Management and Production Engineering Review
Volume 17 • Number 2 • June 2026 • pp. 1–17
DOI: 10.24425/mper.2026.1327

Predictive Maintenance in Serial Production Using Deep Learning: Analysis of Multi-Layered Telemetry Data Streams and Maintenance History

Kamil MUSIAŁ, Artem BALASHOV, Anna BURDUK

Faculty of Mechanical Engineering, Wrocław University of Science and Technology, Wrocław, Poland

Received: 17 December 2025
Accepted: 08 March 2026

Abstract
This article addresses the problem of predicting machine failures in a company conducting serial, multi-variant production, comprising over one hundred workstations, including welding, spot welding, gluing, and milling. We formulate predictive maintenance (PdM) as a binary early-warning task: at time t, the model predicts whether a failure episode – defined as a computerized maintenance management system (CMMS) corrective intervention causing unplanned downtime of at least 10 minutes – will start within the next H hours ($H = 4$ h by default; $H \in \{1, 4, 8, 24\}$ h in the sensitivity analysis), using the preceding W hours of telemetry history. The study integrated historical maintenance data, including service requests, Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR), and failure cause codes, with current machine operation telemetry, such as vibration, temperature, current, load, cycle times, and programmable logic controller (PLC) alarms. A complete pipeline was developed, encompassing synchronization of multi-source data streams, feature engineering, and methods for addressing class imbalance, such as weighted loss functions and focal loss. The analysis considered recurrent neural network architectures – Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) – as well as a one-dimensional convolutional neural network (1D-CNN). Model performance was assessed using the Area Under the Receiver Operating Characteristic Curve (AUROC), the Area Under the Precision-Recall Curve (AUPRC), and the F_1-score, with decision-threshold adjustment reflecting downtime costs. On the chronologically separated test set, the best-performing LSTM model achieved AUROC = 0.924, AUPRC = 0.847, and F_1-score = 0.823, indicating strong potential for predictive maintenance in serial production.

Keywords
predictive maintenance, deep learning, industrial telemetry, time-series analysis, imbalanced classification.

Introduction

Modern manufacturing systems are becoming increasingly vulnerable to unforeseen machine failures, which represent one of the most critical factors influencing industrial productivity and overall operational efficiency. According to recent industrial reports, unplanned downtime remains a major source of operational loss in manufacturing, reducing asset utilization, increasing maintenance costs, and disrupting production continuity (Siemens AG, 2024; Taheri Hosseinkhani, 2025). These effects highlight the need for more effective maintenance strategies capable of detecting and mitigating failures before they escalate into costly stoppages.

In response to these challenges, the concept of predictive maintenance (PdM) has emerged as a promising approach that aims to forecast potential failures before they occur, thereby reducing maintenance costs and minimizing production interruptions. Traditional maintenance paradigms have typically relied on two dominant strategies: time-based maintenance (TBM) and reactive maintenance (RM). The former often results in unnecessary part replacements and over-maintenance costs, whereas the latter is associated with substantial losses caused by unplanned stoppages and emergency interventions (Thomas & Weiss, 2021).

***Corresponding author**: Kamil Musiał – Faculty of Mechanical Engineering, Wrocław University of Science and Technology Wybrzeże Stanisława Wyspiańskiego 27, Wrocław 50-370, Poland, e-mail: kamil.musial@pwr.edu.pl*

Predictive maintenance offers an intermediate and data-driven solution that combines real-time machine condition monitoring with historical maintenance information to optimize maintenance scheduling and resource allocation.

Modern industrial systems continuously generate large volumes of heterogeneous telemetry data, including vibration, temperature, electrical current, mechanical load, and programmable logic controller (PLC) signals. When integrated with historical failure records and maintenance logs, these data form a comprehensive dataset that can be leveraged to construct predictive models based on advanced machine learning (ML) and deep learning (DL) techniques. Previous research and industrial analyses have demonstrated that effective implementation of PdM can reduce downtime by 30%–50% and extend equipment lifespan by 20%–40%, thus significantly improving operational efficiency (Benhanifia et al., 2025; Dilda et al., 2017).

The primary objective of this study is to develop and validate an intelligent predictive maintenance system for a company operating in serial, multi-variant production conditions. The proposed system integrates historical maintenance data with current machine telemetry and employs advanced neural network architectures to predict potential failures. Special emphasis is placed on addressing the class imbalance problem, which is inherent in failure prediction tasks due to the rarity of breakdown events compared to normal operation periods. Appropriate data preprocessing and balancing techniques are investigated to enhance model reliability and robustness.

The remainder of this paper is organized as follows. Section 2 reviews related work on predictive maintenance and machine learning in industrial settings. Section 3 describes the industrial context, data sources, preprocessing, and model development. Section 4 presents the experimental results. Section 5 discusses the findings, limitations, and practical implications. Section 6 concludes the paper and outlines directions for future research.

Literature review

Diagnostics and prognostics are key elements of maintenance management, enabling both the response to ongoing failures and the prediction of their occurrence. Diagnostics deals with detecting, locating, and identifying faults after they occur, allowing for rapid determination of the source of the problem and its nature. Prognostics, on the other hand, focuses on predicting failures, enabling the estimation of when and with what likelihood they may occur, which allows for more effective planning of maintenance activities and minimizing downtime (Jardine et al., 2006). Predictive Maintenance (PdM) has become a key direction in the development of production resource management within the Industry 4.0 paradigm, allowing for reduced downtime, extended machine lifetime, and optimization of maintenance costs (Nunes et al., 2023; Mallioris et al., 2024; Zhang et al., 2019).

Recent bibliometric analyses indicate a dynamic increase in the number of publications addressing the integration of telemetry, the Internet of Things (IIoT), and artificial intelligence methods, confirming the growing importance of PdM in serial production and highly automated manufacturing environments (Nunes et al., 2023; Mallioris et al., 2024). Contemporary PdM studies increasingly use deep learning models to analyze multidimensional, high-frequency industrial sensor data and combine these signals with historical maintenance records. This integrated approach improves failure prediction, reduces false alarms, and supports a more reliable distinction between gradual degradation and sudden faults.

Serial production systems generate heterogeneous, multilayer data streams that include vibration signals, current and load measurements, temperature, pressure, speed and positional data, as well as logs from PLCs and collaborative robots. These data are characterized by varying sampling frequencies, noise levels, and structural properties, while sharing the common feature of high variance resulting from changing operating conditions. High-frequency signals may provide valuable diagnostic insights but require advanced filtering and synchronization to reduce measurement artefacts (Nunes et al., 2023; Cardoso & Ferreira, 2021). At the same time, contextual information – such as service reports, failure histories, downtime logs, repair data, and spare-parts inventories – is gaining importance and is increasingly treated as an integral component of predictive PdM models (Arockia Mary et al., 2024; Ayvaz & Alpay, 2021).

The literature describes a wide spectrum of deep-learning architectures applied to failure prediction and Remaining Useful Life (RUL) estimation (Lei et al., 2018). Particularly popular are LSTM and GRU sequential models, well suited to capturing long-term temporal dependencies and changing operating conditions of mechatronic systems. Recurrent networks achieve high performance when processing quasi-cyclic signals typical for electric motors and drive systems (Li et al., 2024). LSTM and GRU networks – as variants of recurrent neural networks (RNNs) – offer attractive solutions for predictive maintenance, enabling effective use of long time series

to identify degradation patterns. However, their key limitation remains the difficulty of handling irregular time-series data (Adryan & Sastra, 2021).

Growing attention is also being paid to the integration of structured and unstructured data. Maintenance logs often contain descriptive information, requiring the use of natural language processing (NLP) techniques. Text embeddings enable capturing semantic relations among descriptions of failures and linking them with telemetry data (Esteban et al., 2022). The integration of telemetry with historical maintenance information is considered one of the most important yet most challenging aspects of contemporary predictive systems.

Maintenance records are often incomplete, unstandardized, and ambiguous, but combining them with high-frequency signals significantly improves prediction accuracy. Incorporating information on past failures, inspection schedules, and component replacements reduces the number of false alarms and enables distinguishing between natural degradation and sudden faults (Ayvaz & Alpay, 2021). This highlights the need for advanced text-processing methods and representation-level data fusion.

However, research covering the full spectrum of maintenance data available in CMMS/ERP systems under real serial-production conditions remains limited. Most studies rely on laboratory experiments or data from single machines, which restricts the generalizability of results. The literature also points to issues such as model overfitting and insufficient testing of robustness with respect to variability in production processes (Mallioris et al., 2024).

The literature review shows that although contemporary approaches to predictive maintenance in serial production environments are becoming increasingly advanced, significant research gaps remain. A key direction for further development is the comprehensive integration of telemetry data and historical maintenance information using modern deep-learning architectures capable of processing multimodal datasets.

The use of machine learning techniques in industrial diagnostics has evolved from simple statistical models to advanced deep learning architectures. Zhang et al. (2019) presented a systematic review of neural network applications in predictive maintenance, demonstrating the advantage of deep models over traditional classification methods.

Recurrent models such as LSTM (Long Short-Term Memory) and Gated Recurrent Unit (GRU) (Cho et al., 2014), which are capable of modeling long-term dependencies in time series, achieve particularly promising results (Li et al., 2024; Lei et al., 2018; Zhang et al., 2019). Zhao et al. (2019) showed that LSTM models achieve significantly better results in predicting rolling bearing failures compared to conventional spectral analysis methods.

A characteristic feature of failure prediction tasks is a significant imbalance of classes - failures occur relatively rarely compared to the normal functioning of machines (Johnson & Khoshgoftaar, 2019). Johnson and Khoshgoftaar (2019) conducted a detailed analysis of the impact of imbalance on the quality of classification models, pointing to the need to use specialized techniques such as focal loss or weighted loss functions. Lin et al. (2017) proposed focal loss as a solution to the problem of imbalance in object detection tasks, and later studies have shown the effectiveness of this method also in industrial applications (Li et al., 2024; Johnson & Khoshgoftaar, 2019; Lin et al., 2017).

Materials and Methods

The research was conducted in a manufacturing enterprise specializing in the serial production of automotive components. The plant operates as a high-mix, high-volume environment in which process stability, short cycle times, and stringent quality requirements are essential due to automotive-sector compliance constraints. The production system is organized into over 100 heterogeneous workstations representing multiple manufacturing technologies and distinct failure modes, which makes the facility a representative and technically demanding case for predictive maintenance research. The workstation population comprises four principal technology groups:

- Welding stations ($n = 35$): including MIG/MAG, TIG, and resistance welding systems. These workstations typically integrate power sources, wire feeders (where applicable), cooling units, fixturing mechanisms, and safety interlocks. Their operational risk profile is shaped by high thermal loads, cyclic electrical stress, consumable wear, and process sensitivity to parameter drift.
- Spot welding stations ($n = 28$): dedicated spot-welding cells characterized by high-current, short-duration welding cycles and repeated mechanical actuation. Typical degradation mechanisms include electrode wear, transformer and contactor aging, cooling inefficiencies, and mechanical misalignment. The cyclic and high-energy nature of spot welding makes these stations particularly suitable for time-series pattern recognition and anomaly detection.
- Adhesive application stations ($n = 22$): responsible for dispensing and applying structural adhesives. These stations combine dosing equipment, pressure

regulation, nozzle systems, curing-related constraints, and, in many cases, precision positioning. Failure modes commonly involve clogging, viscosity changes, pressure instability, dosing drift, and temperature-dependent process variation. As a result, both process and thermal telemetry are relevant for early detection of quality-relevant deviations.

- Milling stations ($n = 18$): CNC machine tools used for machining operations. These stations introduce failure mechanisms linked to spindle and bearing health, tool wear, axis drive performance, lubrication conditions, and vibration-related phenomena. The complexity of CNC systems and their sensitivity to mechanical dynamics provides a strong motivation for high-frequency monitoring and feature-based condition assessment.

This technology diversity creates a broad spectrum of operational regimes and degradation processes, which is advantageous for evaluating the generalizability of predictive maintenance methods across distinct asset classes.

Each workstation is equipped with a dedicated telemetry monitoring system capable of recording operating parameters in real time. The monitoring infrastructure enables continuous observation of mechanical, thermal, electrical, and process-related signals and supports high-resolution time-series analysis. From a methodological standpoint, this setup provides two critical advantages:

- it enables the extraction of condition indicators and early warning signatures prior to failure, and
- it facilitates cross-station comparability by ensuring consistent data acquisition at the level of individual workstations.

In parallel with telemetry, the company maintains a comprehensive Computerized Maintenance Management System (CMMS) that includes a complete history of service interventions since 2019. The CMMS repository serves as the primary operational ground truth for maintenance and reliability analytics, as it contains documented service requests, corrective and preventive actions, downtime-related information, and repair context. This historical record enables systematic reconstruction of failure events and maintenance timelines, supports calculation of reliability indicators, and provides the organizational context required to interpret telemetry-based anomalies as actionable maintenance insights.

Overall, the combination of a large, heterogeneous workstation base, continuous telemetry at the station level, and a multi-year CMMS intervention history creates a robust empirical foundation for developing and validating predictive maintenance models under realistic industrial conditions.

Predictive Maintenance Framework and Data Foundations

The predictive maintenance framework was designed as a comprehensive five-layer pipeline that integrates data acquisition, processing, machine learning, and

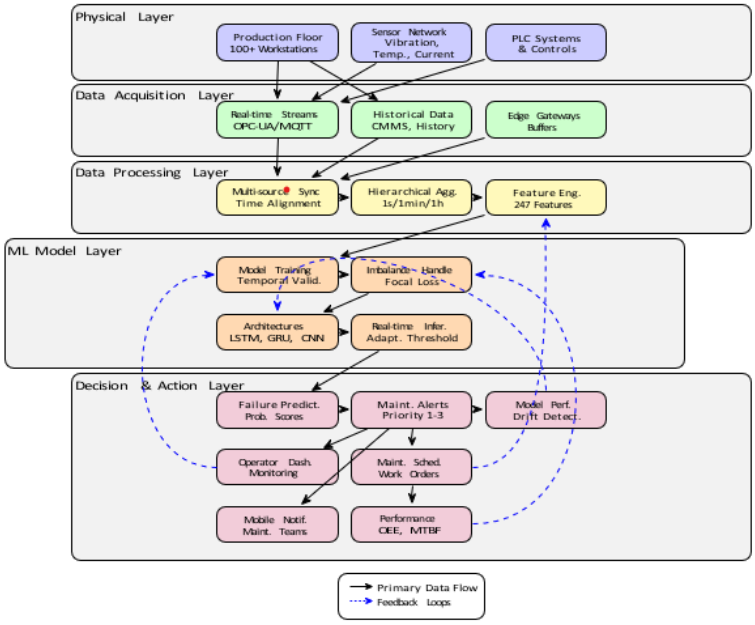

Fig. 1. Simplified system architecture of the predictive maintenance framework

decision support systems. Figure 1 illustrates the complete system architecture, showing how data flows from physical sensors through processing stages to actionable maintenance insights. The architecture addresses the key challenges of multi-source data integration, temporal pattern recognition, and operational decision-making in industrial environments. Each layer serves a distinct function in transforming raw telemetry data into reliable failure predictions.

To support this architecture, the analysis is grounded in two complementary categories of data that jointly capture both the long-term maintenance history and the short-term operational behavior of the investigated assets. The first category represents *Asset Management / Maintenance & Reliability* information collected over multiple years and reflecting organizational maintenance processes, failure reporting practices, and cost structures. The second category consists of *real-time telemetry* streams describing the physical state of the equipment at high temporal resolution. Together, these datasets enable analyses that span from reliability indicators and standardized failure coding to condition monitoring and early fault detection.

Data Sources and Structure

Label definition and sample generation

A key reproducibility element is an unambiguous label definition. In the CMMS, a "service request/work order" is an administrative record and does not necessarily correspond to an actual breakdown. Therefore, we define the prediction target at the level of consolidated failure episodes.

Failure definition (CMMS → failure episode)

A CMMS record is considered failure-related if it meets all criteria:

- $work\_order\_type \in$ {Corrective maintenance, Breakdown};
- $unplanned\_downtime\_minutes \geq 10$ (downtime measured by the production reporting system and written back to CMMS);
- $work\_order\_status$ = Closed (finalized);
- the activity is not a planned PM, inspection, calibration, adjustment, or quality-related intervention (these remain non-failure events).

The failure timestamp is defined as $t\_fail\_start = downtime\_start\_time$ when available; otherwise, the CMMS "reported/created time" is used as a fallback.

Consecutive failure-related work orders for the same workstation are consolidated into a single failure episode if their t_fail_start values are within 90 minutes (to merge multiple tickets opened for the same stoppage). Using this rule, 8,932 failure-related work orders collapse into 1,138 unique failure episodes across 103 workstations.

Prediction target

For each workstation and each evaluation time t (the end of the observation window), the label is:

$$y(t) = \begin{cases} 1 & \text{if } \exists \text{ a failure episode with start time} \\ & t\_fail\_start \in (t, t+H], \\ 0 & \text{otherwise.} \end{cases}$$

where H is the prediction horizon ($H = 4$ h in the main experiments; $H \in \{1, 4, 8, 24\}$ h reported in Section 4.4).

We focus on predicting the start of unplanned downtime (failure onset), not the work-order creation.

Thus, the prediction horizon H is encoded in the target definition rather than in the network architecture itself. For a fixed evaluation time t, the sigmoid output estimates the probability that at least one failure episode will start within the future interval $(t, t+\mathrm{H}]$.

Because labels are assigned on a fixed temporal grid, one consolidated failure episode may generate multiple positive samples at successive pre-failure evaluation points.

The prediction task is therefore binary rather than multiclass: the model predicts the risk of any failure occurrence within the specified horizon, regardless of its subsequent cause category, and does not output separate probabilities for mechanical, electrical/electronic, hydraulic/pneumatic, or control-system failures.

Changing the prediction horizon therefore requires relabeling the samples for a different future interval, while the network architecture remains unchanged.

From events to samples (positive/negative examples).

Samples are generated on a fixed time grid using a sliding-window scheme:

Observation window length: $W = 6$ hours of history (telemetry prior to t).

Stride: $\Delta = 15$ minutes (a new sample is generated every 15 minutes per workstation when the workstation is in production state).

Because the main prediction horizon is $H = 4$ h, a single consolidated failure episode may contribute up to 16 positive evaluation points on this grid, depending on telemetry coverage and machine operating state.

A sample is valid only if at least 80% of the minutes in $[t - W, t)$ have telemetry coverage for the critical channels (vibration, current, temperature); otherwise it is discarded.

Periods when the machine is already stopped for maintenance/downtime are excluded from sampling to avoid trivial leakage (no "prediction" is needed while the asset is already under intervention).

After filtering and resampling, the final dataset contains 3,520,800 samples, including 17,604 positive samples (0.50%) and 3,503,196 negative samples (99.50%), corresponding to a class-imbalance ratio of approximately 1:199.

Given the 15-minute evaluation grid and the $H = 4$ h forecasting horizon, this number of positive samples is consistent with the 1,138 consolidated failure episodes retained for modeling.

Prediction-time availability and strict time alignment

CMMS variables can easily introduce future leakage because some fields are filled or updated after the intervention (e.g., final cause code, actual downtime duration, spare-part costs). To ensure strict time alignment, we define a prediction-time snapshot at time t and use only information that would be known at or before t.

We explicitly restrict CMMS-derived model inputs to static asset metadata and features computed from past, completed events with timestamps $\leq t$.

We do not use any fields that depend on the outcome of the future failure or on post-repair bookkeeping.

In practice, all joins between telemetry and CMMS are performed as-of time t (last observation carried backward in time), and each engineered CMMS feature is computed using only records whose $event\_end\_time \leq t$.

Table 1 summarizes which CMMS fields are allowed at prediction time and how they are used.

The maintenance-management dataset used in this study provides a longitudinal view of asset performance and maintenance actions. It aggregates historical records extracted from the company's CMMS/EAM environment and associated business systems (e.g., parts procurement and cost accounting). The dataset contains both raw event records and derived reliability indicators:

- Service Requests: A total of 11,247 service request records spanning 2019–2023. Each record represents an operationally relevant maintenance interaction (e.g., fault report, initiation of corrective action, or execution of a service intervention). These records constitute the primary event log used to reconstruct failure timelines, quantify maintenance workload, and identify recurring failure patterns.
- MTBF (Mean Time Between Failures): MTBF indicators computed for each position (i.e., for each monitored unit, station, or asset location). These indicators summarize reliability in terms of expected operating time between successive failures and serve as a baseline for comparing reliability across positions and time periods.
- MTTR (Mean Time To Repair): MTTR indicators reported by failure type, enabling a differentiated view of maintainability. This breakdown supports identification of failure modes associated with prolonged downtime and helps prioritize interventions that reduce repair duration and operational disruption.

Table 1
Summary of CMMS fields allowed at prediction time.

CMMS field / feature	**Available at prediction time**	**Used in model**	**Leakage control**
Workstation ID, technology group, installation year	Yes	Yes	Static metadata
Last completed maintenance end time	Yes	Yes	only events with $end\_time \leq t$
Time since last completed maintenance	Yes (derived)	Yes	computed from past only ($end\_time \leq t$)
Count of failures in last 7/30 days	Yes (derived)	Yes	derived from consolidated episodes with $start \leq t$
Open work-order status		No	excluded (may correlate with imminent failure)
Planned PM schedule		No (in v2)	excluded to avoid schedule/decision leakage
Downtime minutes of the next event		No	post-factum outcome
Final ISO 14224 cause code		No	excluded from features, used only for analysis
Spare-parts cost, labor hours		No	used only for economic evaluation, not as features

- Failure cause classification (taxonomy adapted from ISO 14224 (ISO, 2016)): Failure causes were grouped using an internal coding scheme aligned, where applicable, with ISO 14224 principles of reliability and maintenance data structuring. The use of standardized coding ensures consistent labeling across assets and time, strengthens reproducibility of reliability analyses, and facilitates aggregation of results at the level of failure mechanisms rather than ad hoc descriptions.
- Repair and spare parts costs: Economic variables covering repair costs and spare parts expenditures, enabling linkage between technical events and financial impact. These data support cost-of-failure modeling, justification of predictive maintenance investments, and optimization of maintenance planning with respect to total cost of ownership.
- Preventive maintenance schedules: Planned preventive maintenance activities (periodic inspections, servicing intervals, and scheduled tasks). These schedules provide context for interpreting failure occurrence (e.g., failures shortly after PM execution) and enable evaluation of preventive policy effectiveness and compliance.

Overall, this dataset is predominantly event-based and historical, combining timestamps, categorical descriptors (failure types and ISO 14224 causes), and operational/economic attributes. It is well-suited for reliability assessment, maintenance strategy evaluation, and cost-impact analysis.

Mechanical parameters:

- Vibration (3-axis accelerometer signals sampled at 10 kHz) supporting detection of bearing defects, imbalance, misalignment, looseness, and other mechanical anomalies through spectral and time-domain features, as well as loads measured by strain gauges sampled at 1 kHz, capturing dynamic forces and load profiles relevant to fatigue accumulation, overload events, and abnormal mechanical resistance.
- Thermal parameters: Temperatures of critical components, including bearings, motors, and cooling systems, measured using K-type thermocouples at 1 Hz, with thermal trends used to identify overheating, lubrication degradation, cooling inefficiencies, and progressive friction-related faults.
- Electrical parameters: Motor currents, supply voltages, and power consumption sampled at 1 kHz, providing insight into electromechanical load conditions, motor health, and control-system behavior, and supporting detection of overload, phase imbalance, power-quality disturbances, and abnormal drive response.

In contrast to the maintenance dataset, telemetry is continuous, high-dimensional, and time-dependent, requiring careful synchronization, filtering, and feature extraction. It is primarily used for condition monitoring, temporal pattern discovery, anomaly detection, and the identification of precursors to failure events.

The two data categories play distinct but synergistic roles in the predictive maintenance pipeline. Asset Management records provide ground-truth operational outcomes (failures, interventions, downtime, and costs) and enable structured interpretation through standardized cause coding (ISO 14224). Telemetry provides observable physical signatures of degradation and operational stressors that precede or accompany those outcomes. Their integration supports mapping what happened (maintenance and failure events) to how it evolved (signal-level behavior), thereby enabling both reliability-centered analysis and data-driven failure prediction within the five-layer architecture.

At the sample-construction stage, telemetry and maintenance data were integrated through an as-of temporal join keyed by workstation identifier and evaluation time t.

For each workstation, minute-level telemetry from the interval $[t - W, t)$ formed the dynamic model input, whereas CMMS-derived variables were computed exclusively from records with timestamps $\leq t$.

These variables included static asset metadata, time since the last completed maintenance event, and counts of recent failures in the last 7 and 30 days.

The prediction label was then assigned by checking whether a consolidated failure episode started within the future interval $(t, t + H]$.

This integration strategy ensured strict temporal consistency, fused short-term sensor behavior with long-term maintenance history, and prevented future-information leakage from post-failure records.

Preprocessing and Feature Engineering

The preprocessing stage was designed to ensure consistent learning inputs across heterogeneous telemetry sources and to mitigate the strong imbalance between failure and normal-operation observations.

The primary technical challenge concerned synchronization of multi-source time series acquired at different sampling rates; this was addressed through a hierarchical temporal aggregation scheme producing aligned representations at three resolutions.

Each prediction sample was generated on a fixed temporal grid. For every workstation, an evaluation point t was created every 15 minutes (stride $\Delta = 15$ min), and the model input consisted of the preceding $W = 6$ hours of telemetry history, i.e., the interval $[t-6\ \mathrm{h}, t)$.

Thus, each sample represented a rolling six-hour observation window used to predict whether a failure would occur within the subsequent prediction horizon H.

This sliding-window design reflects the intended online deployment scenario, in which the model is re-evaluated repeatedly during machine operation:

- Level 1 (1 s), where high-frequency channels were first aggregated into 1-second descriptors. For vibration and current signals, the 1-second representation included mean, standard deviation, minimum, maximum, RMS, peak-to-peak value, kurtosis, crest factor, and band-energy statistics, thereby preserving short transient behavior while reducing the dimensionality of raw kHz signals;
- Level 2 (1 min), where all telemetry streams were mapped onto a common 1-minute grid. At this level, the model encoded short-term operational dynamics using sliding-window summaries such as rolling mean, rolling standard deviation, short-term linear trend, and local autocorrelation measures;
- Level 3 (1 h), where long-horizon descriptors were computed over trailing one-hour windows to capture progressive degradation, slow thermal drift, and broader variability patterns that are not visible at finer temporal resolutions.

Synchronization across heterogeneous sampling rates was performed in two stages. First, high-frequency telemetry (e.g., vibration at 10 kHz and electrical/current channels at 1 kHz) was reduced to 1-second summaries using fixed non-overlapping one-second bins. Second, all channels, including low-frequency thermal measurements sampled at 1 Hz, were resampled to a common 1-minute time grid. Timestamp alignment relied on the edge-gateway clock synchronized via NTP; when minor timestamp offsets occurred, nearest-neighbor alignment with a tolerance of 250 ms was used at the 1-second stage, followed by exact aggregation into 1-minute bins. This procedure ensured that all modalities contributed temporally aligned features to the downstream models.

On top of the synchronized streams, a feature engineering pipeline generated 247 engineered features from raw signals, including

- time-domain features (windowed statistics, linear trends, autocorrelation measures),
- frequency-domain descriptors (FFT-based spectral characteristics and energy distributed across frequency bands),
- nonlinear indicators (entropy measures, correlation dimension, and Lyapunov-exponent proxies), and
- contextual features integrating maintenance and production context (time since last maintenance, cycle counts, and workload intensity).

The three temporal resolutions were fused by early feature concatenation.

Concretely, at each minute τ within the six-hour observation window, a unified feature vector $x(\tau)$ was constructed by concatenating: (i) Level-1 descriptors pooled from the underlying 1-second statistics over the current minute, (ii) Level-2 features computed directly on the current 1-minute bin, and (iii) Level-3 descriptors obtained from the trailing 1-hour context ending at τ.

As a result, each minute was represented by a 247-dimensional multimodal feature vector, and each model input consisted of a sequence of 360 consecutive minute-level vectors (6 h $\times$ 60 min), i.e., an input tensor of shape 360×247 per sample.

Since failure events were rare relative to nominal operation (approximately 1:200), class imbalance was handled using complementary techniques. For the 1D-CNN model, focal loss (Lin et al., 2017) was used to emphasize hard minority-class examples by down-weighting well-classified instances. In the binary setting, let $y \in \{0, 1\}$ denote the ground-truth label and let $p = \sigma(z)$ be the sigmoid output of the network, interpreted as the predicted probability of failure within horizon H. Let p_t denote the probability assigned to the true class, i.e., $p_t = p$ when $y = 1$ and $p_t = 1 - p$ when $y = 0$. The focal loss is defined as $\mathrm{FL}(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t)$, where α controls class weighting and γ is the focusing parameter. In our experiments, $\gamma = 2.0$ and $\alpha = 1.0$. For the LSTM and GRU models, class imbalance was handled using weighted binary cross-entropy with class weights inversely proportional to class frequencies. In addition, adaptive decision thresholding was used to calibrate alert thresholds to position-specific downtime costs, thereby aligning classification outputs with operational risk and economic impact.

Model Architectures and Temporal Validation Protocol

To capture failure precursors embedded in multivariate time series and to ensure methodological robustness under real production conditions, three neural architectures – LSTM, GRU, and 1D-CNN – were designed and evaluated using a strictly chronological validation scheme. The recurrent models (LSTM and GRU) were selected for their ability to represent long-range temporal dependencies and state evolution, which is essential for predictive maintenance where degradation signatures may develop gradually over extended horizons. In both recurrent variants, the model input was a multivariate sequence of 360 time steps, each represented by 247 engineered features (i.e.,

a 360×247 tensor per sample). Before entering the recurrent blocks, each 247-dimensional minute-level feature vector was projected through an embedding layer mapping $247 \rightarrow 128$ dimensions, reducing redundancy and stabilizing training by providing a compact latent representation at every time step. The LSTM architecture comprised two stacked LSTM layers with 128 hidden units each and dropout of 0.30, followed by a fully connected (dense) layer with 64 units and ReLU activation to model nonlinear combinations of temporal representations, and finally a sigmoid output layer producing a probabilistic estimate of failure risk (Hochreiter & Schmidhuber, 1997). Including the $247 \rightarrow 128$ input projection layer, both recurrent layers, and the dense classification head, the LSTM architecture comprised 304,257 trainable parameters.

Complementary to recurrent modeling, a 1D convolutional neural network (1D-CNN) was implemented to learn local temporal motifs and short-term patterns (e.g., bursts, oscillatory fragments, transient deviations) that may not require explicit long-memory mechanisms. This model consisted of three 1D convolutional layers with 32, 64, and 128 filters and kernel size 3 in each layer, respectively, each followed by max pooling to progressively compress temporal resolution while retaining salient activations. A global average pooling stage was applied to obtain a fixed-size representation with reduced risk of overfitting compared to flattening, followed by a dense layer of 128 ReLU units and a sigmoid output layer for binary risk estimation. The resulting 1D-CNN architecture comprised 71,297 trainable parameters.

Given the temporal nature of both telemetry and maintenance outcomes, model assessment employed a time series split to prevent leakage and to reflect realistic deployment conditions in which models are trained on past data and applied to future operating periods. The dataset was partitioned chronologically into Training (January 2019–June 2022; 70%), Validation (July 2022–December 2022; 10%), and Test (January 2023-December 2023; 20%) subsets. This protocol supports reliable generalization assessment by ensuring that hyperparameter selection, early stopping, temperature-scaling calibration, and threshold optimization are all performed on historically subsequent validation data relative to training, while final performance is reported on an entirely future test window, thereby approximating production forecasting behavior and quantifying predictive stability under non-stationary operating regimes.

Training setup and missing-data handling

All neural models were trained using the AdamW optimizer with weight decay $= 1e-4$ and an initial learning rate of $1e-3$. Learning-rate adaptation was controlled by a ReduceLROnPlateau scheduler monitoring validation AUPRC, with reduction factor 0.5, patience = 4 epochs, and a minimum learning rate of 1e-5. The batch size was set to 128 samples, and each model was trained for up to 60 epochs.

Early stopping was applied based on validation AUPRC with patience = 10 epochs, and gradient clipping with a maximum norm of 1.0 was used to stabilize recurrent training. Dropout regularization was set to 0.30 for LSTM and 0.25 for GRU, while additional L_2-type regularization was provided through AdamW weight decay. For reproducibility, all experiments used fixed random seeds (Python, NumPy, and PyTorch seed = 42). To verify stability, the main models were additionally re-trained for three independent runs with seeds $\{42, 123, 2024\}$, yielding a test AUPRC variability within ± 0.006.

The models were implemented in PyTorch and trained on a single NVIDIA RTX 4090 GPU (24 GB VRAM). Depending on architecture, end-to-end training time ranged from approximately 55 to 75 minutes per model on the full training set.

Missing telemetry values were handled according to gap duration and signal criticality. Short gaps up to 2 minutes were imputed using forward-fill on the 1-minute grid. Medium gaps between 2 and 10 minutes were imputed using the sensor-wise median calculated exclusively on the training split. Long gaps exceeding 10 minutes were treated as structurally missing; if more than 20% of the minutes within a 6-hour observation window lacked critical telemetry (vibration, current, or temperature), the sample was discarded. In addition, binary missingness masks were appended for each channel group so that the models could distinguish between true healthy low-variance behavior and artificial absence of signal.

Downtime-cost-based threshold optimization

In the proposed deployment setting, failure prediction is treated not only as a classification task but also as a cost-sensitive maintenance decision problem. Let $p(x, t)$ denote the calibrated predicted probability that a failure episode will occur within the next prediction horizon H for workstation w at time t.

An alert is issued when the expected cost of inaction exceeds the expected cost of preventive in-

tervention. Downtime-related false-negative cost, denoted $C_{\mathrm{FN}}(w,t)$, was estimated as the product of the workstation-specific hourly downtime rate and the expected duration of an unplanned stoppage.

The hourly downtime rate was derived from internal production-controlling estimates reflecting throughput loss, contribution margin, and line-stoppage penalty risk. These values differed by workstation technology and were optionally adjusted by a time-varying multiplier reflecting operating context (day shift = 1.00, night shift = 0.85, weekend = 1.25).

Expected downtime duration was approximated by the historical mean MTTR for the corresponding workstation group. The false-positive intervention cost, denoted $C_{\mathrm{FP}}(w,t)$, included the labor and consumables needed for inspection as well as the cost of a short planned stoppage.

Planned preventive stoppage was assumed to last 0.5 h for welding and milling stations and 0.4 h for adhesive stations, based on historical maintenance logs. Under calibrated probabilities, the analytically optimal threshold is given by $\tau(w,t) = C_{\mathrm{FP}}(w,t)/(C_{\mathrm{FP}}(w,t) + C_{\mathrm{FN}}(w,t))$.

In addition to this analytical formulation, the threshold was also verified empirically by sweeping τ from 0.01 to 0.99 on the validation split and selecting the value minimizing realized expected cost.

The analytically derived and empirically selected thresholds were closely aligned, confirming the consistency of the cost model with observed operational outcomes.

Accordingly, the final operating threshold for each workstation group was selected on the validation split by minimizing expected cost over a grid of candidate values $\tau \in [0.01, 0.99]$, with the analytical expression $\tau = C_{\mathrm{FP}}/(C_{\mathrm{FP}} + C_{\mathrm{FN}})$ used as the initial reference point.

Results

During the observation period (2019–2023), a total of 11,247 CMMS service requests/work orders were recorded.

Among them, 8,932 records were failure-related (corrective/breakdown interventions with unplanned downtime $\geq$ 10 minutes).

Because multiple work orders may refer to the same physical stoppage, we consolidated failure-related work orders into 1,138 unique failure episodes by merging events occurring within 90 minutes on the same workstation (Section 3.2.1).

All predictive labels are defined at the failure-episode level (failure start within the next H hours), while the remaining CMMS records correspond to minor issues, inspections, adjustments, or planned preventive actions that do not meet the "failure" criterion.

Failure events were categorized into four dominant groups reflecting the underlying physical domain and typical maintenance competencies.

The distribution is summarized in Table 2 and indicates a clear prevalence of mechanical failures (45.2%), followed by electrical/electronic (28.7%), hydraulic/pneumatic (16.3%), and control-system-related failures (9.8%).

Table 2
Failure category distribution in the CMMS dataset (n = 8,932 failure-related CMMS records).

Failure category	Share of failures [%]	Interpretation (typical examples)
Mechanical	45.2	wear, misalignment, bearings, fixtures, tooling mechanics
Electrical / electronic	28.7	drives, sensors, wiring, power modules, contactors
Hydraulic / pneumatic	16.3	leaks, pressure instability, valve faults, actuator issues
Control system	9.8	PLC alarms, interlocks, logic issues, communication faults

Table 2 presents a post hoc descriptive breakdown of failure categories within the positive class; it does not represent the classifier's label space.

From a reliability perspective, the machine park exhibited an average MTBF = 168.3 h and MTTR = 2.4 h (aggregated across positions).

These baseline indicators provide a reference point for both predictive-model evaluation and the later operational impact assessment presented in Section 4.5 and 4.6.

Comparative Performance of Predictive Models

Model performance was evaluated on the chronologically separated test set (January 2023–December 2023). Given the rare-event nature of the problem, discrimination metrics were reported using AUROC and

AUPRC (the latter being particularly informative under class imbalance), along with threshold-dependent measures ($F1$, precision, recall). The core discrimination results are summarized in Tab. 3 and Fig. 2, while deployment-oriented evaluation artifacts (precision-recall curves, threshold sweeps, and calibration analysis) are reported in the following subsection.

Table 3
Comparative model performance on the test set (January–December 2023).

Model	AUROC	AUPRC	$F1$-score	Precision	Recall
LSTM	0.924	0.847	$F_1 = 0.823$	0.798	0.849
GRU	0.918	0.834	$F_1 = 0.811$	0.785	0.838
1D-CNN	0.897	0.798	$F_1 = 0.774$	0.743	0.807
Random Forest	0.856	0.721	$F_1 = 0.698$	0.672	0.726
SVM	0.834	0.689	$F_1 = 0.661$	0.631	0.694

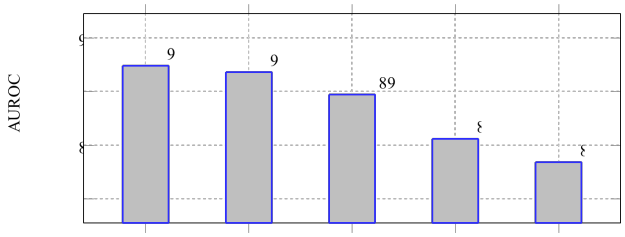

Fig. 2. Comparison of model performance using Area Under ROC Curve (AUROC).

The LSTM architecture achieved the strongest overall performance (AUROC = 0.924; AUPRC = 0.847; $F_1 = 0.823$), followed closely by GRU.

The 1D-CNN yielded competitive recall but lower precision and overall F_1.

Classical models (Random Forest, SVM) showed substantially weaker discrimination, suggesting that temporal representation learning provides a measurable advantage for early-failure pattern recognition in this industrial context.

LSTM GRU. 1D-CNN Random Forest SVM

Higher values correspond to stronger discriminative performance between failure and normal operation classes. Among the evaluated models, LSTM achieved the highest AUROC score of 0.924, followed by GRU with 0.918 and 1D-CNN with 0.897.

To complement the scalar performance metrics reported in Table 3, Table 4 presents confusion matrices for the three neural architectures on the chronologically separated test set (January–December 2023). Binary predictions were obtained using the same validation-selected operating threshold described in Section 3.6 (i.e., the operating point used to compute precision, recall, and $F1$-score in Table 3). The positive class corresponds to the occurrence of at least one failure episode within the prediction horizon.

Table 4
Confusion matrices on the 2023 test set (January–December 2023). Rows denote true classes and columns denote predicted classes.

LSTM	Predicted: no failure	Predicted: failure
Actual: no failure	$TN = 524{,}912$	$FP = 567$
Actual: failure	$FN = 399$	$TP = 2{,}242$
GRU	Predicted: no failure	Predicted: failure
Actual: no failure	$TN = 524{,}872$	$FP = 607$
Actual: failure	$FN = 428$	$TP = 2{,}213$
1D-CNN	Predicted: no failure	Predicted: failure
Actual: no failure	$TN = 524{,}742$	$FP = 737$
Actual: failure	$FN = 510$	$TP = 2{,}131$

Consistent with Table 3, the LSTM model exhibits the most favorable error structure on the test set, combining a high true-positive count with comparatively fewer missed failures (false negatives) and a limited number of false preventive alerts (false positives). The GRU model shows a similar but slightly weaker profile, whereas the 1D-CNN produces more missed failures and more false alerts, in line with its lower precision and F1-score.

Receiver operating characteristic (ROC) analysis on the chronologically separated test set confirmed the ranking observed in Table 3. The LSTM model achieved the highest AUROC (0.924), followed by GRU (0.918) and 1D-CNN (0.897), while the classical baselines reached lower values (Random Forest: 0.856; SVM: 0.834). Across the evaluated models, the ROC profile of LSTM remained the most favorable, indicating the best trade-off between sensitivity and false-positive rate in this industrial setting. These ROC characteristics are consistent with the precision, recall, and $F1$-score values reported in Table 3 and further support the use of LSTM as the main deployment candidate.

Deployment-oriented evaluation: precision-recall, threshold sweeps, and calibration

Because predictive maintenance is ultimately used to support maintenance decisions under severe class imbalance, additional deployment-oriented evaluation was performed beyond AUROC. Consistent with the AUPRC values reported in Table 3, the LSTM model produced the strongest PR envelope, followed closely by GRU, while 1D-CNN showed a more pronounced loss of precision at higher recall levels.

Threshold sensitivity was analyzed for the LSTM model by sweeping the decision threshold from 0.10 to 0.70. The results, summarized in Table 5, show that thresholds in the range 0.30–0.50 provide the most balanced $F1$ values, whereas lower thresholds prioritize recall at the expense of an increased false-alert burden.

Table 5
Threshold sweep for the LSTM model on the 2023 test set.

Threshold	Precision	Recall	$F1$-score
0.10	0.54	0.97	$F_1 = 0.69$
0.15	0.62	0.95	$F_1 = 0.75$
0.20	0.70	0.92	$F_1 = 0.80$
0.25	0.75	0.89	$F_1 = 0.81$
0.30	0.78	0.87	$F_1 = 0.82$
0.35	0.80	0.85	$F_1 = 0.82$
0.40	0.82	0.83	$F_1 = 0.82$
0.50	0.86	0.78	$F_1 = 0.82$
0.60	0.89	0.70	$F_1 = 0.78$
0.70	0.92	0.60	$F_1 = 0.73$

This behavior is operationally relevant because different workstation classes may justify different operating points depending on the relative cost of missed failures and false preventive interventions.

Probability calibration was evaluated using a reliability curve and Expected Calibration Error (ECE, 10 bins).

The raw model outputs exhibited mild overconfidence (ECE = 0.041; Brier score = 0.094).

This confirms that the model probabilities are suitable for downstream cost-based thresholding rather than only for ranking.

Performance by Workstation Type

To assess robustness across heterogeneous processes, the LSTM model was further evaluated by workstation type. As shown in Table 6, predictive effectiveness varied across technologies, reflecting differences in sensor observability and the degree to which degradation manifests as stable precursors.

- Welding stations achieved the strongest results (AUROC = 0.941, $F_1 = 0.856$), consistent with repeatable degradation dynamics in electrodes and cooling subsystems.
- Milling stations also exhibited high predictability (AUROC = 0.938, $F_1 = 0.834$), primarily due to vibration- and load-related signatures associated with tool wear and spindle condition.
- Spot welding stations showed moderately reduced performance (AUROC = 0.915, $F_1 = 0.798$), likely due to process instability and complex electrode degradation effects.
- Adhesive stations produced the lowest results (AUROC = 0.887, $F_1 = 0.743$), which is consistent with polymerization and viscosity-related processes being less directly observable and more sensitive to contextual variables.

Table 6
LSTM performance by workstation type (test set).

Workstation type	AUROC	$F1$-score	Dominant predictive drivers (examples)
Welding stations	0.941	$F_1 = 0.856$	welding temperature, welding current, arc time
Milling stations	0.938	$F_1 = 0.834$	high-frequency vibration, axis loads, spindle temperature
Spot welding stations	0.915	$F_1 = 0.798$	electrical resistance, contact force, electrode temperature
Adhesive stations	0.887	$F_1 = 0.743$	application temperature, adhesive viscosity, gelation-related proxies

Influence of Prediction Horizon

The predictive horizon was systematically varied to quantify the trade-off between early warning time and classification accuracy. Table 7 shows that shorter horizons provide higher discrimination (AUROC and $F1$), while longer horizons extend warning time at the cost of reduced predictive precision.

Table 7
LSTM performance versus prediction horizon.

Prediction horizon	AUROC	$F1$-score	Average warning time
1 hour	0.952	$F_1 = 0.891$	47 minutes
4 hours	0.924	$F_1 = 0.823$	3.2 hours
8 hours	0.896	$F_1 = 0.776$	6.8 hours
24 hours	0.843	$F_1 = 0.687$	18.4 hours

At a 1-hour horizon, the model achieved the best performance (AUROC = 0.952; $F_1 = 0.891$) with an average warning time of 47 minutes.

At 4 hours, the model retained strong performance (AUROC = 0.924; $F_1 = 0.823$) while offering 3.2 hours of average warning time.

For horizons of 8–24 hours, predictive quality decreased, indicating that long-range forecasting is constrained by non-stationary operating conditions and the limited persistence of pre-failure signatures over extended time spans.

Cost-based threshold optimization and economic sensitivity

To translate probabilistic predictions into economically meaningful maintenance actions, calibrated model outputs were converted into alerts using workstation-specific cost-based thresholds. Table 8 summarizes the assumed downtime-rate parameters, mean restoration times, preventive-action costs, and the resulting optimized threshold values for the main workstation groups.

The results indicate that high-throughput bottleneck stations such as spot welding and welding justify lower alert thresholds because the economic consequence of missed failures is disproportionately large. In contrast, lower-cost processes such as adhesive application tolerate slightly higher thresholds without materially increasing total expected loss. Across workstation types, the optimized thresholds ranged from 0.20 to 0.24, which is lower than the globally $F1$-oriented operating point and therefore better aligned with the industrial objective of minimizing downtime-related losses. Using these cost-aware thresholds on the 2023 test period reduced estimated annual downtime-related loss by 9–12% relative to a single global threshold selected only for maximum $F1$. This confirms that threshold optimization should be treated as an explicit economic decision layer rather than as a purely statistical post-processing step.

Estimated operational and economic impact under a deployment scenario

A scenario-based deployment analysis indicated potentially meaningful operational and economic improvements. The reported 34% reduction in unplanned downtime was estimated in a counterfactual deployment analysis rather than measured in a controlled plant-wide trial.

Using the test-set alert outcomes together with workstation-specific mean time to repair (MTTR) and preventive-stop durations, expected downtime after deployment was computed as the sum of: (i) planned preventive stop time for true-positive and false-positive alerts, and (ii) residual unplanned downtime for false-negative cases.

Table 8
Workstation-specific downtime costs, preventive-action costs, and optimized decision thresholds.

Workstation group	Downtime rate (PLN/h)	Mean MTTR (h)	C_FN (PLN)	Planned stop (h)	Preventive labor+parts (PLN)	C_FP (PLN)	Optimized threshold
Spot welding	9500	2.5	$C\_FN = 23750$	0.5	1200	$C\_FP = 5950$	0.200
Welding	8800	2.2	$C\_FN = 19360$	0.5	1000	$C\_FP = 5400$	0.218
Milling	6200	2.0	$C\_FN = 12400$	0.5	900	$C\_FP = 4000$	0.244
Adhesive	4800	1.8	$C\_FN = 8640$	0.4	700	$C\_FP = 2620$	0.233

Formally,

$$DT^{\text{est}}_{\text{after}} = \sum_w (TP_w + FP_w) \cdot T_{\text{PM},w} + \sum_w FN_w \cdot MTTR_w.$$

OEE was estimated as OEE = Availability × Performance × Quality. Availability was adjusted using the scenario-based downtime reduction, while Performance and Quality were estimated from historical line-level production indicators.

Therefore, the reported OEE improvement should be interpreted as an indirect scenario-based estimate rather than a directly measured post-deployment value.

This value was compared with the historical baseline unplanned downtime DT_{baseline}, and the relative reduction was calculated as $1 - DT^{\text{est}}_{\text{after}}/DT_{\text{baseline}}$.

Therefore, the 34% value should be interpreted as a model-based estimate under the stated operational assumptions.

Under this deployment scenario, unplanned downtime was estimated to decrease by 34%, suggesting improved production continuity under the stated assumptions.

Additionally, the average repair time was estimated to decrease from 2.4 h to 1.6 h, indicating faster expected response and improved repair planning.

Annual maintenance costs were projected to decrease by 28%, as summarized in Table 9.

Table 9
Operational KPIs before and after system deployment.

KPI	Baseline	Estimated after deployment	Change
Unplanned downtime	100%	66%	−34%
MTTR (hours)	2.4 h	1.6 h	−0.8 h (−33%)
OEE	73%	84%	+11 pp
Annual maintenance cost	100%	72%	−28%

From an economic standpoint, the total estimated annual savings reached 2.3 million PLN, while the one-time implementation cost was 450 thousand PLN.

This corresponds to an approximate payback period of ≈ 2.35 months and a first-year benefit-to-cost ratio of ≈ 5.11, confirming that the proposed predictive approach is viable not only technically but also financially under the studied production conditions.

Discussion

The results obtained support the suitability of recurrent neural architectures for industrial failure prediction under real-world operating conditions.

Among the evaluated approaches, the LSTM model achieved the highest discriminative performance (AUROC = 0.924), indicating a strong capability to separate nominal operation from pre-failure states across heterogeneous workstations and operating regimes.

The corresponding F_1-score (0.823) should be interpreted as satisfactory given the pronounced class imbalance and the practical requirement to balance missed failures (false negatives) against excessive maintenance alerts (false positives).

In this context, the joint reporting of AUROC and F_1 is essential: AUROC captures threshold-independent ranking quality, whereas F_1 reflects operational performance at a selected decision threshold.

The observed performance advantage of LSTM over GRU, though moderate, is consistent with the hypothesis that degradation processes in complex production assets exhibit both short-term fluctuations and long-term temporal dependencies. The explicit memory cell and gating mechanisms of LSTM, particularly the forget gate, provide a structured means to retain or discard information over extended horizons. This property is beneficial when early degradation signatures are weak, intermittent, or masked by process variability, and when predictive performance depends on integrating evidence across multiple temporal scales. While GRU offers a more parameter-efficient alternative, the LSTM's additional representational capacity appears advantageous in capturing longer-range dependencies and gradual drift phenomena typical for wear-driven or thermally mediated failure mechanisms.

Role of Feature Engineering and Signal Modalities

The observed model behavior suggests that vibration- and temperature-related signals likely play a central role in predictive performance, while process and electrical variables provide complementary context. This interpretation is consistent with the physical nature of degradation in welding, milling, and other mechatronic workstations. However, because no formal explainability analysis (e.g., SHAP, permutation importance, or systematic ablation study) was performed in this work, these observations should be treated as interpretive rather than definitive.

These findings support the methodological decision to employ a multi-domain feature set. Mechanical and thermal variables appear to provide early fault signatures with strong physical interpretability, while process and electrical signals increase robustness by capturing failure pathways that do not manifest purely as vibration anomalies (e.g., control-related instability, increased cycle-time variance, or energy-efficiency drift). Importantly, the results also suggest that feature engineering remains a critical component even when using deep learning, particularly in industrial environments where raw signals are high-dimensional, noisy, and non-stationary, and where the downstream model benefits from domain-informed summarization.

Implementation Challenges in Industrial Deployment

Several practical challenges emerged during implementation, reflecting typical constraints of production-grade predictive maintenance systems. First, data quality issues were non-negligible: on the unified 1-minute grid, 2.9% of bins exhibited missing values in at least one critical telemetry channel, while long gaps exceeding 10 minutes occurred in approximately 0.6% of bins. These artifacts were associated with sensor dropouts, communication instability, buffering issues, and maintenance-related disconnections, and therefore required explicit imputation and sample-filtering rules as described in Section 3.5.

In industrial settings, such artifacts often result from sensor degradation, communication instability, or maintenance-related disconnections, and must be explicitly managed to prevent systematic bias and avoid false alarms.

Second, system latency requirements imposed constraints on algorithmic complexity and deployment architecture. Maintaining end-to-end latency below 30 seconds was essential to preserve operational usefulness, particularly for fast-developing faults or safety-relevant deviations. This requirement affected design choices related to windowing, feature computation, model inference, and the integration layer responsible for data routing and alert delivery.

Third, integration with existing automation infrastructure represented a major engineering effort. The plant environment included heterogeneous SCADA and PLC ecosystems, which typically vary across production lines, vendors, and generations. Ensuring consistent data access, time synchronization, and semantic alignment (e.g., consistent naming of signals and alarms) required dedicated interface design and systematic validation. These integration challenges are often underestimated in laboratory studies but strongly influence real-world adoption and scalability.

Limitations and Generalizability

The study should be interpreted considering several limitations that may affect generalizability. First, the empirical evaluation was conducted within a single manufacturing plant, characterized by a specific combination of technologies, organizational practices, and maintenance procedures. While the dataset is heterogeneous at the workstation level, cross-site variability (e.g., differing production cultures, suppliers, environmental conditions, and maintenance maturity) was not directly represented.

Second, external validation on additional plants, machine types, and production processes was not performed. As a consequence, direct transfer of the reported performance metrics to other industrial environments should be approached cautiously. Differences in sensor configurations, operating regimes, and failure taxonomies can materially influence both feature distributions and the temporal structure exploited by recurrent models.

Third, the test horizon covered 12 months, which provides a realistic operational window but may be insufficient to fully capture longer-term trends and infrequent failure modes, including low-probability, high-impact events. Finally, the analysis incorporated external and contextual factors (e.g., environmental variability and operator practices) only to a limited extent, although such factors can significantly modulate signal patterns, process stability, and failure development trajectories. Extending the framework to incorporate richer contextual variables and multi-site validation constitutes a natural direction for future work.

Conclusions

The study demonstrates that deep learning models, particularly LSTM, can provide strong predictive performance in industrial failure prediction tasks under chronologically separated evaluation.

In the analyzed setting, the LSTM model achieved AUROC $= 0.924$ and F_1-score $= 0.823$, indicating good discrimination between nominal operation and impending failures.

The results also highlight the importance of careful feature engineering, strict time alignment, and explicit handling of class imbalance in predictive maintenance applications.

A scenario-based deployment analysis further suggests that the proposed approach may yield meaningful operational and economic benefits under the assumed plant conditions. In particular, the model-based analysis projected reductions in unplanned downtime and maintenance-related costs. However, these values should be interpreted as estimated outcomes derived from counterfactual analysis rather than as measured results from a full-scale industrial deployment.

The findings of this study indicate practical potential for predictive maintenance in industrial environments, while also emphasizing the need for external validation, multi-site testing, and stronger explainability support before broader generalization.

References

Adryan, F.A. & Sastra, K.W. (2021). Predictive maintenance for aircraft engine using machine learning: Trends and challenges. *International Journal of Aviation Science and Engineering*, *3*(1), 37-44. 10.47355/AVIA.V3I1.45

Arockia Mary, P., Sharma, A., Dekka, S., Gowda, V.D., Singh, M. (2024). Deep Learning Approaches for Real-Time Data Analytics in IoT Sensor Networks. In: Universal Threats in Expert Applications and Solutions, *Lecture Notes in Networks and Systems, 1007*, 239-248. 10.1007/978-981-97-5146-4_21

Ayvaz, S. & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*, 114598. 10.1016/j.eswa.2021.114598

Benhanifia, A., Cheikh, Z.B., Oliveira, P.M., Valente, A. & Lima, J. (2025). Systematic review of predictive maintenance practices in the manufacturing sector. *Intelligent Systems with Applications*, 200501. 10.1016/j.iswa.2025.200501

Cardoso, D. & Ferreira, L. (2021). Application of predictive maintenance concepts using artificial intelligence tools. *Applied Sciences*, *11*(1), Article 18. 10.3390/app11010018

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.* In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724-1734. 10.3115/v1/D14-1179

Dilda, V., Mori, L., Noterdaeme, O., Schmitz, C. (2017). *Manufacturing: Analytics Unleashes Productivity and Profitability.* McKinsey & Company.

Esteban, A., Zafra, A. & Ventura, S. (2022). Data mining in predictive maintenance systems: A taxonomy and systematic review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *12*(5), e1471. 10.1002/widm.1471

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. 10.1162/neco.1997.9.8.1735

ISO. (2016). ISO 14224:2016 *Petroleum, petrochemical and natural gas industries – Collection and exchange of reliability and maintenance data for equipment.* International Organization for Standardization.

Jardine, A.K.S., Lin, D. & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510. 10.1016/j.ymssp.2005.09.012

Johnson, J.M. & Khoshgoftaar, T.M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*, Article 27. 10.1186/s40537-019-0192-5

Lei, Y., Li, N., Guo, L., Li, N., Yan, T. & Lin, J. (2018). Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, *104*, 799–834. 10.1016/j.ymssp.2017.11.016

Li, Z., He, Q. & Li, J. (2024). A survey of deep learning-driven architecture for predictive maintenance. *Engineering Applications of Artificial Intelligence*, *133*, 108285. 10.1016/j.engappai.2024.108285

Lin, T.-Y., Goyal, P., Girshick, R., He, K. & Dollár, P. (2017). Focal Loss for Dense Object Detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980-2988.

Mallioris, P., Aivazidou, E. & Bechtsis, D. (2024). Predictive maintenance in Industry 4.0: A systematic multi-sector mapping. *CIRP Journal of Manufacturing Science and Technology*, *50*, 80–103. 10.1016/j.cirpj.2024.02.003

Nunes, P., Santos, J. & Rocha, E. (2023). Challenges in predictive maintenance – A review. *CIRP Journal of Manufacturing Science and Technology*, *40*, 53–67. 10.1016/j.cirpj.2022.11.004

Siemens AG: The True Cost of Downtime (2024). *How Much Do Leading Manufacturers Lose Through Inefficient Maintenance*? Siemens (2024).

Taheri Hosseinkhani, N. (2025). *Economic Impacts of Artificial Intelligence Integration in Industry 4.0 Manufacturing Systems.* Auburn University. Available at: https://aurora.auburn.edu/handle/11200/50712 (accessed 26 Oct 2025).

Thomas, D., & Weiss, B. (2021). Maintenance costs and advanced maintenance techniques in manufacturing machinery: Survey and analysis. *International journal of prognostics and health management*, *12*(1), 10-36001. 10.36001/ijphm.2021.v12i1.2883

Zhang, W., Yang, D. & Wang, H. (2019). Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Systems Journal*, *13*(3), 2213–2227. 10.1109/JSYST.2019.2905565

Zhao, R. et al. (2019). Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, *115*, 213–237. 10.1016/j.ymssp.2018.05.050