Predicting Machine Failure with Deep Learning

Class: DLNN **Members:** **Group:** 2
AbdulQudus Adebogun (20241355) | Hagar Fisher (20241607)
Md Mijanul Haque (20241356) | Patricia Fabiawari Agha (20242041)

Project Introduction

Machine failures in industrial manufacturing systems are rare but highly disruptive events that often lead to production downtime, safety risks, and substantial financial losses. To mitigate such risks, predictive maintenance systems aim to identify early warning signs of failure using operational data collected during normal machine operation, allowing interventions to take place before a breakdown occurs. However, building effective predictive maintenance models is not straightforward. Failure signals are often weak, and breakdowns typically result from the combined effect of multiple operating factors rather than a single abnormal measurement. These characteristics limit the effectiveness of simple threshold-based rules and shallow or linear modelling approaches.

This project addresses this challenge by developing a deep learning-based binary classification model that estimates the risk of machine failure at the level of individual production cycles. The model is trained and evaluated using the AI4I 2020 Predictive Maintenance dataset, which despite being synthetically generated, is grounded in physics-based failure mechanisms where breakdowns emerge from interactions between thermal conditions, mechanical load, cumulative tool wear, and randomisation. The primary objective is to evaluate how well the model can detect and prioritise rare but critical failure events under realistic industrial constraints. The scope of the work focuses on data exploration and preprocessing, feature engineering, and model development, training, and evaluation.

Dataset Description, Exploration, and Preprocessing

The AI4I 2020 Predictive Maintenance dataset contains 10,000 observations, each representing a single manufacturing production cycle, with 14 recorded variables. These variables include operational sensor measurements, a binary machine failure indicator, and several failure mode indicators describing the physical logic used to generate breakdown events. A failure is recorded whenever at least one failure mode is triggered, reflecting industrial settings where breakdowns arise from multiple interacting causes rather than a single fault.

The dataset contains no missing values, and all variables fall within physically plausible ranges defined by the simulation rules, eliminating the need for imputation or range correction.

Target Variable and Failure Mode Indicators Analysis

The target variable, Machine failure, is a binary indicator denoting whether a failure occurred during a production cycle (1) or not (0). The dataset is highly imbalanced, with only 339 failure events (3.39%) out of 10,000 observations. This imbalance motivates the use of evaluation metrics and modelling strategies that prioritise failure detection over overall accuracy.

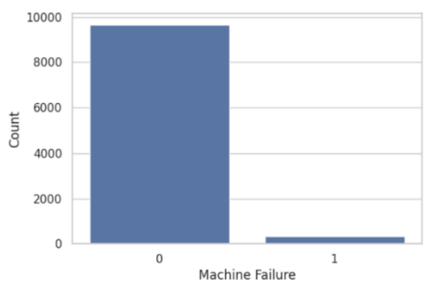

Fig. 1 – Machine Failure Distribution

To reiterate, a machine failure (1) occurs whenever one or more predefined failure mode are triggered during a production cycle. The AI4I dataset includes five failure mode indicators describing the physical mechanisms that can independently trigger machine failure:

- **Tool Wear Failure (TWF):** occurs when tool wear exceeds a randomly assigned failure threshold between 200-240 minutes (120 occurrences).
- **Heat Dissipation Failure (HDF):** occurs when the temperature difference between process and air is below 8.6 K and rotational speed is below 1380 rpm (115 occurrences).
- **Power Failure (PWF):** occurs when mechanical power (torque x angular velocity) falls outside the safe operating range of 3500-9000 W (95 occurrences).
- **Overstrain Failure (OSF):** occurs when torque x tool wear exceeds quality-dependent thresholds (98 occurrences): 11,000 minNm for L, 12,000 minNm for M, and 13,000 minNm for H.

- **Random Failures (RNF):** A rare probabilistic failure mode occurs with a 0.1% chance per record, independent of process parameters (5 occurrences).

Because the failure mode indicators are not observable in real predictive maintenance settings, they are excluded from model inputs to avoid label leakage and are instead used as descriptive metadata to guide exploratory analysis and physics-informed feature engineering.

Input Variables Analysis – Raw and Engineered

The raw input variables capture six observable operation measures of the machine during each production cycle. In addition to these variables, the dataset includes two identifiers: **UDI**, which is set as the dataset index, and **Product ID**, which is dropped during preprocessing due to high cardinality and redundancy with the Type variable.

The following sections examine the six raw operational variables by physical process, explaining how they contribute to failure modes and motivate the engineered interaction features.

Thermal Variables:

- **Air Temperature [K]:** Ambient temperature surrounding the machine, generated via a random walk process and normalised to approximately 300 K ($\sigma \approx 2$ K).
- **Process Temperature [K]:** Internal operating temperature, also generated via a random walk process and constrained to remain approximately 10 K above air temperature ($\sigma \approx 1$ K).

Individually, these variables show limited separation between failure and non-failure cases. However, their difference governs cooling efficiency and directly underlies heat dissipation failures (HDF). To capture this interaction, the following feature is derived:

- **Temperature Gradient [K] = Process Temperature - Air Temperature**

Mechanical Variables:

- **Rotational Speed [rpm]:** Represents spindle speed during operation and is generated from a base mechanical power of 2860 W with added noise. It influences both airflow for cooling and mechanical load.
- **Torque [Nm]:** Represents instantaneous mechanical load and is normally distributed around 40 Nm (σ = 10 Nm). Torque contributes to both instantaneous stress and cumulative overstrain when combined with tool wear.

Neither variable alone is sufficient to indicate failure risk. Instead, their nonlinear interaction determines mechanical load regimes associated with power-related failures (PWF). This motivates the following derived feature:

- **Mechanical Power [W] = Torque x (2π x RPM / 60)**

Degradation Variables:

- **Type:** A categorical variable representing product quality level (H, M, L).
- **Tool Wear [min]:** Represents cumulative tool usage time, increasing monotonically over successive production cycles. Wear increments depend on product quality level (H: +5 min, M: +3 min, L: +2 min per cycle).

These variables capture progressive degradation that interacts with instantaneous mechanical load, directly underpinning Tool Wear Failure (TWF) and Overstrain Failure (OSF). To expose these interaction-driven mechanisms, the following features are derived:

- **Mechanical Strain = Torque x Tool Wear** : Encodes load-induced stress accumulated over tool lifetime.
- **Wear Increment per Cycle = Type → {2, 3, 5}** : Models quality-dependent degradation rates.
- **Effective Wear Load = Tool Wear x Wear Increment** : Scales accumulated wear by product quality to reflect effective degradation.

Feature Selection and Preprocessing Strategy

Both raw operating variables and physics-informed engineered features are retained for modelling. Raw measurements preserve absolute operating regimes and threshold-based behaviour, while engineered features explicitly expose interaction effects underlying known failure

mechanisms. This combination allows the model to learn both established physical relationships and additional latent interactions present in the data.

In preparation for modelling, the dataset is first split into stratified training and test sets (80-20) to preserve the failure rate across splits. Preprocessing is implemented using a unified pipeline to prevent information leakage. The categorical Type variable is one-hot encoded, and all numerical features are standardised using z-score normalisation. All preprocessing transformations are fitted on the training data and applied unchanged to the test data to ensure valid out-of-sample evaluation.

Model Development and Training

The modelling phase aims to learn a reliable mapping from per-cycle operating conditions to machine failure risk. To enable systematic comparison, all experiments are conducted within a consistent global framework that fixes the model family, optimisation strategy, training procedure, and evaluation criteria. Within this framework, a sequence of controlled experiments is used to assess the impact of architectural capacity and regularisation on failure detection performance.

Global Modelling Framework

All models are implemented as fully connected deep neural networks (Multilayer Perceptrons, MLPs), which are well suited to fixed-length tabular feature vectors and to learning nonlinear relationships between operating conditions. The task is formulated as binary classification, with a sigmoid-activated output producing a continuous failure risk score for each production cycle.

Because machine failures account for approximately 3.4% of observations, the learning problem is highly imbalanced. Rather than applying resampling techniques that could distort the physics-based structure of the data, class imbalance is addressed using class-weighted loss functions. Binary cross-entropy loss is used to train probabilistic outputs under this imbalance, while the Adam optimiser is employed for stable and efficient convergence on tabular data. ReLU activations in hidden layers enable the network to learn interaction-driven nonlinear relationships. To control overfitting and promote reliable generalisation, early stopping based on validation loss is applied consistently across all experiments, retaining the model state that performs best on unseen validation data.

Model performance is evaluated using metrics that prioritise sensitivity to rare failure events and reliable risk ranking, with recall and PR-AUC used for model comparison. Accuracy is not considered a meaningful criterion in this setting.

Model Architectures and Experiments

Starting from a baseline MLP, a small number of targeted architectural and regularisation modifications are introduced to assess their impact on failure detection performance. All models are trained and evaluated using identical data splits, preprocessing pipelines, optimisation settings, and evaluation metrics to ensure fair comparison. Model selection is based exclusively on validation performance, with the test set reserved for a single final evaluation.

Baseline Model

The baseline model serves as a reference point for all subsequent experiments. Its purpose is to verify that the raw and engineered features contain sufficient signal to learn meaningful failure patterns. The architecture consists of a moderately sized MLP with two hidden layers and no explicit regularisation, providing a lower bound for deep learning performance.

Architecture:
Input: preprocessed feature vector
Hidden Layer 1: 32 neurons, ReLU
Hidden Layer 2: 16 neurons, ReLU
Output Layer: 1 neuron, Sigmoid

Capacity Expansion Experiment

To assess whether the baseline model underfits the underlying failure mechanisms, network depth and width are increased while keeping all other training conditions fixed. The expanded architecture introduces an additional hidden layer and a larger number of neurons, increasing representational capacity without regularisation.

Architecture:
Input: preprocessed feature vector
Hidden Layer 1: 64 neurons, ReLU
Hidden Layer 2: 32 neurons, ReLU
Hidden Layer 3: 16 neurons, ReLU
Output Layer: 1 neuron, Sigmoid

Dropout Regularisation Experiment

This experiment evaluates whether regularisation improves generalisation and failure risk ranking under class imbalance. Dropout is applied after each hidden layer, with rates of 0.1, 0.2, and 0.3 evaluated. Based on validation performance, a dropout rate of 0.1 provides the best balance between improved robustness and preservation of interaction-driven decision boundaries, yielding the strongest combination of recall and PR-AUC. Only this configuration is retained for further comparison.

Architecture:
Input: preprocessed feature vector
Hidden Layer 1: 32 neurons, ReLU
Dropout Layer
Hidden Layer 2: 16 neurons, ReLU
Dropout Layer
Output Layer: 1 neuron, Sigmoid

Weight Decay Regularisation Experiment

The weight decay experiment examines the effect of L2 regularisation on model generalisation by penalising large weights during optimisation. L2 regularisation is applied to all hidden layers, with values of 1e-5, 1e-4, and 1e-3 evaluated. Among these, the L2 of 1e-4 achieves the best validation performance, providing modest stabilisation without suppressing the sharp nonlinear interactions associated with machine failures. Only this configuration is retained for further comparison.

Architecture:
Input: preprocessed feature vector
Hidden Layer 1: 32 neurons, ReLU (L2)
Hidden Layer 2: 16 neurons, ReLU (L2)
Output Layer: 1 neuron, Sigmoid

Models Comparison

To identify the most suitable architecture for failure risk prediction, validation performance is compared across the baseline model, the capacity-expanded model, and the best-performing regularised variants:

MLP	Recall	Precision	ROC-AUC	PR-AUC	Validation Loss
Baseline	**0.870**	0.336	0.965	0.723	0.153
Higher-Capacity	0.741	**0.465**	0.947	0.737	**0.120**
Dropout = 0.1	**0.870**	0.281	**0.966**	**0.745**	0.193
L2 = 1e-4	**0.870**	0.267	0.954	0.705	0.230

Fig. 2 – Model Comparisons

The baseline model achieves strong recall and solid ranking performance, confirming that the engineered feature set contains sufficient predictive signal. However, its PR-AUC is lower than that of the best regularised model, indicating room for improvement in failure risk ranking. The capacity-expanded model attains the lowest validation loss and highest precision, reflecting more conservative predictions, but this comes at the cost of substantially reduced recall and a higher rate of missed failures. Given the operational priority of failure detection, this trade-off makes the capacity-expanded model unsuitable despite its lower loss.

The L2-regularised model preserves high recall but exhibits weaker PR-AUC and higher validation loss, suggesting that weight penalisation smooths the decision boundary in a way that degrades ranking performance. In contrast, the dropout-regularised model provides the best overall balance, maintaining high recall while achieving the strongest PR-AUC across all models. This improvement is achieved without a significant loss of precision, indicating effective regularisation rather than over-smoothing.

Model Selection

Based on validation performance, the 0.1 dropout-regularised model is selected as the final architecture. To confirm that this selection reflects robust learning behaviour rather than favourable validation metrics alone, the training dynamics of the selected model are examined. Figure 2 presents the training and validation loss and recall curves for the selected model.

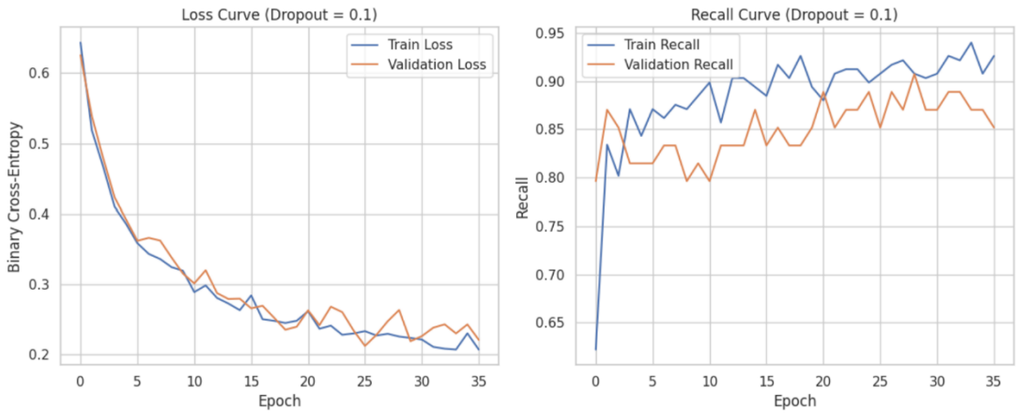

Fig. 3 – Loss and Recall curve for Selected Model

The training dynamics show smooth and stable convergence, with training and validation loss closely aligned throughout training. This behaviour indicates effective regularisation and no evidence of overfitting. Validation loss remains stable across epochs, suggesting that the learned decision boundary generalises well to unseen data. Recall curves further support this conclusion. Validation recall remains consistently high with only minor variability across epochs, while training recall increases gradually without inducing a collapse in validation recall. This pattern indicates improved sensitivity without over-specialisation.

Taken together, these observations confirm that the dropout-regularised model with a rate of 0.1 exhibits stable optimisation behaviour, reliable generalisation, and consistent failure detection performance. It is therefore well suited for subsequent decision-threshold optimisation and final evaluation.

Threshold Optimisation and Final Evaluation

To convert the continuous failure risk scores outputted by the model into actionable predictions, a decision threshold is selected using the validation data. A sweep over candidate thresholds in the range [0.01,0.99] is performed, and the threshold that maximises the validation F2-score is chosen. F2-score is used to place greater emphasis on recall than precision, reflecting the higher operational cost of missed failures relative to false alarms in predictive maintenance settings. The optimal threshold is discovered to be 0.87, and it yields the following validation performance:

- Precision: 0.61
- Recall: 0.72
- F2-score: 0.70

This operating point achieves a strong balance between sensitivity to rare failure events and control of false alarms, making it suitable for final evaluation. Using the selected threshold, the model is evaluated once on the fully held-out test set. This final evaluation reflects the model's expected behaviour in a realistic deployment setting and serves as the concluding assessment of the proposed approach. Test performance is as follows:

- Recall: 0.71
- Precision: 0.64
- F1-score: 0.67
- F2-score: 0.69
- ROC-AUC: 0.97
- PR-AUC: 0.68

The test-set performance closely aligns with the validation performance, indicating stable generalisation and confirming that threshold selection did not overfit the validation data. High recall demonstrates the model's ability to detect the majority of true failure events, while strong ROC-AUC and PR-AUC values confirm effective ranking of failure risk under severe class imbalance.

Overall, these results demonstrate that the proposed modelling approach can reliably detect and prioritise rare but operationally critical machine failures. By combining physics-informed feature engineering, controlled model selection, and threshold optimisation aligned with cost asymmetry, the final system delivers robust failure risk prediction under realistic industrial constraints.