IBIMA Publishing
Communications of International Proceedings
https://ibimapublishing.com/p-articles/47AI/2026/4716526/
Vol. 2026 (12), Article ID 4716526

Deep Learning vs. Machine Learning for Predictive Maintenance under Limited Industrial Data: A Cross-Machine Validation Case Study*

Łukasz PAŚKO and Galina SETLAK

Faculty of Mechanical Engineering and Aeronautics, Department of Computer Science,
Rzeszów University of Technology, Rzeszów, Poland

Correspondence should be addressed to: Łukasz PAŚKO, lpasko@prz.edu.pl

*Presented at the 47th IBIMA International Conference, 29-30 June 2026, Madrid, Spain

Abstract

Predictive maintenance (PdM) has become a strategic factor in modern manufacturing systems. While deep learning (DL) architectures such as Long Short-Term Memory (LSTM) models have demonstrated strong performance on large-scale industrial datasets, their effectiveness in PdM under limited data conditions remains unclear, creating an important gap in the literature. This study addresses that gap by investigating whether LSTM provides advantages over classical machine learning (ML) model, Random Forest (RF), in small-sample industrial environments. The study contributes to the literature by systematically evaluating cross-machine generalization under limited failure data conditions and offers managerial implications for ML and DL adoption in PdM. The empirical analysis focuses on five CNC machines and considers practical data challenges common in production systems: missing observations, irregular sampling intervals, and maintenance-induced downtime. A Leave-One-Machine-Out cross-validation strategy was implemented to evaluate models' performance. The findings show that, despite the temporal nature of the problem, the RF model outperformed LSTM across most evaluation metrics, achieving higher AUC and recall while exhibiting more stable cross-machine generalization. The results suggest that high predictive performance can be achieved without complex recurrent architectures when appropriate feature windowing and ensemble learning are employed. At the same time, LSTM achieved higher precision, indicating fewer false alarms. This highlights a managerial trade-off: RF may be preferable when maximizing fault detection is critical, whereas LSTM may remain attractive when minimizing unnecessary maintenance interventions is a priority. Overall, the study suggests that classical ML can provide a robust and computationally efficient alternative to DL in low-data industrial environments.

Keywords: predictive models' evaluation, failure prediction, sequence analysis.

Introduction

Maintaining production continuity and machine reliability is one of the top priorities for manufacturing plants. Production management has traditionally utilized several maintenance concepts. Jasiulewicz-Kaczmarek et al. (2023) mentioned that the most popular approach has been reactive maintenance, which involves responding only after a failure occurs. The second approach is Preventive Maintenance, which relies on predetermined schedules for inspections, component replacement, and repairs. These traditional maintenance strategies may become ineffective. They generate the risk of unplanned downtime or lead to the replacement of fully functional components, which is a waste of resources. The answer to these challenges is the currently most advanced Predictive Maintenance (PdM) strategy, which uses modern machine learning (ML) algorithms to forecast the technical condition of machines in real time. In the book by Knosala (2017), PdM is an approach based on

Cite this Article as: Łukasz PAŚKO and Galina SETLAK Vol. 2026 (12) "Deep Learning vs. Machine Learning for Predictive Maintenance under Limited Industrial Data: A Cross-Machine Validation Case Study " Communications of International Proceedings, Vol. 2026 (12), Article ID 4716526,
https://doi.org/10.5171/2026.4716526

observation of processes and machine condition to prevent failures and predict the timing of necessary preventive maintenance.

In recent years, deep learning (DL) has gained significant traction in industrial and manufacturing applications, particularly in the domain of PdM. Recurrent neural networks, including Long Short-Term Memory (LSTM) architectures, have been widely adopted due to their ability to model temporal dependencies in sensor data. Numerous review studies have highlighted the growing importance of DL in Industry 4.0 environments, emphasizing its effectiveness in fault detection, remaining useful life estimation, and anomaly detection (Zhao et al., 2019; Serradilla et al., 2022). In addition, according to Carvalho et al. (2019), classical ML techniques are also used in PdM. An example of such techniques is the ensemble learning method known as random forest (RF).

This study deliberately focuses on a data-constrained scenario, as many small and medium-sized manufacturing enterprises operate under similar conditions. In such realistic deployment context, large labeled failure datasets are rarely available. Additionally, this study examines data with numerous shortcomings, characteristic for production processes. Issues such as uneven time intervals between recorded data points, missing individual observations resulting from incorrect measurement equipment recordings, and long episodes of missing data during machine downtime are present in the analyzed data. These issues can also impact the prediction quality of DL and ML models. Therefore, this paper tries to answer the questions of whether DL actually offers an advantage over ML in the presence of limited and challenging industrial data as well as which kind of data-driven methods gives more stable results.

The remaining of this paper is structured as follows. Section 2 reviews the literature to identify research gaps in PdM and data-driven predictive methods. Section 3 presents the research questions, the analyzed dataset, and the experimental methodology used. Section 4 describes the results of the accuracy and stability of the developed models, as well as the managerial implications. Section 5 summarizes the study and proposes further research.

Literature Review

PdM in Manufacturing Systems

Within Industry 4.0, PdM aims to predict machine failures to minimize downtime and optimize operating costs. PdM uses advanced data analysis techniques to make decisions about service interventions precisely when they are necessary. Modern PdM systems increasingly rely on artificial intelligence and ML. This accelerates data collection process and reduces the cost of acquiring information. Dalzochio et al. (2020) mentioned that this process is based on the integration of two main pillars:

- ML: using algorithms to detect patterns and anomalies in data streams obtained from machine and device sensors,
- Reasoning: allowing the interpretation of data analysis results and detected dependencies in the context of expert knowledge and the system's ontology, which increases the understandability of decisions made by autonomous systems.

PdM can serve different objectives depending on the analytical perspective:

- prediction whether specific machine settings or environmental conditions lead to a failure,
- prediction whether a failure will occur within a specified future time horizon t,
- prediction a failure's type (multi-class classification),
- estimation of remaining useful life.

From a data analysis perspective, first and second approaches are binary classification problems, where the positive class represents the occurrence of a failure and the negative class indicates normal operation. The key difference lies in the construction of the target variable. In failure prediction within a future time window t, the target variable includes a larger number of positive observations, covering not only the failure event itself but also the period t preceding the failure.

RF and LSTM in PdM

One of the classical ML techniques commonly applied in PdM is the RF algorithm (Breiman, 2001). RF is a widely used ML method that relies on a collection of decision trees. This ensemble method combines multiple models to improve prediction and stability.

Studies comparing RFs with other ML approaches indicate their superiority in failure prediction tasks (Yang and Wang, 2025). By aggregating decision trees, it performs well in classification and regression. Its effectiveness in fault diagnosis has been widely demonstrated in industrial applications, e.g. in (Cerrada et al., 2026; Li et al., 2016). The combination of bootstrap aggregation (bagging) and random feature selection enables RFs to generalize better than individual decision trees, reducing variance and mitigating overfitting. RFs perform well when dealing with large numbers of observations and input variables (high-dimensional datasets), making them suitable for industrial environments with multiple sensors (Breiman, 2001). In addition, RFs can partially handle missing values through surrogate splits and internal mechanisms, reducing the need for extensive preprocessing (Géron, 2019). Another advantage is automatic feature importance estimation. RFs provide measures of feature importance, enabling identification of the most influential variables in predicting failures, which is particularly valuable in industrial diagnostics (Louppe et al., 2013).

Among DL approaches used in PdM, LSTM neural networks are particularly prominent. LSTM models are designed specifically to overcome the limitations of traditional Recurrent Neural Networks. The goal of LSTM is to enable the modeling of long-term dependencies in sequential data (Hochreiter and Schmidhuber, 1997). LSTM models have been successfully applied to a wide range of problems. They are used widely in time series forecasting and in sequential data prediction problems.

The applicability of LSTM models in PdM is supported by their characteristics. LSTM models are specifically designed to capture long-term dependencies in time series data, making them well-suited for modeling degradation processes in industrial systems (Hochreiter and Schmidhuber, 1997). Their effectiveness in remaining useful life estimation has been confirmed in numerous studies. LSTM models demonstrate strong performance in prognostics tasks, particularly based on sensor data (Wu et al., 2021). Moreover, they can learn directly from raw sequences. Therefore, LSTM models can operate on time-series data without the need for extensive feature engineering (Zhao et al., 2018). In addition, they can capture nonlinear and complex system behavior. This capability is particularly important for modeling relationships in multivariate time series typical of industrial processes (Zhao et al., 2019). Despite their advantages, LSTM models typically require large amounts of high-quality, regularly sampled data to achieve optimal performance.

Despite the rapid growth of DL applications in PdM, a significant portion of the literature focuses on large-scale, high-quality datasets. In contrast, real-world industrial datasets are frequently characterized by limited size, missing values, irregular sampling, and heterogeneous operating conditions. Several studies highlight that DL models may struggle in such constrained data scenarios due to their high data requirements and sensitivity to noise and sparsity (Zhao et al., 2019). Moreover, it has been observed that classical ML methods, such as RF, often outperform DL models on small or medium datasets (Fernandez-Delgado et al., 2014).

Furthermore, the number of available machines in many industrial studies is limited, which restricts the ability to properly evaluate model generalization. This issue is particularly critical in PdM, where the ultimate goal is cross-machine transferability. Consequently, there is a clear research gap in systematically evaluating PdM models under constrained data conditions, including a small number of machines, short observation periods, irregular time intervals, the presence of maintenance-induced gaps, and class imbalance.

Research questions, data and methodology

The aim of the research described in this paper is to analyze the possibility of applying RF and LSTM models to develop predictive models to predict the occurrence of machine failures in the production system. Both models were created under constrained data conditions. The research presented in this paper aims to address the following research questions:

- RQ1: Which predictive model achieves higher performance in the short-term failure prediction problem?
- RQ2: Which of the analyzed models exhibits more stable cross-machine generalization under limited data conditions?

Dataset Description

A dataset that simulates real sensor data collected from industrial machines was used to conduct the research. The dataset containing near 6,000 synthetically generated values of operating parameters from five CNC machines (see details in Table 1). Each machine covers a two-week operational period. The dataset includes a binary target variable indicating whether a machine failure occurred within the next 24 hours.

The target variable is characterized by class imbalance, with a clear dominance of the negative class (no failure). Approximately 85% of observations belong to this class. The number of failure events during the analyzed period is relatively low. For most machines, only two failures occurred within the two-week timeframe. The number of failures corresponds to the number of maintenance interventions, which result in several-hour gaps in the recorded data.

Table 1. Characteristics of the five subsets of the analyzed dataset

Machine number	M1	M2	M3	M4	M5
Number of cases in the dataset	1192	1146	1218	1141	1240
Number of positive class instances (1, failure)	187	108	183	263	93
Number of negative class instances (0, no failure)	1005	1038	1035	878	1147
Number of failures in the analyzed period	2	2	2	3	1

The models developed in this study utilize a consistent set of input variables. These features can be broadly categorized into three groups:

- Sensor-based measurements: vibration amplitude, motor temperature, electrical current, rotational speed, and pressure-related variable,
- Operational context variables: machine's operating mode, which consists of three nominal values (idle, normal, peak), ambient temperature, and the number of hours since the last maintenance intervention,
- Datetime information: a timestamp variable.

The dataset presents several practical challenges that reflect realistic industrial conditions. First, the observation window is limited to only two weeks of operation per machine, which significantly constrains the amount of available training data. Second, the data exhibit irregular sampling intervals, ranging from a few minutes to over thirty minutes, which complicates temporal modeling and violates assumptions of uniform time spacing typically required by sequence-based models.

Additionally, the dataset contains missing values and extended gaps corresponding to maintenance periods, during which machines are offline and no sensor data are recorded. These gaps introduce structural discontinuities that cannot be treated as simple missing observations. Another important limitation is the small number of machines of interest, which resulting in a limited number of failure events. This imposes constraints on model generalization and increases the risk of overfitting, particularly for data-intensive approaches such as DL.

Data Preprocessing

Prior to the main experimental phase, the dataset underwent several preprocessing steps:

1. Imputation of missing observations,
2. Analysis of the distribution of time intervals,
3. Identification of operational episodes,
4. Resampling to a fixed time interval,
5. One-hot encoding of categorical variables.

Steps 1 through 4 were performed separately for each machine. Missing observations may occur due to issues such as sensor malfunctions or data acquisition errors. These missing values were handled using linear interpolation.

The data were generated at irregular time intervals. To distinguish operational sampling variability from maintenance downtime, the empirical distribution of intervals was analyzed. Descriptive statistics were computed

for time intervals expressed in minutes, and visualizations were generated to identify unusually long gaps. In the boxplot (Figure 1), such gaps appear as outliers. Based on this analysis, it was determined that normal operating intervals were typically below 30 minutes, whereas maintenance-related gaps extended to several hours. A total of 10 maintenance-related gaps were identified. In the context of PdM, periods during which the machine is not operating represent important system events rather than missing data that should be imputed.

To address the issue of gaps corresponding to maintenance periods, continuous operational episodes were identified in order to avoid unrealistic interpolation across maintenance events. The boundaries of these episodes are defined by the identified gaps associated with repair activities. Gaps longer than two hours were classified as maintenance downtime. The first operational episode begins at the start of the recorded data and ends at the first maintenance-related gap. Each subsequent episode begins after a repair period and continues until the next gap. The final episode ends at the last available timestamp.

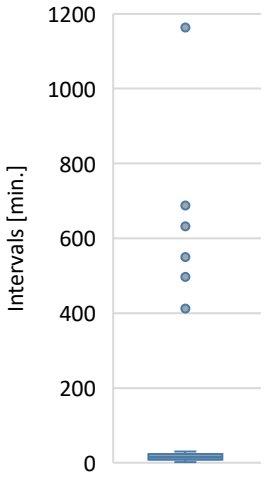


Figure 1. Boxplot of time intervals between timestamps

The fourth step was performed separately for each identified episode of each machine and involved resampling the data to a fixed time interval of 15 minutes (temporal resampling). This step enables the application of standard LSTM models, which assume regularly spaced temporal observations. For continuous variables, values at the new timestamps were obtained using linear interpolation based on neighboring observations. For categorical variables, a back-fill strategy was applied, assigning the most recent previously observed category to the interpolated timestamp.

The final preprocessing step involved full one-hot encoding of the operating mode variable. This transformation was applied globally across the entire dataset. As a result, the original categorical variable was replaced by three binary variables, each representing one category of the original feature.

Experimental Design

The main part of the study was based on the following steps:

1. Splitting the dataset into training and validation subsets,
2. Standardization of data in both training and validation subsets,
3. Construction of sliding window sequences for each machine and each operational episode in the training and validation subsets,
4. Training predictive models on sequences from the training subset,
5. Testing predictive models on sequences from the validation subset.

The experiment employed a validation strategy known as Leave-One-Machine-Out (LOMO). LOMO is based on the classical *k*-fold cross-validation technique, in which the dataset is randomly divided into *k* folds. One fold is used for validation, while the remaining folds constitute the training subset. LOMO is a variant of this approach, where the validation fold consists of all data from a single machine, and the remaining machines form the training subset. The entire procedure is repeated *k* times, where *k* corresponds to the number of machines. Therefore, the

first step of the experiment involved splitting the data into training and validation subsets using a five-fold LOMO strategy.

In the second step, all continuous variables were standardized using the z-score technique. Standardization parameters were computed exclusively on the training subset within each LOMO fold. These parameters were then applied to both the training and validation subsets. This approach prevents data leakage from the validation set into the training process. Binary variables were not standardized.

In the next step, a sliding window approach was used to construct 24-hour input sequences. Sliding windows were generated separately within each LOMO fold and within each operational episode to ensure strict separation between training and testing machines and to avoid temporal leakage. Considering the 15-minute resampling interval, each 24-hour window consisted of 96 timesteps. Each sliding window contains ten time series corresponding to the ten input variables (including dummy variables). Therefore, each window can be represented as a matrix of size 10 × 96 (features × timesteps). The process of sliding window construction is illustrated in Figure 2.

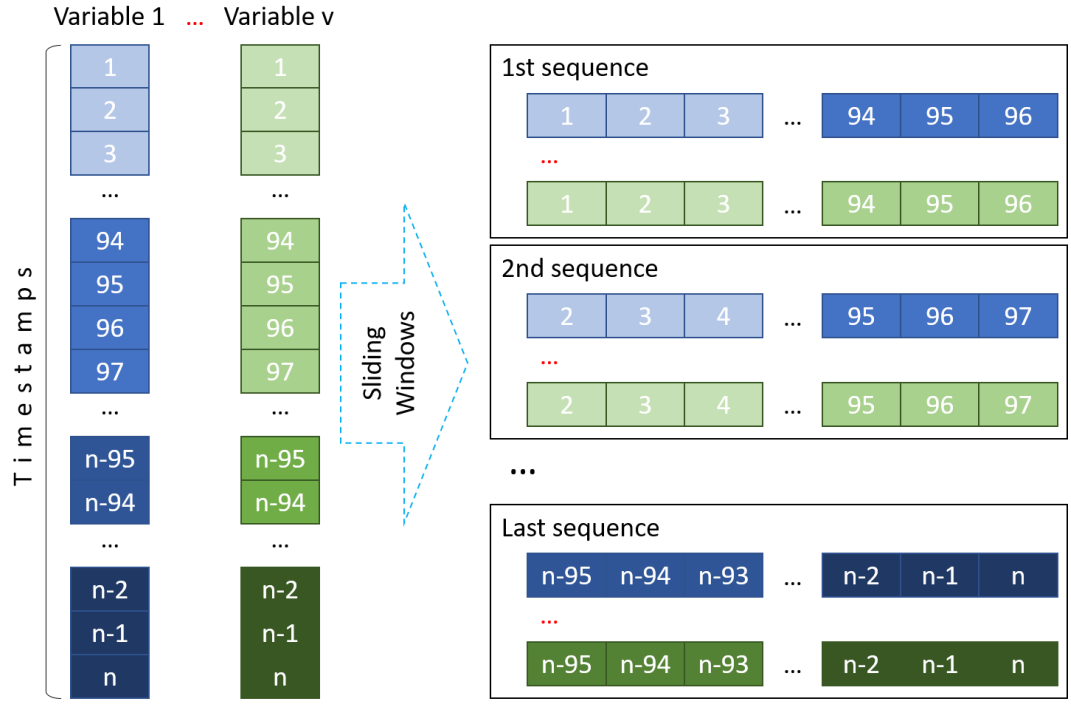


Figure 2. Transformation of the original dataset into sliding window sequences

Due to limitations of the MATLAB environment, the RF does not support sequence inputs. Therefore, each two-dimensional window was transformed into a one-dimensional feature vector (flattened into a vector of length 960). The final two steps of the experiment involved training the LSTM and RF models on the training subset and evaluating them on the validation fold.

Consequently, after completing all steps, the entire procedure was repeated according to the LOMO validation strategy to ensure realistic assessment of cross-machine generalization.

Compared Models

Two types of models were analyzed in the experiment:

- LSTM model with the following architecture:
  - sequence input layer with 10 features,
  - LSTM layer with 80 units,
  - dropout layer with a dropout rate of 0.3 to prevent overfitting,
  - fully connected layer with 32 units followed by a rectified linear unit activation function,
  - fully connected layer with 2 units (corresponding to the number of classes),

    - softmax layer.
- RF model consisting of 200 decision trees.

For each of the five training folds, the following training parameters were used:

- LSTM: Adaptive Moment Estimation (Adam) optimizer, maximum number of epochs equal to 40, mini-batch size of 64, initial learning rate of 0.001, gradient threshold of 1, and class weights proportional to class frequencies in the training subset.
- RF: bagging method with 200 learning cycles.

The architecture and hyperparameters of the models were selected based on preliminary experiments and kept fixed across all folds.

Evaluation Metrics

In each LOMO iteration, the validation fold was used to evaluate both predictive models. The following metrics were employed:

- Accuracy – the proportion of correctly classified instances,
- Precision – the proportion of predicted positive instances that are truly positive,
- Recall – the proportion of actual positive instances correctly identified by the model,
- F1-score – the harmonic mean of precision and recall.

These metrics were computed based on the confusion matrix. A classification threshold of 0.5 was used when generating predictions. Additionally, the Receiver Operating Characteristic (ROC) curve was constructed, and the Area Under the Curve (AUC) was calculated as a measure of the model's ability to rank positive instances higher than negative ones across all possible classification thresholds.

The performance metrics were then averaged across all validation folds. To capture variability in model performance, the standard deviation of each metric was also computed.

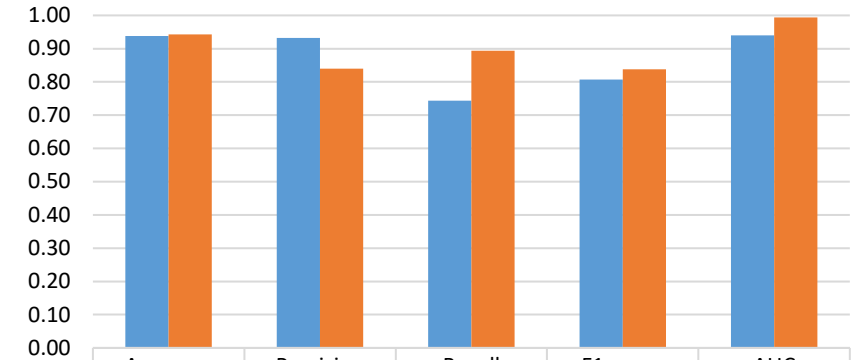
An additional component of the evaluation involved a statistical comparison of the two models using the Wilcoxon signed-rank test. It is a non-parametric statistical procedure used to compare paired measurements originating from the same experimental conditions. It evaluates whether the median of the pairwise differences between two methods differs significantly from zero, without assuming normality of the underlying distribution. This makes it particularly suitable for comparing model performance metrics obtained across cross-validation folds, where sample sizes are small and normality cannot be guaranteed.

Results and Discussion

Overall Models' Performance

The averaged results across all validation folds are summarized in Figure 3. These results allow us to address RQ1. The comparative analysis of the models indicates that the RF achieves overall higher predictive performance. Although both models reach high accuracy, the RF demonstrates a higher recall and a superior AUC. These metrics are particularly important in PdM, where the primary objective is to reliably identify upcoming failures before they escalate into costly breakdowns. The higher recall of the RF suggests that it captures a larger proportion of true failure events, reducing the likelihood of missed detections. Likewise, its near-perfect AUC indicates excellent separability between non-failure and failure states, which is crucial when working with noisy industrial sensor data.

While the LSTM model exhibits higher precision, meaning it generates fewer false alarms, its lower recall implies a greater risk of overlooking actual failures. In industrial environments, such omissions may lead to unplanned downtime, safety hazards, or secondary equipment damage. Therefore, despite the LSTM's strengths in minimizing unnecessary interventions, the RF provides a more robust and reliable prediction. Taken together, the results support the conclusion that the RF model offers superior predictive performance for this specific task.



	Accuracy	Precision	Recall	F1-score	AUC
LSTM	0.938	0.932	0.743	0.808	0.940
RF	0.943	0.840	0.894	0.838	0.994

Figure 3. Mean values of predictive model performance metrics across LOMO validation folds

Additionally, a Wilcoxon signed-rank test was conducted with a significance level of 0.05. The test did not reveal statistically significant differences between the models (p-value = 0.3125). This result should be interpreted with caution. With only five folds, the statistical power of the Wilcoxon test is limited, reducing its ability to detect subtle differences between models and potentially leading to non-significant results even when real differences exist. With such a small sample size, the test does not have sufficient power to reliably detect differences between models, even if they are present.

Cross-Machine Stability

Figure 4 presents the performance metrics obtained for each validation fold. These results contribute to answering RQ2. The LSTM results indicate substantial variability across folds, particularly in terms of recall and F1-score. This suggests that the sequential model is highly sensitive to how the data are partitioned during cross-validation. In practice, this means that LSTM may achieve strong performance on some subsets of data while significantly underperforming on others, reducing its predictability and operational stability.

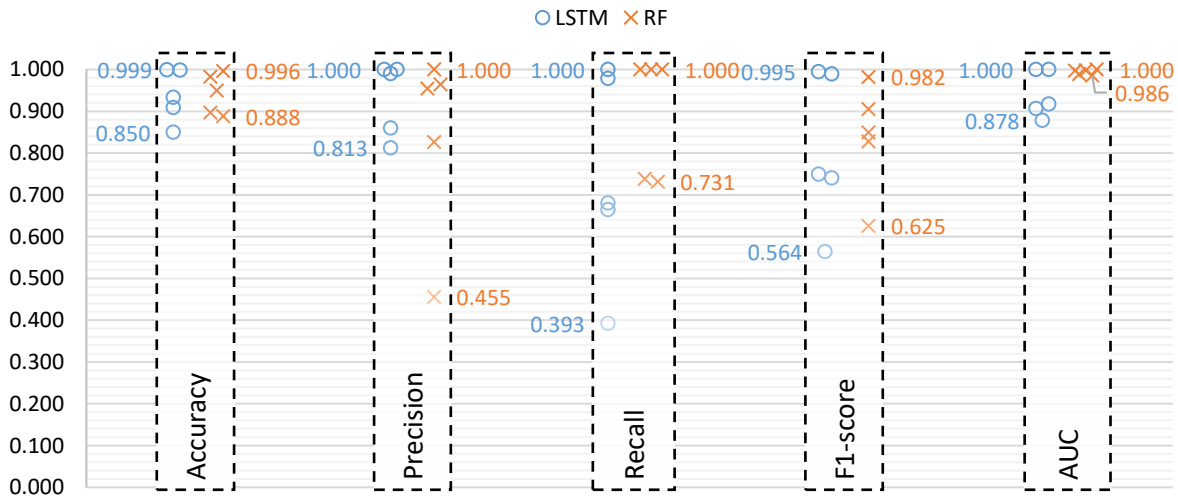
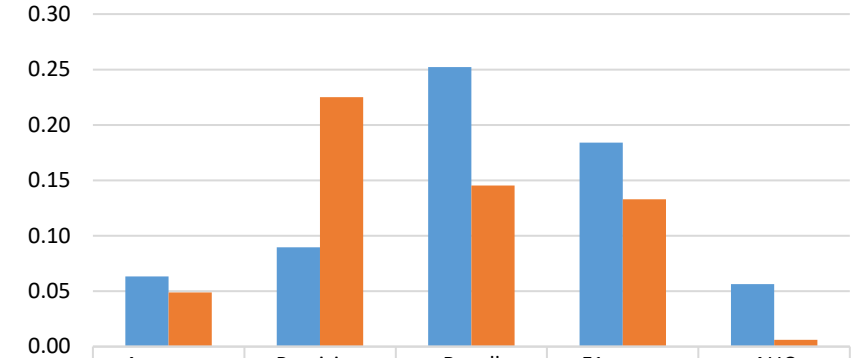


Figure 4. Predictive model performance metrics for each LOMO validation fold

The analysis of standard deviations across the five cross-validation folds (see Figure 5) indicates that the RF exhibits more stable cross-machine generalization. The standard deviations of the RF model are lower for most metrics, with the exception of precision. Therefore, the tree-based model not only achieves higher average values of key performance metrics but also does so in a more consistent and stable manner. This suggests that the RF maintains consistent predictive behavior regardless of how the data are partitioned.



	Accuracy	Precision	Recall	F1-score	AUC
LSTM	0.063	0.090	0.252	0.184	0.056
RF	0.049	0.225	0.145	0.133	0.006

Figure 5. Standard deviation of predictive model performance metrics across LOMO validation folds

The higher variability observed for the LSTM implies that its performance is more dependent on the specific training subset, which may limit its robustness when deployed across different machines or operating regimes. Under limited data conditions, this sensitivity increases the risk of inconsistent predictions, making the RF more dependable choice for stable generalization.

Managerial Implications

The results of the conducted analyses show that both models achieve similar accuracy. However, in the context of imbalanced datasets this metric has limited interpretive value. Evaluating models based on accuracy may lead to overly optimistic conclusions. This risk is particularly pronounced in PdM classification tasks, where class imbalance is common. Therefore, it is essential to assess not only overall model fit but, more importantly, the model's ability to reliably detect upcoming failures while minimizing false alarms.

The study demonstrates that the two models differ in their error profiles, which has direct implications for their practical deployment in industrial environments. The LSTM model achieves significantly higher precision, meaning that it generates fewer false alarms. This is critical for minimizing unnecessary machine downtime, unjustified maintenance interventions, costly preventive actions undertaken without real need, and employee tiredness due to alarms.

In contrast, the RF achieves substantially higher recall, indicating a superior ability to detect actual failure events. In PdM, this capability is particularly critical, as missing a real failure can lead to severe consequences, such as production quality degradation, costly machine breakdowns, or even safety risks for personnel. This makes RF an attractive solution in scenarios where maximizing fault detection is a priority.

From a managerial perspective, decision-makers should align the choice of a model considering the operational priorities of the organization. If the primary goal is to maximize failure detection and avoid costly unplanned downtime, the RF model is more suitable option. Its high recall and exceptional AUC make it a strong candidate for deployment in environments where missing a failure is significantly more expensive than issuing an occasional false alarm. However, in scenarios where false positives carry substantial operational costs the LSTM may still be advantageous. Organizations with tightly optimized production schedules or limited maintenance resources may prefer a model that triggers fewer unnecessary alerts.

Another practical aspect relevant to model selection is the stability of the generated predictions. Models deployed in industrial environments must operate reliably under varying conditions and across diverse machine operating profiles. The higher variability observed in the LSTM model increases the risk of unpredictable behavior after deployment, whereas the RF provides more consistent and trustworthy results. Such stability is especially important in PdM settings, where data availability is often limited and operating conditions differ across machines. A model that generalizes reliably across validation folds is more likely to transfer effectively to machines exhibiting heterogeneous degradation patterns.

Limitations

The dataset used in this study is synthetically generated to mimic realistic industrial IoT behavior. Although sensor ranges and degradation effects were designed to resemble physical processes, synthetic generation may introduce artifacts not present in real-world systems. Furthermore, temporal resampling to fixed 15-minute intervals introduces interpolation assumptions. Although necessary for sequence modeling with LSTM networks, this transformation may smooth abrupt sensor variations and affect anomaly patterns.

The study focuses exclusively on five CNC machines over a two-week period. The limited amount of data restricts the diversity of operational regimes and degradation trajectories. Therefore, the results should be interpreted as evidence from a controlled small-sample industrial scenario rather than universal performance conclusions.

The limited number of failure events (eight across five machines) increases the risk of model overfitting, particularly for DL architectures. Although LOMO cross-validation was applied to mitigate leakage across machines, the small sample size may still bias model estimation.

Conclusion and Future Research

This study investigated the applicability of DL and ML approaches to PdM in a constrained industrial setting. Despite the inherently temporal nature of the problem, the obtained results indicate that a tree-based ensemble model outperforms the LSTM architecture. In particular, the RF achieved higher AUC and recall, suggesting superior capability in detecting impending failures. These findings highlight that:

- High predictive performance can be achieved without complex recurrent architectures.
- Proper feature windowing combined with ensemble learning provides robust cross-machine generalization.
- Temporal DL models may not justify their additional computational complexity when they work in low-data industrial environments.
- RFs effectively exploit nonlinear interactions between sensor data and maintenance-related variables.

From a practical perspective the study shows that RF models are well-suited for maximizing reliability and early detection of failures, making them appropriate for safety-critical or high-availability environments. However, LSTM models may be preferred in scenarios where minimizing unnecessary maintenance actions is critical.

Finally, this study opens directions for future research that can focus on:

- Evaluating the proposed approaches on larger and more diverse industrial datasets.
- Incorporating real-world data from operational environments.
- Investigating whether recurrent architectures provide advantages under higher variability and more complex degradation patterns.

References

- Breiman, L. (2001), ‘Random Forests’, *Machine Learning*, 45 (1), 5–32. https://doi.org/10.1023/A:1010933404324
- Carvalho, TP., Soares, FAAMN., Vita, R., Francisco, R., Basto, JP. and Alcalá, SGS. (2019), ‘A systematic literature review of machine learning methods applied to predictive maintenance’, *Computers & Industrial Engineering*, 137, 106024. https://doi.org/10.1016/j.cie.2019.106024

- Cerrada, M., Zurita, G., Cabrera, D., Sánchez, R.-V., Artés, M. and Li, C. (2016), ‘Fault diagnosis in spur gears based on genetic algorithm and random forest’, *Mechanical Systems and Signal Processing*, 70–71, 87–103. https://doi.org/10.1016/j.ymssp.2015.08.030
- Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J. and Barbosa, J. (2020), ‘Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges’, *Computers in Industry*, 123, 103298, https://doi.org/10.1016/j.compind.2020.103298

- Fernandez-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014), ‘Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?’, *Journal of Machine Learning Research*, 15, 3133–3181.

- Géron, A. (2019) Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.

- Hochreiter S. and Schmidhuber, J. (1997), 'Long Short-Term Memory', *Neural Computing*, 9 (8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
- Jasiulewicz-Kaczmarek, M., Mazurkiewicz, D. and Wyczółkowski, R. (2023) Strategie i metody utrzymania ruchu, PWE, Warsaw.

- Knosala, R. (ed.) (2017), Inżynieria produkcji. Kompedium wiedzy, PWE, Warsaw.
- Li, C., Sanchez, R.-V., Zurita, G., Cerrada, M., Cabrera, D. and Vásquez, RE. (2016), 'Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals', *Mechanical Systems and Signal Processing*, 76–77, 283–293. https://doi.org/10.1016/j.ymssp.2016.02.007

- Louppe, G., Wehenkel, L., Sutera, A. and Geurts, P. (2013), 'Understanding variable importances in forests of randomized trees', *Advances in Neural Information Processing Systems*, 26.
- Serradilla, O., Zugasti, E., Rodriguez, J. and Zurutuza, U. (2022), 'Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects', *Applied Intelligence*, 52 (10), 10934–10964. https://doi.org/10.1007/s10489-021-03004-y

- Wu, J.-Y., Wu, M., Chen, Z., Li, X.-L. and Yan, R. (2021), 'Degradation-Aware Remaining Useful Life Prediction With LSTM Autoencoder, *IEEE Transactions on Instrumentation and Measurement*, 70, 1–10. https://doi.org/10.1109/TIM.2021.3055788
- Yang, Y. and Wang, H. (2025), 'Random Forest-Based Machine Failure Prediction: A Performance Comparison', *Applied* Sciences, 15 (16), 8841. https://doi.org/10.3390/app15168841

- Zhao, H., Sun, S. and Jin, B. (2018), 'Sequential Fault Diagnosis Based on LSTM Neural Network', *IEEE Access*, 6, 12929–12939. https://doi.org/10.1109/ACCESS.2018.2794765
- Zhao, R., Yan, R., Chen, Z., Mao, K., Wang, P. and Gao, RX. (2019), 'Deep learning and its applications to machine health monitoring', *Mechanical Systems and Signal Processing*, 115, 213–237. https://doi.org/10.1016/j.ymssp.2018.05.050