization.

As an illustrative example of these final output tables, Figure 3 presents a sample table demonstrating the structure and content with a 1-minute frequency. In our implemented system, there are typically 3 such final output tables for the processed variables. The optimized dataflow processes the data and delivers an output to these databases every 5 minutes, enabling periodic analysis.

timestamp	CtrlX_2_2_ENER	CtrlX_2_ACT_VE	CtrlX_2_EFF_SY	CtrlX_2_P_A
2025-05-27 11:38:00	153.885778	0.23202535714285713	-1666.016004	0.5997596153846154
2025-05-27 11:37:00		0.5991972123893805	21.990232403846154	1.8698863636363636
2025-05-27 11:36:00	225.875168	5.634104339285714	28.905288056603773	2.052771226415094
2025-05-27 11:35:00	225.380417	-2.447930348214286	27.403432375	1.9752358490566038
2025-05-27 11:34:00	225.229675	3.179788783783784	35.302853039215684	2.073125
2025-05-27 11:33:00	225.427948	-4.652764303571429	27.991247444444443	2.051297169811321
2025-05-27 11:32:00	225.277206	4.748353223214286	31.842161150943397	2.043125
2025-05-27 11:31:00	225.376831	-5.378827428571428	32.27120887037037	2.1166087962962963
2025-05-27 11:30:00	225.50798	3.5999911964285713	30.75191132692308	2.0740625
2025-05-27 11:29:00		7.0883581632653065	37.87267714285714	2.0839460784313726
2025-05-27 11:28:00	225.134109	-5.818910026785714	29.602237692307693	2.1145833333333335
2025-05-27 11:27:00	225.072479	5.700335580357143	29.952798333333334	1.9957682291666667
2025-05-27 11:26:00	225.360825	-2.3347678839285715	23.538321549019606	2.0235795454545453
2025-05-27 11:25:00	225.629822	2.647479214285714	36.63653566666667	2.1553485576923075
2025-05-27 11:24:00		-2.6520401978021977	34.633461608695654	2.1998546511627906

(15 rows)

Fig. 3. Example output table with 1-minute frequency (similar structures apply for 100ms and 1-second intervals).

C. Hydraulic System Description

For our application case, we use this information based on the premise that the hydraulic system has the following configuration:

This is a hydraulic press system monitored by a PLC to detect variations and anomalies when it is not operating correctly. Its monitoring relies on thresholds and operating points that must not be exceeded. Its purpose is to exemplify the digital transformation of a conventional hydraulic system by integrating a controller (CTRLX) from Bosch Rexroth, which displays the hydraulic system's operational variables in real time. This system does not include a cooling system because, despite being an industrial example, its objective is didactic and educational. For safety reasons in educational environments, a cooler was not installed.

As discussed in the Predictive Maintenance section, it is essential to know which definitive failures we must model. This system is a failure simulator where an operating setpoint can be configured, and if the sensor value deviates from this setpoint, alarms are triggered. Consequently, the failures in this system (due to the lack of a cooler) are as follows: the system overheats, and the cylinder retraction speed slows due to component leaks. After approximately 2.5 hours, the system begins operating critically. The next time-correlated failure is that this system can simulate hydraulic cylinder leakage by opening a valve, which mimics cylinder leakage. Its primary objective was to demonstrate how leakage can be detected with the correct operating setpoint. However, it was observed that after 2.5 hours, the leakage becomes so severe that there is insufficient flow to move the cylinder, as the pressures equalize.

In summary, two main time-correlated failures are used, which we aim to prevent: a critical condition and an untimely operational failure (cylinder failure to move). For these two failures, the primary objective is to predict how long the system can continue operating before reaching conditions that could damage the system's integrity (Remaining Useful Life).

V. Machine and Deep Learning

A. Data Description and Preprocessing

Hydraulic actuation exhibits millisecond-scale dynamics; therefore the acquisition layer must preserve fast transients. In our setup, the ETL (Extract, Transform, Load — English) pipeline ingests signals from the PLC (Programmable Logic Controller — English) into two synchronized time-series layers: a $100\,\mathrm{ms}$ layer for control-relevant dynamics and a $1\,\mathrm{s}$ layer for supervisory analytics. This dual-resolution storage, aligned to the database schema described earlier, enables both cycle-level feature extraction and long-horizon trend analysis.

Domain observations indicate that, when the cooler is unavailable, oil temperature rises, viscosity decreases, and internal/external leakages increase (e.g., at valves and throttling devices). Consequences include reduced achievable pressure, slower cylinder motion, and drift in cycle amplitude and period. Guided by these mechanisms, we focused on the cylinder's position and velocity as primary signals.

Feature engineering proceeds as follows. Using sliding windows, we compute cycle-level descriptors such as stroke amplitude (peak-to-peak position), cycle period, and dwell times. To robustly track slow drifts, we estimate the amplitude envelope (e.g., via a Hilbert-transform–based envelope or peak–trough tracking) and short-horizon statistics of velocity (mean, standard deviation, and percentiles). Temperature is included as a covariate to control for viscosity-induced effects. Signals from heterogeneous sampling layers ($100\,\mathrm{ms}$ and $1\,\mathrm{s}$) are time-aligned, de-trended as needed, and validated against the run-status channel to exclude non-operative intervals.

Unsupervised structure is extracted with K-means (K-means clustering) to obtain compact operating regimes. To enforce temporal consistency and detect persistent regimes, we apply HMM (Hidden Markov Model) segmentation on the same feature set. Based on equipment knowledge, four operational

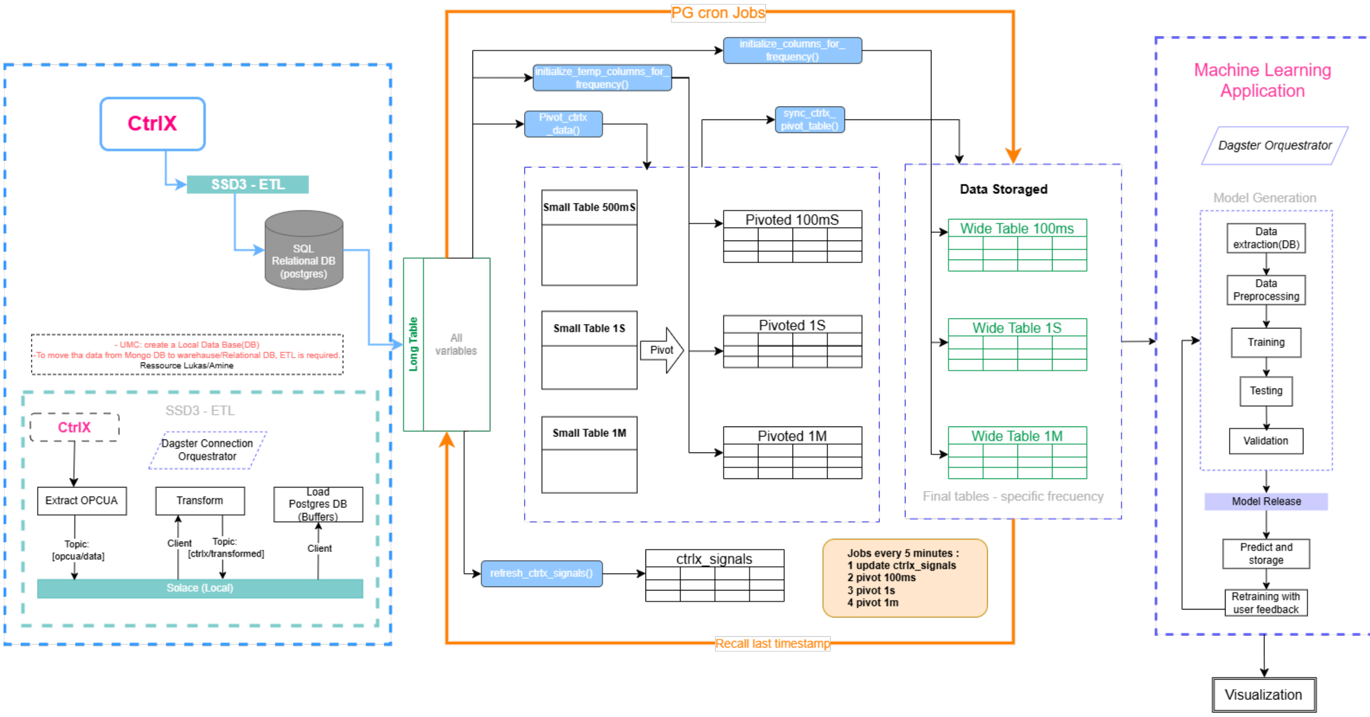

Fig. 2. ETL data flow from PLCs to time-series storage for real-time sensor monitoring and predictive analysis.

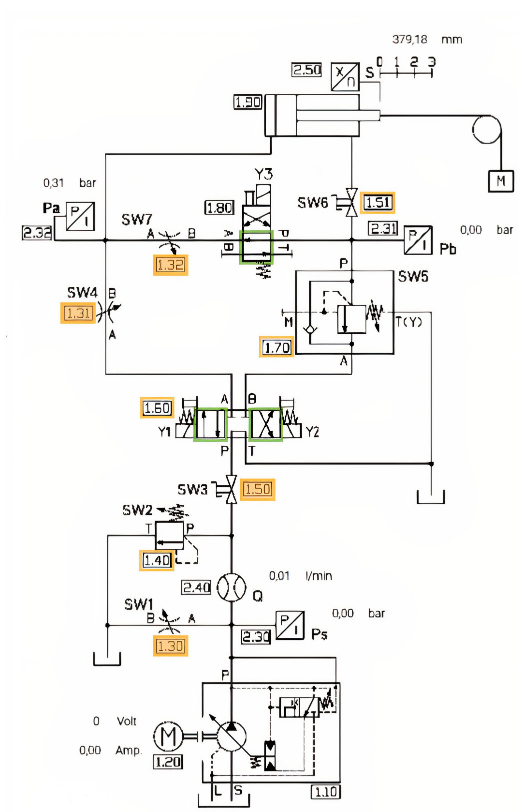

Fig. 4. Hydraulic drawing of the system.

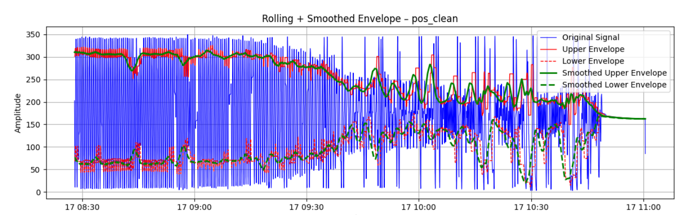

Fig. 5. Envelope of position signal for improved cycle variation characterization

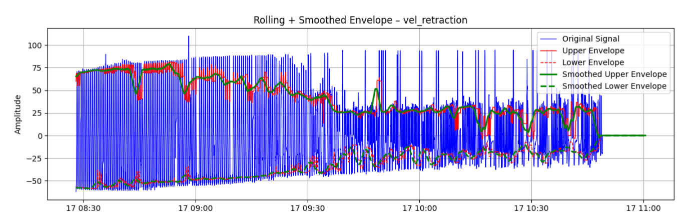

Fig. 6. Envelope of velocity signal for improved cycle variation characterization

states are specified: normal operation, speed degradation, leakage-indicative behavior, and motion failure/stall. These states map directly to the downtime categories introduced earlier (e.g., unplanned failure versus planned maintenance), which facilitates integration with the event log.

Finally, we frame Remaining Useful Life as a time-to-event target. Let t^* denote the first entrance time into a critical

TABLE II
HYDRAULIC PROBLEM TYPE CLASSIFICATION

ID	Problem Description
1	Cylinder fails to extend or retract.
2	Slow or abnormal movement during cylinder operation.
3	Pressure drop detected in the hydraulic circuit.
4	Overheating of hydraulic fluid beyond safe temperature.

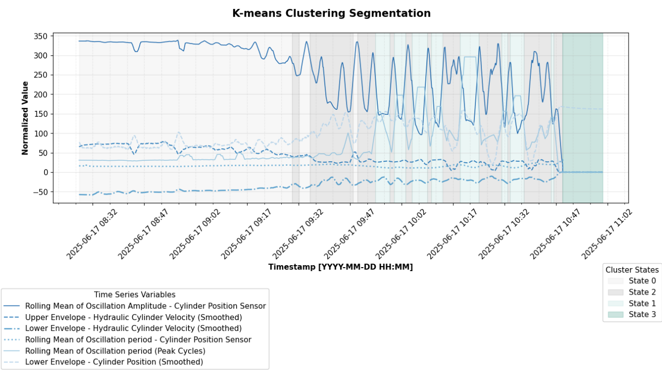

Fig. 7. State segmentation using K-Means clustering on derived variables such as position envelope and velocity.

state (e.g., motion failure or severe leakage) and t the current timestamp; the target is

$$\mathrm{RUL}(t) = t^{*} - t,$$

censored when the critical event is not observed within the horizon. We train a sequence model (e.g., LSTM, Long Short-Term Memory — English) or a survival/regression approach on the engineered features and raw sequences from the $100\,\mathrm{ms}$ layer, while leveraging the $1\,\mathrm{s}$ layer for trend features and health indicators. This design links physics-based effects (temperature–viscosity–leakage) to learned representations, yielding interpretable state classification and actionable RUL estimates suitable for maintenance planning.

B. Machine State Classification

To detect anomalous conditions in a machine, it is crucial to differentiate operational states. Traditionally, this classification requires expert knowledge. However, unsupervised machine learning algorithms, such as K-Means and Hidden Markov Models (HMM), provide an alternative by clustering similar behavior patterns based on derived features.[4]

Figure 7 shows the segmentation of machine states using K-Means clustering applied to derived variables such as envelope and velocity. This allows the identification of behavior clusters without prior labeling.

In contrast, Figure 8 illustrates the application of Hidden Markov Models to the same set of variables, capturing sequential dependencies and probabilistic transitions between states.

A direct comparison between both approaches is shown in Figure 9, where HMM offers a smoother state transition modeling, while K-Means provides clearer cluster boundaries.

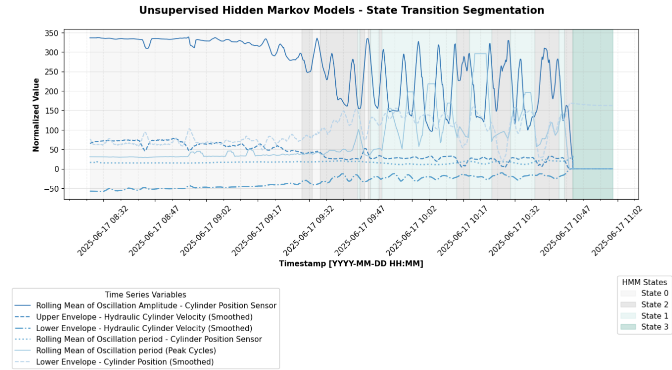

Fig. 8. State segmentation using Hidden Markov Models (HMM) on the same derived variables.

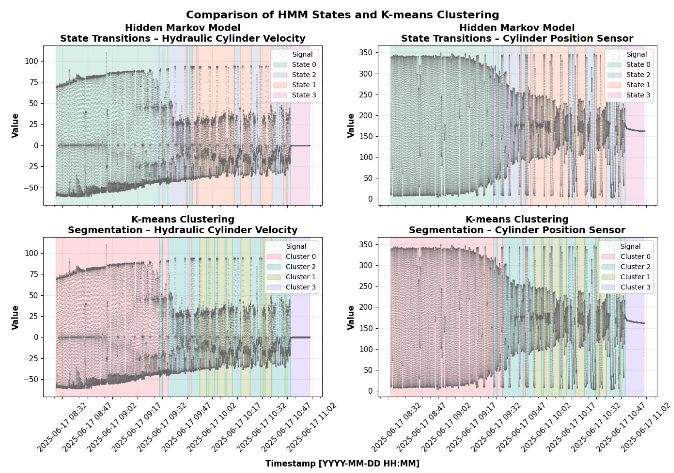

Fig. 9. Comparison of machine state classification using K-Means and HMM approaches.

C. Remaining Useful Life Estimation

To estimate Remaining Useful Life (RUL), an XGBoost regressor and an LSTM neural network were trained using transitions between machine states. Specifically, the shift from state 0 (normal) to state 2 (critical) marks the beginning of degradation, and the first occurrence of state 1 following state 2 defines the point of maximum risk. These transitions are used as temporal anchors to calculate the total and remaining operational time.

Figure 11 visualizes how machine states are mapped and used for RUL modeling. Subsequent figures (Figures 10–17) explain the model's decisions using SHAP values.

The predictions with the LSTM neural network follow [13]

a) Explanation of Individual Predictions.: Figures 12–17 provide instance-level SHAP visualizations for specific timepoints. These reveal the most influential variables such as envelope position, cycle period, and oscillation amplitude used by the model to estimate RUL.

VI. CONCLUSION

The conclusion goes here.

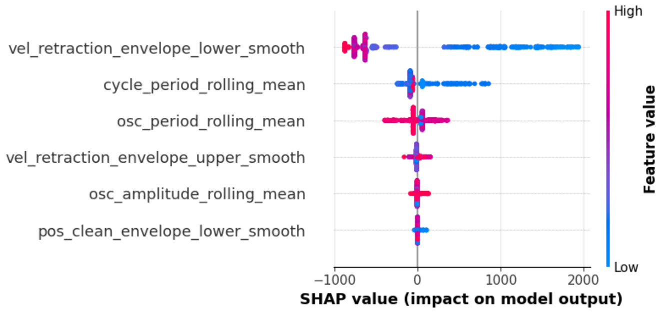

Fig. 10. Global SHAP explanation: key features influencing RUL predictions across all samples.

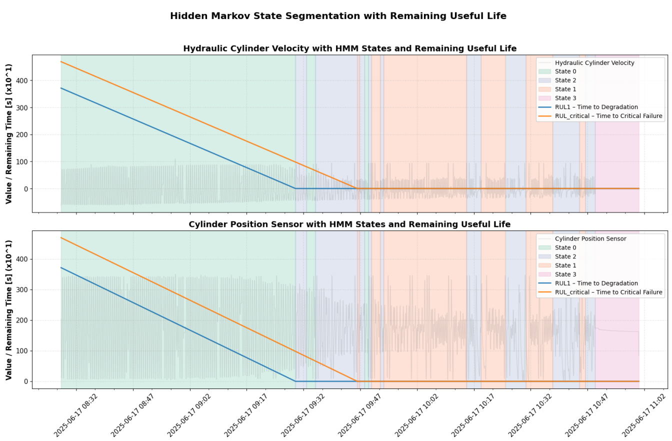

Fig. 11. State-based segmentation used for RUL estimation. Key transitions (state 0→2→1) define total and remaining useful life.

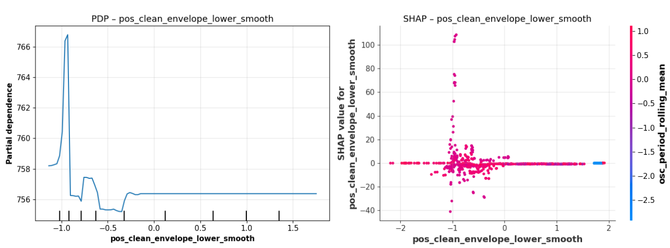

Fig. 12. Instance-level SHAP explanation: lower envelope position as dominant feature.

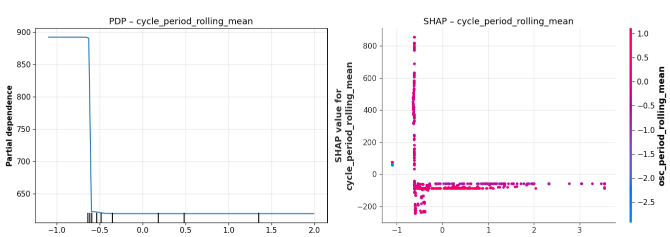

Fig. 13. Instance-level SHAP explanation: influence of cycle period and position sensor.

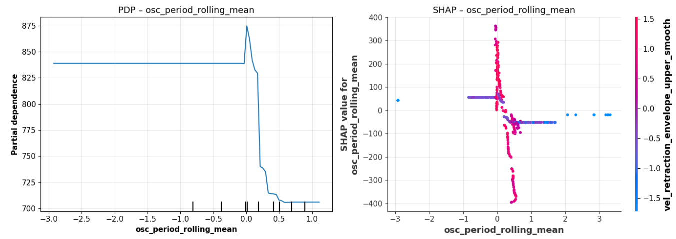

Fig. 14. Instance-level SHAP explanation: oscillation period and rolling mean position as key factors.

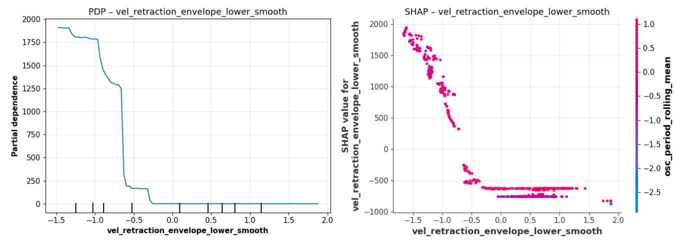

Fig. 15. Instance-level SHAP explanation: velocity and lower envelope importance.

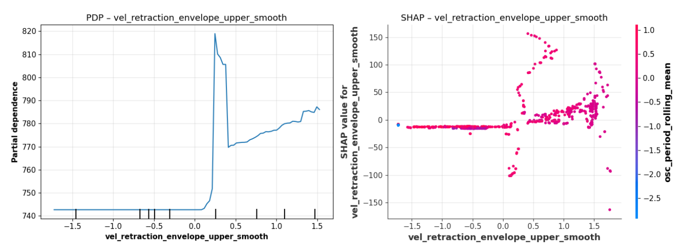

Fig. 16. Instance-level SHAP explanation: effect of envelope and velocity on prediction.

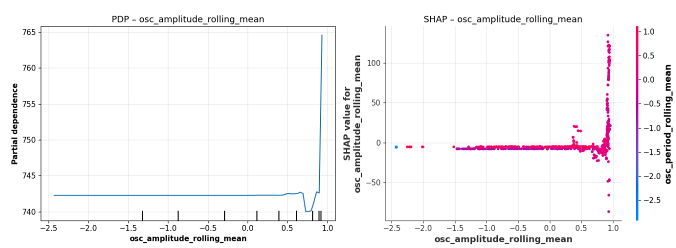

Fig. 17. Instance-level SHAP explanation: oscillation amplitude and position sensor features.

TABLE III
PERFORMANCE METRICS FOR RUL PREDICTION MODELS.

Model	Target	Set	MAE	RMSE	R^2	Sharp
XGBoost	RUL1	Train	1.272	3.223	1.0	1.0
XGBoost	RUL1	Validation	1.618	5.370	1.0	1.0
XGBoost	RUL1	Test	1.655	7.238	1.0	1.0
XGBoost	RUL_critical	Train	2.586	5.040	1.0	1.0
XGBoost	RUL_critical	Validation	3.203	9.425	1.0	1.0
XGBoost	RUL_critical	Test	3.116	8.301	1.0	1.0

REFERENCES

[1] J. Dalzochio, R. Kunst, E. Pignaton, A. Binotto, S. Sanyal, J. Favilla, and J. Barbosa, "Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges," *Computers in Industry*, vol. 123, p. 103298, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361520305327

[2] W. J. Lee, H. Wu, H. Yun, H. Kim, M. B. Jun, and J. W. Sutherland, "Predictive maintenance of machine tool systems using artificial intelligence techniques applied to machine condition data," *Procedia CIRP*, vol. 80, pp. 506–511, 2019, 26th CIRP Conference on Life Cycle Engineering (LCE) Purdue University, West Lafayette, IN, USA May 7-9, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118312988

[3] D. Javier Perez, J. Waltl, L. Prenzel, and S. Steinhorst, "How real (time) are virtual plcs?" in *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2022, pp. 1–8.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[5] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business and Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, 2014. [Online]. Available: https://doi.org/10.1007/s12599-014-0334-4

[6] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li, "Predictive maintenance in the industry 4.0: A systematic literature review," *Computers and Industrial Engineering*, vol. 150, p. 106889, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305787

[7] R. K. Mobley, *An introduction to predictive maintenance*. Elsevier, 2002.

[8] B. Xu, J. Shen, S. Liu, Q. Su, and J. Zhang, "Research and development of electro-hydraulic control valves oriented to industry 4.0: A review," *Chinese Journal of Mechanical Engineering*, vol. 33, no. 1, p. 29, 2020. [Online]. Available: https://doi.org/10.1186/s10033-020-00446-2

[9] D. L. Rivera, M. R. Scholz, M. Fritscher, M. Krauss, and K. Schilling, "Towards a predictive maintenance system of a hydraulic pump$**$this work is funded by the bayerisches staatsministerium fur wirtschaft und medien, energie und technologie in its r and d program bayern digital." *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 447–452, 2018, 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896318314708

[10] L. Wang, Y. Liu, H. Yin, and W. Sun, "Fault diagnosis and predictive maintenance for hydraulic system based on digital twin model," *AIP Advances*, vol. 12, no. 6, p. 065213, 06 2022. [Online]. Available: https://doi.org/10.1063/5.0098632

[11] K. Khan, M. Sohaib, A. Rashid, S. Ali, H. Akbar, A. Basit, and T. Ahmad, "Recent trends and challenges in predictive maintenance of aircraft's engine and hydraulic system," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 43, no. 8, p. 403, 2021. [Online]. Available: https://doi.org/10.1007/s40430-021-03121-2

[12] V. V. Shanbhag, T. J. J. Meyer, L. W. Caspers, and R. Schlanbusch, "Failure monitoring and predictive maintenance of hydraulic cylinder—state-of-the-art review," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 6, pp. 3087–3103, 2021.

[13] O. Serradilla, E. Zugasti, J. Rodriguez, and U. Zurutuza, "Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects," *Applied Intelligence*, vol. 52, no. 10, pp. 10 934–10 964, 2022. [Online]. Available: https://doi.org/10.1007/s10489-021-03004-y

Manuel Caipo received his B.Sc. in Mechanical Engineering from Universidad Nacional de San Agustín (UNSA), Peru, and a postgraduate diploma in Machine and Deep Learning. He is currently pursuing an M.Sc. in Advanced Precision Engineering at Hochschule Furtwangen University, Germany.

He has over four years of experience as a Data Scientist, focusing on predictive maintenance, neural networks, and signal processing. His technical strengths include Python, machine learning, and real-time industrial data architectures using Docker and databases (PostgreSQL), as well as scalable cloud-based solutions (Azure ML) and integration of data warehouses with ERP (SAP) and MES systems.