Machine Learning for Predictive Failure in Industrial Machinery

Riccardo Prosdocimi
Northeastern University
prosdocimi.r@northeastern.edu

Alexander Wilcox
Northeastern University
wilcox.al@northeastern.edu

Abstract

This paper presents the development of a predictive maintenance system for industrial equipment through the application of various machine learning techniques. Utilizing the *Microsoft Azure Predictive Maintenance dataset* [1] ($n = 36,500$), we aim to forecast impending machine failures. Our methodology encompasses a variety of models, including logistic regression, MLP (Multilayer Perceptron), RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory), and GRU (Gated Recurrent Unit). We employ PyTorch as our primary framework. This study addresses various challenges such as computational limitations, feature engineering, and potential overfitting issues. Model effectiveness is rigorously assessed using a structured train-validate-test framework, focusing on metrics such as area under the receiver operating characteristic curve (AUROC), accuracy, precision, and recall. We achieve exceptionally high predictive performance, with some models attaining validation and test AUROCs exceeding 0.99. Through these efforts, this research aims to facilitate proactive strategies that minimize downtime, ensure operational safety, and enhance the longevity of industrial equipment.

1 Introduction

In industrial maintenance, developing strategies for preemptive detection of equipment failures is crucial for enhancing safety and minimizing economic losses. This project explores the use of machine learning techniques, including logistic regression and artificial neural networks (ANNs), to develop a comprehensive predictive maintenance system for industrial machinery. Telemetry data, machine conditions, failure histories, and maintenance records are used to predict future machine failures. This research addresses a critical classification challenge, focusing on deriving actionable insights that promote proactive maintenance strategies. These efforts not only aim to prevent unplanned downtime but also extend the lifespan of equipment, ensuring operational safety and cost-efficiency.

2 Methodology

2.1 Dataset and Preprocessing

The study utilizes the *Microsoft Azure Predictive Maintenance dataset* [1], comprising various data sources from 100 industrial machines over the year 2015. This dataset contains 5 CSV files:

1. Telemetry Data: Hourly data on machine voltage, rotation, pressure, and vibration.
2. Maintenance Records: Logs of both scheduled and unscheduled maintenance activities.
3. Failure History: Records of component replacements due to failures.
4. Error Logs: Logs of operational anomalies that did not result in failures.

5. Machine Features: Static attributes such as machine model and age.

In preprocessing the features, we first merge these datasets to form a unified view, where each machine is represented by hourly records encapsulating all relevant information such as telemetry readings and machine characteristics. We then segment this combined data into discrete 24-hour windows. For each window, we aggregate the data to create structured inputs for our models:

- For non-sequential models (i.e., logistic regression and MLP), we summarize the 24-hour data for each machine into a single vector per window, using statistical measures such as mean, standard deviation, skewness, and kurtosis for telemetry metrics, and maintaining static features like machine age (age does not change throughout the course of 1 year). This resulted in 45 total features per time window.
- For sequential models (i.e., RNN, LSTM, GRU), each 24-hour window contains a sequence of hourly snapshots, preserving the temporal dynamics of the data for each machine. This resulted in 25 total features per hour.

This approach allows us to capture both the static and dynamic characteristics of the machines, providing a comprehensive dataset for training our predictive models. Through careful feature engineering, we address potential issues such as missing values and outliers, ensuring high-quality inputs for subsequent analysis. Using this 24-hour time window, we arrive at $n = 36,500$.

We also explored other feature time windows (12-, 48-, 72-, and 168-hour time windows) and found the 24-hour window to be the smallest window yielding reliable model performance.

Finally, for each machine's time window (features), we create binary labels indicating whether a failure occurred within the subsequent 6 and 24 hours, respectively. Because we included past failure history into the features, we also took great care to avoid any data leakage (i.e., avoid putting labels into the features). In doing so, we performed sanity checks as a guard against this. For the prevalence of each label, please see Appendix A.2.

2.2 Models

Our study evaluates a variety of machine learning models to predict machine failures, ranging from a traditional classifier to ANNs. Each model is chosen to demonstrate different capabilities in capturing relationships within our data:

1. Logistic Regression: Included primarily for illustrative purposes, logistic regression helps establish a baseline by demonstrating the non-linear relationship in our dataset where simpler linear models (such as logistic regression) might fail. It lacks temporal processing but has a clear interpretation.
2. MLP: Serves as a straightforward neural baseline, using both static data (e.g., make) and reshaped time-series data (e.g., voltage) to assess the performance of more complex models.
3. RNN: Adapts to the sequential nature of our dataset, capturing short-term temporal patterns in telemetry data, with static features replicated across each hour. The final hidden layer of the final hidden state is processed through a custom MLP classifier.
4. LSTM: Designed to capture longer-term dependencies by overcoming limitations like the vanishing/exploding gradient problem seen in traditional RNNs, suitable for our dataset with extended time steps. The final hidden layer of the final hidden state is processed through a custom MLP classifier.
5. GRU: Balances between complexity and performance, with a simplified gating mechanism that enhances training efficiency, ideal for computationally constrained scenarios. The final hidden layer of the final hidden state is processed through a custom MLP classifier.

2.3 Training

We split the data into 60% for training, 20% for validation, and 20% for testing using a random split. For our ANN models, we found that training over 200 epochs adequately captures the complexities of the data without overfitting, given the depth and range of model architectures used. In all cases, binary cross entropy was used as the loss function. To fine-tune hyperparameters, we tuned one hyperparameter at a time across experiments, necessitated by time and computational resource constraints.

- Logistic Regression: Applied once for each target (6-hour and 24-hour failure predictions).
- MLP: Hyperparameters fine-tuned/optimized included learning rate, batch size, MLP dropout, and hidden layers. Optimization was performed using the Adam optimization algorithm. ReLU served as the activation function.
- RNN, LSTM, GRU: Hyperparameters fine-tuned/optimized included learning rate, batch size, RNN dropout, RNN hidden layers, MLP dropout, and MLP hidden layers. Optimization was performed using Adam. ReLU served as the MLP's activation function.

3 Results

Please see Appendix A.1 for detailed model performance tables.

Analysis of model performances reveals significant insights into the application of machine learning for predictive maintenance. Surprisingly, logistic regression, which captures only linear relationships, achieved a decent AUROC but suffered from a poor precision-recall AUC, highlighting its limitations in capturing the dataset's complexities and infrequent instances of failure. The MLP performed exceptionally well, achieving slightly superior performance to the RNN and LSTM models. Finally, the GRU model performed best, achieving near-perfect validation and test AUROC scores, showcasing its excellent capability to balance true positive and false positive rates. High AUROC scores are decisive in achieving optimal trade-offs between sensitivity and specificity, which is essential for dependable failure predictions in industrial settings.

4 Discussion

The performance comparison across models illustrates their varied effectiveness in predictive maintenance. Logistic regression showed surprising resilience, likely benefiting from linear separability in some dataset features, despite its general limitation with complex, non-linear relationships. The RNN underperformed, possibly hindered by vanishing or exploding gradients due to the 24 sequence sample size. This issue may also explain why the simpler MLP outperformed the RNN and even the more complex LSTM, demonstrating that less complex models can sometimes capture essential features effectively without the need for handling sequential dependencies.

The GRU model performed the best, blending architectural simplicity (relative to the LSTM) with effective sequence feature capturing, making it quite suitable for this task. This efficiency stresses the need for aligning model complexity with data-specific characteristics. Its success in predicting low prevalence labels underscores its robustness in scenarios where capturing rare events (i.e., failures) is valuable. Future work could investigate hybrid RNN models or advanced ensemble methods to enhance model performance.

Finally, the exceptionally high AUROC scores observed may be attributable to several factors. The data was notably clean and well-preprocessed, lacking extreme outliers and noise, which can significantly enhance model training, validation, and test accuracy. This cleanliness, combined with our structured train-validate-test approach, careful tuning of model hyperparameters, model selection, and thoughtful feature engineering, likely contributed to the impressive performance metrics.

5 Conclusion

This research highlights the effectiveness of machine learning techniques in developing predictive maintenance systems for industrial machinery, offering a robust approach to foresee and mitigate equipment failures. Future efforts may focus on enhancing these models to predict failures at a component level, thus offering more detailed insights into the specific maintenance needs of machines. Additionally, integrating real-time data processing could further improve model responsiveness to sudden changes in machine behavior, potentially leading to more timely and precise failure predictions. This could not only optimize maintenance schedules but also significantly extend the operational life and efficiency of industrial assets.

6 Acknowledgements

This project was completed for CS 6140 (Machine Learning) under the instruction of Professor Paul Hand during the Spring 2024 semester. We would also like to thank our TAs for their time and dedication this semester.

References

[1] Biswas, A. (2020) Microsoft Azure Predictive Maintenance. Kaggle. Accessed April, 2024. https://www.kaggle.com/datasets/arnabbiswas1/microsoft-azure-predictive-maintenance/data.

A Appendix A

A.1 Model Performance Summaries

Model	Target	Validation AUROC	Test AUROC
Logistic Regression	6hr Failure Forecast	0.86	0.86
MLP	6hr Failure Forecast	0.9978	0.9940
RNN	6hr Failure Forecast	0.9441	0.9125
LSTM	6hr Failure Forecast	0.9801	0.9710
GRU	6hr Failure Forecast	>0.9999	0.9999
Logistic Regression	24hr Failure Forecast	0.86	0.86
MLP	24hr Failure Forecast	0.9987	0.9971
RNN	24hr Failure Forecast	0.9489	0.8742
LSTM	24hr Failure Forecast	0.9874	0.9679
GRU	24hr Failure Forecast	>0.9999	>0.9999

Table 1: Comparative performance of models using a 24-hour feature aggregation window ($n = 36,500$). For ANN models, metrics from the best validation epoch were selected.

Model	Target	Test Accuracy at TPR=0.8	Test Accuracy at FPR=0.05
MLP	6hr Failure Forecast	0.9890	0.9567
RNN	6hr Failure Forecast	0.8875	0.9455
LSTM	6hr Failure Forecast	0.9560	0.9446
GRU	6hr Failure Forecast	0.9919	0.9724
MLP	24hr Failure Forecast	0.9901	0.9758
RNN	24hr Failure Forecast	0.9387	0.9537
LSTM	24hr Failure Forecast	0.9764	0.9544
GRU	24hr Failure Forecast	0.9984	0.9990

Table 2: Test accuracies of ANN models corresponding to Table 1 at specified true positive rate (TPR) and false positive rate (FPR) thresholds, demonstrating the models' accuracy under controlled conditions. The thresholds applied were each determined based on the validation set.

	PP	PN
P	7,135	5
N	2	138

Table 3: Confusion matrix for the GRU model predicting 24-hour failure on test data (corresponding to the final row of Table 1), with a specificity constraint (FPR=0.05). The threshold applied was determined based on the validation set. This table selectively illustrates the performance for this specific model and condition, not encompassing other model configurations or outcomes.

A.2 Target Prevalence Metrics

Target	Prevalence
6hr Failure Forecast	0.96%
24hr Failure Forecast	1.94%

Table 4: Prevalence rates of target conditions for the predictive models.