Utrecht University

Mathematical Institute

Risk-Averse Predictive Maintenance Scheduling with Distributional Reinforcement Learning using Data-Driven Probabilistic Prognostics

Master's Thesis

Robin van der Laag

Mathematical Sciences

Supervisors:

Dr. Sjoerd Dirksen
Utrecht University

Dr. Mihaela Mitici
Utrecht University

November 13, 2024

Abstract

Maintenance scheduling for complex industrial systems, such as turbo jet engines, is critical for ensuring operational efficiency and safety. While data-driven prognostics have shown potential for improving predictive maintenance planning, existing approaches often fail to explicitly account for the safety-critical nature of these systems, typically addressing it through assigning high costs to failures. This thesis proposes a novel risk-averse approach to integrating data-driven probabilistic prognostics into predictive maintenance scheduling.

The proposed methodology is applied to NASA's turbofan engine C-MAPSS data set. A threshold-weighted scoring rule is employed as the loss function in a neural network model to induce aversion to downside-risk when estimating the distribution of the remaining useful life (RUL). Building on these estimates, a Distributional Reinforcement Learning (DRL) model is developed for predictive maintenance scheduling. Here, risk-aversion is introduced by optimizing the agent's decision-making based on the Conditional Value at Risk (CVaR) of the return distribution, rather than the mean.

Results show that the forecasting model incorporating the threshold-weighted scoring rule demonstrates a tendency to underestimate RUL, effectively inducing the desired risk-averse behavior with only minor losses in overall performance. The risk-averse maintenance scheduling models exhibited a noticeable, though slightly inconsistent, trend towards preventing engine failures more effectively, with marginally higher average RUL at scheduled replacements compared to their risk-neutral counterparts. The scheduling agents learned to optimize the use of two out of three maintenance actions to balance failure prevention and operational efficiency.

This study demonstrates the feasibility of incorporating downside-risk aversion in both RUL estimation and maintenance scheduling, offering a more robust framework for enhancing safety and performance in predictive maintenance strategies.

Contents

1 Introduction

For complex industrial system the cost of failure and the resulting unscheduled maintenance can be prohibitively high. An accurate predictive maintenance scheduling strategy can significantly decrease these costs by monitoring a health index of the system to schedule maintenance before failure occurs. The remaining useful life (RUL) is a popular health index, as it represents the effective life left of a component measured in some operational time measure, such as number of cycles or hours [1]. There are two main approaches used to model the RUL: physics based and data driven. Physics based models work by mathematically representing the degradation process of the component and using this to predict the RUL. This approach can be time consuming and requires deep knowledge about the system. Data driven models instead use monitoring data obtained from the system to detect the degradation of the components. Over the past decade, developments in machine learning has made the data driven approach more accurate. The majority of the models developed make point predictions of the RUL [2–4]. Whilst these approach have achieved good accuracy they fail to include the probabilistic nature of the degradation process in the form of aleatoric uncertainty as well as the epistemic uncertainty originating from the model parameters [5]. Recently models have been developed that account for one or both of these uncertainties, resulting in probabilistic predictions of the RUL [6–9]. These probabilistic predictions contain crucial information for predictive scheduling models. Based on these probabilistic prognostics alarms can be set up to trigger and maintenance to be scheduled when the probability of the RUL being under a certain value exceeds a chosen threshold [10]. In [7] a deep reinforcement learning model was used to adaptively propose maintenance actions based on the estimated RUL prognostic. For many use cases this can result in a near optimal maintenance strategy, as seen from a cost perspective. However, for safety critical systems, such as jet engines, there are often non-monetary costs associated with failure. In this thesis we will explore risk-averse strategies for the prediction of the RUL by utilising asymmetric loss functions which punish underestimation more than overestimation as well as developing risk-averse strategies for predictive maintenance scheduling by employing distributional reinforcement models which take the RUL prognostics as input. The methods will be trained and tested on NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) [11] data set.

The remainder of this thesis is structured as follows. Section 2 provides an overview of probabilistic forecasting, followed by an introduction to distributional reinforcement learning in Section 3. In Section 4, we present the risk-sensitive framework and discuss its application within probabilistic forecasting and distributional reinforcement learning. Section 5 details our methodology, beginning with a description of the data and preprocessing techniques, and proceeding to illustrate our methodology for the estimation of the remaining useful life (RUL) distribution using probabilistic forecasts and applying it to predictive maintenance planning of turbofan engines. Section 6 evaluates and compares our results on the C-MAPSS dataset against alternative models and strategies. Finally, Section 7 concludes with a discussion of our findings and their implications.

2 Probabilistic Forecasting

This section largely follows Gneiting & Katzfuss (2014) [12] and Gneiting & Ranjan (2013) [13].

A probabilistic forecast represents future quantities or events as a predictive probability distribution. We focus on the case of a real-valued observation Y and a probabilistic forecast F, with its associated cumulative distribution function (CDF) defined on $\mathbb{R}$. We use $\mathcal{L}$ to denote an unconditional or conditional distribution. A prediction space is a probability space specifically tailored for distributional forecasts.

Definition 2.1. Let $k \geq 1$ be an integer. A *prediction space* is a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ together with sub-σ-algebras $\mathcal{A}_1, \ldots, \mathcal{A}_k \subseteq \mathcal{A}$, where the elements of the sample space Ω can be identified with tuples $(F_1, \ldots, F_k, Y, V)$ such that

(P1) for $i = 1, \ldots, k$, F_i is a CDF-valued random quantity that is measurable with respect to the sub-σ-algebra $\mathcal{A}_i$,

(P2) Y is a real-valued random variable, (P3) V is a random variable that is uniformly distributed on the unit interval and independent of $\mathcal{A}_1, \dots, \mathcal{A}_k$ and Y.

Definition 2.2. The CDF-valued random quantity F_i is *ideal* relative to the sub-σ-algebra $\mathcal{A}_i$ if $F_i = \mathcal{L}(Y|\mathcal{A}_i)$ almost surely.

For an example, consider $Y|\mu \sim \mathcal{N}(\mu, 1)$ and $\mu \sim \mathcal{N}(0, 1)$. Then the probabilistic forecast $F = \mathcal{N}(\mu, 1) = \mathcal{L}(Y|\mu)$ is ideal relative to the sub-σ-algebra generated by the random variable μ. Meanwhile the forecast $G = \mathcal{N}(0, 2)$ is ideal relative to the trivial sub-σ-algebra.
More examples that illustrate the concepts of prediction spaces and ideal forecasts can be found in [13].

For a fixed, non-random predictive CDF F for an observation Y, the probability integral transform (PIT) is the random variable $Z_F = F(Y)$. We extend the definition further to allow F to be a CDF-valued random quantity.

Definition 2.3. The *probability integral transform* of the CDF-valued random quantity F is given by the random variable

$$Z_F = \lim_{y \uparrow Y} F(y) + V\left(F(Y) - \lim_{y \uparrow Y} F(y)\right).$$

The PIT is essentially the value that the predictive CDF attains at the observation. Using the PIT we can define the concepts of calibration and dispersion.

Definition 2.4. Let F and G be CDF-valued random quantities with probability integral transforms Z_F and Z_G.

(a) The forecast F is *marginally calibrated* if $\mathbb{E}_{\mathbb{P}}[F(y)] = \mathbb{P}(Y \leq y)$ for all $y \in \mathbb{R}$.

(b) The forecast F is *probabilistically calibrated* if Z_F is uniformly distributed on the unit interval.

(c) The forecast F is *overdispersed* if $\mathrm{Var}(Z_F) < \frac{1}{12}$, *neutrally dispersed* if $\mathrm{Var}(Z_F) = \frac{1}{12}$, and *underdispersed* if $\mathrm{Var}(Z_F) > \frac{1}{12}$.

(d) The forecast F is *as least as dispersed* as the forecast G if $\mathrm{Var}(Z_F) \leq \mathrm{Var}(Z_G)$. It is *more dispersed* than G if $\mathrm{Var}(Z_F) < \mathrm{Var}(Z_G)$.

(e) The forecast F is *regular* if the support of the distribution of Z_F is the unit interval.

In the prediction space setting we immediately have that a probabilistically calibrated forecast is neutrally dispersed and regular. The following theorem shows the relevance of ideal forecasts.

Theorem 2.5. *A forecast that is ideal relative to a σ-algebra is both marginally calibrated and probabilistically calibrated.*

The proof of this theorem can be in Gneiting & Ranjan (2013) [13]. Coming back to the example where $Y|\mu \sim \mathcal{N}(\mu, 1)$ and $\mu \sim \mathcal{N}(0, 1)$, we now have that the forecasts $F = \mathcal{N}(\mu, 1)$ and $G = (0, 2)$ are both probabilistically and marginally calibrated, as they are ideal forecasts. For the forecast $F = \mathcal{N}(0, \sigma^2)$ we have that it is underdispersed if $\sigma^2 < 2$ and overdispersed if $\sigma^2 > 2$.

2.1 Diagnostic checks

2.1.1 Probabilistic calibration

As we saw in Definition 2.4, the forecast F is probabilistically calibrated if Z_F is uniformly distributed. One way to write this condition is

$$\mathbb{P}(Z_F \leq \tau) = \tau, \quad \text{for all } \tau \in [0, 1]. \tag{2.1}$$

When F is continuous almost surely, we can also write this as

$$\mathbb{P}(Y \leq F^{-1}(\tau)) = \tau, \quad \text{for all } \tau \in [0, 1]. \tag{2.2}$$

To assess whether a forecast is probabilistically calibrated we can visually inspect the uniformity of the PIT. This is most commonly done by examining histograms of the PIT values obtained from samples. For a probabilistically calibrated forecast, the PIT histogram will be statistically uniform. Underdispersed or overdispered forecasts will have U-shaped or inverse-U-shaped PIT histograms respectively.
Another way to assess the uniformity of the PIT is by plotting the empirical CDF of the PIT values, as giving by Equation 2.1, and comparing it to the CDF of the uniform distribution.

2.1.2 Marginal calibration

By Definition 2.4, the forecast F is marginally calibrated if $\mathbb{E}_{\mathbb{P}}[F(y)] = \mathbb{P}(Y \leq y)$ for all $y \in \mathbb{R}$. To assess whether a forecast is marginally calibrated we can visually compare the sample-average of the CDF-valued forecast

$$\overline{F}(x) = \frac{1}{N}\sum_{i=1}^{N} F_i(x),$$

where $F_i(x)$ is the CDF-valued forecast for sample x_i, with the empirical CDF of the observations

$$\overline{G}_N(x) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\{x_i \leq x\}}.$$

It is often most instructive [14] to plot the difference $\overline{F}(x) - \overline{G}_N(x)$. If F is marginally calibrated then there should only be minor fluctuations about zero. The same information can be shown by plotting the difference between the quantile functions of $\overline{F}(x)$ and $\overline{G}_N(x)$. Again, here we only expect minor fluctuations about 0 under the hypothesis of marginal calibration.

2.2 Scoring Rules

A scoring rule S assigns a score $S(F, y)$ to each probabilistic forecast and target value pair (F, y). We will use the notation $S(F, Q)$ to denote the expectation of the score over samples $Y \sim Q$:

$$S(P, Q) = \mathbb{E}_{Y\sim Q}[S(P, Y)] = \int S(P, y)\mathrm{d}Q(y).$$

Definition 2.6. The scoring rule $S : \mathcal{F} \times \mathbb{R} \to \mathbb{R} \cup \{-\infty, \infty\}$ is proper relative to the class $\mathcal{F}$ if

$$S(G, G) \leq S(F, G)$$

for all $F, G \in \mathcal{F}$. It is strictly proper if this holds with equality if only if $F = G$.

Given a proper scoring rule S, we refer to the expected score function $e(F) = S(F, F)$ as the associated entropy and to the function $d(F, G) = S(F, G) - S(G, G) \geq 0$ as the corresponding divergence.

Theorem 2.7. *The scoring rule S is proper relative to the class $\mathcal{F}$ if and only if $e(F) = S(F, F)$ is concave and $S(F, \cdot)$ is a supergradient of e at the point F, for all $F \in \mathcal{F}$.*

For a proof of this result, see, e.g., Gneiting & Raftery (2007) [15].
We will now list a number of proper scoring rules. Throughout we will write the scores as functions of densities, CDFs, and quantile functions for convenience.

Logarithmic score (LS). The LS is defined as

$$\mathrm{LS}(p, y) = -\log p(y),$$

where p is a probability density or mass function. To see that the LS is a strictly proper score we compute the divergence:

$$\mathrm{LS}(p, q) - \mathrm{LS}(q, q) = \int q(y) \log \frac{q(y)}{p(y)}\mathrm{d}y. \tag{2.3}$$

This happens to be the Kullback-Leibler divergence, which is known to be non-negative and positive for $p \neq q$.

Quadratic/Brier score (BS). For a probability density or mass function p, the quadratic score is defined as

$$\mathrm{BS}(p, y) = -2p(y) + \|p\|_2^2,$$

where $\|p\|_2^2 = \int p(y)^2 \mathrm{d}y$. By computing the divergence we can see that the QS is also a strictly proper score:

$$\mathrm{BS}(p, q) - \mathrm{BS}(q, q) = \|p\|_2^2 + \|q\|_2^2 - 2\int p(y)q(y)\mathrm{d}y = \|p - q\|_2^2,$$

which is non-negative and zero if only if $p = q$.

Continuous ranked probability score (CRPS). The CRPS is defined in terms of the predictive CDF F for a forecast by

$$\begin{aligned}\mathrm{CRPS}(F, y) &= \int \left(F(x) - \mathbf{1}_{\{y \le x\}}\right)^2 \mathrm{d}x \\ &= \mathbb{E}_F[Y - y] - \frac{1}{2}\mathbb{E}_F[Y - Y'],\end{aligned}$$

where Y, Y' and independent random variables with CDF F and finite first moment.

Dawid-Sebastiani score (DSS). A viable alternative to the CRPS, which is easier to compute, is the DSS, which depends on the forecast only through the first two central moments, μ_F and σ_F^2, of the CDF. It is defined as

$$\mathrm{DSS}(F, y) = \frac{(y - \mu_F)^2}{\sigma_F^2} + 2\log\sigma_F.$$

Quantile score (QS). For a forecast represented as a set of predicted quantiles q_τ with $\tau \in \mathcal{T}$, the QS is defined as

$$\mathrm{QS}_{\mathcal{T}}\left(\{q_\tau\}_{\tau\in\mathcal{T}}, y\right) = \sum_{\tau\in\mathcal{T}} \rho_\tau(y - q_\tau),$$

where

$$\rho_\tau(u) = \begin{cases}\tau|u|, & \text{for } u \ge 0,\\ (1-\tau)|u|, & \text{for } u < 0.\end{cases}$$

The CRPS and QS and closely connected. If F is differentiable, with density f and has a finite expectation, then the CRPS can be written as

$$\mathrm{CRPS}(F, y) = \int \left(F(x) - \mathbf{1}_{\{y \le x\}}\right)^2 \mathrm{d}x = 2\int_0^1 \rho_\tau(y - F^{-1}(\tau))\mathrm{d}\tau.$$

Here the right-hand side is just an integral of the QS over all quantiles $\tau \in [0, 1]$.

2.2.1 Diebold-Mariano test

If we want to compare two forecasts F and G we can do so by comparing their average performance over a test set. The average scores are

$$\overline{S}_n^F = \frac{1}{n}\sum_{i=1}^n S(F_i, y_i) \qquad \text{and} \qquad \overline{S}_n^G = \frac{1}{n}\sum_{i=1}^n S(G_i, y_i).$$

If the forecast cases are independent, a test of equal performance can be based on the statistic

$$t_n = \sqrt{n}\frac{\overline{S}_n^F - \overline{S}_n^G}{\hat{\sigma}_n},$$

where

$$\hat{\sigma}_n^2 = \frac{1}{n}\sum_{i=1}^{n}\left(S(F_i, y_i) - S(G_i, y_i)\right)^2$$

is an estimate of the variance of the score differential. Under the null hypothesis of a vanishing score differential the statistic t_n is asymptotically normal. If the null hypothesis is rejected we have that F is preferred when t_n is negative, and G is preferred when t_n is positive.

However, if the forecast cases are not independent, such as in the case of k-step-ahead time series forecasts one must generalise the variance estimate $\hat{\sigma}_n^2$ to account for autocorrelation. Diebold and Mariano [16] take the following approach for this. Let $d_i = S(F_i, y_i) - S(G_i, y_i)$ and $\overline{d} = \overline{S}_n^F - \overline{S}_n^G$ for convenience. Then denote by

$$\gamma_d(\tau) = \mathbb{E}\left[(d_t - \mu)(d_{t-\tau} - \mu)\right]$$

the autocovariance of the score differential at displacement τ, where μ is the population mean score differential, and let

$$f_d(0) = \frac{1}{2\pi}\sum_{\tau=-\infty}^{\infty}\gamma_d(\tau)$$

be the sample mean loss differential. We then get the statistic

$$\hat{t}_n = \frac{\overline{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{n}}},$$

where $\hat{f}_d(0)$ is a consistent estimator of $f_d(0)$, which we can obtain by

$$2\pi\hat{f}_d(0) = \sum_{\tau=-(n-1)}^{n-1} W\left(\frac{\tau}{L(n)}\right)\hat{\gamma}_d(\tau),$$

where

$$\hat{\gamma}_d(\tau) = \frac{1}{n}\sum_{t=|\tau|+1}^{n}(d_t - \overline{d})(d_{t-|\tau|} - \overline{d}),$$

and $L(n) = k - 1$ is the truncation lag and

$$W\left(\frac{\tau}{L(n)}\right) = \begin{cases} 1, & \text{if } \left|\frac{\tau}{L(n)}\right| \leq 1, \\ 0, & \text{otherwise} \end{cases}$$

is the lag window.

3 Distributional Reinforcement Learning

For this section we largely follow the book Distributional Reinforcement Learning by Bellemare, Dabney, and Rowland [17]. Throughout we use the notation $\mathscr{P}(S)$ to refer to the space of probability distributions supported on the space S.
Reinforcement learning, in a basic setting, can be modelled with a Markov decision process (MDP) $(\mathcal{X}, \mathcal{A}, \xi_0, P_{\mathcal{X}}, P_{\mathcal{R}})$, where

- $\mathcal{X}$ is the set of environment and agent states;
- $\mathcal{A}$ is the set of actions of the agent;
- $\xi_0 \in \mathscr{P}(\mathcal{X})$ the probability distribution of the initial state X_0;
- $P_{\mathcal{X}}$ is the transition kernel, such that $X_{t+1} \sim P_{\mathcal{X}}(\cdot|X_t, A_t)$;

- $P_{\mathcal{R}}$ is the reward distribution, such that $R_t \sim P_{\mathcal{R}}(\cdot|X_t, A_t)$.

The MDP contains all the information needed to describe how the agent's decision influence the environment. The decision of the agent itself arise from a policy, which is a mapping $\pi : \mathcal{X} \to \mathscr{P}(\mathcal{A})$ such that

$$A_t \sim \pi(\cdot|X_t).$$

The goal of reinforcement learning is for the agent to learn a policy which maximises the expected cumulative reward. This expected cumulative reward is called the discounted return, often referred to as just the return. Which is discounted by some discount factor $\gamma \in [0, 1)$ and given by

$$G = \sum_{t=0}^{\infty} \gamma^t R_t.$$

The discount factor encodes a preferences for receiving rewards sooner rather later. This return is the main objective to optimise in reinforcement learning. When the agent is in a setting without a terminal state, the discount factor can also be used to guarantee that G exists and is finite. When the rewards are bounded on an interval $[R_{\min}, R_{\max}]$, we have that the return is bounded as

$$G \in \left[\frac{R_{\min}}{1-\gamma}, \frac{R_{\max}}{1-\gamma}\right].$$

When the rewards are not bounded we can still guarantee the existences of the return if we allow for the following assumption

Assumption 3.1. *For each state $s \in \mathcal{S}$ and action $a \in \mathcal{A}$, the reward distribution $\mathbb{P}_{\mathcal{R}}(\cdot|s, a)$ has finite first moment. That is, if $R \sim \mathbb{P}_{\mathcal{R}}(\cdot|s, a)$, then*

$$\mathbb{E}[|R|] < \infty.$$

Which leads us to the following proposition for the random return.

Proposition 3.2. *Under Assumption 3.1, the random return G exists and is finite with probability 1:*

$$\mathbb{P}_\pi\left(G \in (-\infty, \infty)\right) = 1.$$

A proof of this proposition can be found in Bellemare et. al. (2023) [17].
We refer to the probability distribution of G as the return distribution. This return distribution determines certain quantities such as the expected return

$$\mathbb{E}_\pi[G] = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R_t\right].$$

There is also the variance of the return

$$\mathrm{Var}_\pi(G) = \mathbb{E}_\pi\left[(G - \mathbb{E}_\pi[G])^2\right]$$

and tail probabilities such as

$$\mathbb{P}_\pi\left(\sum_{t=0}^{\infty} \gamma^t R_t \geq 0\right),$$

which is used in risk-sensitive problems as we will discuss later.

3.1 Bellman equation

Different policies can lead to wildly different return distributions. To determine which policy to follow we typically look at the expected value of its random return:

$$\mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t R_t\right].$$

Being able to evaluate the expected return is vital for most reinforcement learning algorithms. A simple approach would be to enumerate each possible realisations of the random trajectory $(X_t, A_t, R_t)_{t\geq 0}$. However, this approach quickly becomes infeasible as the number of trajectories grows exponentially in time.
The Bellman equation provides a concise characterisation of the expected return under a given policy. For this we need to define the value function V^π, which is defined as

$$V^\pi(x) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t R_t | X_0 = x\right].$$

Using the value function we can determine the expected return and decompose it into an immediate reward R_0 and future rewards:

$$\mathbb{E}\left[V^\pi(X_0)\right] = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right] = \mathbb{E}\left[R_0 + \sum_{t=1}^{\infty}\gamma^t R_t\right].$$

Using this we introduce the Bellman equation.

Proposition 3.3 (The Bellman equation)**.** *Let V^π be the value function of policy π. Then for any state $s \in \mathcal{S}$, it holds that*

$$V^\pi(x) = \mathbb{E}_\pi\left[R_0 + \gamma V(X_1) | X_0 = x\right].$$

A proof of the Bellman equation in this form can be found in Bellemare et. al. (2023) [17]. With the Bellman equation we transform an infinite sum into a recursive relation, making it possible to devise efficient algorithms for determining the value function.
There is also an alternative to the value function, for which the following results also hold. Namely, the state-action value function Q^π which is defined as

$$Q^\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t | X_0 = x, A_0 = a\right],$$

where the return is also fixed to a particular action a.

In order to prove Proposition 3.3 and extend the Bellman equation to a distributional form we need to introduce two particular properties of the random trajectories $(X_t, A_t, R_t)_{t\geq 0}$: time-homogeneity and the Markov property.
Denote by $\mathcal{D}(Z)$ the probability distribution of a random variable Z. When Z is real-valued, we have that

$$\mathcal{D}(Z)(S) = \mathbb{P}(Z \in S),$$

for $S \subseteq \mathbb{R}$. We write $\mathcal{D}_\pi$ to refer to the distribution of a random variable derived from the joint distribution $\mathbb{P}_\pi$.

Lemma 3.4 (Markov property)**.** *The trajectory $(X_t, A_t, R_t)_{t\geq 0}$ has the Markov property. That is, for any $k \in \mathbb{N}$, we have*

$$\mathcal{D}_\pi\left((X_t, A_t, R_t)_{t=k}^{\infty} | (X_t, A_t, R_t)_{t=0}^{k-1} = (x_t, a_t, r_t)_{t=0}^{k-1},\ X_t = x\right) = \mathcal{D}_\pi\left((X_t, A_t, R_t)_{t=k}^{\infty} | X_k = x\right)$$

whenever the conditional distribution of the left-hand side is defined.

Lemma 3.5 (Time-homogeneity)**.** *The trajectory $(X_t, A_t, R_t)_{t\geq 0}$ is time-homogeneous, in the sense that for all $k \in \mathbb{N}$,*

$$\mathcal{D}_\pi\left((X_t, A_t, R_t)_{t=k}^{\infty} | X_k = x\right) = \mathcal{D}_{\delta_k, \pi}\left((X_t, A_t, R_t)_{t=0}^{\infty}\right),$$

whenever the conditional distribution of the left-hand side is defined.

For a complete proof of these lemmas, see Bellemare et. al. (2023) [17], Remark 2.4.
It is convenient to define a generative model that only considers the three random variables of the immediate action, reward, and successive state along with the initial state, as these are the only terms needed with the Bellman equation.

Let $\xi \in \mathscr{P}(\mathcal{X})$ be a distribution of the states. We now introduce the sample transition model which assigns a probability distribution to the tuple (X, A, R, X') taking values in $\mathcal{X} \times \mathcal{A} \times \mathbb{R} \times \mathcal{X}$ like

$$\begin{aligned}
&X \sim \xi;\\
&A|X \sim \pi(\cdot|X);\\
&R|(X,A) \sim P_{\mathcal{R}}(\cdot|X,A);\\
&X'|(X,A,R) \sim P_{\mathcal{X}}(\cdot|X,A).
\end{aligned}$$

We use $\mathbb{P}_\pi$ to refer to the joint distribution of these random variables. When considering a single source state x we have that $\xi = \delta_x$ and we write $(X = x, A, R, X')$ for the random tuple. With probability and expectation

$$\mathbb{P}_\pi(\cdot|X = x) \quad \text{and} \quad \mathbb{E}_\pi[\cdot|X = x].$$

With this generative model we can omit the time indices in the Bellman equation and it reduces to

$$V^\pi(x) = \mathbb{E}_\pi[R + \gamma V^\pi(X')|X = x].$$

We can now use the Markov property and time-homogeneity to not just characterise the expected value of the random return for any state x through the Bellman equation, but all aspects of the random return. For this we define the return-variable function

$$G^\pi(x) = \sum_{t=0}^{\infty} \gamma^t R_t, \qquad X_0 = x,$$

which describes the return obtained when following policy π starting from state x.

Proposition 3.6. *Let G^π be the return-variable function of policy π. For a sample transition $(X = x, A, R, X')$ independent of G^π, it holds that for any state $x \in \mathcal{X}$*

$$G^\pi(x) \overset{\mathcal{D}}{=} R + \gamma G^\pi(X'), \quad X = x. \tag{3.1}$$

Note that we have equality in distribution here, that is the random variables on either side have equal probability distribution. The proof is provided in Bellemare et. al. (2023) [17].
In order to write the Bellman equation in distribution form we need to find analogous operations for the indexing into G^π, scaling, and addition in the right-hand side of Equation 3.1.

For the indexing we note that $G^\pi(X')$ describes the random return received at the successor state X' when $X = x$ and A is drawn from $\pi(\cdot|X)$. This can be reformulated as first sampling a state x' from the distribution of X' and then sampling a realised return $G^\pi(x')$. If we let the result be $G^\pi(X')$, we have for subset $S \subseteq \mathbb{R}$ that

$$\begin{aligned}
\mathbb{P}_\pi(G^\pi(X') \in S|X = x) &= \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X' = x'|X = x)\mathbb{P}_\pi(G^\pi(X') \in S|X' = x', X = x)\\
&= \sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X' = x'|X = x)\mathbb{P}_\pi(G^\pi(x') \in S)\\
&= \left(\sum_{x' \in \mathcal{X}} \mathbb{P}_\pi(X' = x'|X = x)\eta^\pi(x')\right)(S),
\end{aligned}$$

where $\eta^\pi(x) = \mathcal{D}_\pi(G^\pi(x))$ denotes the return distribution. This illustrates that $G^\pi(X')$ is a mixture of probability distributions from η^π.

For the analogous operations to scaling and addition we introduce the pushforward distribution and the bootstrap function. For a random variable $Z \sim \nu$ and a transformation $f : \mathbb{R} \to \mathbb{R}$ the pushforward distribution $f_\#\nu$ is defined as

$$f_\#\nu = \mathcal{D}(f(Z)).$$

For two scalars $r \in \mathbb{R}$ and $\gamma \in [0, 1)$ the bootstrap function is defined as

$$b_{r,\gamma} : z \mapsto r + \gamma z.$$

Combining these operations we get

$$(b_{r,\gamma})_{\#} = \mathcal{D}(r + \gamma Z).$$

Applying this to the return distribution $\eta^\pi(x')$, of state x', we get

$$(b_{r,\gamma})_{\#}\eta^\pi(x') = \mathcal{D}(r + \gamma G^\pi(x')).$$

We can now combine this with the mixing to get

$$(b_{r,\gamma})_{\#}\mathbb{E}_\pi[\eta^\pi(X')|X = x] = \mathbb{E}_\pi[(b_{r,\gamma})_{\#}\eta^\pi(X')|X = x].$$

Which follows from the linearity of the pushforward operation. With this we can now write the Bellman equation in distributional form, a derivation of which can be found in Bellemare et. al. (2023) [17].

Proposition 3.7 (Distributional Bellman equation)**.** *Let η^π be the return-distribution function of policy π. For any state $x \in \mathcal{X}$, we have*

$$\eta^\pi(x) = \mathbb{E}_\pi[(b_{R,\gamma})_{\#}\eta^\pi(X')|X = x].$$

Lastly, we introduce the Bellman operator which will be useful for the following sections.

Definition 3.8. The distributional Bellman operator $\mathcal{T}^\pi : \mathscr{P}(\mathbb{R})^{\mathcal{X}} \to \mathscr{P}(\mathbb{R})^{\mathcal{X}}$ is the mapping defined by

$$(\mathcal{T}^\pi\eta)(x) = \mathbb{E}_\pi[(b_{R,\gamma})_{\#}\eta(X')|X = x]$$

Similarly to the Bellman equation we can also write this in terms of the random variables:

$$(\mathcal{T}^\pi G)(x) \overset{\mathcal{D}}{=} R + \gamma G(X'), \quad X = x.$$

3.2 Categorical Distributional RL

There are many reinforcement learning algorithms and strategies that can be used to learn the return distribution and exploit it to find the optimal policy π^* which maximises the expected return:

$$\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty}\gamma_t R_t\right] \geq \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma_t R_t\right], \quad \text{for all } \pi.$$

In this thesis we will solely focus on the extension of Q-learning to the distributional setting in a categorical representation.

In Q-learning, the primary focus is on the state-action value function Q^π, given by

$$Q^\pi(x, a) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t|X_0 = x, A_0 = a\right],$$

as opposed to the value function V^π. The state-action value function satisfies the same (distributional) Bellman equations as the value function. Particularly, for an optimal policy π^*, the associated state-action value function satisfies the Bellman optimality equation

$$Q^{\pi^*}(x, a) = \mathbb{E}_\pi[R + \gamma \max_{a' \in \mathcal{A}} Q^{\pi^*}(X', a')|X = x, A = a].$$

Q-learning works by maintaining an estimate Q of the state-action value function, which is updated as

$$Q(x, a) \leftarrow (1 - \alpha)Q(x, a) + \alpha(r + \gamma \max_{a' \in \mathcal{A}} Q(x', a')).$$

We can extend this to the distributional setting by expressing the maximal action as a greedy policy. If we let η be the return-function estimate over state-action pairs, such that $\eta(x,a)$ is the return distribution associated with $(x,a) \in \mathcal{X} \times \mathcal{A}$, then the greedy action is defined as

$$a_\eta(x) = \arg\max_{a \in \mathcal{A}} \mathbb{E}_{Z \sim \eta(x,a)}[Z].$$

We represent the return-distribution estimate η in categorical form:

$$\eta(x,a) = \sum_{i=1}^{m} p_i(x,a)\delta_{z_i},$$

where $\{z_i\}_{i=1}^m$ represents the fixed, discrete support and p_i the assigned probabilities. It is clear that the categorical distributions are not closed under the distributional Bellman operation. Thus it is necessary to project the distribution back onto the fixed support $\{z_i\}_{i=1}^m$.

Definition 3.9. Given a probability distribution ν in categorical representation with fixed support $\{\zeta_i\}_{i=1}^m$ and probabilities q_i, $i = 1, \ldots, m$, its *categorical projection* is

$$\Pi_c \nu = \Pi_c \sum_{i=1}^{m} q_i \delta_{\zeta_i} = \sum_{i=1}^{m} q_i \Pi_c(\delta_{\zeta_i}),$$

where

$$\Pi_c(\delta_{\zeta_j}) = \begin{cases} \delta_{z_1}, & \zeta_j \leq z_1, \\ \frac{z_{i+1}-\zeta_j}{z_{i+1}-z_i}\delta_{z_i} + \frac{\zeta_j - z_i}{z_{i+1}-z_i}\delta_{z_{i+1}}, & z_i < \zeta_j \leq z_{i+1} \\ \delta_{z_m}, & \zeta_j > z_m. \end{cases}$$

Note that we interchangeably use Π_c as an operator on probability distributions and as a function of Dirac measures for categorically represented distributions.
The categorical Q-learning update rule then is given by

$$\eta(x,a) \leftarrow (1-\alpha)\eta(x,a) + \alpha\left(\Pi_c (b_{r,\gamma})_\# \eta(x', a_\eta(x'))\right),$$

3.2.1 C51 Algorithm

The C51 algorithm [18] is an extension of a deep-Q network (DQN) [19] to the distributional setting.

The C51 algorithm models the value distributions using a discrete distribution parameterised by $V_{\min}, V_{\max}$ and $m \in \mathbb{N}$, with support $\{z_i = V_{\min} + i\frac{V_{\max} - V_{\min}}{m-1}\}$. The probabilities on the support are given by the main parametric model $\theta : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^m$

$$\eta_\theta(x,a) = \sum_{i=1}^{m} p_i^\theta(x,a)\delta_{z_i},$$

with probabilities

$$p_i^\theta(x,a) = \frac{\exp(\theta_i(x,a))}{\sum_{j=1}^m \exp(\theta_j(x,a))}.$$

The induced state-action value estimates are then given by

$$Q_\theta(x,a) = \sum_{i=1}^{m} p_i^\theta(x,a) z_i.$$

The C51 algorithm also employs a second target model $\tilde{\theta} : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^m$, defined identically to the main model.
For a sample transition (x, a, r, x'), the sample target is given by

$$\overline{\eta}(x,a) = \sum_{j=1}^{m} \overline{p}_j \delta_{z_j} = \Pi_c\left((b_{r,\gamma})_\# \eta_\theta(x', a_{\tilde{\theta}}(x'))\right),$$

where

$$a_{\tilde{\theta}}(x') = \arg\max_{a' \in \mathcal{A}} Q_{\tilde{\theta}}(x', a')$$

is the greedy action for the induced state-action value function of the target model.
The sample target is then used to formulate a cross-entropy loss

$$\mathcal{L}(\theta) = -\sum_{i=1}^{m} \overline{p}_i \log p_i^{\theta}(x, a).$$

Which is minimised by updating the parameters θ in the direction of the negative gradient of this loss. This yields the following semi-gradient update rule:

$$\theta \leftarrow \theta + \alpha \sum_{i=1}^{m} \left(\overline{p}_i - p_i^{\theta}(x, a)\right) (\nabla_{\theta} z_i)(x, a)$$

4 Risk-Sensitive Setting

Both forecasting and distributional reinforcement learning (DRL) are usually done in the so called risk-neutral setting, where we optimise over the expectation of the scores of proper scoring rules in forecasting or the expectation of the returns in DRL. However, it is often desirable that our forecasts or decisions are informed by the variability in the process in which our outputs are used. We refer to this variability as risk. This is especially evident in the data set and problem considered in this thesis. An airplane engine failing mid-flight can have much more severe consequences than just the monetary costs associated with it, which is typically the objective that is optimised.

The risk-sensitive setting does take this variability into account. Both probabilistic forecasting and DRL can be done in this manner and we will introduce some methods to achieve this in this section.

4.1 Forecasting

In Section 2 we have seen that in (probabilistic) forecasting tasks we typically use the expectation of the score over some samples, determined by a (proper) scoring rule, to determine the quality or performance of a forecast. Logically these same metrics are also used to optimise forecasting models using training data.

Recall that by Definition 2.6 a scoring rule is proper relative to the class $\mathcal{F}$ is $S(G, G) \leq S(F, G)$ for all $F, G \in \mathcal{F}$. For a density forecast f this condition is equivalent to

$$\mathbb{E}_f[S(f, Y)] = \int f(y) S(f, y) \mathrm{d}y \leq \int f(y) S(g, y) \mathrm{d}y = \mathbb{E}_f[S(g, Y)]$$

which has to hold for all density functions f and g. Similarly it is strictly proper if the condition holds with equality if and only if $f = g$ almost surely.

One way to do forecasting in this risk-sensitive setting is by using weighted scoring rules. The weight function is used to emphasise certain regions of interest, such as the left or right tails or the centre of the forecast distribution. Amisano and Giacomini [20] used a weighted logarithmic scoring rule

$$S(f, y) = w\left(\frac{y - \mu}{\sigma}\right) S_0(f, y),$$

where w is a fixed, non-negative weight function, μ and σ are estimates of the mean and standard deviation, and S_0 is the logarithmic scoring rule.
This approach however has a problem. As Gneiting and Ranjan [21] showed this results in the use of an improper scoring rule. In fact, the following result [21] shows that it is the case that for any strictly proper scoring rule $S_0(f, y)$, its product with a weight function $w(y)$ is improper, unless the weight function is constant.

Emphasis	Threshold weight function	Quantile weight function
Center	$u(z) = \phi_{a,b}(z)$	$\nu(\alpha) = \alpha(1-\alpha)$
Tails	$u(z) = 1 - \phi_{a,b}(z)/\phi_{a,b}(a)$	$\nu(\alpha) = (2\alpha - 1)^2$
Right tail	$u(z) = \Phi_{a,b}(z)$	$\nu(\alpha) = \alpha^2$
Left tail	$u(z) = 1 - \Phi_{a,b}(z)$	$\nu(\alpha) = (1-\alpha)^2$

Table 1: Proposed weight functions from [21] for the threshold and quantile weighted version of the CRPS. Here $\phi_{a,b}$ and $\Phi_{a,b}$ are the PDF and CDF of the normal distribution with mean a and standard deviation b.

Theorem 4.1. *Suppose that f is the sampling density of the random variable Y. Let S_0 be any proper scoring rule and let W be a weight function such that $0 < \int w(y)f(y)dy < \infty$. Then the expected value of the weighted score*

$$S(g, Y) = w(Y)S_0(g, Y)$$

is minimised for the density forecast

$$g(y) = \frac{w(y)f(y)}{\int w(y)f(y)\,dy}.$$

Proof. Let h be any density forecast. Then

$$\begin{aligned}
\mathbb{E}_f[g, Y] &= \int w(y)f(y)S_0(g, y)\mathrm{d}y \\
&= \int w(y)f(y)\mathrm{d}y \int g(y)S_0(g, y)\mathrm{d}y \\
&\leq \int w(y)f(y)\mathrm{d}y \int g(y)S_0(g, y)\mathrm{d}y \\
&= \int w(y)f(y)S_0(h, y)\mathrm{d}y \\
&= \mathbb{E}_f[S(h, Y)]
\end{aligned}$$

□

Gneiting and Ranjan [21] then define the alternative approach of a threshold or quantile weighted version of the CRPS. The threshold weighted version is defined as

$$\text{twCRPS}(f, y) = \int_{-\infty}^{\infty} \left(F(z) - \mathbf{1}_{\{y \leq z\}}\right)^2 u(z)\mathrm{d}z,$$

where $u(z)$ is a non-negative function on $\mathbb{R}$, if $u(z) = 1$ this reduces to the normal CRPS. Analogously, they define the quantile weighted version as

$$\text{qwCRPS}(f, y) = 2\int_0^1 \left(\mathbf{1}\{y \leq F^{-1}(\alpha)\} - \alpha\right)\left(F^{-1}(\alpha) - y\right)\nu(\alpha)\mathrm{d}\alpha,$$

where $\nu(\alpha)$ is a non-negative function on the unit interval $[0, 1]$. Matheson and Winkler [22] showed that both these scoring rules are proper.
Table 1 shows some possible weight functions with different emphases. By optimising and testing forecasts over these weighted functions we can operate in a risk-sensitive setting by putting the emphasis on either of the tails of the distributions.

4.2 Distributional Reinforcement Learning

As we have seen in Section 3 DRL agents are optimised over the expectation of the returns

$$\mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^t R_t\right].$$

However, with distributional reinforcement learning we have access to (an estimate of) the entire distribution. Thus allowing us to account for risk in the decision making. We do this by replacing the expectation by a risk measure.

Definition 4.2. A risk measure is a mapping $\rho : \mathscr{P}(\mathbb{R}) \to [-\infty, \infty)$ defined on a subset $\mathscr{P}_\rho(\mathbb{R}) \subseteq \mathscr{P}(\mathbb{R})$ of probability distributions.

Thus yielding the objective

$$\rho\left(\sum_{t=0}^{\infty} \gamma^t R_t\right).$$

Some examples of risk measures are:

Example 4.3 (Mean-variance criterion)**.** Let $\lambda > 0$. The variance-penalized risk measure penalizes high variance outcomes and is given by

$$\rho_{\mathrm{MV}}^{\lambda}(Z) = \mathbb{E}[Z] - \lambda \mathrm{Var}(Z).$$

△

Example 4.4 (Entropic risk)**.** Let $\lambda > 0$. Entropic risk emphasises low-valued outcomes:

$$\rho_{\mathrm{ER}}^{\lambda}(Z) = -\frac{1}{\lambda} \log \mathbb{E}[e^{-\lambda Z}].$$

△

Example 4.5 (Value-at-risk)**.** Let $\tau \in (0,1)$. The value-at-risk measure corresponds to the τth quantile of the return:

$$\rho_{\mathrm{VaR}}^{\tau}(Z) = F_Z^{-1}(\tau).$$

△

Whilst the mean-variance criterion seems like an excellent way to do risk-sensitive learning it does have a few issues. Most notably it can not differentiate between "downside risk", meaning greater than expected losses, and "upside risk", involving greater than expected gains. In many situations we might only care about one of the two. Particularly in risk-averse settings we want to minimise the downside risk, whilst the upside risk does not matter too much.

One way to account for this is by consider the conditional value-at-risk (CVaR) [23] measure.

Definition 4.6. Let Z be a random variable with cumulative distribution function F_Z and its inverse F_Z^{-1}. For $\tau \in (0,1)$ the CVaR of Z is defined as

$$\mathrm{CVaR}_\tau(Z) = \frac{1}{\tau} \int_0^\tau F_Z^{-1}(u) \mathrm{d}u.$$

For strictly increasing F_Z^{-1} the definition is equivalent to

$$\mathbb{E}[Z | Z \leq F_Z^{-1}(\tau)].$$

For a discrete probability distribution $\eta = \sum_{i=1}^m p_i \delta_{z_i}$, as we have in our categorical representation of DRL in Section 3.2, with sorted support $z_1 \leq \cdots \leq z_m$ and assigned probabilities p_i such that $p_1 + \cdots + p_m = 1$ we have the following explicit expression for the CVaR:

$$\begin{aligned} \mathrm{CVaR}_\tau(\eta) &= \frac{1}{\tau} \sum_{i=1}^m \left| [0, \tau] \cap \left[\sum_{j \leq i-1} p_j, \sum_{j \leq i} p_j \right] \right| z_i \\ &= \frac{1}{\tau} \sum_{i=1}^m \left(\min\left(\tau, \sum_{j \leq i} p_j\right) - \max\left(0, \sum_{j \leq i-1} p_j\right)\right)^+ z_i, \end{aligned} \tag{4.1}$$

where $(\cdot)^+ = \max(0, \cdot)$.

5 Methodology

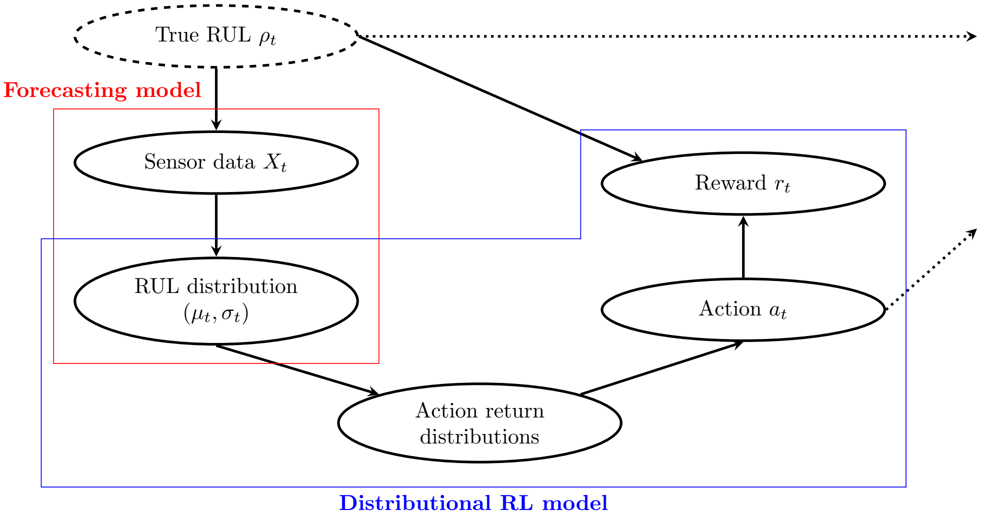


Figure 1: Overview of the proposed pipeline using RUL prognostics from probabilistic forecasts for predictive maintenance scheduling with a distributional reinforcement learning model.

The primary goal of this thesis is to investigate a risk-averse strategy for maintenance scheduling. To do this we will construct a relatively simple pipeline in both risk-neutral and risk-averse settings and compare them. The pipeline will consist of two parts (Figure 1); a parametric probabilistic forecasting model, which will predict a distribution of the RUL given a limited window of sensor data, and a distributional reinforcement learning model which takes these probabilistic forecasts of the RUL and determines the best replacement policy for the engines. Since we can do both the forecasting and the maintenance scheduling in risk-neutral and averse settings we will end up comparing four different models:

- **Neutral-Mean**: neutral forecast and neutral scheduling,
- **Neutral-CVaR**: neutral forecast and averse scheduling,
- **Averse-Mean**: averse forecast and neutral scheduling,
- **Averse-CVaR**: averse forecast and averse scheduling.

We will train and test this pipeline on the C-MAPSS data set.

5.1 Data set

The C-MAPSS data set consists of degradation data of aircraft turbofan engines obtained through run-to-failure simulations. It is divided into four subsets: FD001, FD002, FD003, and FD004 with different operating conditions and fault modes. Each subset contains a set of simulations of an engine ran till failure for training and a testing set consisting of segments of simulations (Table 2).
Each instance consists of a time-series of 21 sensor measurements, three operating conditions, and two vectors containing the current flight cycle and the engine unit number (Table 3). An instance is given by a $n \times 26$ matrix, where n is the total number of flight cycles for that instance.

Each data set will have its training instances split into three subsets: the forecast training subset containing 70% of the training instances, the maintenance scheduling subset with 20% of the training instances, and the remaining 10% will make up the testing set for maintenance scheduling. Furthermore, 5% of the forecast and maintenance scheduling training subsets will be used for validation instead of training.

	FD001	FD002	FD003	FD004
Training instances	100	260	100	249
Testing instances	100	259	100	248
Operating conditions	1	6	1	6
Fault modes	1	1	2	2

Table 2: C-MAPSS data sets.

Column	Content
1	Engine unit number
2	Time in cycles
3	Operational Setting 1
4	Operational Setting 2
5	Operational Setting 3
6	Sensor 1
⋮	⋮
26	Sensor 21

Table 3: C-MAPSS data set format

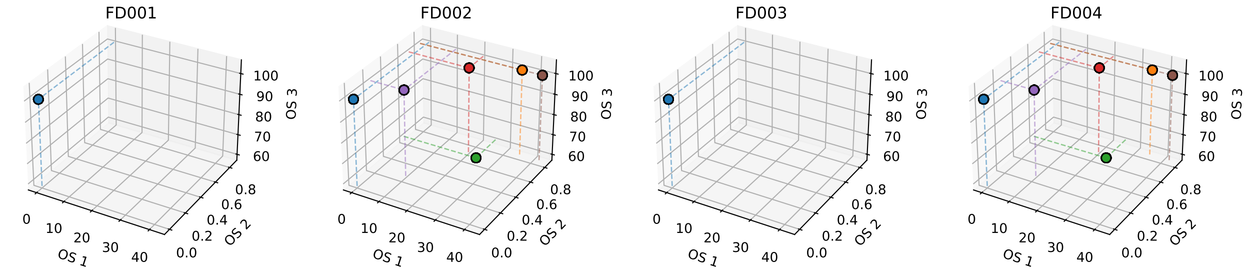


Figure 2: Operational Settings for each subset clustered.

5.2 Preprocessing

The sensors 1, 5, 6, 10, 16, 18, and 19 are constant valued for the same operating conditions, meaning that they do not contain any information that is not already contained in the operational settings. We thus exclude these sensors from here on and refer to the remaining sensors as sensors 1 through 14. Using K-Means clustering on the three operational settings we identified six different operating conditions. The operating conditions in subsets FD002 and FD004 are the same, and the single operating condition of FD001 and FD003 is also contained in FD002 and FD004 as can be seen in Figure 2. With this we reduce the three operational settings to a single time series $o_t \in \{1, \ldots, 6\}$, denoting the operational condition at cycle t. We then normalize the measurements from the 14 useful sensors with respect to their operating conditions. Denote the measurement of sensor i at cycle t by $s_{i,t}$ and the operating condition at cycle t by o_t. The measurements are then normalized as

$$s_{i.t} = \frac{s_{i,t} - L_i^{o_t}}{U_i^{o_t} - L_i^{o_t}},$$

where $L_i^{o_t}$ and $U_i^{o_t}$ are the minimal and maximal values that sensor i attains whilst in operating condition o_t over all the training instances in the data set.

Lastly, we use a sliding window of size W for each training and testing instance. This results in the following matrices that will be used as input to the forecasting models:

$$X_t = \begin{bmatrix} o_{t-W+1} & s_{1,t-W+1} & \cdots & s_{14,t-W+1} \\ \vdots & \vdots & & \vdots \\ o_t & s_{1,t} & \cdots & s_{14,t} \end{bmatrix}.$$

Note that we label the matrix by the cycle of the last row of data included.

5.3 RUL Forecasting

For the forecasting we opt for a model with a similar architecture to the one used in [8]. The model will take inputs in the shape of a $W \times 15$ matrix X_t, where W is the window size and 15 corresponds to the 14 sensors and 1 input for the operating condition. For an engine, with a data trajectory of length n, we defined the RUL at cycle t as $\text{RUL}_t = min\{C, n-t\}$. We will use $C = 128$ for all experiments. This splits the degradation process into two parts; a constant phase where there is little to no degradation and a linearly decreasing phase. The goal is then to forecast a distribution over the RUL given a window of data X_t, where the time index t corresponds to the time of the final data point in the window. For simplicity and since the lengths of the trajectories are distributed similar to this, we will use the log-normal distribution to model the distribution of the RUL. This is also a reasonable choice to include the non-negativity of the RUL, as the log-normal distribution has a support of $(0, +\infty)$. The log-normal distribution is defined as the distribution of the random variable

$$X = e^{\mu + \sigma Z},$$

where Z is a standard normal random variable and $\sigma > 0$. The probability density function (PDF) is given by

$$f_X(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left(-\frac{(\log x - \mu)^2}{2\sigma^2}\right),$$

and the cumulative distribution function (CDF) by

$$F_X(x) = \Phi\left(\frac{\log x - \mu}{\sigma}\right).$$

Lastly the mean, median, and variance are defined as

$$\begin{aligned}\mathbb{E}[X] &= \exp\left(\mu + \frac{\sigma^2}{2}\right),\\ \text{Med}[X] &= \exp(\mu),\\ \text{Var}[X] &= \exp\left(2\mu + \sigma^2\right)\left[\exp(\sigma^2) - 1\right].\end{aligned}$$

5.3.1 Network Architecture

To compute μ and σ we will employ a neural network with two stacked LSTMs with a dropout layer in-between and a fully connected layer for the final output (Figure 3).

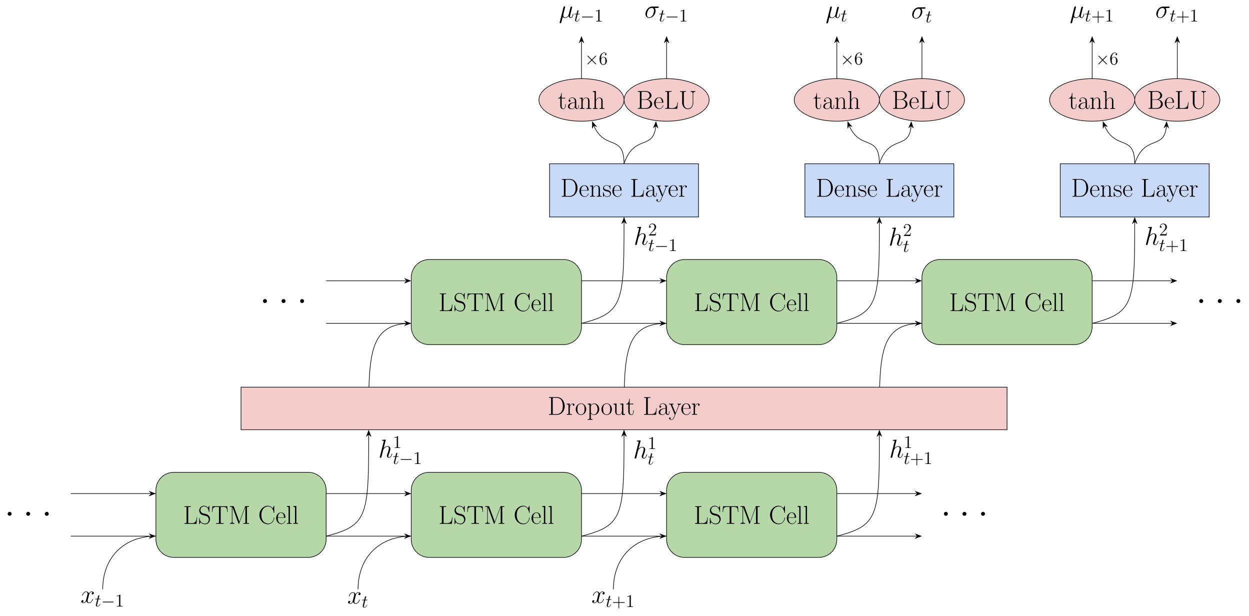


Figure 3: Forecast Network architecture.

We use the tanh activation function, multiplied by 6, for the final output of μ, resulting in μ being between -6 and 6. Thus we have that the minimum value of the median $\exp(\mu)$ is approximately 0.00248 and the maximum is approximately 403.429. Similarly, we will use a bounded version of ELU activation function for σ:

$$\mathrm{BELU}(x) = \begin{cases} e^x, & \text{if } x \leq 0 \\ x+1, & \text{if } 0 < x < U \\ U+1, & \text{if } x \geq U \end{cases}$$

Loss Function
We employ a weighted sum of the CRPS (or threshold weighted CRPS for the risk-averse setting) over each of the W outputs:

$$L_{\text{neutral}}(X_t, \hat{y}_t) = \sum_{i=1}^{W} \frac{i}{\frac{1}{2}W(W+1)} \mathrm{CRPS}(\mu_i, \sigma_i; \min(C, y_t + W - i)),$$
$$L_{\text{averse}}(X_t, \hat{y}_t) = \sum_{i=1}^{W} \frac{i}{\frac{1}{2}W(W+1)} \mathrm{twCRPS}(\mu_i, \sigma_i; \min(C, y_t + W - i)),$$

where

$$\mathrm{CRPS}(\mu_i, \sigma_i; y) = \int_{-\infty}^{\infty} \left(\Phi\left(\frac{\log x - \mu_i}{\sigma_i} \right) - \mathbf{1}_{\{y \leq x\}} \right)^2 \mathrm{d}x,$$
$$\mathrm{twCRPS}_b(\mu_i, \sigma_i; y) = \int_{-\infty}^{\infty} \left(\Phi\left(\frac{\log x - \mu_i}{\sigma_i} \right) - \mathbf{1}_{\{y \leq x\}} \right)^2 \Phi\left(\frac{x-y}{b} \right) \mathrm{d}x.$$

In these loss functions the later time steps, within the same window, are weighted more than the earlier ones in a linearly increasing way. For the threshold weighting function we use the CDF of the normal distribution with mean y and variance b^2 from Table 1 to put an emphasis on the right tail. The parameter b is a hyperparameter that can be tuned.

5.3.2 Evaluation Metrics

For the evaluation we will only the predicted forecast for the final time step in the window. We will evaluate the calibration of the risk-neutral and risk-averse forecasting models by means of calibration plots using Equation 2.2, PIT histograms, and by comparing the sample-average CDF to the empirical CDF of the observations as described in Section 2.1.
To compare both models to ones from the literature we will use the following commonly used metrics computed for the test sets. For point predictions we will make use of the following:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2},$$
$$\mathrm{SF} = \begin{cases} \sum_{i=1}^{N} e^{-(\hat{y}_i - y_i)/10} - 1, & \text{if } \hat{y}_i - y_i < 0 \\ \sum_{i=1}^{N} e^{(\hat{y}_i - y_i)/13} - 1 & \text{otherwise,} \end{cases}$$

where $\hat{y}_i = \exp\left(\mu + \frac{\sigma^2}{2}\right)$ is the (mean) predicted RUL and y_i the true RUL. For probabilistic predictions we use

$$\mathrm{PICP} = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 1, & \text{if } y_i \in [U_\alpha(\hat{p}_i), L_\alpha(\hat{p}_i)] \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathrm{NMPIW} = \frac{1}{N(\max\{y\} - \min\{y\})} \sum_{i=1}^{N} U_\alpha(\hat{p}_i) - L_\alpha(\hat{p}_i),$$

here $\hat{p}_i = \text{Lognorm}(\mu_i, \sigma_i^2)$ is the estimated distribution of the RUL and $U_\alpha(\hat{p}_i), L_\alpha(\hat{p}_i)$ the upper and lower bounds for confidence level α.
Lastly, we also want to evaluate whether the risk-averse model underestimates the RUL more than the risk-neutral model, as this coincides with a more risk-averse approach. To do this we will look at the probability mass below the true RUL $\mathbb{P}(x < \text{RUL})$ and the mass above $\mathbb{P}(x > \text{RUL})$, and compare the averages over the test sets.

5.4 Maintenance Scheduling

For the maintenance scheduling we will consider a planning window of D cycles into the future where the agent can choose to plan the engine replacement. The distributional reinforcement learning agent will take inputs from a forecasting model in the form of

$$(F_t(1), \ldots, F_t(D)) = \left(\Phi\left(\frac{\log(1) - \mu_t}{\sigma_t}\right), \ldots, \Phi\left(\frac{\log(D) - \mu_t}{\sigma_t}\right) \right).$$

Here μ_t and σ_t is the output of the final time step from a forecasting model that was given a window of sensor data X_t. We then employ a network which will output three discrete distributions supported on a fixed number of atoms in the range $\left[\frac{R_{\min}}{1-\gamma}, \frac{R_{\max}}{1-\gamma}\right]$. The three distributions correspond to the actions of doing nothing, replacing the engine immediately, and replacing the engine at the end of the planning window. Initial tests with actions to replace at any cycle in the planning window showed that the agents struggled to differentiate between them, resulting in a few of them being randomly favoured over the others. We will use a simple fully connected neural network for this consisting of four layers with dimensions D, n_1, n_2, and $3n_{\text{atoms}}$ (Figure 4). We then use an ϵ-greedy policy over the expected value of the action distributions in the risk-neutral setting. In the risk-averse setting we instead do this over the CVaR of the action distributions. This entails that the agent chooses the action with the highest expected value (or CVaR) with probability ϵ, and a random action with probability $1 - \epsilon$.

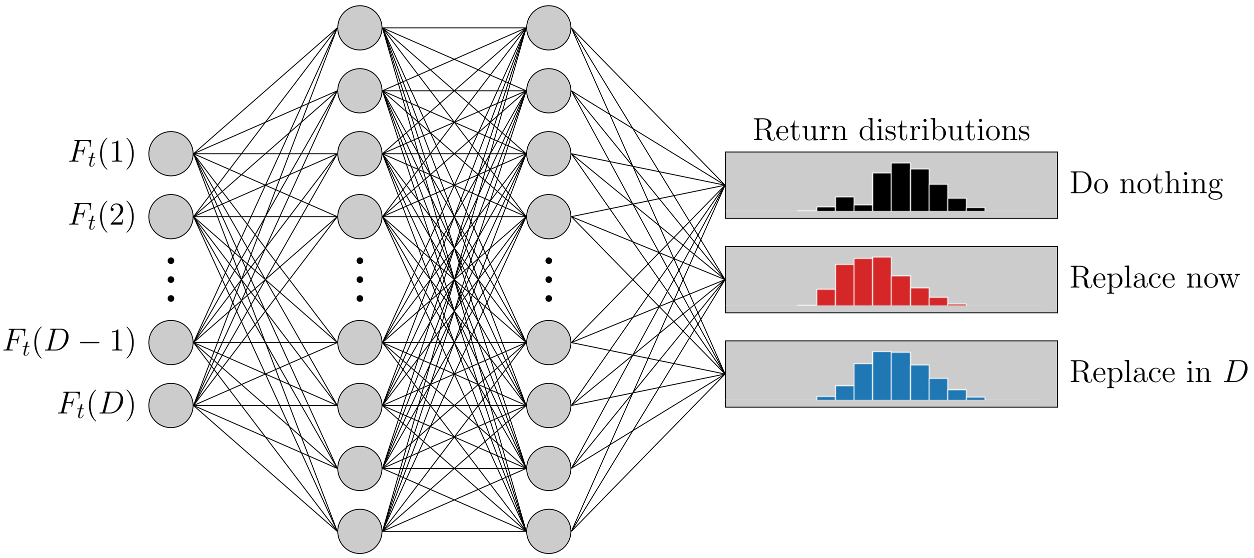


Figure 4: Network architecture of the distributional RL agent. F_t refers to the CDF of a forecasting model.

For each decision the agent is given a reward. We define the reward r_t obtained by the agent for choosing action a_t as

$$r_t = \begin{cases} c_n & \text{if } \mathrm{RUL}_t > D \text{ and } a_t = \textit{do nothing} \\ -c_f & \text{if } \mathrm{RUL}_t \leq D \text{ and } a_t = \textit{do nothing} \\ -c_0 & \text{if } a_t = \textit{replace now} \\ Dc_1 - c_0 & \text{if } \mathrm{RUL}_t \geq D \text{ and } a_t = \textit{replace in } D \textit{ cycles} \\ -c_f & \text{if } \mathrm{RUL}_t < D \text{ and } a_t = \textit{replace in } D \textit{ cycles} \end{cases} \tag{5.1}$$

When choosing these constants we make sure that $c_f > c_0$, $Dc_1 - c_0 < c_n$, and all constants are positive. The agent will be trained and updated according to the C51 algorithm [18] as explained in Section 3.2. The computation of the loss for a sample transition X_t, a_t, X_{t+1} is shown in Algorithm 1. Note here the use of the secondary target network, in the form of the output probabilities $\overline{p}_i$, in the computation of the projected distribution m_i. The primary network, which is to be updated using this loss, is only used to compute the cross-entropy loss in the final line, in the form of the probabilities $p_i(X_t, a_t)$. In the risk-averse setting we instead define $Q(X_{t+1}, a)$ in line 1 of Algorithm 1 as the CVaR (Equation 4.1). The secondary target network is itself updated in regular intervals by copying the weights of the primary network.

Algorithm 1 Categorical algorithm from [18]

```
Input: $X_t, a_t, r_t, X_{t+1}, \gamma_t \in [0, 1]$
 1: $Q(X_{t+1}, a) := \sum_i z_i \overline{p}_i(X_{t+1}, a)$
 2: $a^* \leftarrow \arg\max_a Q(X_{t+1}, a)$
 3: $m_i = 0, \quad i \in \{0, \dots, N-1\}$
 4: for $j \in \{0, \dots, N-1\}$ do
 5:     $\hat{\mathcal{T}} z_j \leftarrow [r_t + \gamma_t z_j]_{V_{\mathrm{MIN}}}^{V_{\mathrm{MAX}}}$
 6:     $b_j \leftarrow \left(\hat{\mathcal{T}} z_j - V_{\mathrm{MIN}}\right) / \Delta z$
 7:     $l \leftarrow \lfloor b_j \rfloor, \quad u \leftarrow \lceil b_j \rceil$
 8:     $m_l \leftarrow m_l + \overline{p}_j(X_{t+1}, a^*)(u - b_j)$
 9:     $m_u \leftarrow m_u + \overline{p}_j(X_{t+1}, a^*)(b_j - l)$
10: end for
Output: $-\sum_i m_i \log p_i(X_t, a_t)$
```

5.4.1 Evaluation

The maintenance scheduling agents will be evaluate and compared using the training instances set aside for testing only for each data set. We reiterate that we can not use the testing sets of each data set as they only often stop short of the last few cycles, which is precisely where actions other than doing nothing will occur. We will evaluate the agents based on two key metrics: the RUL at scheduled replacement (or failure) and the obtained reward until scheduled replacement or failure. For the obtained rewards we will also compare the agents against an optimal strategy. This strategy consists of doing nothing until the RUL is equal to D cycles at which point we schedule the engine replacement in D cycles time.

To investigate why the agents choose each action we will look at which action is chosen for each mean and standard deviation of the forecasted RUL distributions, as well as the frequency of the actions compared to the true RUL and the probability that RUL $\leq D$ based on the RUL forecasts.

6 Results

6.1 RUL Forecasting

We opted to use different window sizes for each of the four data sets. Specifically we used a window size of 31 for FD001, 21 for FD002, 38 for FD003, and 19 for FD004. These correspond to the lengths of the smallest testing samples in each of the four data sets, avoiding the need to pad the data at any point. For both forecast models in all four data sets we used an upper-bound of $U = 0.5$ in the BELU activation function for the σ output, yielding that $\sigma \in (0, 1.5]$. As well as a fixed learning rate of 10^{-3}. All the averse forecasting

FD001	Point Prediction		Probabilistic Prediction		
	RMSE	SF	PICP	NMPIW	CRPS
Risk-Neutral (mean)	9.62	245.92	0.87	0.38	**7.30**
Risk-Neutral (median)	**9.57**	254.25			
Risk-Averse (mean)	10.76	444.27	0.88	0.40	8.19
Risk-Averse (median)	11.19	509.81			
Liu et al. (2019) [9]	14.72	331.9	**0.995**	0.540	
Nguyen et al. (2022) [8]	12.227	243.8	0.950	**0.316**	
Wang et al. (2024) [4]	11.43	**201.26**			

Table 4: Point and probabilistic prediction metrics for FD001 compared to three of the best performing models in the literature.

models used a value of $b = 50$ for the threshold weighted CRPS loss function. The models were trained for a total of 50 epochs.

We note that studies in the literature rarely do RUL forecasts for the FD002, FD003, and FD004 data sets, and we thus lack models to compare to. Therefore, we will mostly focus on the results for the FD001 data set. The results for the other data sets are shown in Appendix A.1.

Figure 5 shows the final mean prediction and the 95% confidence intervals for all the engines in the FD001 test set for the risk-neutral forecaster and the risk-averse forecasters. The values of the point and probabilistic prediction metrics computed for the FD001 testing set are shown in Table 4. For the FD001 data set we see that our risk-neutral model outperforms all other models when it comes to the RMSE. However, it does lack behind with the other metrics, specifically the probabilistic metrics. We do note that for the PICP the optimal value is actually 0.95, as this corresponds to the confidence level chosen. From this we can conclude that the mean (or median) of the predictions of our models is on average very close to the true RUL, at least compared to other models. However, when the prediction is off the true RUL does fall outside of the 95% confidence interval more often than some of the other models in the literature. In Figure 5 we can see that the majority of the instances where the true RUL falls outside of the 95% confidence intervals (the blue and orange errorbars) happen when the RUL is between about 75 and 125. In this region the piecewise linear RUL we use as the target during training has just transitioned from being constant to linearly decreasing. An example, with engine 67, is shown in Figure 6e and 6f, where the forecasters are still predicting a constant RUL. Whilst this is still not ideal it has the upside of not really affecting the maintenance scheduling, as the replacement scheduling actions should ideally only be chosen when the true RUL is quite low and the engine is close to failure.

Figure 7 shows the calibration plots for both forecasters and each data set and in Figure 8 the PIT histograms are shown. For all data sets the models seem decently well calibrated based on the calibration plots. However, all the PIT histograms, besides the one for the risk-averse forecast for FD001, do show a higher than expected value in the first bin. This indicates that more often than expected the observations, meaning the true RUL, fall outside of the confidence intervals. Note that we also observed this in Figure 5 and the lower scores for the PICP. We can see the same for data sets FD002, FD003, and FD004 in Figure 14 and Tables 7, 8 and 9 in Appendix A.1.

Lastly, Table 5 shows the average percentage of probability mass under or over the true RUL for each of the predictions in the test set. We can clearly see the desired effect of the weighted CRPS loss function here with the risk-averse forecast underestimating more than the risk-neutral forecast.

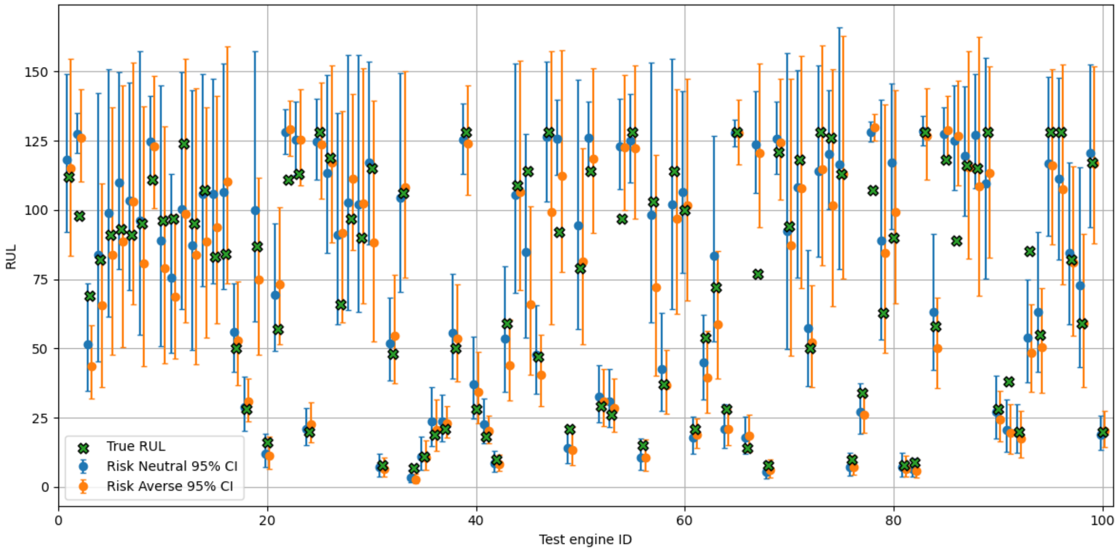
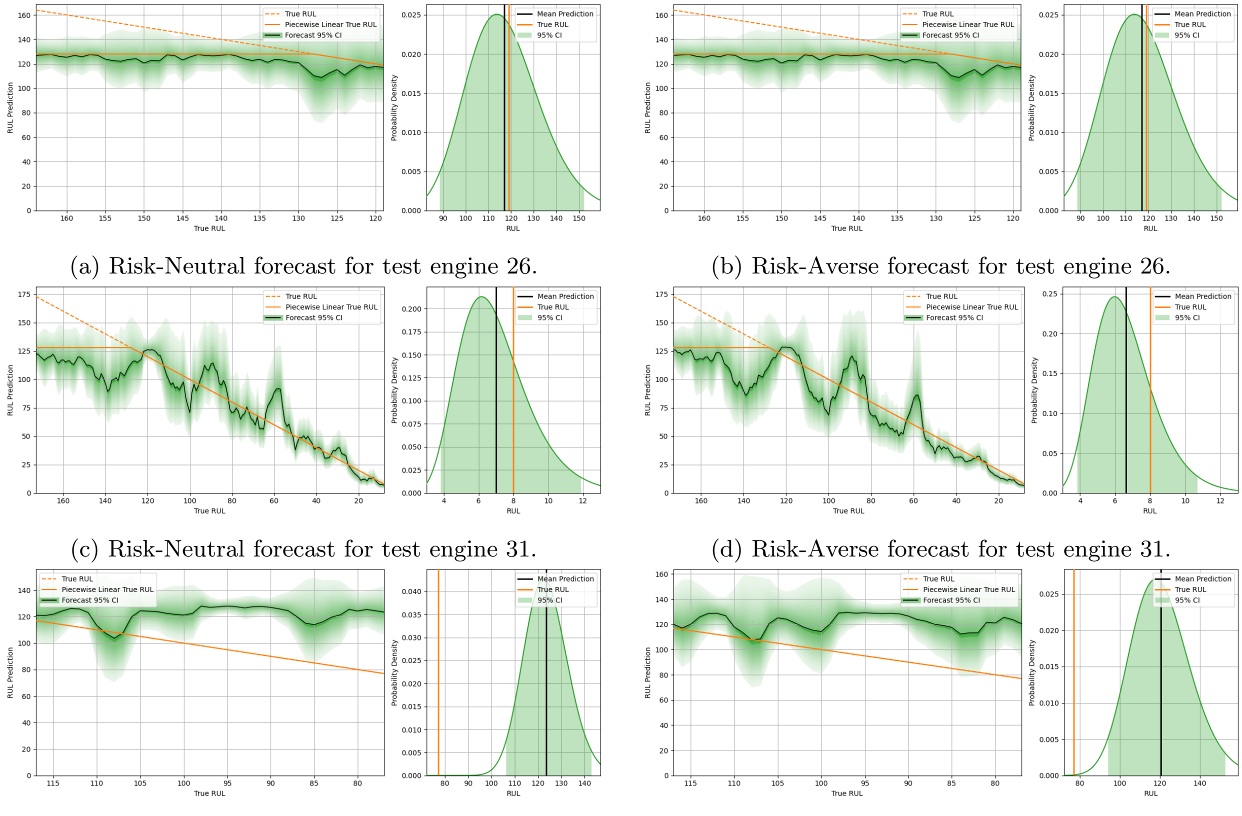


Figure 5: RUL prognostics and 95% confidence intervals for the FD001 test data set with the Risk-Neutral (blue) and Risk-Averse (orange) forecasters.



(a) Risk-Neutral forecast for test engine 26.

(b) Risk-Averse forecast for test engine 26.

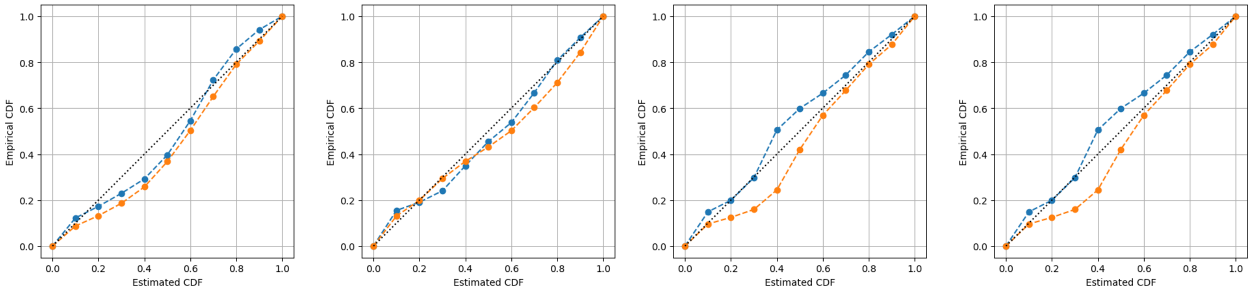
(c) Risk-Neutral forecast for test engine 31.

(d) Risk-Averse forecast for test engine 31.

(e) Risk-Neutral forecast for test engine 67.

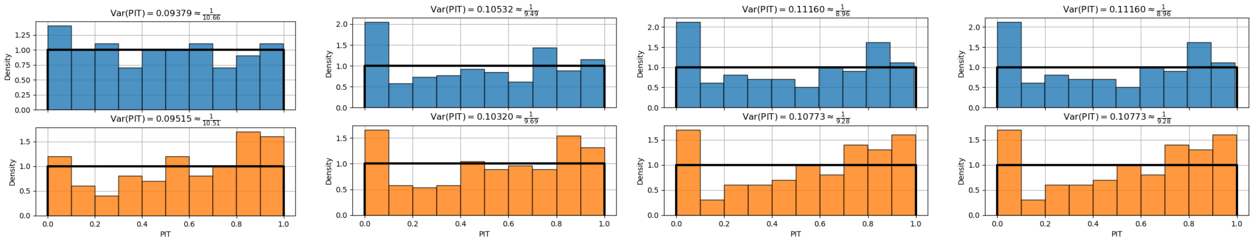
(f) Risk-Averse forecast for test engine 67.

Figure 6: Mean prediction and the 95% confidence intervals, including the prediction at the end of the monitoring period on the right, for three selected test engines of FD001 by the Risk-Neutral forecaster (left) and the Risk-Averse forecaster (right).



(a) FD001 (b) FD002 (c) FD003 (d) FD004

Figure 7: Calibration plots of the estimated CDF of the RUL by the Risk-Neutral (blue) and Risk-Averse (orange) forecasts for the four test data sets.



(a) FD001 (b) FD002 (c) FD003 (d) FD004

Figure 8: Probability integral transform (PIT) histograms for the Risk-Neutral (blue) and Risk-Averse (orange) forecasts of the four test data sets.

	Risk-Neutral		Risk-Averse	
Data set	Under	Over	Under	Over
FD001	47.50%	52.50%	57.50%	42.50%
FD002	48.37%	51.63%	52.76%	47.24%
FD003	48.27%	51.73%	54.87%	45.13%
FD004	45.40%	54.60%	48.32%	51.68%

Table 5: Average percentage of probability mass lower than (under) or greater than (over) the true RUL on the test data sets.

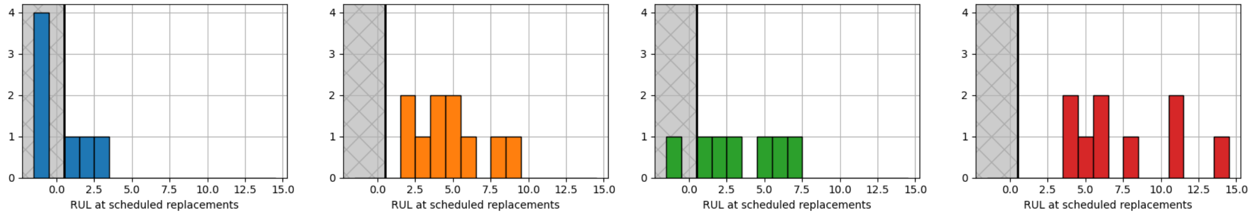
6.2 Maintenance Scheduling

For the maintenance scheduling we used a planning window of $D = 10$ cycles and the constants in the reward function were set to $c_f = 8$, $c_0 = 4$, $c_1 = 0.35$, and $c_n = 0.1$. Furthermore, we set the discount factor to $\gamma = 0.9$ resulting in the range of $[-80, 1]$ for the return distributions. The number of atoms were set to $n_{\text{atoms}} = 20$ and the dimensions of the second and third layer of the networks as $n_1 = n_2 = 128$. These values were chosen based on the agents performance on the validation sets and are equal for all four data sets.
The agents were trained for a total of 40000 episodes, where one episode follows a training engine until failure or replacement. The target network was updated every 200 episodes and the ϵ parameter was set to decrease each episode following the function

$$\epsilon(i) = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min})e^{i/7000},$$

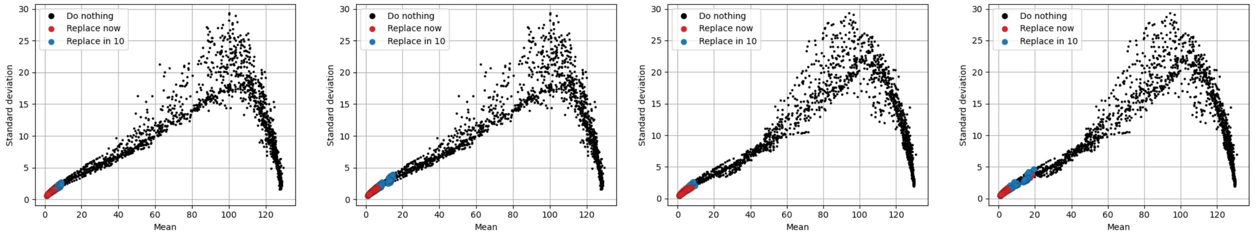
where $\epsilon_{\min} = 0.05$ and $\epsilon_{\max} = 0.9$. The main network was optimized using the Adam optimizer with a learning rate of 0.001. As we have four models per data set we will primarily focus on the results from the data set FD001. The missing results for the other three data sets are shown in Appendix A.2. Videos showing the forecasts, return distributions, and chosen actions for each cycle of all the testing engines in each data set can be found at https://github.com/RvdLaag/master-thesis.

In Figure 9 we show the RUL at the scheduled replacement for the engines selected for testing in FD001. The bar in the grey hatched area on the left contains all engines that failed. Table 6 shows the mean, standard deviation, maximum, and minimum of these same values for all four models. The Neutral-Mean model has the most failures with four and the Averse-Mean model is the only other model with a failure, having only one. We can see in Table 6 that both methods of including an aversion to risk result in an increase in the mean RUL at replacement and prevent failures. The same holds for the models trained on the FD003 data set, however, it does not hold for the other data sets as can be seen in Tables 10, 11 and 12 in Appendix A.2. Specifically, for FD002 the Averse-CVaR model has a lower mean RUL at replacement than both the Neutral-CVaR and Averse-Mean models (Table 10). For FD004 we instead have that the Averse-Mean model has a lower mean RUL at replacement than the Neutral-Mean model (Table 12).
Figure 10 shows the means of the forecasts plotted against their standard deviations for each cycle of the testing engines. Each point is coloured according to the action chosen by the agents. The forecasts and chosen actions for the last 50 cycles of two engines are shown in Figure 11. In Figure 12 we can see the chosen actions for each possible mean ≤ 40 and standard deviation ≤ 10 combination inside the convex hull of the points from the testing engines. There are clear red regions in the bottom left corners in Figure 12, where the agents will always choose to replace the engine immediately. We also see differently shaped blue areas directly to the right of this red region, corresponding to the action of replacing the engine in 10 cycles. These blue areas however, are irregularly interrupted by black regions, corresponding to the do nothing action. This can, for example, have the undesired effect of the agent switching from replacing in 10 to doing nothing when the mean of the forecast decreases. Which clearly happens for the Neutral-CVaR model if we look at Figure 12b and follow the dashed white lines (or the area in between) from right to left. We suspect that this is predominately caused by the definition of our cost function (Equation 5.1), since with this definition the replace in 10 action is only optimal for 1 cycle, namely when $\text{RUL}_t = 10$. This is a similar problem to the one we encountered in the initial tests, mentioned in Section 5.4, with actions to replace at each point in the planning window. Preliminary results for these initial tests showing this can be found in Appendix B.
For these same engines we also show the frequency that each action occurs with for a given true RUL in Figure 13. Here we see that the action of replacing immediately occurs almost entirely when the true RUL is less than 10, and before the true RUL reaches zero it becomes the only action being chosen. Similarly, we see that the action of replacing in 10 cycles occurs primarily when the true RUL is between 5 and 20. The failures we saw in Figure 9 for the Neutral-Mean and Averse-Mean models come from the replacing in 10 cycles actions when the true RUL is less than 10.
We see similar results for the other data sets in Appendix A.2, where for low enough RUL only the replace immediately action is picked. However, just like for FD001 the replace in 10 cycles action is picked sporadically when the true RUL is low, but still greater than about 5. This has the undesired consequence of resulting in occasional failures.



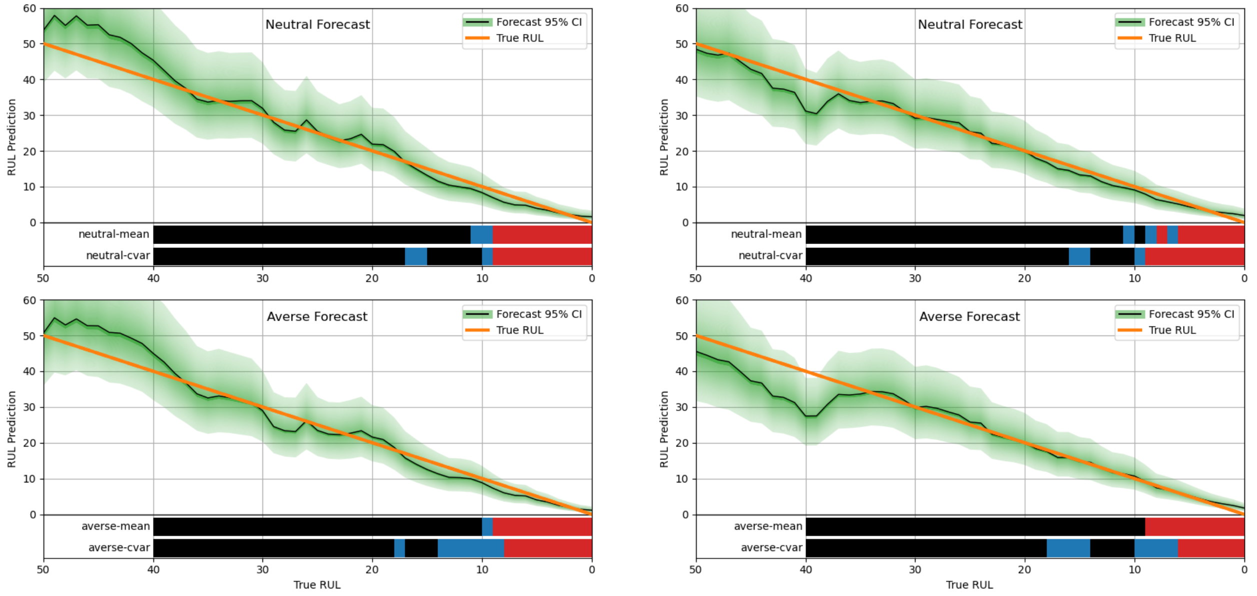
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 9: The remaining useful life (RUL) at scheduled replacement for the test engines in FD001.



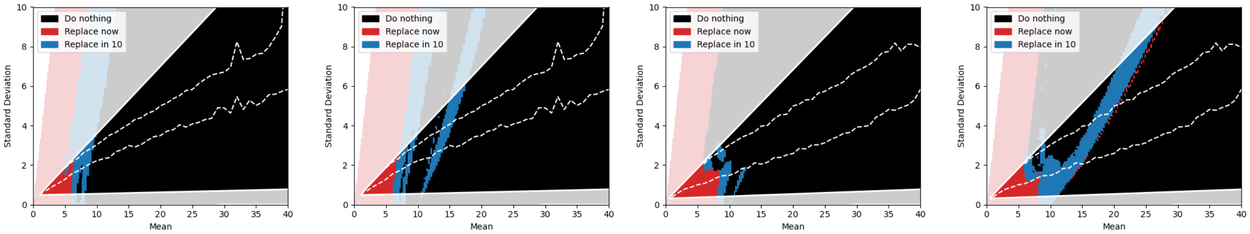
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 10: Means plotted against the standard deviations of the forecasts for the test engines in FD001, coloured by the actions chosen by the models.



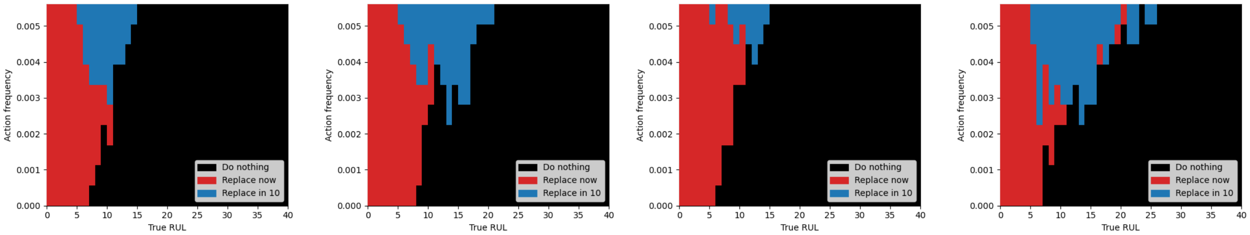
(a) RL test engine 3 (b) RL test engine 10

Figure 11: Forecasts and actions chosen by the models for the last 50 cycles of two engines from FD001 set aside for testing. The actions chosen for each cycle at indicated by the black (doing nothing), blue (replacing in 10 cycles), and red (replacing immediately) bars under the respective forecasts.



(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 12: The actions chosen by the model for each mean (≤ 40) and standard deviation (≤ 10) of the forecasts inside the convex hull of the test points from FD001 shown in Figure 10. The area between the white dashed lines contains all the points that the models have seen during training.



(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 13: The frequency of the chosen actions for the test engines in FD001 plotted against the true RUL.

	Mean	SD	Max	Min
Neutral-Mean	-0.90	2.26	3	-4
Neutral-CVaR	4.80	0.23	9	2
Averse-Mean	1.20	3.99	7	-6
Averse-CVaR	8.40	3.88	15	4

Table 6: Mean, standard deviation, maximum, and minimum of the RUL at scheduled replacement for the test engines in FD001.

7 Discussion

In this thesis we have proposed a risk-averse approach for the probabilistic forecasting of prognostics, such as the remaining useful life, as well as predictive maintenance scheduling using these prognostics. The focus of this work is to investigate whether these risk-averse approaches are feasible and not necessarily to create the best performing models. For this aim, we proposed a simple pipeline consisting of two models. We first make a probabilistic forecast for the RUL of engines in the form of a log-normal distribution using an LSTM. These forecasts are updated after each time step using the sensor measurements from the engines. Using the forecasts we then schedule maintenance of the engines in the form of their replacement using a distributional reinforcement learning agent. This agent takes as input the log-normal distribution of the forecasted RUL and outputs a return distribution for each action. This approach allows for much flexible decision making than using a fixed threshold of the predicted RUL to trigger maintenance.

Our results show that the risk-averse strategy to forecasting has the desired effect of underestimating the RUL more than the risk-neutral approach. This only comes at a slight cost in performance when looking at our point and probabilistic prediction metrics. With both approaches the models perform comparatively to some of the best performing models in the literature. The effect of the risk-averse strategies in maintenance scheduling are less clear, however, they do demonstrate a trend in preventing failures more frequently and have a higher average Remaining Useful Life (RUL) at scheduled replacement compared to their risk-neutral counterparts. The maintenance scheduling agents correctly learned to use the do nothing and replace immediately actions to prevent engine failure whilst optimizing their use. However, the agents struggled to effectively use the replace in 10 cycles actions. Often choosing this action too late and regularly switching back and forth between it and doing nothing. We suspect that the cost function we defined is largely responsible for this, as there is only one cycle per engine life time that this action incurs the lowest cost. This makes the decision of choosing this action complicated since the agent only has access to an approximated distribution of the RUL.

Overall, this thesis has shown that optimizing over a threshold weighted scoring rule, such as the CRPS used here, can be used to effectively create a risk-averse forecasting model. Furthermore, by using these risk-averse forecasts or by choosing actions based on the CVaR of the return distributions instead of the mean, we can induce a slight risk-averse tendency in the maintenance scheduling models.

Future work could investigate the specific impacts of the risk-averse approaches on maintenance scheduling, as the findings in this thesis indicate only a minor tendency. Enhancing the training environment of these models could address this by enabling the agents to better utilize additional actions, such as carrying out replacements at any point within the planning window, which may help to more clearly reveal the effects of the risk-averse strategies.

References

[1] Jorben Sprong, Xiaoli Jiang, and Henk Polinder. “Deployment of Prognostics to Optimize Aircraft Maintenance - A Literature Review: A Literature Review”. In: *Annual Conference of the PHM Society* 11 (Sept. 2019). DOI: 10.36001/phmconf.2019.v11i1.776.

[2] Shuai Zheng et al. “Long Short-Term Memory Network for Remaining Useful Life estimation”. In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 2017, pp. 88–95. DOI: 10.1109/ICPHM.2017.7998311.

[3] Linchuan Fan, Yi Chai, and Xiaolong Chen. “Trend attention fully convolutional network for remaining useful life estimation”. In: *Reliability Engineering & System Safety* 225 (2022), p. 108590. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2022.108590. URL: https://www.sciencedirect.com/science/article/pii/S0951832022002356.

[4] Lubing Wang, Butong Li, and Xufeng Zhao. “Multi-objective predictive maintenance scheduling models integrating remaining useful life prediction and maintenance decisions”. In: *Computers & Industrial Engineering* 197 (2024), p. 110581. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2024.110581. URL: https://www.sciencedirect.com/science/article/pii/S0360835224007022.

[5] Eyke Hüllermeier and Willem Waegeman. “Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods”. In: *Machine Learning* 110 (Mar. 2021). DOI: 10.1007/s10994-021-05946-3.

[6] Abhishek Srinivasan, Juan Carlos Andresen, and Anders Holst. “Ensemble Neural Networks for Remaining Useful Life (RUL) Prediction”. In: *PHM Society Asia-Pacific Conference* 4.1 (Sept. 2023). ISSN: 2994-7219. DOI: 10.36001/phmap.2023.v4i1.3611. URL: http://dx.doi.org/10.36001/phmap.2023.v4i1.3611.

[7] Juseong Lee and Mihaela Mitici. “Deep reinforcement learning for predictive aircraft maintenance using probabilistic Remaining-Useful-Life prognostics”. In: *Reliability Engineering & System Safety* 230 (2023), p. 108908. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2022.108908.

[8] Khanh T.P. Nguyen, Kamal Medjaher, and Christian Gogu. “Probabilistic deep learning methodology for uncertainty quantification of remaining useful lifetime of multi-component systems”. In: *Reliability Engineering & System Safety* 222 (2022), p. 108383. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2022.108383. URL: https://www.sciencedirect.com/science/article/pii/S0951832022000606.

[9] Chongdang Liu et al. “Multiple Sensors Based Prognostics With Prediction Interval Optimization via Echo State Gaussian Process”. In: *IEEE Access* 7 (2019), pp. 112397–112409. DOI: 10.1109/ACCESS.2019.2925634.

[10] Ingeborg de Pater, Arthur Reijns, and Mihaela Mitici. “Alarm-based predictive maintenance scheduling for aircraft engines with imperfect Remaining Useful Life prognostics”. In: *Reliability Engineering & System Safety* 221 (2022), p. 108341. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2022.108341. URL: https://www.sciencedirect.com/science/article/pii/S0951832022000175.

[11] Abhinav Saxena et al. “Damage propagation modeling for aircraft engine run-to-failure simulation”. In: *International Conference on Prognostics and Health Management* (Oct. 2008). DOI: 10.1109/PHM.2008.4711414.

[12] Tilmann Gneiting and Matthias Katzfuss. “Probabilistic Forecasting”. In: *Annual Review of Statistics and Its Application* 1.Volume 1, 2014 (2014), pp. 125–151. ISSN: 2326-831X. DOI: https://doi.org/10.1146/annurev-statistics-062713-085831. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-statistics-062713-085831.

[13] Tilmann Gneiting and Roopesh Ranjan. “Combining predictive distributions”. In: *Electronic Journal of Statistics* 7.none (2013), pp. 1747–1782. DOI: 10.1214/13-EJS823. URL: https://doi.org/10.1214/13-EJS823.

[14] Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. "Probabilistic Forecasts, Calibration and Sharpness". In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 69.2 (2007), pp. 243–268. ISSN: 13697412, 14679868. URL: http://www.jstor.org/stable/4623266 (visited on 06/29/2024).

[15] Tilmann Gneiting and Adrian E Raftery. "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378. DOI: 10.1198/016214506000001437. eprint: https://doi.org/10.1198/016214506000001437. URL: https://doi.org/10.1198/016214506000001437.

[16] Francis X. Diebold and Roberto S. Mariano. "Comparing Predictive Accuracy". In: *Journal of Business & Economic Statistics* 20.1 (2002), pp. 134–144. ISSN: 07350015. URL: http://www.jstor.org/stable/1392155 (visited on 07/01/2024).

[17] Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. The MIT Press, May 2023. ISBN: 9780262374026. DOI: 10.7551/mitpress/14207.001.0001. eprint: https://direct.mit.edu/book-pdf/2111075/book_9780262374026.pdf. URL: https://doi.org/10.7551/mitpress/14207.001.0001.

[18] Marc G. Bellemare, Will Dabney, and Rémi Munos. "A Distributional Perspective on Reinforcement Learning". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, June 2017, pp. 449–458.

[19] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *Nature* 518.7540 (Feb. 2015), pp. 529–533. ISSN: 1476-4687. DOI: 10.1038/nature14236. URL: https://doi.org/10.1038/nature14236.

[20] Gianni Amisano and Raffaella Giacomini. "Comparing Density Forecasts via Weighted Likelihood Ratio Tests". In: *Journal of Business & Economic Statistics* 25.2 (2007), pp. 177–190. DOI: 10.1198/073500106000000332. eprint: https://doi.org/10.1198/073500106000000332. URL: https://doi.org/10.1198/073500106000000332.

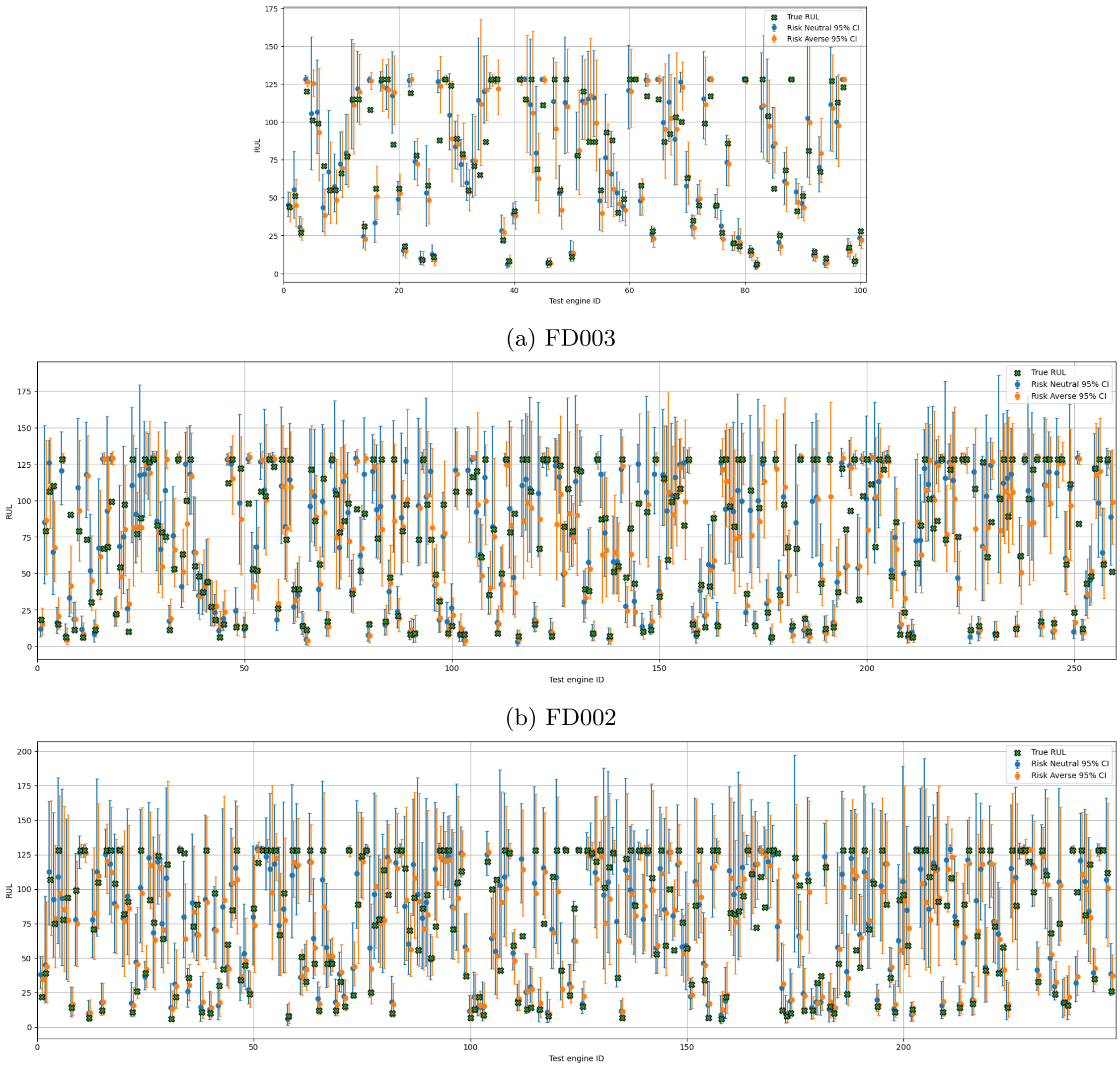
[21] Tilmann Gneiting and Roopesh Ranjan. "Comparing Density Forecasts Using Threshold- and Quantile-Weighted Scoring Rules". In: *Journal of Business & Economic Statistics* 29.3 (2011), pp. 411–422. DOI: 10.1198/jbes.2010.08110. eprint: https://doi.org/10.1198/jbes.2010.08110. URL: https://doi.org/10.1198/jbes.2010.08110.

[22] James E. Matheson and Robert L. Winkler. "Scoring Rules for Continuous Probability Distributions". In: *Management Science* 22.10 (1976), pp. 1087–1096. ISSN: 00251909, 15265501. URL: http://www.jstor.org/stable/2629907 (visited on 07/04/2024).

[23] R.Tyrrell Rockafellar and Stanislav Uryasev. "Conditional value-at-risk for general loss distributions". In: *Journal of Banking & Finance* 26.7 (2002), pp. 1443–1471. ISSN: 0378-4266. DOI: https://doi.org/10.1016/S0378-4266(02)00271-6. URL: https://www.sciencedirect.com/science/article/pii/S0378426602002716.

A Additional Results

A.1 RUL forecasting



(a) FD003

(b) FD002

(c) FD004

Figure 14: RUL prognostics and 95% confidence intervals for the test sets of FD002, FD003, and FD004 with the Risk-Neutral (blue) and Risk-Averse (orange) forecasters.

FD002	Point Prediction		Probabilistic Prediction		
	RMSE	SF	PICP	NMPIW	CRPS
Risk-Neutral (mean)	**12.54**	1847.70	0.85	**0.39**	**9.40**
Risk-Neutral (median)	**12.54**	2159.99			
Risk-Averse (mean)	13.00	2656.26	0.86	**0.39**	9.95
Risk-Averse (median)	13.44	3345.06			
Liu et al. (2019) [9]	24.81	4245.4	**0.955**	0.557	
Wang et al. (2024) [4]	15.69	**1214.47**			

Table 7: Point and probabilistic prediction metrics for FD002.

FD003	Point Prediction		Probabilistic Prediction		
	RMSE	SF	PICP	NMPIW	CRPS
Risk-Neutral (mean)	**9.28**	243.11	0.79	**0.26**	**7.03**
Risk-Neutral (median)	9.32	244.37			
Risk-Averse (mean)	9.61	313.35	0.82	0.28	7.31
Risk-Averse (median)	9.77	330.11			
Liu et al. (2019) [9]	14.99	355.2	**0.963**	0.445	
Wang et al. (2024) [4]	11.28	**181.99**			

Table 8: Point and probabilistic prediction metrics for FD003.

FD004	Point Prediction		Probabilistic Prediction		
	RMSE	SF	PICP	NMPIW	CRPS
Risk-Neutral (mean)	13.43	**2562.43**	0.86	0.49	**9.77**
Risk-Neutral (median)	**13.39**	2848.23			
Risk-Averse (mean)	13.57	2721.57	0.86	**0.47**	9.83
Risk-Averse (median)	13.90	3348.45			
Liu et al. (2019) [9]	28.61	6280.8	**0.919**	0.491	
Wang et al. (2024) [4]	18.35	**2627.11**			

Table 9: Point and probabilistic prediction metrics for FD004.

A.2 Maintenance Scheduling

	Mean	SD	Max	Min
Neutral-Mean	3.46	4.84	16	-5
Neutral-CVaR	14.81	6.96	28	3
Averse-Mean	11.12	7.32	33	0
Averse-CVaR	6.92	6.07	19	-5

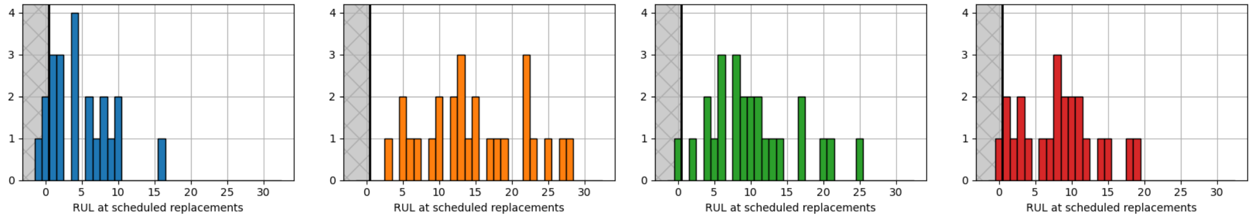
Table 10: Mean, standard deviation, maximum, and minimum of the RUL at scheduled replacement for the test engines in FD002.

	Mean	SD	Max	Min
Neutral-Mean	1.60	2.54	5	-2
Neutral-CVaR	2.80	4.07	11	-3
Averse-Mean	2.10	2.34	7	-2
Averse-CVaR	5.80	4.58	13	-1

Table 11: Mean, standard deviation, maximum, and minimum of the RUL at scheduled replacement for the test engines in FD003.

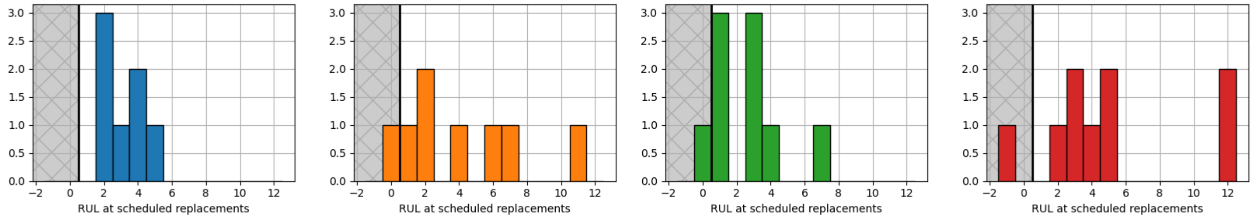
	Mean	SD	Max	Min
Neutral-Mean	6.08	5.53	16	-6
Neutral-CVaR	15.38	12.66	43	-5
Averse-Mean	1.46	4.13	11	-5
Averse-CVaR	18.62	11.06	48	5

Table 12: Mean, standard deviation, maximum, and minimum of the RUL at scheduled replacement for the test engines in FD004.



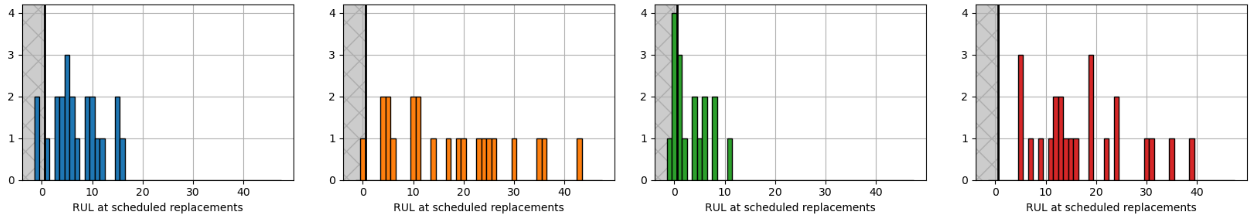
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 15: The remaining useful life (RUL) at scheduled replacement for the test engines in FD002.



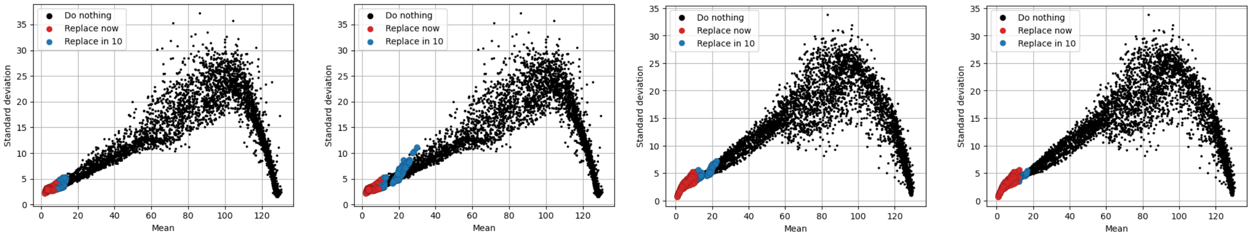
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 16: The remaining useful life (RUL) at scheduled replacement for the test engines in FD003.



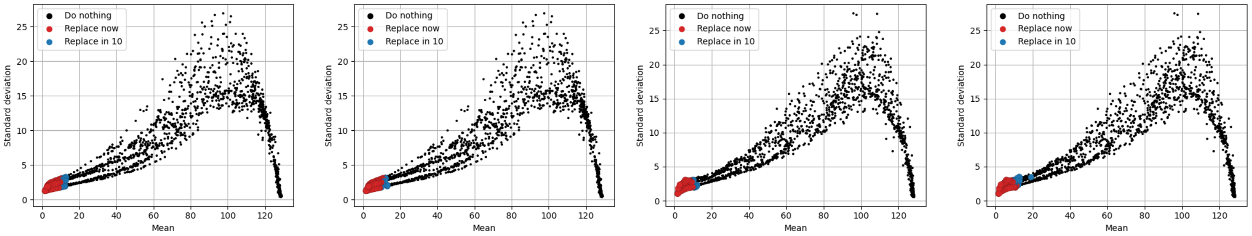
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 17: The remaining useful life (RUL) at scheduled replacement for the test engines in FD004.



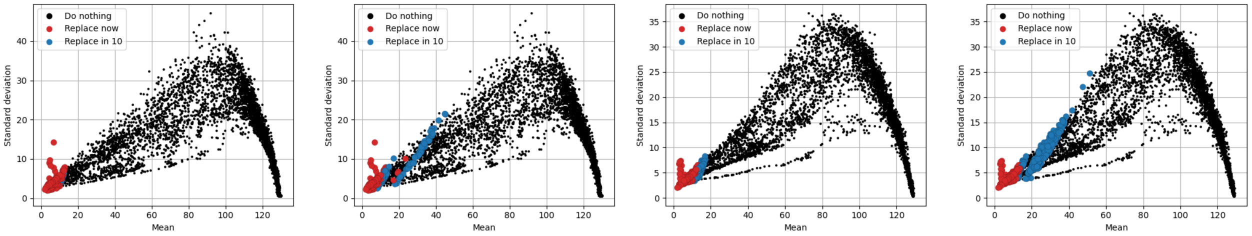
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 18: Means plotted against the standard deviations of the forecasts for the test engines in FD002, coloured by the actions chosen by the models.



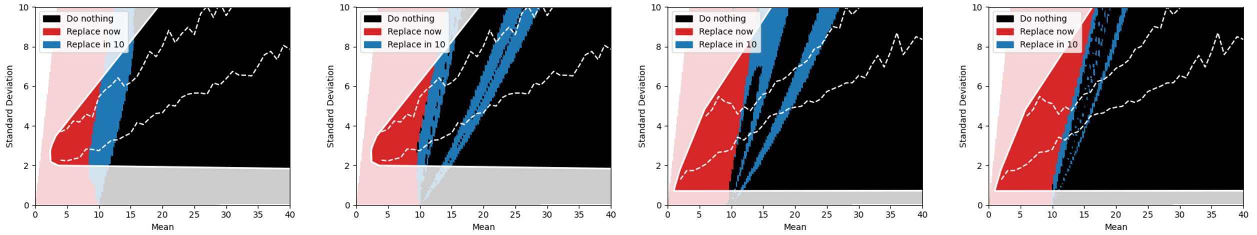
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 19: Means plotted against the standard deviations of the forecasts for the test engines in FD003, coloured by the actions chosen by the models.



(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 20: Means plotted against the standard deviations of the forecasts for the test engines in FD004, coloured by the actions chosen by the models.



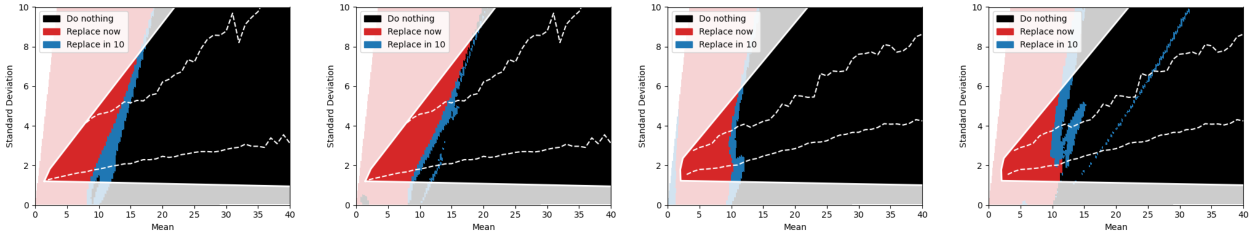
(a) Neutral-Mean model

(b) Neutral-CVaR model

(c) Averse-Mean model

(d) Averse-CVaR model

Figure 21: The actions chosen by the model for each mean (≤ 40) and standard deviation (≤ 10) of the forecasts inside the convex hull of the test points from FD002 shown in Figure 10.



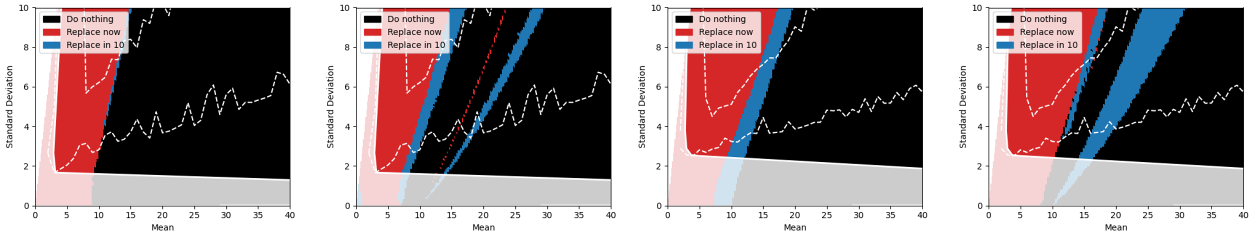
(a) Neutral-Mean model

(b) Neutral-CVaR model

(c) Averse-Mean model

(d) Averse-CVaR model

Figure 22: The actions chosen by the model for each mean (≤ 40) and standard deviation (≤ 10) of the forecasts inside the convex hull of the test points from FD003 shown in Figure 10.



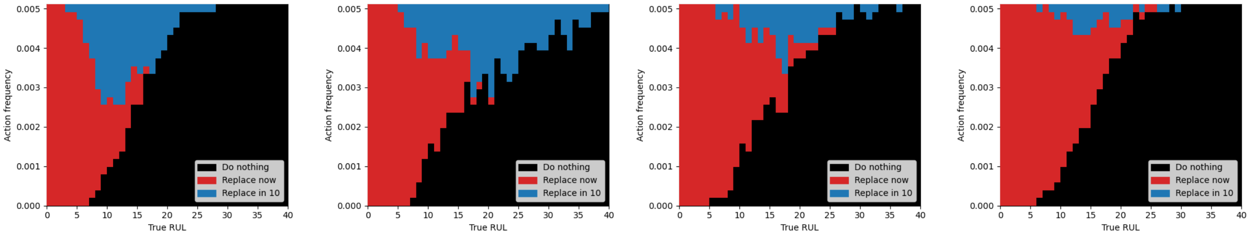
(a) Neutral-Mean model

(b) Neutral-CVaR model

(c) Averse-Mean model

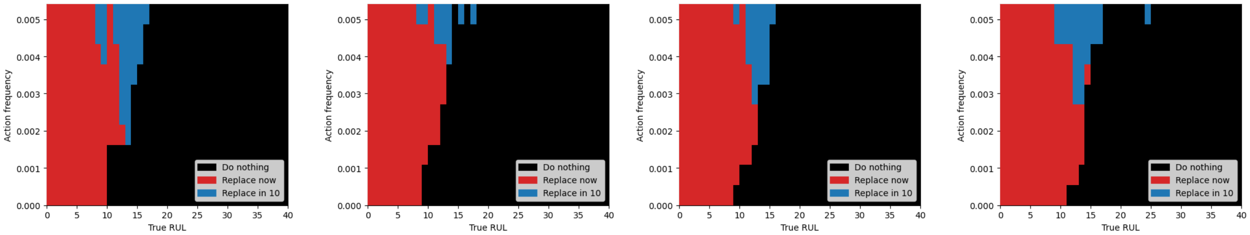
(d) Averse-CVaR model

Figure 23: The actions chosen by the model for each mean (≤ 40) and standard deviation (≤ 10) of the forecasts inside the convex hull of the test points from FD004 shown in Figure 10.



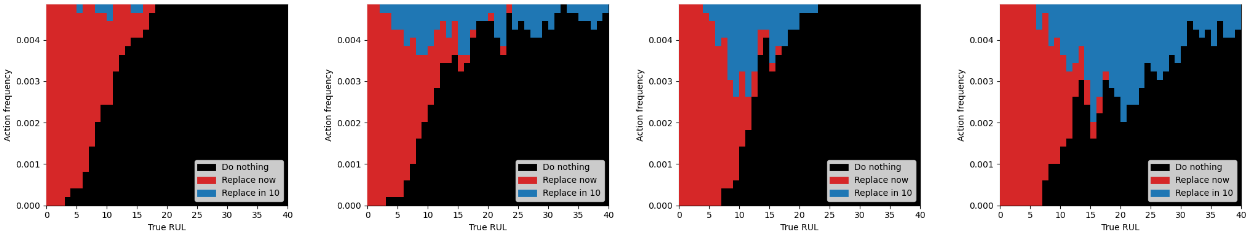
(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 24: The frequency of the chosen actions for the test engines in FD002 plotted against the true RUL.



(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

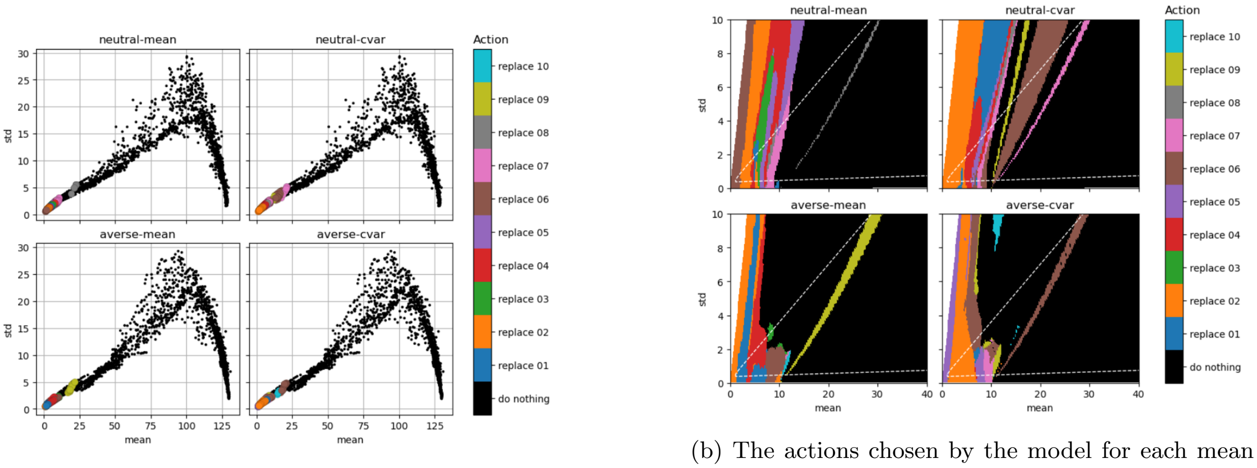
Figure 25: The frequency of the chosen actions for the test engines in FD003 plotted against the true RUL.



(a) Neutral-Mean model (b) Neutral-CVaR model (c) Averse-Mean model (d) Averse-CVaR model

Figure 26: The frequency of the chosen actions for the test engines in FD004 plotted against the true RUL.

B Preliminary Tests



(a) Means plotted against the standard deviations of the forecasts for the test engines in FD001, coloured by the actions chosen by the models.

(b) The actions chosen by the model for each mean (≤ 40) and standard deviation (≤ 10) of the forecasts inside the convex hull of the test points from FD001 shown in Figure 27a.

Figure 27: Initial tests with replacing actions at each cycle in the planning window.